Computer Networks 105 (2016) 22–32

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Regrouping algorithm to alleviate the hidden node problem in 802.11ah networks

Sung-Guk Yoon[a], Jeong-O Seo[b], Saewoong Bahk[c,*]

[a] School of Electrical Engineering, Soongsil University, Seoul, Republic of Korea
[b] TmaxSoft R&D Center, Gyeonggi-do, Republic of Korea
[c] INMC, School of EECS, Seoul National University, Seoul, Republic of Korea

ARTICLE INFO

Article history:
Received 29 June 2015
Revised 3 March 2016
Accepted 13 May 2016
Available online 20 May 2016

Keywords:
802.11ah
Group contention
Hidden node problem

ABSTRACT

An IEEE 802.11ah network has been designed to service a wide range of sensor network applications where a single access point (AP) covers a transmission range of up to 1 km and needs to support more than 8000 nodes. As a result, it has much more hidden node pairs compared to conventional 802.11a/b/g/n/ac networks. In addition, since nodes in power saving mode wake up simultaneously to send frames after receiving a beacon frame from the AP, the hidden node problem can become worse, resulting in frequent packet collisions and performance degradation. The 802.11ah standard proposes a group-based contention scheme to resolve the performance degradation problem, but it cannot resolve it appropriately.

In this paper, we analyze the harmful impact of the hidden node problem on network performance and propose a new grouping algorithm to alleviate the performance degradation. Our proposed hidden matrix based regrouping (HMR) algorithm first finds hidden node pairs, and generates a hidden node matrix accordingly. Then it regroups hidden nodes to alleviate the hidden node problem using the hidden node matrix. Through extensive simulations, we show that our HMR algorithm eliminates most of hidden node pairs, thereby improving the performance of the 802.11ah network significantly in terms of the number of hidden node pairs and power-save poll (PS-Poll) transmission end time.

1. Introduction

An IEEE 802.11 Wireless Local Area Network (WLAN) is one of most popular wireless communication systems in use. 802.11 WLAN systems operate in industrial, scientific, and medical (ISM) bands, such as 2.4 GHz and 5 GHz bands, which are error prone due to the interference from other devices that use the same band. In addition, as many 802.11 WLAN systems are deployed at hot-spot areas, they suffer from severe congestion, and the backward compatibility is a big issue whenever a new WLAN system is introduced.

The IEEE 802.11ah Task Group (TGah) has defined a new WLAN standard, which operates at sub 1 GHz ISM bands. It aims to service a wide range of sensor network applications such as smart grid, and supports an efficient power save mode. Due to the superior propagation characteristic of a low frequency spectrum, the 802.11ah provides a much longer transmission range, i.e., 1 km, compared to 802.11a/b/g/n/ac WLANs, enabling an access point (AP) to support more than 8000 nodes [1].

One of the most important challenges in the 802.11ah is that it suffers from the severe hidden node problem [2]. The probability that any two nodes become hidden from each other increases up to 41% in the case when the network is randomly deployed [3]. For the deployment of 8000 nodes, the expected number of hidden node pairs reaches 1,311,836. In addition, this problem can be exacerbated especially when the power saving mode is active. Since most of nodes in an 802.11ah network operate in power save mode, they wake up and listen to the AP's beacon at the same time, and attempt to transmit power-save poll (PS-Poll) frames, resulting in many collisions.

To minimize such collisions, the 802.11ah proposes a group based contention scheme [4] where each node is allocated to a group, and contends with other nodes within the same group. Previous studies have focused on improving the performance of the group based contention scheme [5–7]. They adjust the number of contention groups or the duration of a restrict access window (RAW) size to achieve optimal performance. In these works, it is assumed that the network is fully connected, i.e., so no hidden

* Corresponding author.
E-mail addresses: sgyoon@ssu.ac.kr (S.-G. Yoon), sjo@netlab.snu.ac.kr (J.-O. Seo), sbahk@snu.ac.kr (S. Bahk).

http://dx.doi.org/10.1016/j.comnet.2016.05.011
1389-1286/

node problem occurs. This means that currently available group-based contention mechanisms are not able to solve the hidden node problem properly although the performance enhancement of the 802.11 access scheme is important research area [8–10].

Extensive researches on solving the hidden node problem in 802.11 networks have been performed, and most of these works have been designed to find hidden node pairs first in a given network [11,12]. In [11], the authors have proposed a hidden node pair detection method that uses a clear channel assessment (CCA), but its effectiveness is limited due to the imperfect CCA information. In [12], a time difference-based detection mechanism that needs the help from neighboring APs has been proposed, but having such a configuration is not practical in 802.11ah networks.

In this paper, we first analyze the performance degradation due to the hidden node problem in 802.11ah networks, and propose a new grouping algorithm, termed hidden matrix-based regrouping (HMR) algorithm, to alleviate this problem. Our proposed algorithm running at the AP first detects hidden node pairs by using the time difference of PS-Poll transmission times. Then it creates a hidden node matrix where each element value is easily obtained from PS-Poll transmission times. Finally, it moves nodes experiencing the hidden node problem into another group one by one, in the order of experiencing hidden nodes in number. Through extensive simulations, we confirm that our proposed algorithm significantly reduces the total number of hidden nodes, leading to performance improvement in terms of PS-Poll transmission end time and the number of retransmissions.

The rest of the paper is organized as follows. In Section 2, we briefly describe the 802.11ah standard and the system model in Section 3. We analyze the PS-Poll transmission end time in Section 4, and propose our hidden matrix based regrouping (HMR) algorithm in Section 5. After evaluating the proposed scheme in Section 6, we conclude our paper in Section 7.

2. IEEE 802.11ah: overview

The IEEE 802.11ah standardization project has been in progress and its fifth draft was published in 2015 [4].

A key difference of 802.11ah from conventional WLANs is the band in which it operates. While conventional 802.11 WLANs operate in 2.4 GHz and 5 GHz bands, 802.11ah operates in sub 1 GHz license-exempt bands. Owing to the good channel propagation characteristics of the sub 1 GHz bands, the transmission range of an 802.11ah AP is larger than those of conventional 802.11 WLAN APs [13]. For instance, a single 802.11ah AP can cover up to 1 km in an outdoor area [14], which is much longer than other 802.11 APs in 2.4 GHz and 5 GHz bands. The unique features of 802.11ah open its use cases in smart grid, indoor health-care systems, and cellular systems.

The following are major requirements for an 802.11ah system [15]: i) Its maximum physical (PHY) data rate is of at least 100 kbps with a coverage of 1 km, ii) an 802.11ah AP supports a maximum of 8,191 nodes, iii) an 802.11ah system harmonizes with IEEE 802.15.4 and IEEE 802.15.4g systems, and iv) it supports an enhanced power saving mode.

2.1. PHY layer

Basically, the PHY layer in an IEEE 802.11ah system is a down-clocked version of an IEEE 802.11ac system. The 802.11ah standard supports 1 MHz, 2 MHz, 4 MHz, 8 MHz and 16 MHz bandwidths, where the 2 MHz bandwidth is the basic one consisting of 64 sub-carriers and the subcarrier spacing is of 31.25 kHz. The 802.11ah system uses normal- and double-length cyclic prefixes (CPs) to cover wide outdoor areas [16]. It also supports a 1 MHz mode PHY with 32 sub-carriers and a 2 × repetition code that enables to extend its transmission range further.

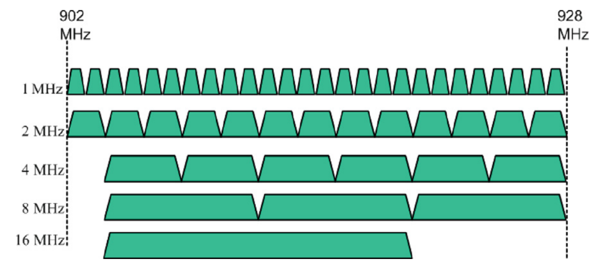


Fig. 1. Channelization for IEEE 802.11ah in the United States.

Table 1
Time Parameters of IEEE WLANs.

μs	11a	11b	11n	11ac	11ah
T_{SIFS}	16	10	16	16	160
T_{DIFS}	34	50	34	34	264
Backoff slot time (σ)	9	20	9	9	52

The license-exempt bands in sub 1 GHz are different in different nations. For instance, they are 902–928 MHz with bandwidths of 1, 2, 4, 8, and 16 MHz in the United States, and 917.5–923.5 MHz with bandwidths of 1, 2, and 4 MHz in Korea. Fig. 1 shows an channelization example in the United States [17]. Because of a small bandwidth per channel, 13 independent channels exist with 2 MHz bandwidth.

2.2. MAC layer

The media access control (MAC) layer of the 802.11ah standard includes several enhancements and tailored features that target each use case.

2.2.1. Timing parameters

The extended coverage has a disadvantage due to the prolonged propagation delay. Therefore, 802.11ah has to use longer time intervals for short inter-frame spaces (SIFS), distributed coordination function (DCF) inter-frame spaces (DIFS), and idle slot times, compared to those of IEEE 802.11a/b/n/ac systems. Table 1 compares the parameters modified in the 802.11ah standard.

2.2.2. Maximum number of nodes

Another enhancement in 802.11ah is the number of nodes covered by an AP. To cover a large area, an 802.11ah AP needs to support a large number of nodes. To this end, the 802.11ah system uses a 13-bit association identifier (AID), thereby supporting 8191 nodes (i.e., $2^{13} - 1$) at its maximum, which is about four times greater than those of conventional WLAN networks, i.e., 2007 nodes [18].

2.2.3. Frame format

The 802.11ah system reduces the sizes of the PHY and MAC headers. Actually, a modified frame format[1] was proposed as an optional feature in the 802.11ac standard. However, it is not widely used due to the backward compatibility. On the other hand, since 802.11ah is free from this issue, it can use a reduced frame format. Owing to the new frame format, the size of the acknowledgement (ACK) frame has been also reduced to the ACK transmission time of 240 μsec.[2]

[1] The modified frame format is called a green field system.
[2] If 802.11ah uses a normal ACK, its transmission time is 480 μsec.

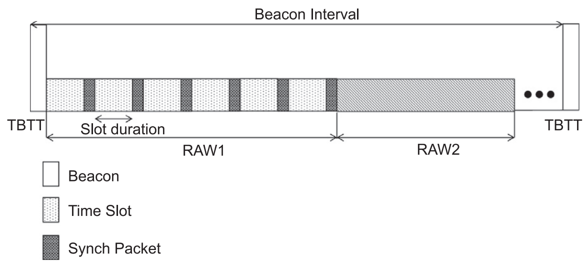


Fig. 2. Example of a beacon interval in 802.11ah. The beacon contains a TIM. RAW1 and RAW2 are used to send PS-Poll frames and buffered data frames, respectively. There are six slots in RAW1 because the considered number of groups is six.

2.2.4. Channel access

The channel access in 802.11ah is a combination of a centralized scheduling and the DCF medium access technique. A beacon interval is divided into several RAWs. All the nodes wake up every target beacon transmission time (TBTT) to listen to the beacon frame that contains the start timing and duration of each RAW within the beacon interval. This is the centralized scheduling part by the AP. Each node enters sleep state and wakes up again at the assigned RAW, and it uses DCF within the RAW.

2.2.5. Power save mode

In power save mode, a node is in sleep state most of the time. It wakes up every TBTT to check whether the AP has waiting packets destined for it. The AP broadcasts a traffic indication map (TIM) through a beacon frame at TBTT, and the TIM carries the information on nodes that the AP has packets destined for. To receive a packet, each node in the list sends a PS-Poll frame to the AP during the allocated RAW.[3] When the AP successfully receives a PS-Poll frame, it replies with an ACK to confirm its reception. Then the node that has received the ACK goes into sleep state, and wakes up to receive the buffered packet from the AP when the next allocated RAW comes.

2.2.6. Group based contention

Since there are a large number of nodes in power save mode, too many nodes wake up simultaneously and send PS-Poll frames, resulting in increased collision and degraded throughput. To solve this problem, 802.11ah defines a grouping method, where the AP divides all the nodes into several groups and allocates a non-overlapping period to each group. Therefore, each node contends with nodes within the same group during the assigned time period of a slot.[4]

Each node wakes up at the beginning of the assigned slot. The AP sends a synch frame when the channel is idle. Then all the nodes in a same group synchronize with each other and become ready to access the channel.

Using a simple modulo operation, the AP divides all the nodes into several groups. Let G_n denote the group index for node n, defined as

$$G_n = (A_n \bmod N_{\text{group}}) + 1, \tag{1}$$

where A_n and N_{group} denote the AID of node n and the number of groups, respectively. The group number for each node is informed through the TIM [16].

Fig. 2 depicts an example of a beacon interval. In this example, there are two RAWs and six slots in RAW1, i.e., $N_{\text{group}} = 6$. RAW1 and RAW2 are used to send PS-Poll frames in uplink and buffered data frames in downlink, respectively. The AP broadcasts a beacon including TIM at TBTT. Through TIM, each node knows its own group number and whether the AP has buffered packets for it. The nodes for which the AP has packets buffered wake up at their corresponding slots to send PS-Poll frames, and only the nodes that have received ACK frames from the AP wake up again at the RAW2 to receive the buffered packets. The other nodes enter sleep state and sleep until the next TBTT.

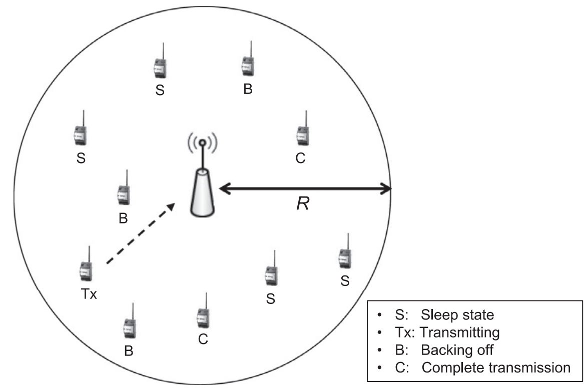


Fig. 3. Example of an 802.11ah network. The AP and all the nodes have the transmission range of *R*.

3. System model

We consider an 802.11ah network with N nodes and a single AP. The AP is plugged into an electric outlet while all the nodes are battery powered. We assume the nodes operate in power save mode, and they are uniformly distributed within the AP coverage with no mobility.

To investigate the hidden node problem, we consider PS-Poll frame transmissions (i.e., uplink) that are normally sent in the first RAW. Each node periodically wakes up at TBTT to check whether the AP has buffered packets for it. The node with packets to receive from the AP transmits a PS-Poll frame. Otherwise, it goes back to sleep state until the next TBTT. After sending the PS-Poll frame, if the node fails to receive an ACK frame, it retransmits the PS-Poll frame again and again until the number of retransmissions reaches a predetermined retry number. When the node receives an ACK frame, it waits in sleep state and wakes up to receive the buffered packet at the next RAW. This means that the number of nodes that attempt to transmit PS-Poll frames is reduced by whenever the PS-Poll and ACK exchange is successful.

Fig. 3 shows an example of an 802.11ah network with one AP and ten nodes where R denotes the AP's transmission range. Six nodes wake up to receive buffered packets from the AP, and the other four nodes remain in sleep state. Among the six nodes awake, the two nodes have already finished the PS-Poll and ACK exchange, three nodes are backing off, and one node is transmitting its PS-Poll frame. The nodes that have successfully sent the PS-Poll frame remain in sleep state again until the next RAW for data transmission.

4. PS-Poll transmission end time analysis

This section analyzes the PS-Poll transmission end time with and without hidden nodes in an 802.11ah network. In the 802.11ah network, most of nodes are in power save mode and many of them may be hidden from each other, and nodes can only transmit packets during their allocated RAWs. Therefore, the PS-Poll transmission end time is important in allocating a proper time period to each RAW.

[3] Other nodes that are not listed enter sleep state instantly and sleep until the next TBTT.

[4] A conventional 802.11 system uses the term 'slot' as idle slot for channel access. In this section, we use this term as subperiod in a RAW.

4.1. Probability model and duration parameters

The 802.11ah system still uses the same channel access protocol of DCF as in the existing standards. Our analysis model is based on Bianchi's throughput analysis [19] and Chatzimisios's delay analysis [20] that use the Markov chain model. This section briefly summarizes several probabilities and duration parameters to get the PS-Poll transmission end time.

Let p and τ denote the conditional collision probability that a packet transmission experiences collision and the probability that a node attempts transmission in a randomly chosen time slot,[5] respectively. We get τ from the steady state probabilities $b_{i,k}$ at the backoff stage i and the backoff counter k of the Markov chain, given as

$$\tau = \sum_{i=0}^{m} b_{i,0} = \frac{2(1-2p)}{(1-2p)(C_{\min}+1) + pC_{\min}(1-(2p)^m)}, \tag{2}$$

where $C_{\min}$ and m are the minimum contention widow size and the maximum backoff stage, respectively. In general, p is a function of τ, which is given by

$$p = 1-(1-\tau)^{N-1}. \tag{3}$$

From this, we can calculate the probabilities P_{tr}, P_s, and P_c, which denote the probabilities that at least one node transmits a packet in a slot time, the packet transmission is successful, and the transmission collides with other transmissions, respectively, as follows.

$$P_{tr} = 1-(1-\tau)^N, \tag{4}$$

$$P_s = \frac{N\cdot\tau\cdot(1-\tau)^{N-1}}{P_{tr}}, \tag{5}$$

$$P_c = 1-P_s. \tag{6}$$

Let T_s and T_c denote the durations of a successful packet transmission and a packet collision, respectively. We obtain the duration parameters as

$$T_s = T_{DIFS} + T_H + T_D + T_{SIFS} + T_{ACK}, \tag{7}$$

$$T_c = T_{DIFS} + T_H + T_D + T_{SIFS}, \tag{8}$$

where T_{DIFS}, T_H, T_D, T_{SIFS}, and T_{ACK} are the durations of the DIFS, packet header, data, SIFS, and ACK frame, respectively.

4.2. End time analysis

Since a RAW for PS-Poll transmissions allows each node to transmit only one PS-Poll frame, a node that succeeds in PS-Poll transmission goes into sleep state again until the next RAW begins. Therefore, we define the PS-Poll end time as the time when all the nodes in a group successfully exchange their PS-Poll and ACKs with the AP.

Let F_N denote a random variable for the PS-Poll transmission end time when there are N nodes, which is the sum of each node's PS-Poll transmission end time. Then we obtain the average $E[F_N]$ as

$$E[F_N] = \sum_{n=1}^{N} E[D_n], \tag{9}$$

where D_n denotes the duration for a first successful PS-Poll and ACK exchange when there are n nodes in the network. The duration consists of the waste time due to idle slots and collisions, and the period for a successful PS-Poll transmission. Then we have

$$E[D_n] = E[X^n]\cdot E[T^n_{slot}] + T_s, \tag{10}$$

where X^n and T^n_{slot} represent random variables for the number of slot times and the length of a slot time, respectively, that are required for a first successful PS-Poll transmission when n nodes are contending.[6] Note that the durations for idle slots and collisions are counted in X^n and T^n_{slot}. Since X^n consists of idle slots and collisions, we have

$$E[T^n_{slot}] = P(\text{idle}|\text{no success})\cdot\sigma + P(\text{collision}|\text{no success})\cdot T_c, \tag{11}$$

where σ is the duration of a backoff slot time and

$$P(\text{idle}|\text{no success}) = \frac{1-P_{tr}}{1-P_sP_{tr}}, \tag{12}$$

$$P(\text{collision}|\text{no success}) = \frac{P_{tr}(1-P_s)}{1-P_sP_{tr}}. \tag{13}$$

After obtaining $E[X^n]$ for $1\le n\le N$, we then calculate the average PS-Poll transmission end time. We begin with the initial case $E[X^N]$ where all N nodes start to contend after receiving a synch frame. Let X^n_i denote a random variable that represents the number of slot times that are required for a first successful PS-Poll transmission when n nodes are contending at the backoff stage i. Since all the nodes simultaneously start to contend at the backoff stage 0 in this case, the average number of slot times becomes $E[X^N] = E[X^N_0]$.

According to the definition of X^n_i, its expectation is given by

$$E[X^n_i] = \sum_{j=i}^{m} P^n_{s,j}\cdot E[W^n_j], \tag{14}$$

where $P^n_{s,j}$ represents the probability that the first PS-Poll transmission succeeds in the jth backoff stage when there are n contending nodes, and W^n_j represents a random variable for the number of slot times wasted before the first successful PS-Poll transmission when there are n nodes at the backoff stage j. We obtain $P^n_{s,j}$ as

$$P^n_{s,j} = p^j\cdot P_s. \tag{15}$$

The average number of slot times $E[W^n_j]$ is given by

$$E[W^n_j] = \sum_{x=0}^{C_j-1} \frac{x}{C^n_j}\left((C_j-x)^n - (C_j-(x+1))^n\right), \tag{16}$$

where C_j is the contention window size at the backoff stage j. The detailed procedures are given in Appendix A. Therefore, we obtain $E[X^N] = E[X^N_0]$.

Next, we move on to obtaining $E[X^n]$ for $1\le n\le N-1$. We cannot obtain these using the same procedures as those used for $E[X^n_0]$ since nodes may undergo contention at any backoff stage. We use the probability P^n_i that a node is at backoff stage i when n nodes are left and the previous PS-Poll transmission under $n+1$ contending nodes succeeds. Then we have

$$E[X^n] = \sum_{i=0}^{m} P^n_i\cdot E[X^n_i], \tag{17}$$

where we obtain P^n_i by using the steady state probabilities of the Markov chain model [19] under the saturation assumption, i.e.,

$$P^n_i = \sum_{k=0}^{C_i} b_{i,k}, \tag{18}$$

where $b_{i,k}$ is the steady state probability at the backoff stage i and the backoff counter k.

[5] The term 'slot' in this section indicates an idle slot for channel access.

[6] It is assumed that X^n and T^n_{slot} are independent.

Finally, we get the average PS-Poll transmission end time in the case of N nodes as

$$E[F_N] = \left(E[X_0^N] \cdot E[T_{slot}^N] + T_s\right) + \sum_{n=1}^{N-1} \left(E[X^n] \cdot E[T_{slot}^n] + T_s\right). \qquad (19)$$

4.2.1. Case with no hidden node pair

In [21], an approximation for p was proposed with a conditional collision probability that is independent of the number of retransmissions. The authors assumed that no hidden pair exists in the network. In their model, the collision probability is given as

$$p = \frac{2C_{\min}(n-1)}{(C_{\min}^2 + 2C_{\min}(n-1))}. \qquad (20)$$

We use this result to get the PS-Poll transmission end time for no hidden node pair case.

4.2.2. Case with hidden node pairs

The conditional collision probability in this case is higher than that in the case with no hidden node pair since some hidden nodes attempt their transmissions regardless of on-going transmissions. In [22], the authors obtained the collision probability considering hidden node pairs as

$$p = 1 - (1-\tau)^{N_c - 1}[(1-\tau)^{N_{hdd}}]^k, \qquad (21)$$

where N_c and N_{hdd} represent the numbers of contending nodes and hidden nodes, respectively. k indicates the approximate number of slot times for $2T_s$, which is given by

$$k = \frac{T_s/\sigma}{1 + (1-(1-\tau)^n)(T_c/\sigma - 1) + n\tau(1-\tau)^{n-1}(T_s/\sigma - T_c/\sigma)}. \qquad (22)$$

The term $[(1-\tau)^{N_{hdd}}]^k$ is the probability that hidden nodes should not access the channel until the current transmission ends. Let P_{hdd} denote the probability that any of the two nodes becomes hidden from each other. Then we have

$$N_c = N(1 - P_{hdd}), \qquad (23)$$

$$N_{hdd} = NP_{hdd}. \qquad (24)$$

We use the probability p to get the PS-Poll transmission end time when there are some hidden node pairs.

4.3. Model validation

To validate the analytical model, we compare our analysis results with simulation results that are obtained using an event-driven simulator written in C++. In the case with hidden node pairs, nodes are randomly distributed within the AP coverage and deployed within their mutual coverage as in the case with no hidden node pair. Since the PS-Poll transmission end time mainly depends on the node deployment, all the results are averaged over 10,000 runs.

Simulation environments follow the IEEE 802.11ah specification [4], and detailed simulation parameters are listed in Tables 1 and 2. With these parameters, we set $T_s = 1249$ μsec and $T_c = 849$ μsec, and vary the number of nodes from 1 to 20. Since the probability of any two nodes being hidden from each other is 0.41 under a randomly deployed scenario,[7] we let $P_{hdd} = 0.41$.

Fig. 4 shows the average PS-Poll transmission end time against the number of nodes associated with the AP. The analytical result shows almost the same result with that of simulation. In the cases with no hidden node pair and with hidden node pairs, the average

Table 2
Simulation parameters.

Simulation parameter	Value
Data rate	0.65 Mbps
Bandwidth	2 MHz
PHY header (T_H)	240 μsec
Length of PS-Poll	20 bytes
Length of proposed PS-Poll	28 bytes
Minimum contention window (C_{min})	32
Maximum contention window	1024
Number of groups	6

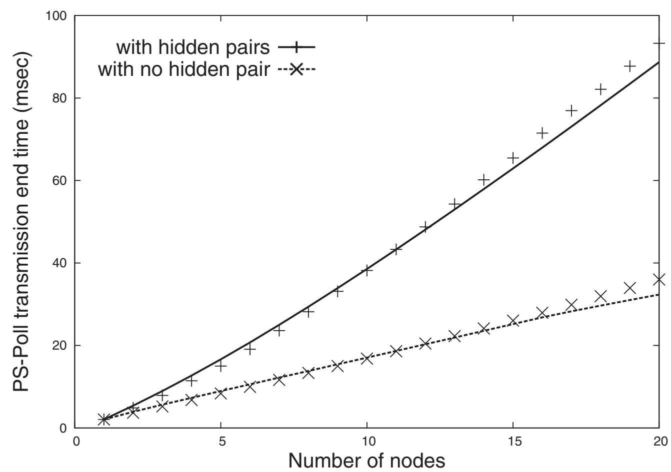


Fig. 4. The average PS-Poll transmission end time according to the number of nodes. The lines and marks represent the analysis and simulation results, respectively.

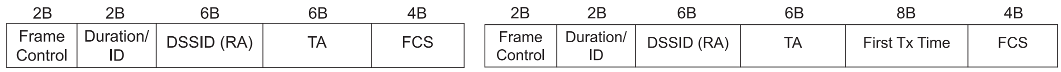
gaps between the analytical and simulation results are 4.6% and 5.3%, respectively. In the case of 20 nodes with hidden node pairs, the average PS-Poll end times are 93.3 msec and 36.0 msec with and without hidden node pairs, respectively.

Currently, commercial Wi-Fi APs have a default beacon interval of 100 msec [23]. Since 802.11ah is designed for use in wide area sensor networks, its beacon interval is configured to be much larger, e.g., 5 s [24]. The PS-Poll transmissions take place in a specific RAW where, for instance, assuming the period of a RAW is one second and there are 10 groups, one slot lasts for 100 msec. Then all the nodes may not be able to finish their PS-Poll transmissions in a given time slot when 20 nodes for a group are randomly chosen. However, if there is no hidden node pair in the group, the same slot duration of 100 msec is sufficiently long enough to exchange PS-Poll and ACK transmissions. This leads us to conclude that the existence of hidden node pairs heavily affects network performance.

5. Hidden Matrix based Regrouping (HMR) algorithm

The existing grouping algorithm for 802.11ah reduces the frequency of collisions by simply lowering the allocated number of nodes in each group. The algorithm, however, does not solve the hidden node problem properly since each node is randomly allocated to a group member. To reduce the number of nodes suffering the hidden node problem, we propose a new grouping algorithm that aims to alleviate the hidden node problem using hidden node relation observed during the network operation. Our HMR algorithm consists of three parts: hidden node detection, hidden node matrix generation, and hidden node regrouping.

[7] In [3], the authors have numerically calculated this probability.



(a) PS-Poll frame structure of the 802.11ah standard

(b) Proposed PS-Poll frame structure

Fig. 5. PS-Poll frame structures of the 802.11ah standard and our proposed scheme. Our proposed structure includes additional eight bytes to feed the first PS-Poll transmission time of each node back to the AP.

5.1. Hidden node detection

To detect the hidden node relation between two nodes, we use a similar method to the time difference-based detection method introduced in [12], which detects a hidden node pair through the help of neighboring APs. Different from this, our scheme does not rely on neighboring APs.

Since hidden node pairs are detected in uplink transmission only, we use the PS-Poll transmission time as the hidden node detection metric. Let t_n and $T_{\text{PS-Poll}}$ denote the start time of the first PS-Poll transmission of node n and the duration of the PS-Poll frame transmission, respectively. We infer most likely that two nodes x and y are hidden from each other if the following condition is met,

$$\epsilon < |t_x - t_y| < T_{\text{PS-Poll}}, \tag{25}$$

where ϵ is a small timing variable to cover the propagation delay and clock drift offset between nodes. That is, ϵ should be in between the propagation delay for 1 km and one backoff slot time. Since they are 3.33 μsec and 52 μsec, respectively, we set ϵ to 20 μsec. In case of $|t_x - t_y| \le \epsilon$, nodes x and y are not hidden from each other in the proposed detection method. The PS-Poll transmission fails due to the random backoff collision, i.e., a same random number is selected by nodes x and y.

If the AP knows t_n for all $n \in N$, it can find a hidden node relation between any two nodes. [8] However, this information cannot be obtained from a PS-Poll transmission in failure because the AP has no idea about which node has sent the failed packet. Therefore, the transmitter ID and t_n should be fed back to the AP. We consider two feedback approaches: i) modifying the PS-Poll frame structure in the 802.11ah standard and ii) sending explicit control frames in other RAWs or best effort region during the same beacon period. Regarding the first approach, we need to have an additional header field of eight bytes for t_x. Fig. 5 compares the PS-Poll frame structures of the 802.11 standard and our proposed structure. That is, while the original PS-Poll frame has the header of 20 bytes, our modified frame has 28 bytes. The former is more efficient than the latter, but requires a modification to the 802.11ah specification. The latter one is more practical and standard compliant.

Fig. 6 shows an operation example of the proposed hidden node detection method. In this example, it is assumed that nodes 1 and 2 are hidden from each other. Node 2 transmits a PS-Poll frame while node 1 is transmitting, resulting in collision. Later on, nodes 2 and 1 succeed in exchanging PS-Poll and ACK frames, and the AP knows t_1 and t_2. Since the time difference satisfies the condition (25), the hidden node relation is detected.

5.2. Hidden node matrix generation

After receiving all t_x, the AP determines the hidden node relation between nodes according to Eq. (25). Then it generates a hidden node relation matrix where element (i, j) has '1' if nodes i and j are hidden from each other, and '0' if their hidden node relation has not been detected yet. The hidden node matrix is symmetric and updated every TBTT. All the elements are initially set to 0.

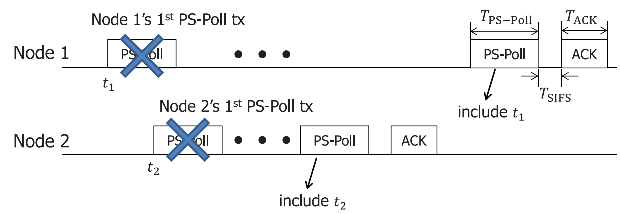


Fig. 6. Operation example of collision and hidden node detection. A PS-Poll collision between nodes 1 and 2 occurrs when they are hidden from each other. In this example, t_1 and t_2 are fed back to the AP through the modified PS-Poll structure. If the second feedback method is assumed, these time information is fed back through control frames in the following data exchange RAW.

Table 3
Example of a hidden node matrix.

	1	2	3	4	5	6
1	0	0	1	0	0	0
2	0	0	1	1	1	1
3	1	1	0	0	0	0
4	0	1	0	0	1	0
5	0	1	0	1	0	0
6	0	1	0	0	0	0

Table 3 shows an example of a hidden node matrix when there are six nodes in a group. In this example, nodes 1 and 3 and nodes 4 and 5 are hidden from each other, respectively, and nodes, 3, 4, 5, and 6 are hidden from node 2.

5.3. Hidden node regrouping

The HMR algorithm uses the hidden node relation matrix to regroup hidden nodes[9] that have been detected during each TBTT. It runs for each group in turn at each TBTT. Initially the HMR chooses a node experiencing the hidden node problem, and moves it into some other group where it does not have the hidden node problem. Each node is allocated for a group randomly according to (1).

Algorithm 1 shows how our proposed hidden node regrouping algorithm works. Our algorithm runs for each group sequentially (line 1). The algorithm initially selects a group and creates the list of nodes for the group (line 2). If there is no node experiencing the hidden node problem or no node left in the list for selection, the algorithm stops (line 3). The AP starts the regrouping algorithm from a node that has the highest number of hidden nodes (line 4). The algorithm tries to put the selected node into a group where it does not have any hidden node.

The AP selects a group m for the selected node (line 5), and checks whether the node has any hidden node in group m (line 6).

[8] The proposed detection method can report false positives or false negatives because of the deep fading or inaccurate RSSI measurement.

[9] The group ID change is supported by the 802.11ah specification, named Dynamic AID Assignment.

Algorithm 1 Hidden node regrouping algorithm.

```
1:  for All groups do
2:    Create the list of nodes for the current group.
3:    while there exists any node suffering the hidden node problem in the list do
4:      Select a node that has the highest number of hidden nodes.
5:      Select a target group, say m, for the selected node.
6:      if group m has no node hidden from the selected node then
7:        Put the selected node into group m and remove the selected node from the list.
8:      else
9:        if group m is the last group then
10:         Remove the selected node from the list.
11:       else
12:         m ← m + 1 modulo N_group and go to line 5.
13:       end if
14:     end if
15:   end while
16: end for
```

If there exists, the AP attempts a next group for the node (line 12). If not, the node will be moved into group m and removed from the list (line 7). If all the groups for the selected node have been tried but there is no group that helps the selected node avoid the hidden node problem, the selected node will be removed from the list and remain in the current group (line 10). This means that it is not possible for this node to avoid the hidden node problem anyway. Then the AP selects a next node for regrouping that has the next highest number of hidden nodes (line 4).

Then the algorithm runs for the next group if there is neither a hidden node nor any node that needs regrouping. After running for all the groups, the algorithm stops and will run again at the next TBTT.

Algorithm 1 is heuristic and the information on hidden node relation is gradually obtained by self-learning procedures. As a result, the number of hidden node pairs can temporarily increase but gradually decreases in the long run because the HMR algorithm always chooses a group with the highest number of hidden nodes. The statistical results of the proposed HMR algorithm is presented in Section 6.

5.4. Discussion: multiple APs case

The proposed HMR algorithm is not limited to the case of a single AP, and can be applied for the case of multiple APs, which is called overlapping BSS (OBSS) in the 802.11ah standard. The algorithm is designed for each AP to work independently in solving the hidden node problem. When neighboring APs use a same channel with no communication link between them, hidden node pairs may co-exist with OBSS nodes, named OBSS hidden nodes. Because OBSS nodes are not able to feed their failed transmission times back to other APs, the AP cannot find hidden node relation between its associated nodes and OBSS nodes. Packet transmission failures due to transmission from OBSS hidden nodes cannot be distinguished from those caused by error-prone channels. As a result, hidden node relations are correctly detected only within a network while the delay increases in the PS-Poll transmissions.

To reduce the increased delay, the AP changes its transmission time slot for each group occasionally. By doing so, the AP can allocate time slots well to achieve a less transmission end time by putting a smaller number of OBSS hidden nodes into each time slot.

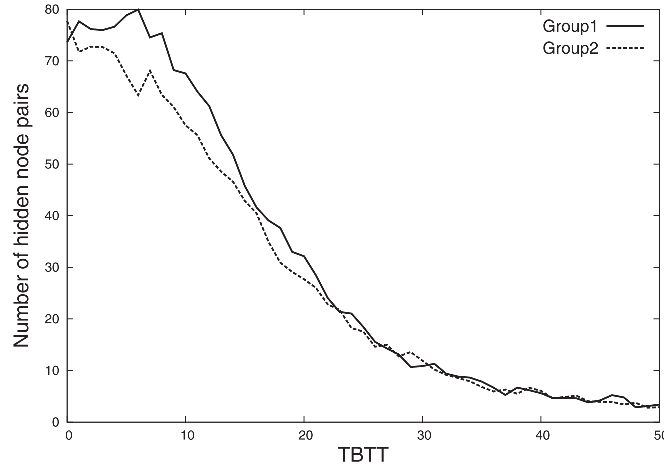


Fig. 7. The number of hidden node pairs in groups 1 and 2 according to TBTT.

Note that, if a neighboring AP uses a different channel, it does not interfere with the other AP at all. Because there are 13 independent channels, each with 2 MHz bandwidth, the neighboring AP can find a different channel easily. In case the neighboring AP uses a same channel, the 802.11ah standard defines sectorization to solve the OBSS problem through a communication link between APs[4]. APs can adjust their time slots to avoid interference or to improve spatial reuse.

6. Simulation results

In this section, we evaluate the performance of the proposed HMR algorithm against the 802.11ah standard, i.e., random grouping, with respect to the number of hidden node pairs, the PS-Poll transmission end time, and the number of retransmissions.

6.1. Simulation settings

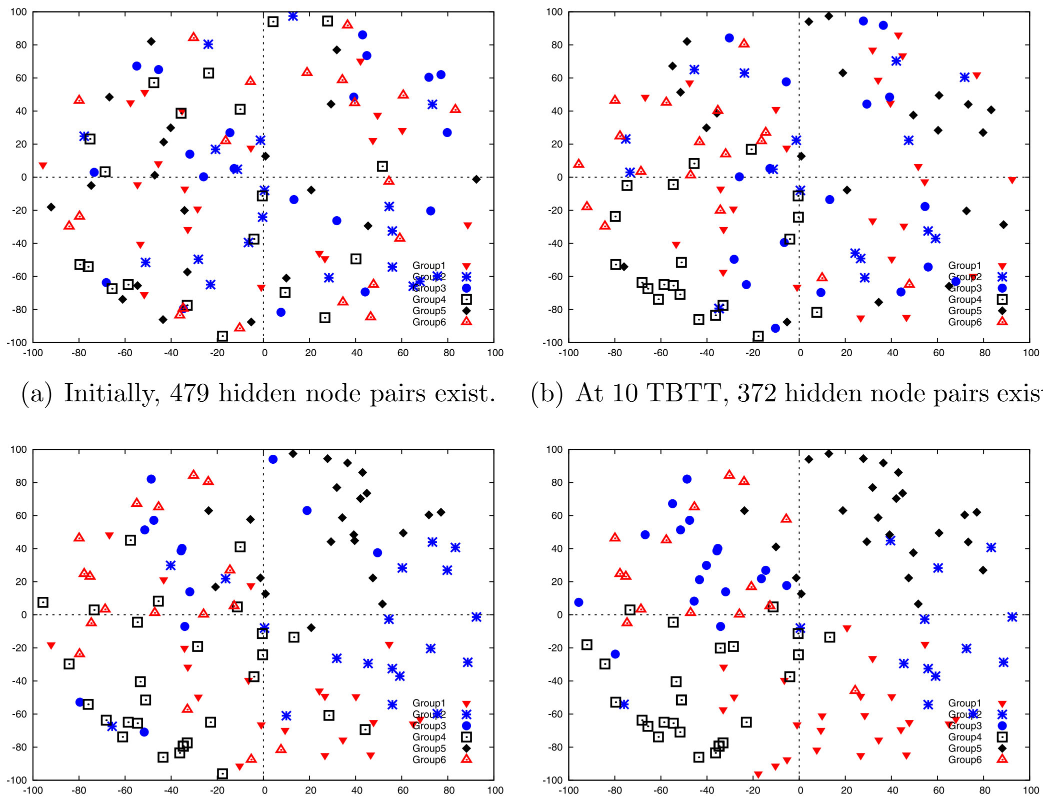
We consider an 802.11ah network with an AP whose transmission range is 1 km. Within the network, 120 nodes are uniformly distributed. The AP has buffered packets for all the nodes. The path loss exponent is set to 3. We assume an ideal channel condition that has neither channel error nor capture effect. All the nodes operate in the power saving mode. Each node transmits a PS-Poll frame in the first RAW, and data transmissions follow in the next RAW of the same TBTT as shown in Fig. 2.

As a modulation and coding scheme (MCS), we use a basic rate with BPSK modulation at a 1/2 coding rate, which transmission rate is 0.65 Mbps when a 2 MHz bandwidth is applied. The system parameters for the 802.11ah standard [4] and other simulation parameters are listed in Table 2. Each simulation runs for 100 beacon intervals (i.e., TBTTs). Since the performance of our proposed HMR algorithm heavily depends on node positions, simulations are performed 100 times with random node distributions, and their results are averaged.

6.2. Simulation results

The performance metric we are interested in is the number of hidden node pairs as the TBTT goes by. Fig. 7 shows the number of hidden node pairs for groups 1 and 2 according to the TBTT. All the six groups show a similar tendency of rapid decrease in the number of hidden node pairs with the TBTT.

The decreasing slope is steep at the early stage, and gradually becomes gentle after 20 TBTTs. At 48 TBTT, the number of hidden node pairs is reduced by more than 95% compared to that of the



(a) Initially, 479 hidden node pairs exist. (b) At 10 TBTT, 372 hidden node pairs exist.

(c) At 20 TBTT, 203 hidden node pairs exist. (d) At 30 TBTT, 35 hidden node pairs exist.

Fig. 8. Example of regrouping results for each group.

initial hidden node pairs, and the reduction rate is marginal afterwards. Sometimes the number of hidden nodes does not decrease according to TBTT, e.g., at 1 TBTT for group 1 and at 7 TBTT for group 2. This is because our HMR algorithm heuristically changes the allocated group one by one without using the whole hidden node pair information. As the algorithm runs, it always shows the same tendency of dramatic reduction in number of hidden node pairs.

Fig. 8 shows an example of regrouping results for each group. The AP is located at (0, 0), and 120 nodes are randomly distributed within the AP's coverage. Each group is marked with a different symbol. Initially, nodes are randomly grouped for six groups resulting in 479 hidden node pairs. After 20 TBTTs, the number of hidden node pairs has been reduced by half, i.e., 203 and nodes that are geographically close tend to be grouped together as shown in Fig. 8(c). After 30 TBTTs, there are only 35 hidden node pairs left.

In terms of the number of hidden node pairs and the PS-Poll end time, we compare our algorithm against the 802.11ah standard algorithm, and present the results in Table 4. While the 802.11ah standard has 478 number of hidden node pairs in total, our proposed algorithm has only 8. This means that our regrouping proposal reduces hidden node pairs by 98% compared to the standard grouping algorithm. Without grouping, the average total number of hidden node pairs reaches 2927.

Owing to a huge reduction of our proposal regarding the number of hidden node pairs, it achieves the PS-Poll end time performance of 178.4 msec while the 802.11ah standard algorithm achieves 566.0 msec, i.e., 68.5% reduction. We also compare the two algorithms in terms of the number of retransmissions. The average numbers of retransmissions in our proposal and the 802.11ah standard algorithm are 15.0 and 56.6, respectively. Our algorithm reduces the number of retransmissions by 73.5% compared to the standard algorithm. Therefore, our algorithm is able to save energy significantly, which is an important factor in sensor network design.

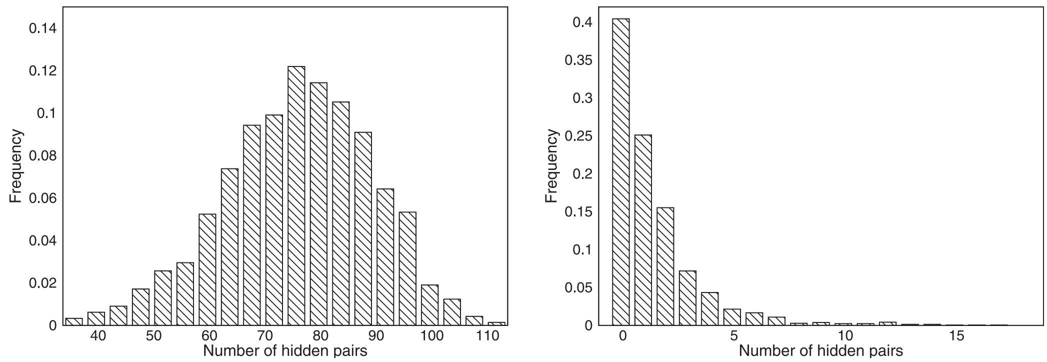
Table 4
Simulation results.

	# of Hidden pairs			Avg. PS-Poll end time (msec)		
	11ah	HMR	HMR w/ O	11ah	HMR	HMR w/ O
Group 1	79.9	1.5	1.9	95.4	41.8	50.3
Group 2	78.7	1.3	1.4	95.5	34.3	40.8
Group 3	80.7	1.1	1.2	91.7	25.5	35.9
Group 4	79.5	1.3	1.1	95.1	29.0	30.9
Group 5	80.1	1.3	1.4	92.7	23.5	31.4
Group 6	79.3	1.3	1.5	95.6	24.0	29.1
Total	478.2	8.2	8.4	566.0	178.4	218.5

The "HMR w/ O" in Table 4 represents that OBSS hidden nodes exist in the network. It is assumed that the OBSS hidden nodes always have packets to transmit, i.e., saturated traffic condition. The results show that the proposed HMR the HMR algorithm works stably in this scenario, i.e., the number of hidden node pairs does not change much. However, the average PS-Poll transmission end time of the HMR with OBSS hidden nodes increases by 22.5% compared to that without OBSS hidden nodes.

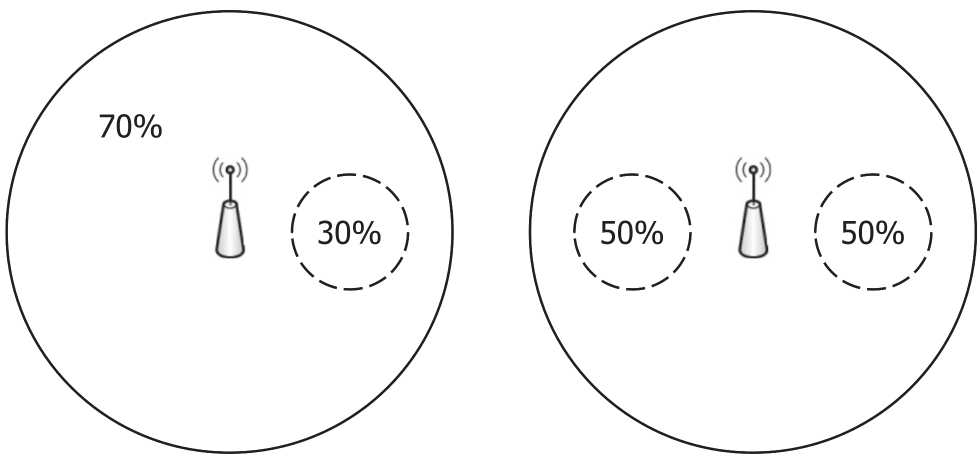
Note that the end time increase is not small because of the saturated traffic condition for the OBSS hidden nodes, i.e., the worst case. Therefore, the PS-Poll end time will be reduced with a more realistic traffic assumption.

Fig. 9 shows the frequency distribution for the number of hidden node pairs per group. We run 350 simulations and the num-



(a) **802.11ah grouping** (b) **Proposed HMR algorithm**

Fig. 9. Frequency distribution for the number of hidden node pairs per group.



(a) **Random and one hot spot topology** (b) **Two hot spots topology**

Fig. 10. Hot spot topologies.

ber of groups is six, resulting in 2100 samples in the graphs. The 802.11ah grouping method shows a Normal distribution for the frequency distribution of hidden node pairs according to the central limit theorem and the average number of hidden node pairs in a group is 78.5. On the other hand, our proposed HMR algorithm shows an exponential distribution. It reveals that about 95% of groups have less than or equal to 5 hidden node pairs. The average and maximum numbers of hidden node pairs are 1.6 and 36, respectively. It is confirmed that our algorithm significantly outperforms the 802.11ah grouping method. To reduce the number of hidden node pairs when it seems high, the AP can re-initiate the HMR algorithm by choosing members of each group randomly.

We simulate two other topologies which are shown in Fig. 10. For the topology shown in Fig. 10(a), 30% of nodes are deployed in the hot spot area and the other 70% are randomly deployed in the network. In the other topology, i.e., Fig. 10(b), half of the nodes are in each hot spot.Table 5 shows the simulation results that are averaged for 100 simulations. For the 802.11ah grouping method, the one hot spot topology shows a similar number of hidden node pairs to that of random deployment, and a two hot spot topology has less hidden node pairs than a one hot spot topology. For the proposed HMR algorithm, the average numbers of hidden node pairs for the both topologies are much smaller, i.e., about 10, than those of the 802.11ah grouping method.

Table 5
Average number of hidden node pairs.

	One hot spot		Two hot spots	
	11ah	HMR	11ah	HMR
Group 1	73.6	3.1	51.5	1.7
Group 2	74.8	1.7	51.5	1.0
Group 3	75.9	1.6	51.0	2.0
Group 4	75.9	1.4	51.0	1.7
Group 5	77.6	1.5	51.0	1.7
Group 6	74.8	2.4	50.6	1.4
Total	452.6	11.6	306.5	9.4

7. Conclusion

The IEEE 802.11ah standard has been proposed to have a much longer transmission range and an enhanced power save mode to service a wide area for sensor networks applications. However, the hidden node problem can be exacerbated especially when many nodes in sleep mode are activated simultaneously. As a solution, the 802.11ah standard proposed a group-based contention algorithm, but it is not a satisfactory solution to the problem.

To evaluate the performance degradation, this paper has analyzed the PS-Poll transmission end time with and without hidden node pairs. Then we have proposed a hidden node matrix based regrouping (HMR) algorithm that consists of time difference-based hidden node detection, hidden matrix generation, and hidden node regrouping. Through extensive simulations, we have confirmed that our proposed HMR algorithm significantly outperforms the 802.11ah standard grouping algorithm. Compared to the standard algorithm, the HMR algorithm reduces the number of hidden node pairs by 98.3% and the PS-Poll transmission end time by 68.5% when the AP covers 120 nodes. The HMR algorithm also performs well with OBSS hidden nodes and hot spot topology.

Acknowledgement

This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. 2015R1A2A2A01008240) and in part by ICT R&D program of MSIP/IITP (B0126-15-1017, Spectrum Sensing and Future Radio Communication Platforms).

Appendix A. Derivation details for $E[W_j^n]$

$E[W_j^n]$ is obtained by taking the minimum of the random variables. Let $W_j^n = \min(W_1, W_2, \cdots, W_n)$. We have $W_j^n > x$ if W_1, W_2, $\cdots$, $W_n > x$. Since W_j has a discrete uniform distribution in (1,C_j) where C_i is the maximum contention window size for backoff stage i, we have

$$P[W_j > x] = \frac{C_j - x}{C_j}.$$

We also have

$$P[W_j^n > x] = \left(\frac{C_j - x}{C_j}\right)^n.$$

Then the cumulative distribution function of $W_j^n > x$ is given by

$$F_{W_j^n}(x) = 1 - \left(\frac{C_j - x}{C_j}\right)^n.$$

The probability mass function of $W_j^n > x$ is

$$\begin{aligned} f_{W_j^n}(x) &= F_{W_j^n}(x) - F_{W_j^n}(x-1) \\ &= \frac{1}{C_j^n}\Big((C_j - x)^n - (C_j - (x+1))^n\Big). \end{aligned} \tag{A.1}$$

Finally, we obtain

$$E[W_j^n] = \sum_{x=0}^{C_j - 1} \frac{x}{C_j^n}\Big((C_j - x)^n - (C_j - (x+1))^n\Big).$$

References

[1] W. Sun, M. Choi, S. Choi, IEEE 802.11ah: a long range 802.11 WLAN at sub 1 GHz, J. ICT Standard. 1 (1) (May 2013) 83–107.
[2] M. Park, et al., Uplink channel access. Doc.: IEEE 802.11-12/0606r1, May 2012. https://mentor.ieee.org/802.11/dcn/12/11-12-0606-01-00ah-uplink-channel-access.pptx.
[3] Y.C. Tseng, S.Y. Ni, E.Y. Shih, Adaptive approaches to relieving broadcast storms in a wireless multi-hop mobile ad hoc network, IEEE Trans. Comput. 52 (5) (May 2003) 545–557.
[4] Apr. 2015, IEEE P802.11ah/D5.0 Draft 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 6: Sub 1 GHz License Exempt Operation.
[5] L. Zheng, M. Ni, L. Cai, J. Pan, C. Ghosh, K. Doppler, Performance analysis of group-synchronized DCF for dense IEEE 802.11 networks, IEEE Trans. Wireless Commun. 13 (11) (Nov. 2014) 6180–6192.
[6] R. Liu, G. Sutton, I. Collings, WLAN power save with offset listen interval for machine-to-machine communications, IEEE Trans. Wireless Commun. 13 (5) (May 2014) 2552–2562.
[7] C. Park, D. Hwang, T.J. Lee, Enhancement of IEEE 802.11ah MAC for m2m communications, IEEE Commun. Lett. 18 (7) (Jul. 2014) 1151–1154.
[8] Y. Hammal, J. Ben-Othman, L. Mokdad, A. Abdelli, Formal modeling and verification of an enhanced variant of the IEEE 802.11 CSMA/CA protocol, J. Commun. Netw. 16 (4) (Aug. 2014) 385–396.
[9] Y.J. Choi, S. Park, S. Bahk, Multichannel random access in OFDMA wireless networks, IEEE J. Sel. Areas Commun. 24 (3) (Mar. 2006) 603–613.
[10] H. Kim, S. Yun, I. Kang, S. Bahk, Resolving 802.11 performance anomalies through qos differentiation, IEEE Commun. Lett. 9 (7) (Jun. 2005) 655–657.
[11] Y. Kim, J. Yu, S. Choi, K. Jang, Novel hidden station detection mechanism in IEEE 802.11 WLAN, IEEE Commun. Lett. 10 (8) (Aug. 2006) 608–610.
[12] K. Nishide, H. Kubo, R. Shinkuma, T. Takahashi, Detecting hidden and exposed terminal problems in densely deployed wireless networks, IEEE Trans. Wireless Commun. 11 (11) (Nov. 2012) 3841–3849.
[13] S. Aust, T. Ito, Sub 1 GHz wireless LAN propagation paht loss models for urban smart grid applications, in: Proceedings of the IEEE ICNC, Hawaii, USA, Jan. 30-Feb. 2, 2012.
[14] Y. Zhou, H. Wang, S. Zheng, Z. Lei, Advances in IEEE 802.11ah standardization for machine-type communications in sub-1GHz WLAN, in: Proceedings of theIEEE ICC, Budapest, Hungary, Jun. 9-13, 2013.
[15] M. Cheong, TGah functional requirements and evaluation methodology, Doc.: IEEE 802.11-11/0905r5, Jan. 2012. http://mentor.ieee.org/802.11/dcn/12/11-11-0905-05-00ah-tgah-functional-requirements-and-evaluation-methodology.doc.
[16] M. Park, Specification framework for TGah. Doc.: IEEE 802.11-11/1137r15, May 2013. http://mentor.ieee.org/802.11/dcn/12/11-11-1137-15-00ah-specification-framework-for-tgah.docx.
[17] R. Porat, et al., US channelization. Doc.: IEEE 802.11-12/0613r0, May 2012. http://mentor.ieee.org/802.11/dcn/12/11-12-0613-00-00ah-us-channelization.pptx.
[18] T. Adame, A. Bel, B. Bellalta, J. Barcelo, M. Oliver, IEEE 802.11ah: the wi-fi approach for m2m communications, IEEE Wireless Commun. 21 (6) (Dec. 2014) 144–152.
[19] G. Bianchi, Performance analysis of the IEEE 802.11 distributed coordination function, IEEE J. Sel. Areas Commun. 18 (3) (Mar. 2000) 535–547.
[20] P. Chatzimisios, A. Boucouvalas, V. Vitsas, IEEE 802.11 packet delay . a finite retry limit analysis, in: Proceedings of the IEEE GLOBECOM, San Francisco, USA, Dec. 1-5, 2003.
[21] M. Carvalho, J. Garcia-Luna-Aceves, Delay analysis of IEEE 802.11 in single-hop networks, in: Proceedings of the IEEE ICNP, Atlanta, Georgia, USA, Nov. 4-7, 2003.
[22] O. Ekici, A. Yongacoglu, IEEE 802.11a throughput performance with hidden nodes, IEEE Commun. Lett. 12 (6) (Jun. 2008) 465–467.
[23] H. Velayos, G. Karlsson, Techniques to reduce the IEEE 802.11b handoff time, in: Proceedings of the IEEE ICC, Paris, France, Jun. 20-24, 2004.
[24] S. Abraham, et al., Short beacon. Doc.: IEEE 802.11-12/0129r3, May 2012. https://mentor.ieee.org/802.11/dcn/12/11-12-0129-03-00ah-short-beacon.pptx.

Sung-Guk Yoon received B.S. and the Ph.D degree from Seoul National University, Seoul Korea, in 2006 and 2012, respectively. From 2012 through 2014, he was with the same university as a post-doctoral researcher. In 2014, he joined the school of electrical engineering at Soongsil University as an assistant professor. His research interests include next generation wireless networks, power line communications, and smart grid.

Jung-O Seo received B.S. and the M.S. degree from Seoul National University, Seoul Korea, in 2012 and 2014, respectively. Since 2014, he has been with TMAX Soft, Gyeonggi-do, Korea. His research interests include Wi-Fi.

Saewoong Bahk is a professor at Seoul National University (SNU). He served as Director for the Institute of New Media and Communications during 2009-2011. Prior to joining SNU, he was with AT&T Bell Laboratories as a member of technical staff from 1991 to 1994 where he had worked on network management.He received KICS Haedong Scholar Award in 2012. He has been leading many industrial projects on 3G/4G/5G and IoT connectivity supported by Samsung Electronics, LG Electronics, SK Telecom, etc and published more than 200 technical papers and holds 72 patents.He was TPC Chair for IEEE VTC-Spring 2014 and general chair of JCCI 2015. He is co-EIC of IEEE/KICS Journal of Communications and Networks (JCN), and was on the editorial board for Computer Networks Journal (COMNET) and IEEE Transactions on Wireless Communications (TWireless). He is an IEEE senior member and a member of Whos Who Professional in Science and Engineering.He received B.S. and M.S. degrees in Electrical Engineering from Seoul National University (SNU) in 1984 and 1986, respectively, and the Ph.D. degree from the University of Pennsylvania in 1991.