

A QoS-aware Scheduling with Node Grouping for IEEE 802.11ah

Nurzaman Ahmed · Md. Iftekhhar Hussain

Received: date / Accepted: date

Abstract The recent IEEE 802.11ah amendment has proven to be suitable for supporting large-scale devices in Internet of Things (IoT). It is essential to provide a minimum level of Quality of Service (QoS) for critical applications such as industrial automaton and healthcare. In this paper, we propose a QoS-aware Medium Access Control (MAC) layer solution to enhance network reliability and reduce critical traffic latency by *an adaptive station grouping* and *a priority traffic scheduling* scheme. The proposed grouping scheme calculates the current traffic load and distributes among different RAW groups considering different requirements of the stations. The RAW scheduling scheme further provides priority slot access using *a novel backoff* scheme. Markov-chain model is developed to study the throughput and latency behaviours for the traffic generated from the critical application. The proposed protocol shows significant delay improvement for priority traffic. The overall throughput performance improves up to 12.7% over the existing RAW grouping scheme.

Keywords Internet of Things · IEEE 802.11ah · Restricted Access Window · Quality of Service

1 Introduction

The deployment of Internet of Things (IoT) involves the installation of a huge number of nodes for allowing seamless communication. The nodes deployed over a large-area are capable of sensing, actuating, or tracking [17]. The IEEE 802.11ah [2] is a recent standard having different new concepts which are suitable for IoT. Features like *improved range* (up to 1Km in a single hop), *low power consumption*, *rich data rate* (up to 78Mbps), *hierarchical addressing* and *grouping of nodes* are wonderfully incorporated in 802.11ah . The new

N.Ahmed, M. I. Hussain
Department of Information Technology, North-Eastern Hill University, Shillong
E-mail: nurzaman@cse.iitkgp.ac.in

hierarchical address used in the scheme is 13 bits long, leading to a maximum number of associated stations to 8191. To reduce collisions among such a large number of stations, 802.11ah proposes a Restricted Access Window (RAW) mechanism.

In IoT, data flows are mainly of three types – *event-driven*, *query-driven*, and *continuous* [13]. Event-driven and query-driven traffic flows are generated only when there is an event or query, respectively. While supporting communication in an IEEE 802.11ah network with 8191 STAs, which is more for multiple Basic Service Sets (BSSs), such traffic diversity is obvious. Furthermore, there are IoT applications such as health-care and fire-alarm, where data are sensitive in terms of reliability and delay, so provisioning QoS is important but challenging. Also, real-time audio or graphics data streaming are delay-sensitive and has a certain requirement of bandwidth. The overall IoT control loop (sensor to the actuator) may take a huge delay with an increasing number of relays. Unfortunately, the design of 802.11ah does not specifically consider these dynamic behaviors.

The RAW mechanism of IEEE 802.11ah divides the time frame and allows a particular group of STAs for transmission. Several issues evolve while dealing with such a massive number of STAs with heterogeneous traffic flow. The grouping of STAs is dependent on the AID, which is assigned according to their association time. After the association, STAs are grouped in a sequence. Therefore, in the case of STAs with higher AID, it may differ a possible transmission. Further, critical and rare traffic demand special processing over continuous data flows to meet different requirements. The current research works are primarily focused on estimating the RAW size based on the number of STAs or traffic loads for a particular group. However, the solutions do not consider traffic types, QoS requirements, and distribution of traffic loads among the groups. In such a case, dynamic and priority-based RAW allocation and adaptive grouping mechanisms can help a network to achieve better performance.

The state-of-the-art solutions such as [20], [18] and [16] achieve QoS up to a certain extent by slot scheduling over the RAW frame. However, the scheduling problem is limited to the size of the RAW frame. There is a need for consistent QoS mechanism concerned by different groups and time frames in network. A QoS-aware priority RAW scheduling and adaptive load aware node grouping scheme, named *QS-MAC*, are proposed. The proposed solution can optimize channel access among different groups in a highly dense network. Contributions of this paper can be summarized as:

- ◆ A QoS-aware RAW slot assignment and fairness method to handle the traffic flow of critical applications. Concerning the criticality of traffic in terms of delay and reliability, the proposed scheduling algorithm provisions priority to the frame belong to the needy applications.
- ◆ A load balancing scheme among different RAW groups using dynamic AID allocation to improve network reliability.
- ◆ A traffic modeling

The rest of the paper is organized into four sections. Background and related works are discussed in Section 2. Section 4 presents the proposed MAC layer solution for IEEE 802.11ah. Section 5 gives the performance evaluation and analysis of the proposed scheme. Finally, Section 6 concludes the paper.

2 Background and Related works

This section discusses the different QoS requirements of various IoT applications and their support in 802.11ah. We discuss some of the related solutions while provisioning different QoS supports. Finally, we provide an analysis of the existing RAW scheme.

2.1 QoS Provisioning in IoT Networks

The QoS is defined as a set of quality criteria for a particular service [4]. QoS in IoT depends on the characteristics of the objects and applications. Different rules/criteria need to set due to their various characteristics. Looking at the traffic in IoT applications, there can be event-driven, query-driven, and continuous flows. Low rate traffic is usually generated by the alarms/commands from the control center to the corresponding authorities/actuators. For example, there may be a massive amount of on-demand reporting type of traffic from thousands of smart-grid STAs [11]. In the power outage situation, a vast number of STA simultaneously tries to report the failure before the battery dies [9]. Prior scheduling is essential for improving reliability for such event-driven traffic. Also, many critical applications need to be processed within a given time-bound. Expecting the behavior of event-driven and query-driven traffic, scheduling MAC protocol can improve the performance up to a great extent. Further, a dynamic node grouping scheme can help to utilize network resources efficiently. The MAC solution for 802.11ah does not consider these requirements.

2.2 Related Works

The current research works are primarily focusing on estimating RAW size in different network conditions. For example, Park *et al.* [14] estimated the number of uplink STAs to determine the size of RAW. Zhao *et al.* [24] optimized the performance of the RAW-mechanism in terms of power consumption, and showed that energy efficiency in the sensor nodes improves with the increasing number of RAW groups. In [6], based on the delay requirements of STAs, RAW is divided and allocated. Tian *et al.* [20] suggested an optimization algorithm to judiciously define the grouping parameters based on real-time traffic and improved network efficiency. Šljivo *et al.* [18] proposed an immediate reply scheme for 802.11ah AP to reduce downlink latency.

Table 1: Some of the important notations used in this paper

Notations	Description
W_i	Backoff window size at stage i
$f(c, i, j)$	Probability density function distributed over backoff stage i and backoff window j for class c
Γ_{G_i}	Ease of transmission for group G_i
φ_c	A real number which gives the increase and decrease of the distribution for class c
$f(d)$	Probability density function for an event d
μ^d	Expected rate of occurrence of an event d
τ_c	Probability of successful transmission of a station from class c
$P_{c,i,k}$	State transition probability of (i,k) for class c
$E(D_c)$	Average saturation delay for class c in a group
Th_c	Average saturation throughput for class c in a group

The above mechanisms improve RAW performance to achieve a better result in terms of success rate. Tian *et al.* [20] and Šljivo *et al.* [18] improve delay performance considering real-time nature of traffic. Seferagic *et al.* [16] enhances QoS by considering the overall control loop frame. It considers delay and reliability for sensitive and periodic traffic over the RAW frame. Further, Kia *et al.* [10] improves energy-efficiency with the help of a load-aware grouping scheme without considering the QoS requirements of an application. However, for real IoT scenarios, traffic classification, and priority scheduling for delay-sensitive applications according to their requirements are needed to be considered. Furthermore, load balancing among the complete network is essential to utilize the available bandwidth optimally. The analysis of state-of-art makes out a research space on provisioning QoS for real-time priority traffic over limited channel bandwidth in 802.11ah. To address this issue, we propose RAW scheduling and load-aware node grouping scheme for 802.11ah called, *QS-MAC*.

2.3 Problem Formulation

The AP node in IEEE 802.11ah transmits beacon, which also broadcast RAW Parameter Set (RPS) information in the preceding. In the interval, there may be one or more number of RAW for a group of STAs. The STAs belonging to a RAW group are allowed for contention within the assigned slot duration. To get a slot from RAW is decided based on a mapping function, as shown in Eq. (1),

$$x = (i + \text{offset}) \% S_{\text{RAW}} \quad (1)$$

where, x is the slot number in a RAW frame of size S_{RAW} , the offset value is for improving fairness among the STAs in a RAW, and i is the position index or AID of the STA. If STA is already paged, it uses AID. Otherwise, the

position index is used. If the RAW is restricted to STAs with AID bits in the TIM element set to 1. The slot duration (T_x) is calculated from slot duration count (S_c) specified in RPS as:

$$T_x = 500\mu s + S_c \times 120\mu s$$

where, S_c depends on the value of k ($S_c = 2^k - 1$), which is the number of bits in sub-filed. If the slot format field is set to 0, $k = 11$ otherwise for 1, $k = 8$.

- (i) For $k = 11$, the minimum value of $T_x = 500\mu s$ when $S_c = 0$, whereas the maximum of T_x is $500\mu s + (2^{11} - 1) \cdot 120\mu s = 246.14ms$
- (ii) For $k = 8$, the maximum value of T_x is 31.1ms.

Again, size of S_{RAW} is calculated as 2^{14-y} from a 14 bit field, hence the maximum number of slots in a RAW is 8 for $k = 11$, and 64 for $k = 8$. In this way, group size will be $T_{RAW} = T_x \times S_{RAW}$, and maximum time frame size of group for $k = 11$ is 1.96s and 1.99s for $k = 8$. The complete time frame (T_{FRAME}) for a network with all associated STAs can be calculated as:

$$T_{FRAME} = N_{RAW} \times T_{RAW}$$

where N_{RAW} is the total number of RAW groups in the network. If we consider $N_{RAW} = 10$, a STA may need to wait for $T_{wait} = 19.9 - T_x$ Second for its transmission. However, such delay is not tolerable for critical and rare traffic flows. Once STA finds its RAW slot, it uses a DCF based contention mechanism. The second approach needs T_{ac} access time to get its slot for transmission. Access delay in the DCF scheme depends on the probability of successful transmission in a slot. An increase in the number of groups can lead to the reduction of collision. Hence throughput improves beyond a threshold number of groups. Again, the latency of a STA will be very high if the number of groups is high, as it may need to wait for T_{wait} time until assigned RAW and slot. In such a case, the primary concern in a network is to find the optimal RAW and group size based on throughput, delay, and energy consumption. Also, for uplink traffic, the main issue in a large-scale network is to reduce collisions due to contention.

Further, event-driven traffic is generated randomly, which can create congestion in the network. This is considered as a bottleneck of the existing RAW mechanism. For any event, if a STA is not in the current RAW, it is not allowed to transmit. Although, predicting the patterns of such traffic is challenging, but a suitable priority scheduling and adaptive grouping scheme may solve the problem up to a certain extent. As shown in Fig. 1, although an event has occurred in Group 5 for a STA, however it cannot transmit because Group 1 is currently transmitting.

3 The Proposed Protocol

The proposed protocol provides QoS in terms of delay and reliability in large-scale 802.11ah network. Considering the requirements of critical IoT applications, the proposed scheme dynamically schedules and regroups STAs. It works

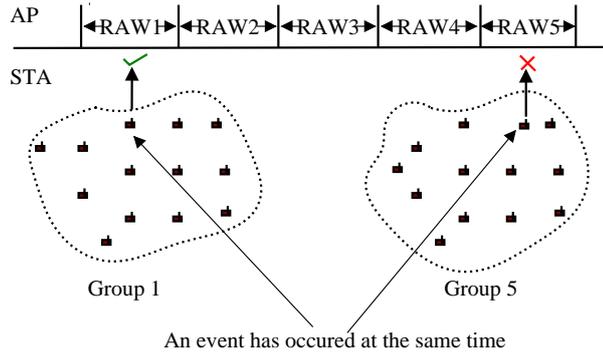


Fig. 1: An illustration of RAW operation for real-time application

primarily in two phases: i) *Priority slot scheduling within a RAW group*, and ii) *Adaptive grouping within the network*.

3.1 Network Model

The 802.11ah technology operates in sub-1GHz channel band which extends connectivity over distances upto 1km in single hop. Relay node support may increase this distance further. The 802.11ah network can be represented as $N(A, S)$ is a combination of an AP (A) node and a set of groups $S = \{G_1, G_2, \dots, G_{N_{RAW}}\}$, where $G_i = \{S_1, S_2, \dots, S_R\}$ is a set of STAs. For a network with n associated STAs, the total number of groups will be $\lceil n/R \rceil$. These STAs are categorized into C priority classes, $c \in [0, C - 1]$ where, $c_i > c_{i+1}$, based on different QoS performance metrics used in IoT (discussed in Sec. 2.1). Here, a higher value of c indicates higher priority. Each of the priority classes are assigned with a contention window W with successful transmission probability p . The AP node is responsible for assigning a lower W to higher priority class to ensure better probability of successful transmission. For example, a critical traffic may acquire an initial value from a list of c . Due to the heterogeneity and QoS restrictions, there are different loads $\{L_1, L_2, L_3, \dots, L_{N_{RAW}}\}$ and $L_{G_i} \neq L_{G_j}$. Hence, AP adaptively balances loads and it increases the number of groups for additional AID or swaps an unused AID j with i from an existing group, if needed. The proposed scheme take the priority value of traffic as input to schedule it accordingly. For example, considering bit value '1' for QoS and '0' for non-QoS traffic, classification can done with the QoS restriction such as latency-sensitive and packet-loss-sensitive. The possible traffic classification can be:

1. QoS Traffic: $C_1 \in \{1, 1\}, C_2 \in \{1, 0\}, C_3 \in \{0, 1\}$
2. Non-QoS Traffic: $C_4 \in \{0, 0\}$

The binary and decimal representation of the classified traffic can be seen in Table 2.

Table 2: Binary and decimal representation of traffic classes

Class	$\alpha\beta$	Value
1	11	3
2	10	2
3	01	1
4	00	0

4 The Proposed Protocol

The proposed protocol provides QoS in terms of delay and reliability in large-scale IEEE 802.11ah network. Considering the requirements of critical IoT applications, the proposed scheme dynamically schedules and regroups stations. It works primarily in two phases: i) *Priority slot scheduling within a RAW group*, and ii) *Adaptive grouping within the network*.

4.1 Network Model

The 802.11ah technology operates in sub-1GHz band which extends connectivity over distances upto 1Km in a hop. Relay node support may increase this distance further. A possible IoT architecture using IEEE 802.11ah is shown Fig. 2. The 802.11ah network can be represented as $N(A, S)$ is a combination of an AP (A) node and a set of groups $S = \{G_1, G_2, \dots, G_{N_{RAW}}\}$, where $G_i = \{S_1, S_2, \dots, S_R\}$ is a set of stations. For n associate stations, total number of groups will be $\lceil n/R \rceil$. These STAs are categorized into θ priority classes, $c \in [0, C - 1]$ where, $c_i > c_{i+1}$, based on different QoS performance metrics used in IoT (discussed in Sec. 2.1). Each of the priority classes are assigned with a contention window W of probability p . The AP node is responsible for assigning a lower W to higher priority class to ensure better probability of successful transmission. For example, a critical type of class may acquire an initial value from a list of C . Due to the heterogeneity and QoS restrictions, there are different loads $\{L_1, L_2, L_3, \dots, L_{N_{RAW}}\}$ and $L_{G_i} \neq L_{G_j}$. Hence, AP adaptively balance loads and if required, it increases the number of groups for additional AID or swap an unused AID j with i from an existing group. Considering $\alpha \leftarrow$ Latency and $\beta \leftarrow$ Reliability as the primary QoS requirements, traffic are classified as follows:

1. QoS Traffic: $C_1 \in \{\alpha, \beta\}, C_2 \in \{\alpha, -\}, C_3 \in \{-, \beta\}$
2. Non-QoS Traffic: $C_4 \in \{-, -\}$

The binary and decimal representation of the classified traffic can be seen in Table 3.

4.2 A QoS-aware RAW Scheduling Scheme

A slot scheduling scheme is run for uplink and downlink traffic from STA and AP, respectively. The scheme starts with Target Beacon Transmission

Class	$\alpha\beta$	Value
1	11	3
2	10	2
3	01	1
4	00	0

Table 3: Binary and decimal representation of traffic classes

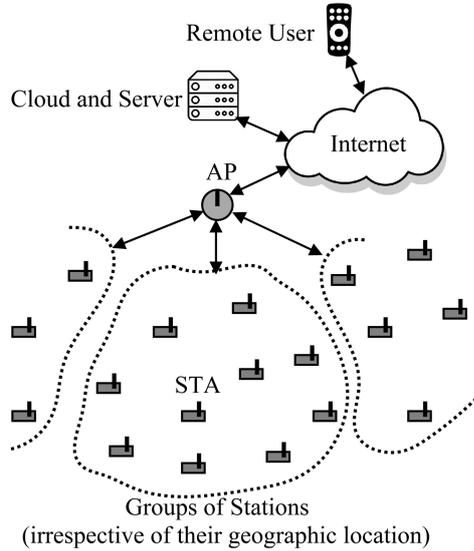


Fig. 2: IEEE 802.11ah-based IoT network architecture

Time (TBTT) carrying RPS information from AP. Although most of the IoT applications periodically generates traffic, however, other types of traffic, as discussed above, may be present in a single network. The AP collects transmission behavior from the different received information and predict their next transmission time. We identify the traffic in the network mostly into two types – periodic and non-periodic, based on their transmission behavior. The past information is stored in a database to predict the expected next packet arriving. AP can easily find the periodic traffic by monitoring the initial transmissions from a STA. However, finding the next transmission for non-periodic STA is challenging. To find the next transmission for non-periodic (i.e., event-based) traffic, poisson distribution is commonly used, where probability mass function $f(d)$ can be calculated as [3]:

$$f(d) = \frac{\mu^d e^{-\mu}}{d!}$$

Here, $f(d)$ is the probability for d events that occur in a time-bound, and μ is the expected rate of occurrence of an event. Probability of several packets

arrived in range (between a and b , where $a < b$) is given as:

$$f(a \leq d \leq b) = \sum_{t=a}^b f(t); t \in I \quad (2)$$

where I is the average service interval of traffic. The value of I is dependent on the requirements of IoT applications. We store flow rates associated with different links from STAs in an event database. A prediction model decides the expected service interval using Eq. (2). If a STA is ready to transmit, the AP node finds an available slot over the RAW frame. The AP node checks priority class C_j , and their priority value j for scheduling. Meanwhile, the proposed protocol ensures an earlier slot for low priority STAs if higher priority STAs are currently not requesting. Then, a smaller backoff window is assigned to a higher priority STA so that it can win an earlier slot. Instead of uniform random backoff time selection, we use a non-uniform random backoff time selection using a truncated distribution defined in $[0, W_i-1]$ where i is the actual backoff stage. Most of the existing mechanisms use continuous distribution to model the backoff process. But, due to the discrete nature of slots, we need a discrete backoff distribution technique for different classes. Therefore, we use a random time selection scheme that utilizes all the slots at any backoff stage. For any backoff stage i , window size, W_i is calculated as [7]:

$$W_i = \begin{cases} 2^i W_0 & \text{for } i \in [0, m' - 1] \\ 2^{m'} W_0 & \text{for } i \in [m', m] \end{cases}$$

where m is the maximum retry limits and m' is the backoff stage, after which the window size remains constant. The probability density function, $f(x)$ of the mentioned geometric distribution can be calculated as [12]:

$$f(c, i, j) = \varphi_c^j \frac{1 - \varphi_c}{1 - \varphi_c^{W_i}} \quad (3)$$

where, $i \in [0, m]$, $j \in [0, W_i - 1]$, $\tau \in [0, C - 1]$ and $\varphi_c \in \mathbb{R}$ gives the increase and decrease of the distribution for a given class c . The variation of $f(c, i, j)$ gives different W value which further defines the priority of class c . For example, if $\varphi_{c_1} < \varphi_{c_2}$ then priority of class c_1 is higher than c_2 . Algorithm 1 describes the proposed traffic classification and priority scheduling scheme. Line #5-14 classifies the traffic into different categories. Once the STAs are classified, RAW scheduling is carried out in line #15-22.

We provide fairness among critical traffic belongs to the STA, which has failed in recent attempts; the proposed scheme keeps track of transmission count T_c . On every failure, this count value is incremented by 1. Higher the value of T_c , higher is the chance of getting a slot in the near future. Algorithm 2 describes a QoS class promotion scheme. A STA with higher transmission count is promoted to higher priority class if the current RAW is not saturated, as discussed from line #2-6.

Algorithm 1 Priority-based RAW scheduling scheme**Initialize:** uRPS \leftarrow modified RPS, $\alpha \leftarrow$ Latency, $\beta \leftarrow$ Reliability

```

1: while TBTT do
2:   Transmit BEACON (uRPS)
3:   Collect and update information on past transmission
4:   Predict STAs time interval and next transmission time
5:   Classify STAs into QoS (Q) and non-QoS (nQ)
6:   if Q then
7:      $c_0 \in \{\alpha, \beta\}, c_1 \in \{\alpha, -\}, c_2 \in \{-, \beta\}$  ▷ QoS STAs
8:   else
9:      $c_3 \in \{-, -\}$  ▷ Non-QoS STAs
10:  Distribute backoff window into using truncated probability distribution,  $f(c)$  over
    class c ▷ Eq. (3)
11:  if  $f(c_1) > f(c_2)$  then
12:     $c_1$  has more chance to win an earlier slot than  $c_2$ 
13:  else
14:     $c_2$  has more chance to win an earlier slot than  $c_1$ 
15:  for  $i = 0 \ \& \ i < 4$  do ▷ Allocation of backoff window sizes for different traffic classes
16:    Allocate  $W_i \rightarrow c_i$ 
17:     $i = i + 1$ 
18:  if Ready to transmit then
19:    Calculate  $T_{FRAME}, N_{RAW}, T_{RAW}$ 
20:    Allow  $G_i$  to transmit in  $T_{RAW}$ 
21:  else
22:    Go to Step (1)

```

Algorithm 2 Fairness for Critical Applications**Initialize:** $W_s^c \leftarrow$ average window size for class C, S \leftarrow a set of STAs, $T_c \leftarrow$ is the failure count

```

1: if Ready to transmit then
2:   for (S  $\in C_i$ ) do
3:     for Every failure do
4:        $T_c = T_c + 1$  ▷ Increases failure counts
5:       if ( $W_a(C_{i-1}) < W_m^{i-1}$ ) then
6:         Promote S  $\rightarrow C_{i-1}$  with highest  $T_c$  ▷ QoS class promotion for STA S
7:   else
8:     Wait for the next beacon

```

4.3 Adaptive Load-aware Grouping

The proposed grouping algorithm starts whenever any request arrives from the previous phase of the protocol. Meanwhile, the AP periodically collects all the load information of RAWs for both types of traffic. For all received frames, the AP node keeps records of the window (W) and timestamp (t_s), which they have piggybacked. A circular queue is used to store this information for every node in a group. Whenever the queue becomes full, the oldest entry is automatically removed to allow the latest frame to enter. For maximum permissible window sizes $W_m = \{W_1, W_2, W_3, W_4\}$, AP finds a set of X , where X_m is the number of counts for same type of entries in the queue. Enhancing PigWin [8], the ease of transmission (Γ), which is the reciprocal of the average W size. The Γ is

calculated as:

$$\Gamma_{G_i} = \frac{\sum_{m=1}^4 X_m}{\sum_{m=1}^4 W_m X_m}$$

The value of a large and small Γ indicates that the STAs are currently using a small and large W , respectively. Consequently, the transmission failure is low and high for a large and small W , respectively. Therefore, we define the load in the network is the reciprocal of Γ . This can further explain the difficulty of transmission. Hence, the average W size, considering the recent past communication, is vital for understanding the current network load.

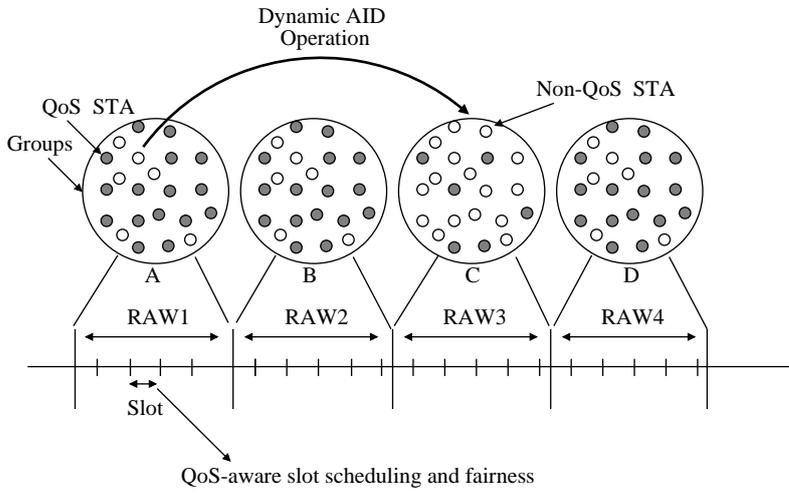


Fig. 3: An example scenario of the proposed scheme

The theoretical maximum value of the load is W_m , whereas, the minimum permission value of it is W_0 if the frames are being transmitted successfully. Similarly, loads of other groups are also monitored. However, according to the current network condition, a threshold value ($\Gamma_{G_i}^t$) is decided. For example, value of $\Gamma_{G_i}^t$ is calculated considering all the saturated window sizes received only once, i.e., $X = \{1, 1, 1, 1\}$. So, $\Gamma_{G_i}^t$ is calculated with $C = \{1, 2, 3, 4\}$ and $W_m = \{16, 64, 256, 1024\}$ as:

$$\Gamma_{G_i}^t = \frac{1 + 1 + 1 + 1}{16 \times 1 + 64 \times 1 + 256 \times 1 + 1024 \times 1} = \frac{4}{560} = 0.00294$$

- ◆ Case 1: $X = \{1, 1, 2, 3\}$, $\Gamma_{G_i}^1 = 0.0019$; $\Gamma_{G_i}^1 < \Gamma_{G_i}^t$ hence, group is heavily loaded
- ◆ Case 2: $X = \{1, 1, 0, 0\}$, $\Gamma_{G_i}^2 = 0.025$; $\Gamma_{G_i}^2 > \Gamma_{G_i}^t$ hence, group is lightly loaded

Algorithm 3 Adaptive load balancing for different groups**Initialize:** Associated STAs are grouped as: $G_1, G_2 \dots G_N$

```

1: for  $G = 1$  to  $N$  do
2:   Calculate ease of load  $\Gamma_{G_i}^C$  ▷ For class C in group  $G_i$ 
3: for  $C = 4$  to  $1$  do
4:   if  $\Gamma_i^C < \Gamma_t^C$  then ▷  $\Gamma_t^C$  is the threshold value for ease of load
5:     Find  $G_j^C$  with highest  $\Gamma_j^C$ 
6:     if  $\Gamma_j^C$  then
7:       Swap ( $ID_{STA}(L_i^C), ID_{STA}(L_j^C)$ ) ▷ Exchanging AID values with a STA from low
load group
8:     else
9:       Create new group  $G_k$ 
10:  else
11:    Goto Step (3)
12:  if Transmission failed then
13:    Goto Step (3)
14:  else
15:    Initiate priority slot scheduling ▷ As described in Algorithm 1

```

In general, dynamic AID allocation is initiated by a non-AP node by sending an AID switch request. To enable this, STAs are programmed with *dot11DynamicAIDActivated* equal to true. The proposed adaptive grouping scheme is discussed in Algorithm 3. In our scheme, an AID switch frame is sent to a particular STA from a heavily loaded group with a new, unused, or AID from the lesser loaded group. Most of the major operations are carried out by the AP node, which further reduces the load of resource constraint STAs. Fig. 3 shows an example scenario of the proposed QoS scheme. RAW1, RAW2, RAW3, and RAW4 are allocated for groups– A, B, C, and D, respectively. We consider the CSB type of slots. As discussed, the QoS aware scheduling happens within a slot. However, as group A has more number of QoS STAs, there is a chance of congestion in the slot. In such a case, a non-QoS STA may need to follow different AID, preferably from a group with a lesser number of QoS-STAs. This allows the scheme to automatically reduce the load in group A by transferring the STA to group C.

5 Performance Evaluation

The performance of the proposed priority RAW and grouping scheme is measured using theoretical and simulation analysis. We compare results with the traditional schemes considering both QoS and non-QoS traffic. The existing solutions optimise RAW size based on delay and energy consumption [21], reduce association delay [19], and control loop latency [16]. To the best of our knowledge, this work comes as the first for proposing a traffic class-based QoS with dynamic grouping in 802.11ah. Therefore, the proposed protocol is compared with traditional 802.11ah.

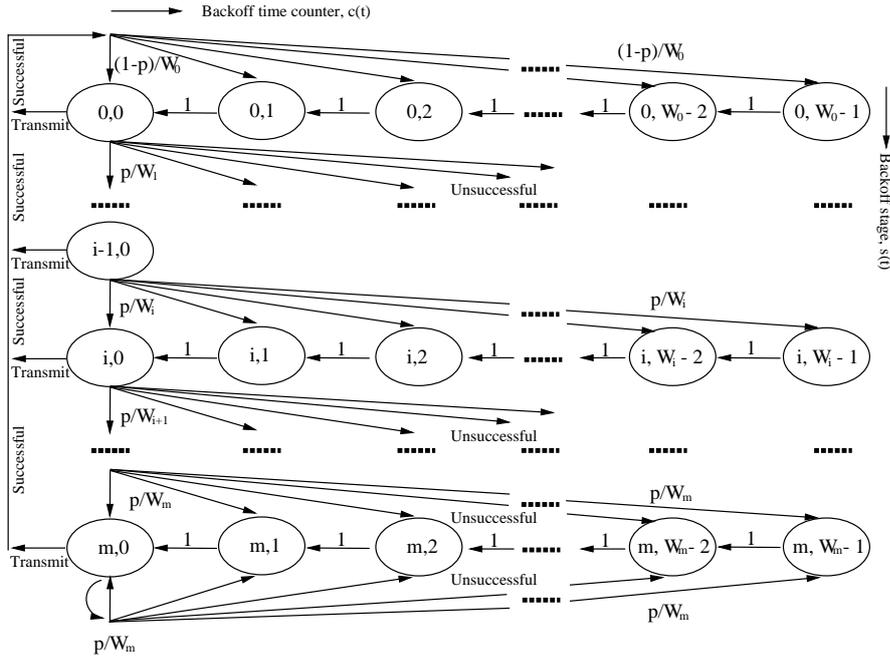


Fig. 4: Diagrammatic representation of 2D Markov Chain Model

5.1 Analytical Model

The 2-Dimensional (2D) discrete Markov Chain model (proposed by Bianchi [5]) is used to analyze the performance of the proposed protocol. The probability of transition to a next state is only dependent on the current state without any regard for the past states. The probabilistic picture for a possible transition is dependent on two stochastic functions: $s(t)$ and $c(t)$ at time slot t . The 2D-Markov chain model as depicted in Fig. 4.

5.1.1 Probability of successful transmission for priority classes

The network with n STAs are divided into C classes with class $c \in [0, C - 1]$ having n_g STAs in each group. A random 2D process $(s(t), c(t))$ where $s(t)$ is the backoff stages in $[0, m]$ and $c(t)$ is the backoff counter in $[0, W - 1]$ is represented by (i, k) . The state transition probability of (i, k) for class c is denoted as $P_{c,i,k}$. Different backoff transition probabilities can be described as:

- ◆ From state $(i, k+1)$, if channel is found to be idle, k is decremented by 1 and moves to state (i, k)

$$P\{i, k|i, k+1\} = 1; k \in [0, W_i - 2], i \in [0, m]$$

- ◆ At state (i,k) , if the channel is busy, backoff counter stays at the same state with probability, p_c

$$P\{i, k|i, k\} = p_c; k \in [1, W_i - 1], i \in [0, m]$$

- ◆ If the transmission is successful, initialize $i=0$ to the minimum contention window W_0 with probability, $1 - p_c$

$$P\{0, k|i, 0\} = (1 - p_c)f(c, 0, k); k \in [0, W_i - 1], i \in [0, 2]$$

- ◆ After an unsuccessful transmission attempts at stage $i-1$, the STA selects its backoff counter value from the range $(0, W_i - 1)$ in the next stage i with the probability p_c

$$P\{i, k|i - 1, 0\} = p_c f(c, 0, k); k \in [0, W_i - 1], i \in [1, m]$$

- ◆ After being unsuccessful is transmission attempts at all the stages, the STA reaches the last stage m , and it remains in the same stage with the probability p until the transmission succeeds.

$$P\{m, k|m, 0\} = p_c f(c, 0, k), k \in (0, W_m - 1)$$

The steady distribution of the Markov Chain model can be calculated as:

$$b_{c,i,k} = \lim_{i \rightarrow \infty} P_c(s(t) = i, c(t) = k)$$

For class c , $b_{c,i,0} = b_{c,i-1,0}; i \in [1, m] \implies b_{c,i,0} = p_c^i b_{c,0,0}; i \in [0, m]$ and $b_{c,0,0} = (1 - p_c) \sum_{i=0}^{m-1} b_{c,i,0} + b_{c,m,0}$. Finally, once STA's backoff counter reaches zero ($k=0$), probability τ_c that STA can transmit in the randomly chosen slot can be calculated as:

$$\tau_c = \sum_{i=0}^m b_{c,i,0} = b_{c,0,0} \frac{1 - p_c^{m+1}}{1 - p_c}; i \in [0, m] \quad (4)$$

5.1.2 Throughput Analysis

Let $n \subset N$ be the total number STA of class c , so during allocated RAW slot duration, the following events may occur [23]:

1. Channel may be idle, so probability that no STA is currently contending for a RAW slot is given by

$$P_{idl} = (1 - \tau_c)^n$$

2. Probability that at least one STA gets access to a slot for transmission

$$P_{txn} = 1 - (1 - \tau_c)^n$$

3. Probability that exactly one STA gets access and transmits successfully can be written as

$$P_{\text{suc}} = \frac{n\tau_c(1 - \tau_c)^{n-1}}{P_{\text{txn}}}$$

4. If more than one STAs try to transmit in a single slot at the same time, collision occurs with probability,

$$\begin{aligned} P_{\text{col}} &= 1 - (P_{\text{idl}} + P_{\text{txn}} + P_{\text{suc}}) \\ &= P_{\text{txn}}(1 - P_{\text{suc}}) \end{aligned}$$

Throughput (T_{h_c}) is defined as the successful transmission of a frame bits for a given time for class c ,

$$\begin{aligned} T_{\text{h}_c} &= \frac{E[\text{Payload size transmitted in a slot}]}{E[\text{Length of slot time}]} \\ &= \frac{P_{\text{txn}}P_{\text{suc}}E[\text{Payload}]}{(1 - P_{\text{txn}})t_{\text{slot}} + P_{\text{txn}}P_{\text{suc}}T_{\text{suc}} + P_{\text{col}}T_{\text{col}}} \end{aligned}$$

where, T_{suc} is the busy time for successful transmission, t_{slot} is the average duration of a slot, and T_{col} is the busy time when a collision occurs. In 802.11ah, it can be calculated as below:

$$\begin{aligned} T_{\text{suc}} &= T_{\text{FH}} + T_{\text{DATA}} + T_{\text{SIFS}} + T_{\text{P}} + T_{\text{ACK}} + T_{\text{DIFS}} + T_{\text{P}} \\ T_{\text{col}} &= T_{\text{FH}} + T_{\text{DATA}} + T_{\text{DIFS}} + T_{\text{P}} \end{aligned}$$

Where, $T_{\text{FH}} = T_{\text{PHY}} + T_{\text{MAC}}$ is the frame header duration, and T_{DATA} , T_{SIFS} , T_{P} , T_{ACK} , and T_{DIFS} are the data, SIFS, propagation, ACK and DIFS duration respectively. Again, the system throughput (T_{h}) is the summation of throughput for all classes of STAs, i.e.,

$$T_{\text{h}} = \sum_{c=0}^{C-1} T_{\text{h}_c}; c \in [0, C - 1]$$

5.1.3 Delay Analysis

We calculate the average delay as discussed in [22]. If $E(D_b)$, $E(D_s)$, and $E(D_r)$ are the delays due to backoff, number slots before the backoff counter freezes, and retries respectively, then the average delay for class c can be calculated as:

$$E(D_c) = E(D_b)\sigma + E(D_s)T_1 + E(D_r)T_2 + T_{\text{suc}}$$

where, $T_2 = T_{\text{col}} + T_{\text{DIFS}} + T_{\text{ACKtimeout}}$, $T_{\text{ACKtimeout}}$ is the ACK time out, and $T_1 = P_{\text{suc}}T_{\text{suc}} + P_{\text{col}}T_{\text{col}}$. So, the average delay including all classes is

$$E(D) = \frac{1}{C} \sum_{c=0}^{C-1} E(D_c); c \in [0, C - 1]$$

Table 4: Saturation throughput: analytical vs. simulation

Traffic Class	Protocol	A/S	$N_{C_1}=100$	$N_{C_1}=100$	$N_{C_1}=150$	$N_{C_1}=250$
			$N_{C_2}=100$ (Kbps)	$N_{C_2}=150$ (Kbps)	$N_{C_2}=200$ (Kbps)	$N_{C_2}=300$ (Kbps)
C1	QS-MAC	A	104.9	111.7	116.3	121.4
		S	101.4	107.2	113.4	120.5
C2	QS-MAC	A	102.3	100.1	97.1	100.7
		S	91.0	93.2	94.9	92.7
Default	802.11ah	A	60.3	63.9	66.1	70.9
		S	56.0	58.2	59.8	61.8

Table 5: Saturation delay: analytical vs. simulation

Traffic Class	Protocol	A/S	$N_{C_1}=100$	$N_{C_1}=100$	$N_{C_1}=150$	$N_{C_1}=250$
			$N_{C_2}=100$ (Sec.)	$N_{C_2}=150$ (Sec.)	$N_{C_2}=200$ (Sec.)	$N_{C_2}=300$ (Sec.)
C1	QS-MAC	A	1.902	1.954	2.077	2.167
		S	1.941	2.105	2.167	2.311
C2	QS-MAC	A	3.101	3.201	3.658	3.814
		S	3.112	3.402	3.756	3.987
Default	802.11ah	A	9.081	11.102	12.982	14.088
		S	9.120	11.201	13.321	14.568

5.2 Analytical Results and Validation

We analyze the RAW performance considering traffic classes- C_1 , C_2 , C_3 , and C_4 , where priority of $C_i > C_{i+1}$; $1 \leq i \leq 4$. We compare the results of the proposed QoS-aware MAC protocol (*QS-MAC*) with the DCF mechanism of 802.11ah. Further, a simulation analysis is carried out in the same environment. The values of different parameters are mentioned in Table 6. The duration of the data frame for MCSs used in 802.11ah can be calculated by the equation mentioned in [15] as:

$$T_{\text{DATA}} = \left\lceil \frac{8 \times (L + m_h)}{\frac{R}{D_r} \times L^{D_{rs}}} \right\rceil \times T_{\text{sym}} + T_{\text{PHY}} \quad (5)$$

where, m_h , R , D_r , $L^{D_{rs}}$, T_{sym} and T_{PHY} are MAC header size, basic data rate, number of bits in one OFDM, symbol duration of OFDM, and PHY header size respectively. For 256 bytes (MCS0, 2MHz), $T_{\text{DATA}}=4.56$ ms. To see the saturation throughput in a condition where the number of STAs from different classes is not the same (among 1000 STAs), we use tabular data representation. We consider 4, 6, and 10 as the maximum backoff stages for C_1 , C_2 , and C_4 classes respectively. Otherwise, a graphical representation is used for common case scenarios. Initially, we analyze the throughput and delay performance of the proposed scheme in analytical (A) and simulation (S) environments and compare the results with traditional 802.11ah. Considering the theoretical analysis from Sec. 5.1.2, the calculated saturation throughput can be seen

as presented in Table 4. Throughput achieved by the QoS scheme is much higher than the conventional DCF used in 802.11ah. The required bandwidth is optimally utilized for more reliability and low latency in priority traffic. Further, a saturation delay is calculated from both types of traffic, which can be seen in Table 5. The saturation delay is greatly improved in the proposed protocol. About 700% and 500% (on average) lesser latency can be noticed in C1 and C2 respectively as compared to the default DCF scheme.

Table 6: Default parameters used for simulation and analysis

Parameters	Value
Bandwidth	2MHz (MCS0)
Data rate (D_r)	650 Kbps
Payload size (L)	256 bytes
CWmin/ CWmax	15/1023
Slot time	52 μ s
SIFS	160 μ s
DIFS	SIFS+2 *Slot time (μ s)
RTS/CTS	Not enabled
STA distribution	Random
Beacon Interval	0.1s
Number of STA	1000
No. of data bits in one OFDM (L^{Drs})	26 bits
Symbol duration (T_{sym})	40 μ s
Coding rate for Binary Phase Shift Keying (BPSK)-MCS0 (R)	0.5
PHY header	6 * symbol duration
MAC header (m_h)	14 B
Number of slot in each RAW	10
RAW slot duration	15 ms
Queue size	100 packets
No. of QoS STAs	50%
No. of groups	2-10
Simulation area	1000 \times 1000 m^2
Simulation time	5 min

5.3 Simulation Analysis

We use NS-3 [1] for simulation analyses of the proposed protocol. Initially, the effect on throughput and delay are analyzed for different priority classes with 50% of QoS traffic in the traditional scheme. Other simulation parameters are mentioned in Table 6. The following throughput and delay analysis are carried out to portray the essence of the proposed scheme:

- ◆ QoS in the traditional scheme: Throughput and delay results are analyzed.
- ◆ QoS in proposed and traditional scheme: Throughput results are compared in the proposed and traditional scheme.
- ◆ QoS of Different Traffic Categories: Throughput and delay of the proposed scheme considering different classes of traffic are measured.

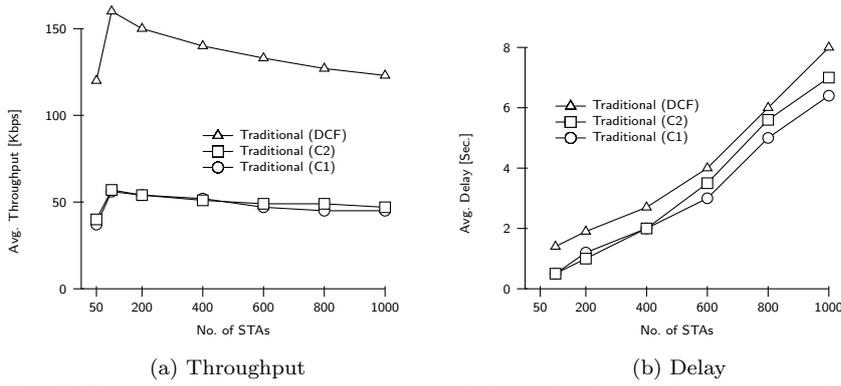


Fig. 5: Throughput and delay characteristics with increasing number of STAs in traditional scheme

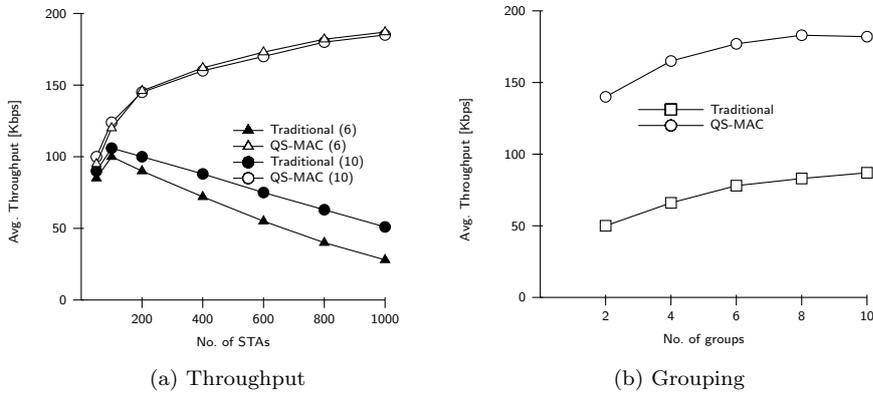


Fig. 6: Throughput characteristics with increasing number of STAs and groups

- ◆ Overall QoS performance in the proposed and traditional scheme: Throughput performance are measured to see the efficiency of QS-MAC for dynamic grouping.

5.3.1 QoS in Traditional Scheme

Before applying the proposed scheme, we measure the performance of the traditional 802.11ah MAC protocol with the considered traffic classes. As shown in Fig. 5a, the tradition scheme with more number of non-QoS STAs show higher throughput due to the higher possibility of transmission. Similarly, the delay is not reduced for the QoS traffic as shown in Fig. 5b.

5.3.2 QoS in Proposed Vs. Traditional Scheme

We compare the performance of *QS-MAC* with the traditional scheme. We measure the effect on the priority of traffic over non-priority traffic is mea-

sured. In Fig. 6a, considering a number of groups like 6 and 10, with an increasing number of STAs, throughput decreases for the DCF scheme. However, in the case of the priority scheme, saturation throughput increases. With an increasing number of STAs, a higher number of priority STAs try for channel access. Hence, the probability of getting a channel for the non-QoS STA is lesser. Secondly, without any priority scheduling, chances of collisions are higher in case of a higher number of STAs in DCF. The remaining bandwidth can be utilized to support the requirements of class STAs, which is done by the proposed adaptive grouping scheme. Again, the DCF mechanism performs better in case of a higher number of groups. This is because of the reduction in collisions (due to contention). Fig. 6b considers all the traffic classes for analysis. Throughput is measured with an increasing number of groups considering QoS bound traffic. For the only group in the DCF mechanism, throughput decreases drastically. However, using 6 groups a better result can be seen especially in a case for a large number of STAs.

5.3.3 QoS of Different Traffic Categories

Throughput achieved over increasing number of STAs considering all classes C1, C2, C3, and C4 of traffic at the same time. If we consider only class-based traffic in the same scenarios, throughput achieved by the higher priority class is better. Also, a small decline in throughput is observed as compared to DCF scheme. In addition to all the priority classes, throughput shows 125% higher than the tradition DCF scheme. Further, throughput performance for class STAs is also measured against different groups sizes. The proposed protocol shows similar results in almost all the group sizes (can be seen in Fig. 7a). Throughput achieved over increasing number of STAs considering all classes C1, C2, C3, and C4 of traffic at the same time. This is due to the adaptive grouping scheme, which balances the available bandwidth among all the groups. The proposed scheme reduces the simultaneous channel access by early priority scheduling. Finally, Fig. 7b shows the saturation delay performance. The proposed scheme reduces delay up to a huge margin. It applies RAW scheduling and adaptive node grouping to provide QoS for the critical STAs. The preliminary delay in a large-scale network is the channel access delay, which is reduced in this scheme. However, non-class STAs incur higher delay in spite of relief for more number of groups.

5.3.4 Effect of Grouping in Proposed Vs. Traditional Scheme

We analyzed the overall delay improvement in the proposed scheme and compared it with the traditional scheme. As shown in Fig. 8a, the average throughput is measured with the increasing number of STAs. Our solution efficiently utilizes the available channel bandwidth, which is distributed among different RAW groups. Due to this, the total throughput ($C1+C2+C3+C4$) in the proposed scheme is higher than the traditional scheme. Fig. 8b shows the average delay performance of the proposed scheme with increasing loads in a group.

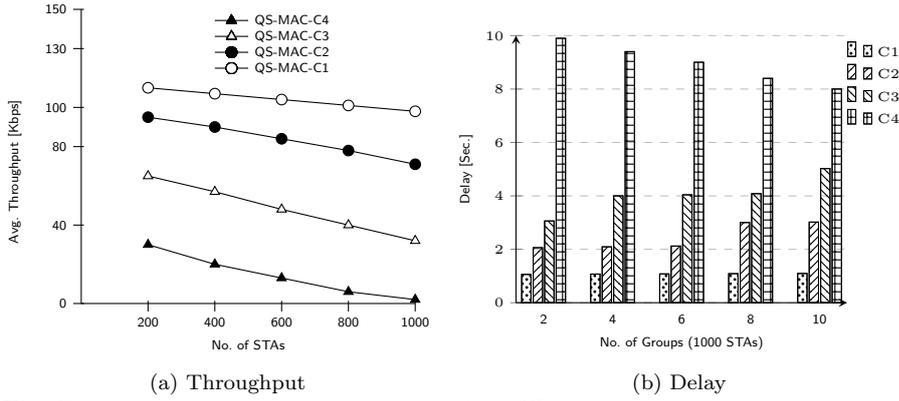


Fig. 7: Average throughput and delay for different network and node group sizes respectively

Considering 50% of QoS traffic in a RAW group, we take an average of delays of all the STAs that are initially considered to be in the same group. It may be noted that a STA may not be in the same group for the entire duration. Due to the dynamic grouping scheme, the average delay of the STAs is reduced significantly as compared to the traditional scheme. Due to the saturation capacity of the links, the delay drastically changes when the loads increase beyond 300 Kbps.

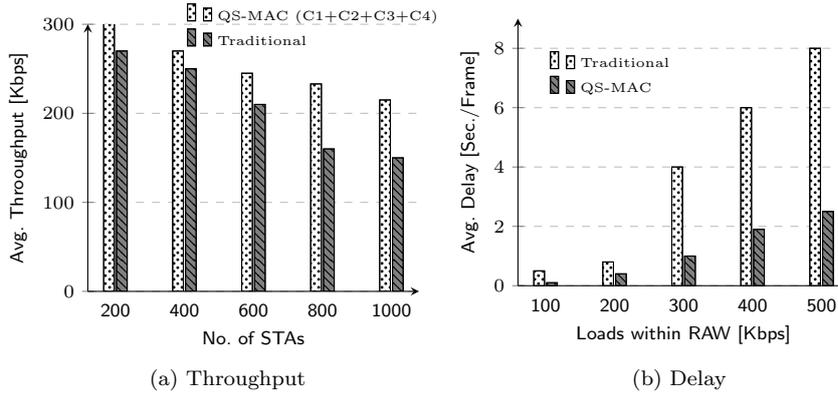


Fig. 8: Average throughput and delay performance

6 Conclusion

This work presented a priority scheduling and adaptive grouping scheme for improving QoS performance in terms of latency and throughput over a highly

dense network. With the use of the proposed scheme, the utilization of a large-scale network is significantly improved over the existing RAW grouping. From the experimental results, it is apparent that a higher number of groups gives better performance given different requirements of STAs. This study suggested that it is more advisable to classify all the STAs of the network into different priority classes according to their requirements and schedule their transmission opportunities accordingly. The proposed protocol used three algorithms, viz., priority scheduling, fairness scheme, and adaptive grouping to achieve QoS. These algorithms use the same computation on backoff operations for different purposes, such as window-based priority setup, load balancing, and achieving fairness. Hence, computational latency is lesser, and scalability is higher as compared to the existing related works. Although the proposed QoS scheme can ensure guaranteed services for critical applications up to a certain extent, with an increasing number of STAs and critical traffic, congestion will be high. In such a case, the total network capacity needs to be enhanced further.

References

1. What is ns-3. <https://www.nsnam.org/overview/what-is-ns-3/>. online.
2. IEEE Approved 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 2: Sub 1 GHz License Exempt Operation. *IEEE P802.11ah/D10.0, Sep 2016*, pages 1–660, Jan 2016.
3. Abdelnaser Adas. Traffic models in broadband networks. *IEEE communications Magazine*, 35(7):82–89, 1997.
4. M Zulfiker Ali, Jelena Mišić, and Vojislav B Mišić. Performance Evaluation of Heterogeneous IoT Nodes With Differentiated QoS in IEEE 802.11ah RAW Mechanism. *IEEE Transactions on Vehicular Technology*, 68(4):3905–3918, 2019.
5. Giuseppe Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal on selected areas in communications*, 18(3):535–547, 2000.
6. Navroz Firoz Charania, Mukesh Kumar Giluka, Bheemarjuna Reddy Tamma, and Antony Franklin. DEARF: Delay and Energy Aware RAW Formation Scheme to Support Delay Sensitive M2M Traffic in IEEE 802.11ah Networks. *arXiv preprint arXiv:1709.03723*, 2017. Available at: <https://arxiv.org/abs/1709.03723>.
7. Youssef Dehbi, Hafssa Benaboud, and Noufissa Mikou. A Geometric Backoff Time Distribution of IEEE 802.11 DCF: An Analytical Study. *International Journal of Communication Networks and Information Security (IJCNIS)*, 5(3):192–200, 2013.
8. Aditya Dhananjay and Lu Ruan. PigWin: Meaningful load estimation in IEEE 802.11-based wireless LANS. In *IEEE International Conference on Communications*, pages 2541–2546. IEEE, 2008.
9. IEEE802.16p. IEEE 802.16p Machine to Machine (M2M) Evaluation Methodology Document (EMD). *IEEE 802.16 Broadband Wireless Access Working Group (802.16p)*, 2011. EMD 11/0005.
10. Caihong Kai, Jiaojiao Zhang, Xiangru Zhang, and Wei Huang. Energy-Efficient Sensor Grouping for IEEE 802.11ah Networks With Max-Min Fairness Guarantees. *IEEE Access*, 7:102284–102294, 2019.
11. Germán Corrales Madueño, Čedomir Stefanović, and Petar Popovski. Reliable and efficient access for alarm-initiated and regular M2M traffic in IEEE 802.11ah systems. *IEEE Internet of Things Journal*, 3(5):673–682, 2016.
12. David Malone, Ken Duffy, and Doug Leith. Modeling the 802.11 distributed coordination function in nonsaturated heterogeneous conditions. *IEEE/ACM Transactions on Networking (TON)*, 15(1):159–172, 2007.

13. Zhou Ming and Ma Yan. A Modeling and Computational Method for QoS in IoT. In *International Conference on Software Engineering and Service Science (ICSESS)*, pages 275–279. IEEE, 2012.
14. Chul Wan Park, Duckdong Hwang, and Tae-Jin Lee. Enhancement of IEEE 802.11ah MAC for M2M Communications. *IEEE Communications Letters*, 18(7):1151–1154, 2014.
15. Orod Raeesi, Juho Pirskanen, Ali Hazmi, Toni Levanen, and Mikko Valkama. Performance evaluation of IEEE 802.11ah and its Restricted Access Window mechanism. In *International Conference on Communications Workshops (ICC)*, pages 460–466. IEEE, 2014.
16. Amina Seferagić, Ingrid Moerman, Eli De Poorter, and Jeroen Hoebeke. Evaluating the suitability of IEEE 802.11ah for low-latency time-critical control loops. *IEEE Internet of Things Journal*, 2019. Available at: <https://doi.org/10.1109/JIOT.2019.2916579>.
17. Pallavi Sethi and Smruti R Sarangi. Internet of Things: Architectures, protocols, and applications. *Journal of Electrical and Computer Engineering*, 2017, 2017. <https://doi.org/10.1155/2017/9324035>.
18. Amina Šljivo, Dwight Kerkhove, Le Tian, Jeroen Famaey, Adrian Munteanu, Ingrid Moerman, Jeroen Hoebeke, and Eli De Poorter. Performance Evaluation of IEEE 802.11ah Networks With High-Throughput Bidirectional Traffic. *Sensors*, 18(2):325, 2018.
19. Pranesh Sthapit and Jae-Young Pyun. Station grouping strategy for minimizing association delay in IEEE 802.11 ah. *IEICE Transactions on Communications*, 100(8):1419–1427, 2017.
20. Le Tian, Evgeny Khorov, Steven Latré, and Jeroen Famaey. Real-time station grouping under dynamic traffic for IEEE 802.11ah. *Sensors*, 17(7):1559, 2017.
21. Yanru Wang, Yun Li, Kok Keong Chai, Yue Chen, and John Schormans. Energy-aware adaptive restricted access window for IEEE 802.11ah based Smart Grid networks. In *International Conference on Smart Grid Communications (SmartGridComm)*, pages 581–586. IEEE, 2015.
22. Yang Xiao. Backoff-based priority schemes for IEEE 802.11. In *Communications, 2003. ICC'03. IEEE International Conference on*, volume 3, pages 1568–1572. IEEE, 2003.
23. Sung-Guk Yoon, Jeong-O Seo, and Saewoong Bahk. Regrouping algorithm to alleviate the hidden node problem in 802.11ah networks. *Computer Networks*, 105:22–32, Aug 2016.
24. Yue Zhao, Osman NC Yilmaz, and Anna Larmo. Optimizing M2M energy efficiency in IEEE 802.11ah. In *Globecom Workshops (GC Wkshps)*, pages 1–6. IEEE, 2015.