

A Novel Adaptive Random Access Window (RAW) Slot Adjustment Model for IoT Network Using an Integrated Scheme.

Nwogu, UchennaObioma

Department of Electrical Electronics Engineering, Federal Polytechnic Nekede, Owerri Nigeria

ABSTRACT: *The RAW (Restricted Access Window) component of the IoT network is deployed to reduce traffic and channel contention in dense and heterogeneous sensor network environment. It divides sensor nodes into groups and slots, allowing channel access only to one RAW slot at a time. Several algorithms and improved channel utilization optimization models have been proposed to optimize the RAW parameters, to ensure a contention free network or at least, minimally reduce it. These techniques often rely on previous traffic demands schedules, collision analysis and send/receive matrices to accurately predict the future of stations' interactions in an IoT environment. Thus systematically adjusting its operations to reduce contention among the stations and the Access Point(AP), thereby ensuring a flexible transmission even in a dense network environment. This paper critically investigated, analysed and proposed a novel approach to RAW size adjustment. The new approach will invoke data mining instruments and optimisation algorithms to improve performance of stations in an IoT environment and thereafter simulated (with respect to some specified QoS parameters) to ascertain it's (output) performance index with the already existing models.*

KEYWORDS: *Resource allocation, station, network, nodes, simulation.*

I. INTRODUCTION

A system group based contention scheme known as restricted access window, used for accessing a shared medium of communication is an IEEE ah protocol technique in use. RAW slot mechanism is used to control the collusion that may arise due to heavy contention in a heterogeneous network [1]. Various RAW size adjustment schemes were examined, to ascertain their strengths and weaknesses.

Their various modeling parameters and algorithms were analyzed. Most of which contains complex (mathematical) computations and large amount of time to achieve result.

Since the heterogeneous environment is bedeviled with the task of arranging the object sets (stations) into groups or clusters to aid seamless data transmission and contention free communication, we use the **Enhanced Transitive Heuristic (ETH algorithm)** [2] which brings an improvement, to the original heuristic algorithm for clustering large dataset called transitive heuristic, to improve calculations in cluster representation and management and also the manner in which each data object is processed.

Ada-TDMA algorithm [3] supports contention free link in mobile networks. Most information required by the protocol is gathered exclusively from the stations environment, global coordination of stations is not necessary and changes in the state information at the station do not need to be distributed to all stations in the network. The TDMA algorithm operates effectively in its environment in a dual capacity; where one part of the algorithm adjust the slot assignments if an added station is detected, while the other modifies the TDMA slot assignment to improve on the slot assignment anytime two or more terminals are out of range[3].

During the respective time slot of the RAW of a given network, the network should adjust itself based on the number of groups or clusters it has. So as to allocate the appropriate data set accordingly to stations. For this purpose, we present an **improved version of Hidden Markov Model**, which invokes Markov state chain models and allows for arbitrary distribution of station sizes and also helps their estimation to be known[4].

The result is a **novel robust algorithm (VS-HMM)** that requires little user input to obtain the result if well simulated. The major advantage of the VS-HMM algorithm is that it is less sensitive to initial estimates of step size [5]. The usage of this algorithm (VS-HMM),is best in Kinetic Models where the state of the system is not known because of the silent kinetic transition and also helps to reduce noise in the system[5].

As a result of the issues/problems encountered in the existing techniques, our modified scheme (TGI²) contributes the following processes:

- To avoid the hidden node problem in the network, we convert the AP coverage area into the different sizes of sectors.
- We group the stations that are present within each sector using the ETH algorithm. To the best of our knowledge, we are first in utilizing the Transitive Heuristic algorithm for grouping stations in the IEEE 801.11ah based IoT network.
- We perform the registration process intending to reduce the time required throughout the network. We achieve this by setting the best authentication control threshold (ACT) in parameter settings.
- We allocate RAW slots to each RAP through AP in an adaptive manner using the Ada-TDMA algorithm. Our novelty is present in adjusting RAW slot size where we utilize VM-HMM based state prediction method. None of the works have performed VM-HMM based RAW slot adjustment procedures.

Furthermore, we show our work betterment by comparing simulation results of the modified scheme with the existing techniques. It is acquired through upcoming validation metrics such as network delay, throughput, channel utilization, registration time and successful transmission probability.

II. PROBLEM STATEMENT IN THE EXISTING RAW TECHNIQUES

Here we highlighted the problems that are present in the RAW slot allocation in previous works related to IEEE 802.11ah based IoT network. The network comprises of three major entities such as *AP*, *RAP* → {*RAP*₁, *RAP*₂, .. *RAP*_{*n*}} and *Stations* → {*STA*₁, *STA*₂, .. *STA*_{*n*}}. The main issues encountered from existing works are channel contention overhead, network delay and hidden node problem.

We have also defined problems from the individual works that are discussed as follows: ***RAW size Adjustment (RA_d) in IEEE 802.11ah*** based scalable network architecture is proposed for IoT environment [6]. Here, TDMA slot is allocated to the RAP that may lead to contention due to the same channel allocation for two different RAP. Furthermore, Overlapping is high in station grouping due to available channel based group formation. ***AID based back off timer is estimated for throughput enhancement*** in the IEEE 802.11ah network [7].

More effects are required under sectorization, since it plays a vital role in reducing interference among different stations and hidden node problem. Here, the traffic demand of station is a significant metric in group formation in order to enhance channel utilization. ***Traffic demand based stations grouping (TSG)*** is the introduced in IEEE 802.11ah standard based IoT network [8]. In this, Greedy based grouping doesn't provide an optimal solution and also consumes more time to find an optimum solution. Because it's searching behavior is not effective and choosing the choice from the current state. Efficient massive IoT device registration is performed using a ***hybrid slotted CSMA/CA and TDMA (HSCT) algorithm*** [9].

More analysis is required in parameters setting of threshold estimation in order to provide effective authentication/association process to the stations. Here, the authentication and association request process runs with the usage of both traditional CSMA/CA and TDMA algorithms that increases the mathematical computations with complex procedures (Authentication/Association (Req/Res)). ***Markov chain and M/G/1 model based RAW slot Allocation mechanism (MRA)*** is introduced in IEEE 802.11ah networks [10]. Markov chain based queue state prediction takes more time and also doesn't provide the exact state of the stations. The contention is more since part of the beacon interval is kept open where all stations transmit their data without proper scheduling.

III. MODELING OF THE EXISTING RAW SLOT TECHNIQUES

The above mentioned schemes are individually modeled using their peculiar network parameters that are used to prepare a ***pseudo code*** to generate an algorithm which is simulated using network simulators. ***The hybrid slotted CSMA/CA and TDMA (HSCT) algorithm*** is modeled using ***the up and down algorithms*** [9] for optimal value threshold selection. And ***MAC layer parameters*** [9] are used in the simulations to generate the results for analysis.

Traffic demand based stations grouping (TSG) uses the ***Traffic-aware grouping algorithm*** [8] to improve channel utilization. Furthermore, a ***regression-based analytical model*** is derived to estimate the contention success probability with consideration of sensors' heterogeneous traffic demands as parameters used in the model [8]. The evaluation is done via NS-3 simulations and results generated are used for the analysis.

RAW size Adjustment (RA_d) in IEEE 802.11ah uses scalable network architecture where an RAP allocation scheme provides dynamic selection and positioning of relay nodes over a large-scale network. [6] here, 802.11ah module developed by [6] for NS 3 network simulator to observe the performance of the network using the network parameters like bandwidth, modulation and coding schemes, RAW size, RAW slot duration, Nos. of relay and RAW groups, etc.

AID based back off timer is estimated for throughput enhancement in the IEEE 802.11ah. Here, the network is modeled using **hierarchical clustering** [7] and a **modified Welsh-Powell algorithm** [7] that is combined to create hidden node-free group. An analytical model is developed for a deterministic backoff technique that uses association identifiers to set the backoff in order to eliminate collisions. Some of the simulation parameters used for the modeling includes; bandwidth, carrier frequency, noise figure and density, fade merging, SIFS and DIFS, etc.

3.1 MODELLING OF THE ADAPTIVE MODEL

This section describes the adaptive processes such as; grouping, parameter settings for registration, RAW slot allocation in details. Also, we discuss each algorithm respective to each process in IoT network.

Conceptual Overview

We designed an effective network which is robust to the network delay, hidden node problem and contention overhead issues. Our designed network comprises the AP, RAP and Stations (IoT nodes) as shown in figure 1

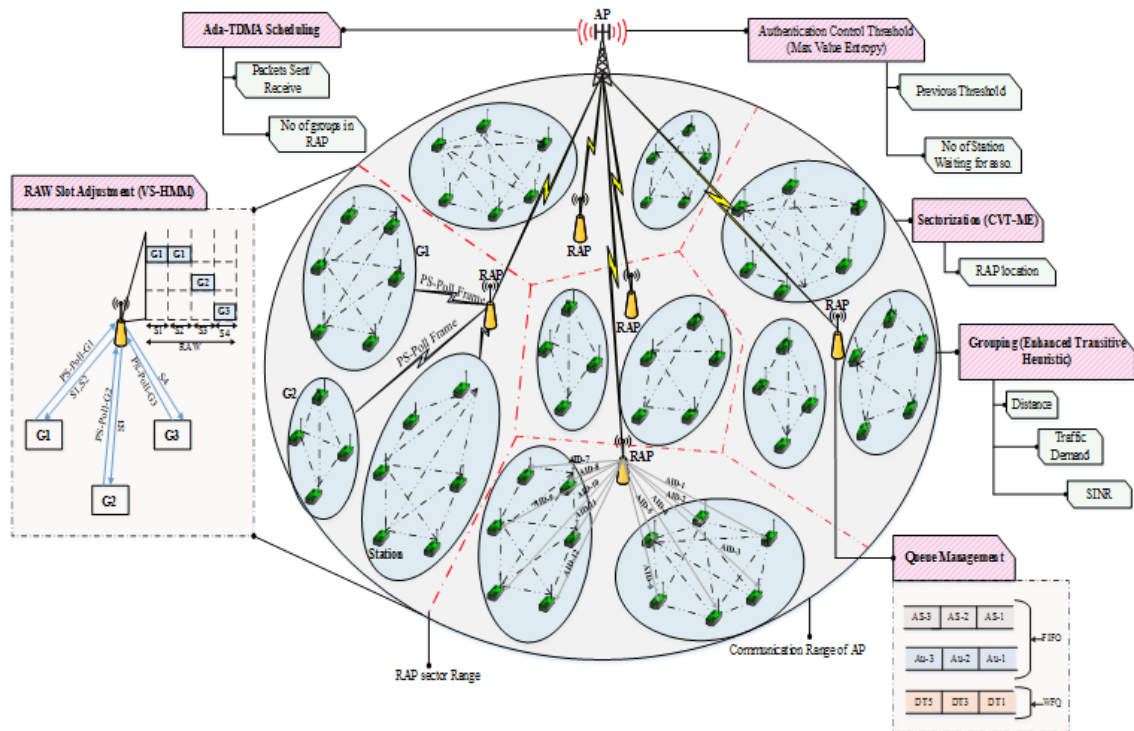


Figure 1: AP, RAP and Stations (IoT nodes)

The above discussed issues are resolved through the processes discussed as follows: Primarily, our network is sectorized by considering the RAP location, to increase the scalability of the AP. In each sector, we perform station grouping processes using **Enhanced Transitive Heuristic (ETH) algorithm**. Here, **three metrics are used to group the stations** that are **distance, SINR and Traffic demand**. The registration process is started when we complete the sectorization and grouping in the network. The registration process, we set ACT (Authentication Control Threshold) using the Max Value Entropy function. To avoid data collisions during transmission between AP and RAP, we provide an adaptive time slot to each RAP. It is achieved through the **Ada-TDMA algorithm**. Using their respective slot, RAP nodes gather RAW slot from the AP. To avoid the contention within each sectored group, RAP present in each sector adjusts its RAW slot. It adjusts RAW slots using the **VS-HMM algorithm by considering traffic load and collision rate**.

3.2 TRANSITIVE GROUPING

The grouping of stations in the IEEE 802.11ah based IoT network is a significant process; since grouping the stations reduces the contention overhead issue and increases the throughput of the network. Hence in our network, we perform stations grouping in each sector respective to the RAP.

Our **adaptive network utilizes the ETH algorithm to perform grouping** in sector respect each RAP. To the best of our knowledge, we are first in utilizing this algorithm for grouping the station in IEEE 802.11ah based IoT network. **The reason behind selecting this algorithm is that it provides better performance even in a huge number of stations. It provides better performance compared to the existing Transitive**

heuristic algorithm [40]. It clusters by estimating the Condorcet's criterion of station present in the network. For clustering the station, it considers three major factors that are distance, SINR and traffic demand. These metrics are designated as follows:

Definition 1: Distance (\mathcal{D}) - This metric is used to reduce the delay and communication cost in each group. It is the distance between the station 'i' and station 'j' in the network. It is expressed in mathematical form as follows:

$$\mathcal{D} = \sqrt{(S_{ix2} - S_{ix1})^2 + (S_{iy2} - S_{iy1})^2} \quad (1)$$

Here, S_{ix1}, S_{ix2} represents the $x1, x2$ position of the station S_i . S_{jx1}, S_{jx2} Represents the $x1, x2$ position of the station S_j .

Definition 2: SINR-This metric is considered to reduce the interference occur during data transmission between RAP and station. By considering this metric in station grouping, we ensure that our work robust against collision in the network. It is expressed in mathematical form as follows:

$$SINR_{S_i, S_j} = \frac{P}{N+I} \quad (2)$$

Where, P represents the power from the incoming signal of interest, N represents the noise and I represents the interference power of other signal.

Definition 3: Traffic Demand (T_d): This metric is taken to group the stations that are in a similar context with respect to the traffic it belongs to. By utilizing this metric in station grouping, our work satisfies the traffic demand of the stations present in the network. Thus increases the throughput of the network.

With the use of these three metrics, proposed ETH algorithm groups the stations within the sectors. It estimates the co-efficient between stations in order to group it accordingly. The estimation of co-efficient depends on the Condorcet's criterion which is used to measure the similarity between the stations present in the network. The Condorcet's criterion is expressed in mathematical form as follows:

$$\text{Condorcet } (CO, G) = \sum_{k=0}^N \sum_{l=0}^N CO_{kl} G_{kl} + \sum_{k=0}^N \sum_{l=0}^N \overline{CO}_{kl} \overline{G}_{kl} \quad (3)$$

Here, CO represents the collective relational matrix, G denotes the collection of the groups. Likewise, $\overline{CO}_{kl}, \overline{G}_{kl}$ signifies the complementary of the collective relational matrix and clusters respectively. Here, the co-efficient between the stations are estimated using the below equation,

$$\text{Co-eff } (S_i, S_j) = \sum \frac{\mathcal{D} + SINR}{T_d} \quad (4)$$

By estimating this equation, the ETH algorithm identifies the similarity between the stations in the network. The stations that are having high co-efficient with each other are clustered into one group. The pseudocode for the ETH based clustering is described as follows:

Pseudocode /algorithm 1: ETH based Station Grouping

Require: Set of Stations ($S_{1..n}$)

Ensure: Grouped Stations

Select → Representative stations (S_j);

Initialize → $S_n \leftarrow \text{null}, c \leftarrow 0$;

For ($i=0 \rightarrow |S|$) **do**

If (S_i is not clustered) **Then**

Calculate → Co-eff(S_i, S_j) using equ. (4);

If (Co-eff(S_i, S_j) ≥ 0) **Then**

Assign → $S_i \rightarrow$ Current Cluster;

Update → S_j ;

$C++$;

Else If ($S_n \rightarrow \text{null}$) **Then**

$S_n \leftarrow S_i$;

Else If (S_i is clustered) **Then**

Calculate → Co-eff(S_i, S_j) and Co-eff(S_i, S_j');

If (Co-eff(S_i, S_j) $>$ Co-eff(S_i, S_j'))

Add → $S_i \rightarrow$ Current Cluster;

End If;

End For

Pseudocode Description: The above pseudocode describes how our network groups the stations in each sector. Initially, it selects one representative to estimate the similarity between them to form groups. For this purpose, it estimates the performance of the co-efficient between two stations using the equ. 4. Based on the estimated co-efficient value, it groups the stations that are in the same context with one other. This way of grouping the stations in the network evades the contention overhead and also improves the throughput in the network. The time complexity of the proposed ETH based grouping is $O(Zn)$ where Z represents the number of clusters and n represents the number of iterations. From this, we conclude that our proposed ETH based grouping method has less time complexity which is less compared to other grouping algorithms such as K-Means, FCM and so on. Besides, the proposed ETH algorithm is flexible to high dense network and also exhibit better performance.

3.3 ADAPTIVE RAW SLOT ALLOCATION

In this portion, we are going to describe how we allocate the RAW slot to the stations present in the network. Our TGI² network, we follow the adaptive RAW slot allocation process. To avoid data collisions during the RAP and AP transmission process, we utilize the Ada-TDMA algorithm based slot allocation process. Here, we allocate slots to each RAP by considering the number of stations group it has and packets sent or receive count.

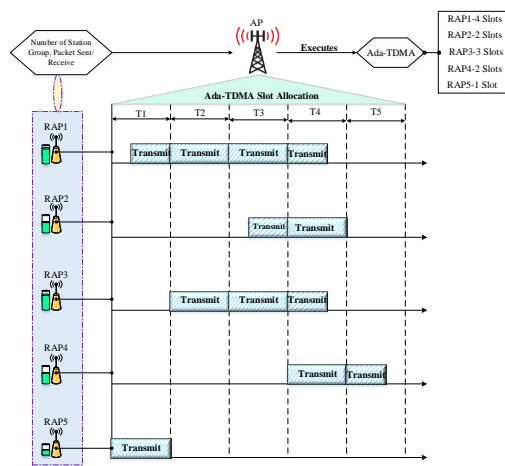


Figure 2: Ada-TDMA based Time Slot Allocation

Figure 2 depicts the Ada-TDMA based slot allocation process for RAP data transmission to the AP. The main purpose of slot allocation RAP data transmission is to avoid the data collisions among RAP. Here, AP considers each RAP’s number of station groups and the packet transmit/receive count during the last slots. During the allocated time slot only each RAP transmits its data to the AP. This way communicating with the AP reduces the data collisions and packet losses in the network effectually.

Pseudocode /algorithm 2: The Psuedocode for Adaptive Raw Slot Allocation Is Shown below

```

/*  $G_C = (V_C, E_C)$ : the graph of cluster  $C$  */
1: for each cluster  $C_i$  do
2:   for each flow  $F_i \in \mathcal{F}$  do
3:      $F_k \leftarrow F_i$  by using the CH for  $dst_k$ 
4:      $\varphi_k \leftarrow$  a path from  $src_i$  to the CH in  $\varphi_i$ 
5:     for  $k$  from 1 to  $|\varphi_k|$  and  $k \leq N_{IS}$  do
6:        $\vec{uv} \leftarrow$  the  $k$ -th edge in  $\varphi_i$ 
7:     endfor
8:      $\mathcal{F}_C \leftarrow \mathcal{F}_C \cup (r_i, F_k, \vec{uv})$ 
9:   endfor
10:   $\mathcal{T}_{IS_C} \leftarrow IA-FP-Scheduling(G_C, \mathcal{F}_C)$ 
11: endfor
12: for each flow  $F_i$  do
13:   Calculate  $r_i^{SH}$  from the result of  $\mathcal{T}_{IS_C}$ 
14: endfor
15: return  $\mathcal{T}_{IS_C}$ 

```

Pseudocode Description: The pseudo-code of IntraSend scheduling is described in Algorithm 4 (41). The flows are scheduled in the cluster for NIS slots (line5). Each CH manages the IntraSend scheduling result by calling the function IA-FP-Scheduling 4 (line10). After scheduling, then expected arrival time of flows that is required to reach their CHs are computed from the result of IntraSend scheduling (line13).

Pseudocode /algorithm 3: InterComm Scheduling (G_{header}, F, N_{IC})

```

/*  $G_{header}$ : the graph consisting of only CHs */
1: for each flow  $F_i \in \mathcal{F}$  do
2:    $F_k \leftarrow F_i$ 
3:    $src_k \leftarrow$  the source CH of  $F_i$ 
4:    $dst_k \leftarrow$  the destination CH of  $F_i$ 
5:    $\varphi_k \leftarrow$  a path from  $src_k$  to  $dst_k$  in  $\varphi_i$ 
6:   for  $k$  from 1 to  $|\varphi_k|$  and  $k \leq N_{IC}$  do
7:      $\vec{uv} \leftarrow$  the  $k$ -th edge in  $\varphi_i$ 
8:   endfor
9:    $\mathcal{F}_C \leftarrow \mathcal{F}_C \cup \{(rframe_i^{IC}, F_k, \vec{uv})\}$ 
10: endfor
11:  $\mathcal{T}_{IC} \leftarrow IA-FP-Scheduling(G_{header}, \mathcal{F}_C)$ 
12: for each flow  $F_i$  do
13:   Calculate  $r_i^{DH}$  from the result of  $\mathcal{T}_{IC}$ 
14: endfor
15: return  $\mathcal{T}_{IC}$ 

```

Pseudocode Description: In this scheduling, the flows that arrive to the source CH are transmitted to the destination CH (42). For this purpose, we build a sub-graph G_{header} consisting of only CHs. The InterComm scheduling is used to allocate NIC slots and its scheduling result is shared among CHs. As shown in Algorithm 5, the routing path contains only the sub-path from source CH to destination CH (line5). Each flow is scheduled for NIC slots (line6). The function IA-FP-scheduling 4 is used to adjust flows for InterComm scheduling (line11). After scheduling, the expected arrival time of flows is obtained from the result of InterComm scheduling (line13).

Pseudocode /algorithm 4: IntraRecv Scheduling (G_C, F, N_{IR})

```

/*  $G_C = (V_C, E_C)$ : the graph of cluster  $C$  */
1: for each cluster  $C_i$  do
2:    $\mathcal{F}_C \leftarrow \emptyset$ 
3:   for each flow  $F_i \in \mathcal{F}$  do
4:      $F_k \leftarrow F_i$  by using the CH for  $src_k$ 
5:      $\varphi_k \leftarrow$  a path from the CH to  $dst_i$  in  $\varphi_i$ 
6:     for  $k$  from 1 to  $|\varphi_k|$  and  $k \leq N_{IR}$  do
7:        $\vec{uv} \leftarrow$  the  $k$ -th edge in  $\varphi_i$ 
8:     endfor
9:      $\mathcal{F}_C \leftarrow \mathcal{F}_C \cup \{(rframe_i^{IR}, F_k, \vec{uv})\}$ 
10:   endfor
11:  $\mathcal{T}_{IR_C} \leftarrow IA-FP-Scheduling(G_C, \mathcal{F}_C)$ 
12: for each flow  $F_i$  do
13:   Calculate  $r_i^{\alpha_i}$  from the result of  $\mathcal{T}_{IR_C}$ 
14: endfor
15: return  $\mathcal{T}_{IR_C}$ 

```

Pseudocode Description: In IntraRecv scheduling, the flows arrived at the destination C Hare transmitted to destination node (42). Like Intra Send scheduling, each cluster manages the IntraRecv scheduling result through the pseudo-code shown in Algorithm 6. The scheduling is done by calling the function IA-FP-scheduling 4(line11).

During the respective time slot, AP allocates the RAW slot for the stations belong to respective to each RAP. After getting RAW slots from the RAP, each RAP in the network adjusts its slot based on the states of the group it has. For this purpose, it utilizes the VS-HMM algorithm. VS-HMM algorithm consumes less parameter to reduce the tedious processing in the existing state prediction algorithm. Hence, we select this algorithm to evaluate the current condition of the station groups present in each sector belonging to each RAP. In our work, we consider two states relate to RAW slot allocation that requires high RAW slot and requires a low RAW slot. Figure 5 illustrates the RAW slot allocation process handled in the RAP entity with the aid of the VS-HMM algorithm. The VS-HMM finds the state of each group in the sector and it is estimated by considering the traffic load and collision rate. It is described as follows:

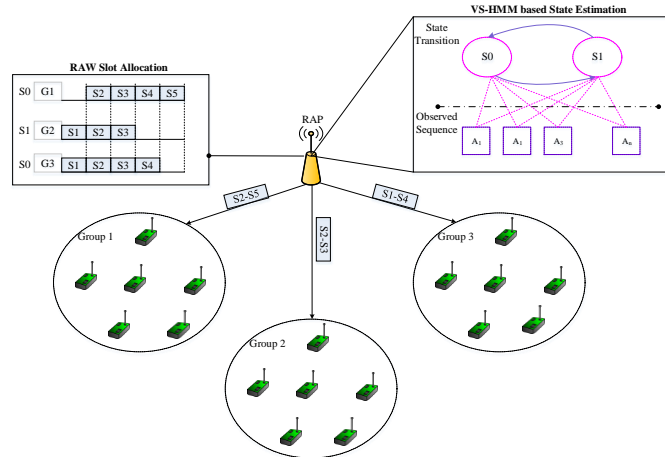


Figure 3: RAW Slot Allocation

Definition 1- Traffic Load- This metric is used to know the current traffic load status of the station groups in the network. By considering this metric during state estimation provides better performance in the RAW slot allocation.

Definition 2- Collision rate- This metric is used to avoid the collision during the data transmission in the network. The collision rate is estimated by considering the data transmission performance of the stations in the network.

Let the parameters of the VS-HMM are described as,

$$\gamma = [\pi, \tau_0, C] \quad (5)$$

Here, π represents the vector of the initial probabilities, C denotes the matrix of transition probabilities and τ_0 denotes the noise. The likelihood in the VS-HMM algorithm is designated as the probability $\mathcal{P}[A|\gamma]$ of the observed data sequence $A = A_1, A_2, \dots, A_n$. The log likelihood of the data sequence is listed as follows:

$$\mathcal{L} = \ln[\mathcal{P}[A|\gamma]] \quad (6)$$

The probability that the sequence (s) underlies the following observations,

$$\mathcal{P}[A|\gamma] = \sum_s \mathcal{P}[s|\gamma] \mathcal{P}[A|s, \gamma] \quad (7)$$

In this, the probability $\mathcal{P}[A|s, \gamma]$ is further expressed in mathematical form as follows:

$$\mathcal{P}[A|s, \gamma] = \prod_{i=1}^n b_i[A_i, v_i] \quad (8)$$

Where, v_i represents the position variable and A_i denotes the data sequence. With the aid of these equations, the VS-HMM algorithm estimates the state of each station group belongs to each sector in the IEEE 802.11ah based IoT network. The time complexity of the VS-HMM algorithm is $O(nT)$ where n represents the number of states and T represents the sequence length. The time complexity of the proposed Markov model is less than the other existing Markov models.

Pseudocode /algorithm 5: The pseudocode for the VS-HMM Algorithm (forward/Backward Algorithm 2 is shown below:

<p>Inputs: Length m, set of possible states \mathcal{S}, function $\psi(s, s', j)$. Define $*$ to be a special initial state.</p> <p>Initialization (forward terms): For all $s \in \mathcal{S}$,</p> $\alpha(1, s) = \psi(*, s, 1)$ <p>Recursion (forward terms): For all $j \in \{2 \dots m\}, s \in \mathcal{S}$,</p> $\alpha(j, s) = \sum_{s' \in \mathcal{S}} \alpha(j-1, s') \times \psi(s', s, j)$ <p>Initialization (backward terms): For all $s \in \mathcal{S}$,</p> $\beta(m, s) = 1$ <p>Recursion (backward terms): For all $j \in \{1 \dots (m-1)\}, s \in \mathcal{S}$,</p> $\beta(j, s) = \sum_{s' \in \mathcal{S}} \beta(j+1, s') \times \psi(s, s', j+1)$ <p>Calculations:</p> $Z = \sum_{s \in \mathcal{S}} \alpha(m, s)$ <p>For all $j \in \{1 \dots m\}, a \in \mathcal{S}$,</p> $\mu(j, a) = \alpha(j, a) \times \beta(j, a)$ <p>For all $j \in \{1 \dots (m-1)\}, a, b \in \mathcal{S}$,</p> $\mu(j, a, b) = \alpha(j, a) \times \psi(a, b, j+1) \times \beta(j+1, b)$
--

Pseudocode Description: The forward-backward algorithm is shown above [5]. Given inputs consisting of a sequence length m , a set of possible states \mathcal{S} , and potential functions $\psi(s, s', j)$ for $s, s' \in \mathcal{S}$, and $j \in \{1 \dots m\}$, it computes the following quantities:

1. $Z = \sum_{s_1 \dots s_m} \psi(s_1 \dots s_m)$.

2. For all $j \in \{1 \dots m\}, a \in \mathcal{S}$,

$$\mu(j, a) = \sum_{s_1 \dots s_m: s_j = a} \psi(s_1 \dots s_m)$$

3. For all $j \in \{1 \dots (m-1)\}, a, b \in \mathcal{S}$,

$$\mu(j, a, b) = \sum_{s_1 \dots s_m: s_j = a, s_{j+1} = b} \psi(s_1 \dots s_m)$$

IV. PERFORMANCE EVALUATION

Furthermore, we set up simulation experiments to evaluate the above algorithms with the performance metrics of network delay, throughput, scalability and registration time.

A simulation program will be developed, to evaluate the proposed model and also investigate its efficacy and effectiveness as regards to the already existing models. We will use a network simulator(NS-3.23 version), which is a discrete driven simulator, to generate the network- modeled topology where up to 50 stations are randomly generated in a specified test bed size with a specified time frame. The reason for selecting this tool for implementation is that it provides a better performance in any type of network with the proposed algorithms.

Finally we summarize the behavior of the proposed adaptive scheme (proposed algorithms) and compare it with that of the already existing techniques to ascertain its level of outperformance.

V. CONCLUSION

This paper proposes and adaptive RAW Slot adjustment model TGI. It is also a station-based grouping algorithm and RAW slot adjustment model. This model uses network parameters to perform sectoring and registration first CVT-ME, before grouping the stations (intuitively) in such a calculated and seamless manner for ease data transmission using the ETH algorithm. Thereafter, it uses scheduling algorithm Ada TDMA, depending on the state of the traffic in network, where slots are allocated based on the size of the station in the group. It fills a critical gap created by the other existing techniques as it attracts computational efficiency in IoT station's administration. Finally, the simulation results of our TGI² model was compared with the RAd, TSG, HSCT and MRA methods. From the result comparison, it proved that our TGI² method outperforms the other existing methods in IEEE 802.11ah based network.

APPRECIATION

Special thanks to Tertiary Education Trust Fund (TETFund) and the Research and Development Unit of Federal Polytechnic Nekede for making this research project possible.

REFERENCES

- [1]. Tian, L., Famaey, J., & Latré, S. "Evaluation of the IEEE 802.11ah Restricted Access Window mechanism for dense IoT networks", IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM), PP-1-10, 2016.
- [2]. Slaoui, S. C., Dafir, Z., & Lamari, Y. (2018). E-Transitive: an enhanced version of the Transitive heuristic for clustering categorical data. *Procedia Computer Science*, 127, 26–34.
- [3]. Gohar, A., Kyong H.K., & Ki-II, K. (2002). Adaptive TDMA Scheduling for Real-Time Flows in Cluster-Based Wireless Sensor Networks. *Computer Science and Information System* 13(2):475-492.
- [4]. U., S., & A. V., B. (2018). Performance analysis of IEEE 802.11ah wireless local area network under the restricted access window-based mechanism. *International Journal of Communication Systems*, e3888
- [5]. Michael Collins (2015). The Forward-Backward Algorithm. *International Journal of Communication System*, Volume 32, Issue 7, PP-1-20, 2019
- [6]. Tian, L., Khorov, E.M., Latré, S., & Famaey, J. (2017). Real-Time Station Grouping under Dynamic Traffic for IEEE 802.11ah. *Sensors*. 17(7), 1559
- [7]. Seferagic, A., Moerman, I., De Poorter, E., & Hoebeke, J. (2019). Evaluating the suitability of IEEE 802.11ah for low-latency time-critical control loops. *IEEE Internet of Things Journal*, 1–1.
- [8]. Lakshmi, L. R., & Sikdar, B. (2019). Achieving Fairness in IEEE 802.11ah Networks for IoT Applications with Different Requirements. *ICC 2019 - 2019 IEEE International Confere*
- [9]. Sthapit, P., & Pyun, J. (2017). Station Grouping Strategy for Minimizing Association Delay in IEEE 802.11ah. *IEICE Transactions*, 100-B, 1419-1427.
- [10]. Bhandari, S., Sharma, S.K., & Wang, X. (2018). Device Grouping for Fast and Efficient Channel Access in IEEE 802.11ah Based IoT Networks. *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*, 1-6.