Society for Science and Education
United Kingdom

TNC Transactions on Networks and Communications

Volume 3, Issue 6
ISSN: 2054 -7420

Resource Allocation Algorithm for Improving Performance of the OFDMA Based Connection Oriented Networks

Md. Nazmus Saadat[1], Borhanuddin Mohd Ali[1], Aduwati Sali[2], Raja Syamsul Azmir Raja Abdullah[2] and Hafizal Mohamad[3]
[1,2]*Faculty of Engineering, University Putra Malaysia, Serdang, 43300, Selangor, MY*
[3]*MIMOS Berhad, Technology Park Malaysia 57000,*
nazmus.saadat@yahoo.com, {borhan, aduwati, rsa}@eng.upm.edu.my
hafizal.mohamad@mimos.my

ABSTRACT

To enhance network performance, PHY and MAC layer has direct influence besides other factors as these are major layers of OSI based communication system. One way of enhancing network performance is the managing the radio resources intelligently. As cross layer based systems might be faster responding in case of network resource distribution and due to the spectrum limitation for commercial use, there are active researches in this area that targets to enhance the network users' experience, though RA might be considered as an evergreen topic for all evolving communication systems. This paper aims to focus specifically on how to increase throughput and delay performance leading to overall higher system performance and fairness. We use techniques of graph theoretic tools and optimization mechanism in our solution to improve radio resource allocation. After we optimize the subcarrier allocation using cross layer interaction of mainly MAC and PHY, the final assignment is done along with power allocation to all users. Then we reevaluate each new incoming resource request and use threshold based allocation techniques to cater for more users. Besides showing the performance enhancement we also show the fairness comparison to other existing state of the art research as benchmarking by means of simulation.

Key words: OFDMA, Resource Allocation, Radio Resource, Connection Oriented Networks, Wireless Networks, Algorithms, Graphs, QualNet, Cross Layer.

1 Introduction & Literature:

Various resource allocation algorithms has been proposed based on different opinions on how to save resources and enhance system performance. For example in [1] authors described optimized and suboptimal solution to manage multiuser diversity and resource allocation which has similar objective as we do but differs in approach to the solution. In [2], several cross layer based ideas are discussed and it is clear that cross layer design can bring higher performance for wireless networks comparing to conventional systems. In this paper, our proposed solution also make use of it and comes with better and intelligent algorithm. Besides, there are works on adaptive allocation. Adaptive nature of allocation is important due to that static allocation might not really handle all different situations in huge network loads. For example, in [3], authors defined an adaptive algorithm to solve the OFDMA mapping problem

DOI: 10.14738/tnc.36.1280
Publication Date: 21st November 2015
URL: http://dx.doi.org/10.14738/tnc.36.1280

in IEEE 802.16e networks. Authors showed that with the price of lesser throughput the active time of the SS can be reduced. They have proposed few different mapping algorithms to enhance system performance mainly using the resource mapping efficiency and they showed it by means of simulation tools.

In [4] authors considered highly QoS sensitive applications like multipoint video conferences and interactive videos games etc. and proposed optimum discrete bit loading aiming to satisfy all users. During resource allocation, when allocation of subcarrier gets updated at the starting of a time windows and channel gain is not known accurately at that moment and slow adaptive systems needs to have adaptive algorithms for better performance as per authors. They have formulated proportional resource allocation based on chance constrained programming for such slow adaptive systems that maximizes the average sum rate. Also they claim to maintain Jain's Fairness Index with target probability. The proposed solution is a hybrid of ant colony optimization and support vector machine. However satisfying all users comes to an expense of less number of users to use the network at the same time which might restrict the growth of number of users in the network. Authors in [5] proposed a decoupled solution like [1] but having an iterative and semi-distributed approach to implement a frequency domain scheduler. Their approach implements packet scheduler for all cells and users and interfaces of the wireless network in frequency domain to determine the global resource allocation. In [6] authors proposed an opportunistic scheduling algorithm considering power and subchannel allocation. They formulate the optimization problem targeting maximizing average sum rate for users and also claim to provide QoS requirement. They address non-convexity and coupled optimization same as this paper addresses but in different way to solve it. They also proposed two heuristic algorithms to reduce computational complexity. But the work may not be directly comparable to this paper as they consider device to device communication when planning for resource distribution which is not of the similar aspect as ours. In [8] authors aimed for sum capacity maximization by using MIMO OFDMA structure. They have proposed Lagrange dual based method first. This method is computationally expensive and hence they also proposed sub optimal solution to reduce complexity. However, though MIMO has definite benefits, for this paper we have proposed solutions based on SISO system and our work targets objectives similar to [1] but again our approach to the problem is very different than that of [1]. Due to that it is very close to our focus, we have mainly evaluated and benchmarked against this article comparing our outcomes.

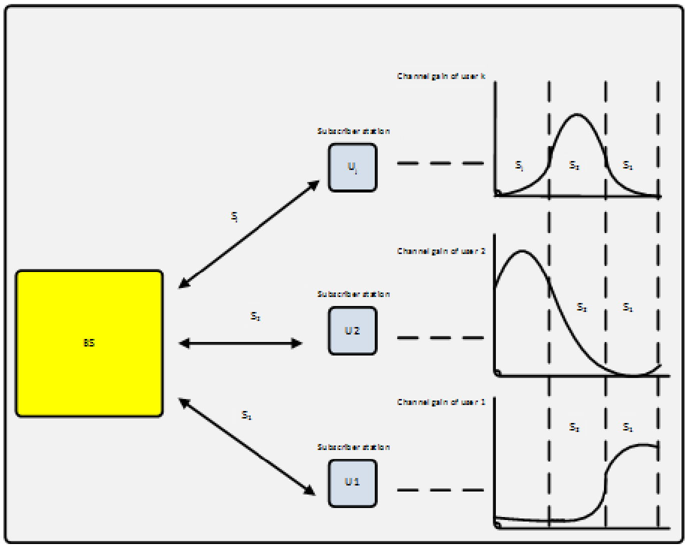

Figure 1: OFDMA based point to multipoint networks

Normally a cell or a site is prepared with its probable load capacity of that specific area. Also the resources starts to be used up incrementally rather than suddenly even though this case is also considered. Hence the necessity of the extra resources comes after the distribution of existing resources. But also, if the system is not ready before finishing of 100% of its resources, it might face a QoS crisis specially in terms of delay when new requests start to pop into the system because optimized resource decision usually are computationally expensive than non-optimized solution. However, most of the existing works focus on instantaneous sum rate or delay improvement or both. This is why we propose to do a long term system wide optimization with our proposed solution model which clearly shows significant improvement in performance and stability. Then we also perform our experimentation based on the proposed solution in a full-fledged network simulation with modeling using cross layer mechanism (Figure 2).

This paper describes the methods to enhance both speed (delay) and volume of data transfer/time slot (throughput), from system wide average performance perspective. As radio resource management (RRM) has direct impact on network performance, this paper proposes an effective solution that can handle higher number of users than that of existing reference systems with support of better QoS in terms of system wide fairness. Here we also consider the problem of fairness in this solution even though maximizing system sum data rate but many times the process shut off the user during scheduling time due to that they don't have good channel gain. We propose a threshold based resource distribution where the system sum data rate is increased but without shutting off current users, rather we propose to use a certain portion of available bandwidth from high rate consuming users by means of threshold calculation algorithm and share with starving users along with adaptive allocation to balance system wide performance. However in the beginning we would show

2 The Proposed Solution with System Model

The system model for the proposed solution is a cross layer model for OFDMA based connection oriented networks. Multilayer communication and contribution to accurate decision making on resource distribution is considered as depicted in the figure 2.

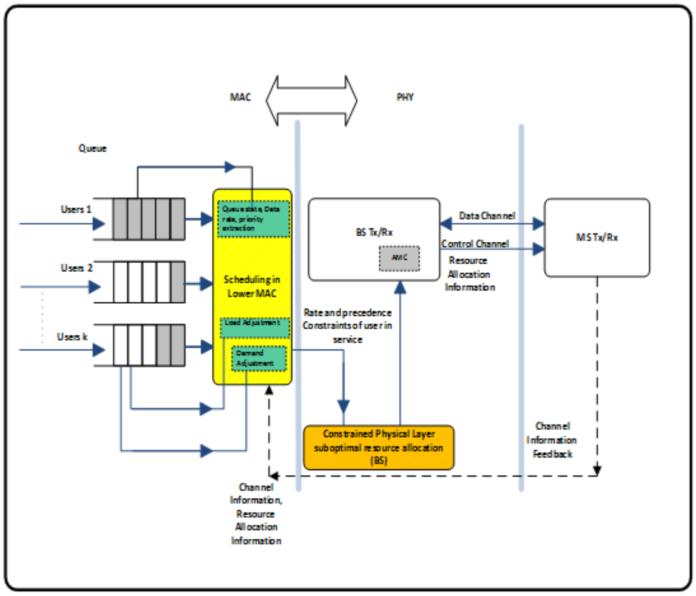

Figure 2: Base cross layer system model for proposed solution

This paper proposes solution for a long term effectiveness and stability for the system. The system model is based on figure 2. This paper introduces a new resource allocation strategy based on threshold optimized distribution and re-allocation approach. Initially, the system allocates resources based on cross layer system model which is adaptive. As the load increases, it triggers a new re-evaluation process based on proposed threshold model and most recent updated history of resource distribution. We consider the system to look into existing connected users' allocation from the history of the system to understand the minimum rate requirement (Rminimum) for current services in place for the user. Usually resources are not 100% occupied on 100% timescale. So the idle queue information is also included into decision making to stop the idle user and use that resource for the one requesting or, for the user currently starving. It may look a bit harsh from fairness perspective from the first look, but it has really good effects on the system performance as we will show by means of simulation based on the mathematical formulation proposed along with the algorithms.

Mathematically, the optimization problem considered here can be formulated based on [1] and shown below:

$$\max_{p_{k,n}\rho_{k,n}} \sum_{k=1}^{K}\sum_{n=1}^{N} \frac{\rho_{k,n}}{N} \log_2\left(1 + \frac{p_{k,n}h_{k,n}^2}{N_0\frac{B}{N}}\right) \quad (1)$$

Subject to

$$C_1: \sum_{k=1}^{K}\sum_{n=1}^{N} p_{k,n} \leq P_{total},$$

$$C_2: p_{k,n} \geq 0 \; for \; all \; k, n,$$

$$C_3: \rho_{k,n} = \{0,1\} for \; all \; k, n,$$

$$C_4: \sum_{k=1}^{K} \rho_{k,n} = 1 \; for \; all \; n,$$

$C_5: E[T_k] \leq \tau_k$.

Here,

- K is the total user number,
- N denotes the total subchannel number,
- B and P_{total} are the total bandwidth and power.
- $p_{k,n}$ denotes the power that is allocated for user k on the subchannel n
- $\rho_{k,n}$ is to be either 0 or 1, that indicates if user k is using subchannel n.
- fading and channel gain of user k on subcarrier n is $g_{k,n}$, having AWGN or additive white Gaussian noise, $\sigma^2 = N_0 \frac{B}{N}$, and N_0 denotes noise power spectral density [1]
- and its channel to noise ratio for the subchannel, $h_{k,n} = \frac{g_{k,n}^2}{\sigma^2}$
- user k receives the SNR on subcarrier n, $\gamma_{k,n} = p_{k,n}h_{k,n}$
- C4 shows each subchannel can be used by one user only.
- user k has the channel capacity of R_k which is given below:

$$R_k = \sum_{n=1}^{N} \frac{\rho_{k,n}}{N} \log_2\left(1 + \frac{p_{k,n}h_{k,n}^2}{N_0 \frac{B}{N}}\right) \tag{2}$$

Users bits are modulated in BS into N M-level QAM and then combined using IFFT into OFDMA symbols [34], [35], the subchannel-to-noise ratio using [37] be,

$h_{k,n} \geq 4$ and BER$\leq 10^3$,

and, $BER_{M-QAM}(\gamma_{k,n}) \approx 0.2exp\left[\frac{1.6\gamma_{k,n}}{2^{\gamma_{k,n}-1}}\right]$.

then, solving for number of bits, $r_{k,n}$, we have

$R_{k,n} = \log_2\left(1 + \frac{\gamma_{k,n}}{\varphi}\right)$,

where, $\varphi = \left(\frac{-\ln(5BER)}{1.6}\right)$, which is a constant (SNR gap) and $H_{k,n} = \frac{h_{k,n}^2}{N_0\frac{B}{N}}$.

We vary users by high priority users and general users (low priority). Thus if the users are of type high priority users, the rate calculation would be,

$$R_k = R_k + \frac{B}{N}\log_2\left(1 + \frac{pH_{k,n}}{\varphi_{high}}\right) \tag{3}$$

if the user is of low priority, the rate calculation will be

$$R_k = R_k + \frac{B}{N}\log_2\left(1 + \frac{pH_{k,n}}{\varphi_{low}}\right) \tag{4}$$

Equation 3.3 and 3.4 will be used to calculate rate for the user during allocation cycle.

- with the Physical layer resource allocation, we include the delay bindings of a user that can be extracted from MAC layer when a request is heard at the MAC in connection oriented networks such as WiMAX. We assume, if $E[T_k]$ is the average system time of user k and τ is its delay bound, the delay requirement of the user k can be formed as C5 considering a M/G/1 queue in a Poisson distribution based system [27]. The value of τ, is inversely proportional to the priority of the user.

To take this into consideration, we introduce an index of the users' delay bound and accordingly sort it according to the index; the lowest delay bound first and subsequently the rest of the index in increasing order. If the same delay bound is found for multiple users, first-in first-out mechanism is considered to serve them.

As the optimization of (2.1) involves both continuous variables $p_{k,n}$ and binary variables $\rho_{k,n}$, it becomes very hard to solve in computing environment. Furthermore, non-linear constraintsgives rise to higher complexities in order to find an optimal solution [1]. If we separate the power and subcarrier allocation in suboptimal manner, the complexity would reduce to almost half, since the number of variables of the objective function reduces to half [1]. So, we use maximum weight matching techniques from graph theory for subcarrier matching and allocation. For power allocation the proposed algorithm in [1] has very low complexity which can be used to get minimized power distribution among users and hence the higher possible values for (3.2). However, power optimization is not the focus of this work even though it can bring definite improvement in performance. For simplicity, we adopt greedy water filling method from [28]. The solution is different in many ways to [1] such as, it considers priority and QoS parameters such as delay whereas [1] does not. Secondly it uses graph theory whereas [1] uses Lagrangian relaxation which may require high computational resource for optimal solution. As subcarrier matching with constraints can be categorized as combinatorial optimization because it needs to find optimal allocation by matching from a finite set of subcarriers and exhaustive search is not feasible as it is expensive in operations. Thus Hungarian Algorithm is used along with the proposed solution. It is also called Kuhn-Munkres Algorithm (KMA) which works on bipartite graph to find a match optimally, that is, it is guaranteed that it would find an optimal solution and this is the reason we adopt this to get the subcarrier of maximum matching during the subcarrier allocation part. Also, it considers queue state of the connected SSs but [1] does not and it has different set of constraints to [1] which is another reason to be distinct as a solution. In this paper the core is of subcarrier allocation, assignment and related issues with detailed design. However, there is a commonality which is that both address the issue of enhancing system throughput but it does it in distinctive ways using the techniques and mechanism described in the next section that provide higher performance of the overall system and this would be demonstrated by simulations in WiMAX based network simulator. In the next part the subcarrier allocation algorithm will be described.

2.1 Subcarrier allocation algorithm

During subcarrier allocation, the first thing that the system will need to see is what is the rate and requirements. We formulate here, assuming B bits per symbol loaded for every subcarrier, if a user requested R bps with subcarrier spacing to be D_f Hertz, the total number of subcarriers to be allocated to the user would be,

$$S = \left\lceil \frac{R}{(BD_f)} \right\rceil$$

There are a few steps involved in the proposed subcarrier allocation algorithm, they are firstly collecting MAC layer information of how many subcarriers is needed by a specific user, labeling users and subcarriers to form bipartite graph, assignment of weight, optimizing the assignment and collecting optimized allocation information. This completes the bit and power allocation. After that this resource allocation information is finally sent to SS through dedicated control channels.

The proposed solution converts the allocation problem into a weighted bipartite matching problem. The matching graph is denoted as G=(U,V,E). U is referred to as subset of users, V is subset of vertices that represents the subcarriers to be assigned, E is the subset that has the set of selected subcarriers, U and V are disjoint subsets. Next we find a matching from U to V on a one-to-one basis. A valid matching is constrained by requirements like C4 in (2.1) which restrict that one subcarrier can be used by one user at a given time, which eliminates interference probability. We need to assign the weight for each edge expressed as w(e). Before assigning the weight, we need to sort the users according to priority and delay bound to meet the delay requirement expressed by C5 to ensure that the serving user can have appropriate QoS. Then after assigning the channel gain as weight w(e) for each edge e∈E, the problem now is converted to constrained weighted bipartite matching problem. After weight assignment, the calculation for matching starts. However, the bipartite graph produced does not provide optimized assignment, which means that the number of subcarrier a user gets, might be higher if we do further optimization on the graph. To get optimal solution for a match, Kuhn-Munkres [29] algorithm (KMA) is used in our solution for highest possible cardinality to achieve highest performance in terms of data rate because for single objective optimization, it is known that KMA can always find the optimal matching for a bipartite graph with O(n3) computational complexity [30]. The KMA is based on the procedure of the Hungarian algorithm [31]. The subcarrier allocation algorithm we came up with which includes matching part using KMA is provided here next.

Assume that,

A = {1, 2, …, N}, the subchannels in set.

S(A)=size of *A.*

N = total number of subchannels.

N_R =*Remaining channels*

R_i = set of user requests, i=1 to M

R_i = {R_1, R_2, …, R_M}.

Last request=$R_{L,}$

q = the request of user k for rate, $q\neq 0$

FS_k = the set for user k consisting of allocated subchannels

Z=Size (FS_k) = total subchannels of FS_k.

R_k = the total allocated rate/capacity for k

The steps of the allocation are as follows:

Start

Initialize:

$\boldsymbol{R_i = 0}$.

1. Sort the users as per the delay bound and priority information received from MAC layer in ascending order, and start processing from top to bottom. Get the number of subcarriers required to serve the current request: assign n_k = Z, for every user and $R_k > q$ and n_k is minimum, for all FS_k.
2. Repeat process 1 and 2 where total of $\sum n_k > S(A)$. $Ri = \{Ri - R_L\}$. R_L not accepted, as it is R while $S(A) <= \sum n_k$. Otherwise, N_R= size of $(A) - \sum n_k$
3. Run KMA for getting optimized FS_k for every admitted k having the number of the subcarriers in step 1, *S(A)* is unchanged at this stage.
4. For all admitted k and $R_k > q$, update *A, with* $(A - \{FS_k\})$; R with $(R - q)$
5. $R_i=R_{(i+1)}$ if $R_i \neq R_L$
6. When $S(A) \neq 0$, $R \neq \{\}$, $n_k = n_k + 1$. Otherwise, if SFC = 0 then $R = R - R_{Lastone}$.
7. Repeat 1 - 7 while size of (A)> n_k
8. Assign p for each element in R_i using Greedy waterfilling
9. Call module **rate_allocation(selected_subcarrier)**

End

//module rate_allocation pseudo code

rate_allocation(selected_subcarrier)

{//start

if user priority == high{

use equation 2.3 to calculate rate assignment

update system of the assignment

}

else if user priority == low{

use equation 2.4 to calculate rate assignment

update system of the assignment

}

} //end

3 Threshold Based Optimal Resource Re-Distribution Algorithms

In this paper we consider a special problem or scenario where the number of users might get high typically higher than what is originally planned. Putting extra BSs is expensive, infeasible and time consuming. We propose a solution to this problem which exploits the idea of greedy type of resource distribution and re-distribution over time. We assume that any user with higher priority enjoys the best possible resource in terms of sub channels and power allocation, and the user has lower need of bandwidth because it already has it more than what it requires. This type of users are easily identifiable from the history of the allocation and also in MAC layer allocation log. However, we put the constraints that if the delay bound of a user (τ_k) is lower than other users in the same type queue, the previous one would be served first if the required power is available and there are enough subcarriers to entertain the request. If available power is not enough, it needs know it can manage it from within the existing resources used or unused. If it does not get that, the user would be rejected. However, when it needs more subcarrier if there is not enough available, it would start to search for it. This is the point where the algorithm would start to subtract 1 subcarrier from each of the highest rate consuming nodes from the sorted index in the MAC layer as per the proposed formula. This process continues until the highest consumers' rate becomes reduced to threshold maximum (threshold value T_h will be derived in this paper next), and the new users are put under the continuous system flow of assignment and allocation. Finally, when the threshold T_h in (3) is reached, no further re-assignment is done and the user request for resources might be queued or dropped.

3.1 The major steps of the algorithm

3.1.1 Initialization:

Besides system parameters described earlier we need to add few more parameters to run the proposed solution. The new main parameters here that need to be added are threshold values (to be used in the next section) T_h, the set $\{\sigma_i\}_{i=1}^{K}$, where σ indicates the value for user *i* which signifies the proportion of resources fixed for his service as per the service layer agreement (SLA), K is the number of users.

3.1.2 Run distribution:

If the system has enough resources unutilized then this part still would be running the resource allocation algorithm described earlier. However, as soon as the resources are utilized with assumed 100% of the total resources, it starts the re-distribution process as the systems proactively calculates and run algorithm for saving and reusing existing resources described next. This process can start earlier for example when the resources are 50%-80% consumed to reduce computational delay as there will be ample history data of existing users to be used.

3.1.3 Run re-distribution:

This section will start the full optimal solution once the resources are already in use to a full scale. The system will re-examine with the existing user list and its allocation. The system starts the following new process in the system. The solution in paper 4 performs requirement formation and assignment of subcarrier in one phase and power allocation in another phase. However, when the number of users exceeds the limit, and they need to manage subcarriers as well as to know the power availability and required power level for each subcarrier $p_{k,n}$ (from section 4.4), the total maximum power P_{total} is fixed for a BS and it is being used by users already admitted in the system. This process will trigger once the resources reach $\frac{N}{2}$ or $\frac{P_{total}}{2}$ whichever comes first in this case. The reason is that it can get ample time to process the steps below to avoid processing delay that would affect the QoS. At this stage the systems acts like its resources are already exhausted but in reality it still has resources. This heuristic process adds benefits to the system resource management in terms of entertaining more users' request and maintaining the QoS performance in course of time.

Sub Steps of the algorithm for reallocation re-distribution

- Run extra-user accommodation process in case of (total resources)/2 <= (total usage at the moment),
    - Search through all users current allocation information
    - extract current usage information

- ✓ if service queue is idle, remove the connection and delete the whole resource of that connection type from the allocation
- Extract users with 0 requirement, and these would be the ones that already have the highest amount of resources for their SLA. We define a requirement index for all the users indicated by the following equation:

$$R_k^{index} = \begin{cases} q, & R' \leq R_{minimum} \\ 0, & R' > R_{minimum} \end{cases} \quad (5)$$

Where R' is the already allocated resources to the user and $R_{minimum}$ is the required minimum rate for the service currently using, q is defined in paper 4, section 4.3 which is the rate requirement of a new requesting user. We also assume, a user k having current rate beyond the $R_{minimum}$ has a necessity of rate request indicated byR_k^{index}.

- Subtract $N^{\oplus}$ subcarrier from each of the highest rate consuming range
  - ✓ Follow the new equation of subtracting

$$N_k = \left[N \left(\frac{\sigma_k}{\sum_{i=1}^{k} \sigma_i} \right) - 1 \right] \quad (6)$$

and,

$$N^{\oplus} = \sum_{i=1}^{K} IntegerOf(N_i) \quad (7)$$

Where, k=1, 2, 3, …, K, N^{int} is the nearest rounding, and $\{\sigma_i\}_{i=1}^{K}$the set of constant values as per SLA to provide proportional part of resources for service. N_k indicates deduction of 1 subcarrier from the required total number of subcarrier for his service.

- Determine if

$$\sum N^{\oplus} \geq R_k^{index} \quad (8)$$

  - ✓ if this condition is met, then it is enough to serve current rate request, assign it to the requesting user, at the same time distribute required power for bit loading using greedy water-filling.
- if resources get freed anywhere in system, assign first to the users that contributed to the previous process with subcarrier deduction until their previous allocation as per SLA, provided if they are requesting more resources. The reason is that they are usually of higher priority users.
- if more resources are available, keep it as reserved resource to serve new resource request incoming by incrementing resource indicators.
- if resources are available to serve the whole user base, use suboptimal solution.

At this stage we need to know the threshold value and how it is calculated. This is described next.

3.2 Calculation of threshold value Th

To calculate the threshold value T_h, first (2) can be written in terms of data rate as:

$$\boldsymbol{T_h = \frac{R_k}{2}} \quad (9)$$

This depends on the number of subcarriers assigned (F_k in section 4.4.1) to user k, and power assigned to user k is expressed as:

$$\boldsymbol{P_k = \sum_{i=1}^{x} p_{k,n_i}} \quad (10)$$

Where,

P_k is the total power assigned to user k,

p_{k,n_i} is power assigned to subcarrier n of user k and i=1 to x, and

x is the number of subcarriers assigned to user k in set F_k (from section 3.4.1).

Figure 3.8 shows the whole process of the solution, and the dotted box shows the threshold based re-distribution module.

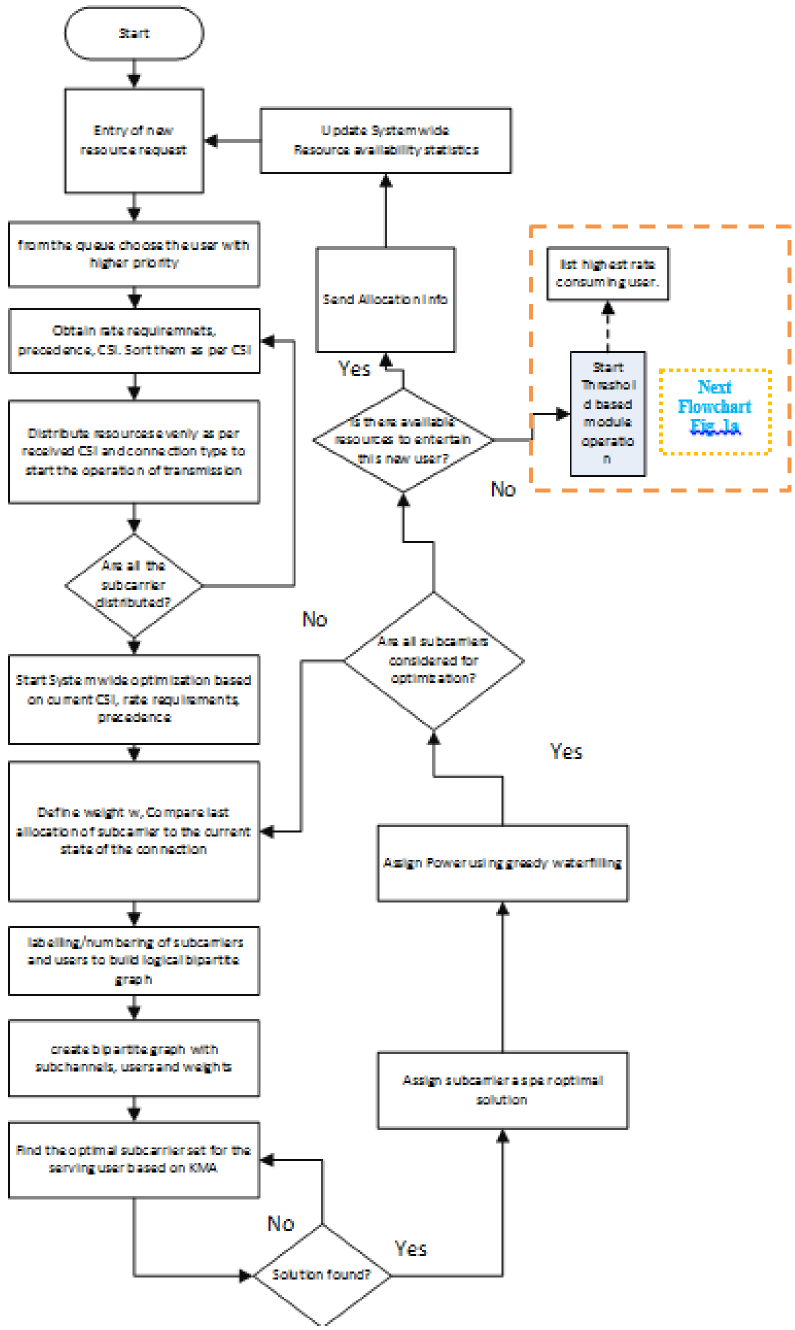

Figure 1: Total algorithm flow of the proposed solution along with threshold based distribution module (added flow chart next due to space)

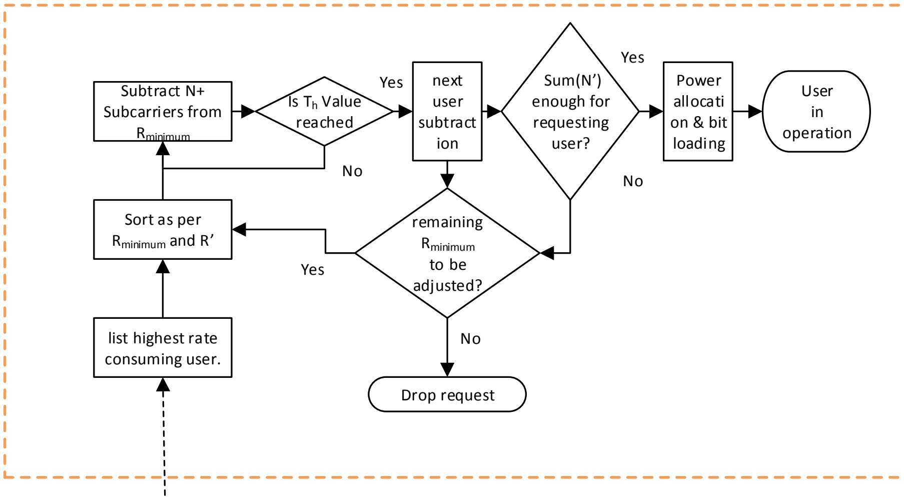

Figure 1a: Th based resource re-distribution module flowchart

4 Performance Evaluation

The results obtained from carrying out several simulation experiments in the current study were examined and evaluated in order to prove the performance of the proposed solution under various conditions. The aim of such experiments is to assess the ability of the proposed solutions and test its effectiveness.

In the following experiments, the threshold based optimal resource reallocation and re-distribution increases the total sum rate and as well as QoS performance. It is achieved by reusing existing subcarriers and power more efficiently to allow low priority nodes to use adequate resources while not affecting the higher priority nodes. It creates system wide better network experience contributing to higher overall sum rate and QoS benefits. To evaluate the performance of the proposed solution, we compare first with Zukang Shen et al. (Ref [1]) under the same conditions here.

5 Simulation and Results

This paper investigates different cases with the simulation. We mark the proposed algorithm to be AORAA in the graphs because the algorithm works in conjunction with AORAA proposed in paper 4, though the process of paper 4 has a number of changes to adopt the proposed algorithms in this paper. To distinguish, paper 4 proposes the optimized subcarrier assignment and allocation algorithm in some details, while paper 5 proposes a threshold based resource balancing algorithm with mathematical formulation that improves the long term network performance.

5.1 Effects of Various Network Load

We implement the steps shown above in QualNet WiMAX simulator (Advanced Wireless Model, Version 1), the results produced shows improvement that we will explore now. However, for smaller number of nodes performance variation is not too much as the resources are enough to support the load and this is why we will show the simulation during benchmarking. Here we start from higher number of nodes contributing to the scenarios so that the improvement gets clearer with the provided graphs.

A. Scenario with 60 nodes

In this scenario there is a maximum of 60 nodes. The base parameters are as those described in paper 4. The plots are presented here for the two algorithms separately and benchmarked with other existing solutions at the later sections.

1. Using reference [1]

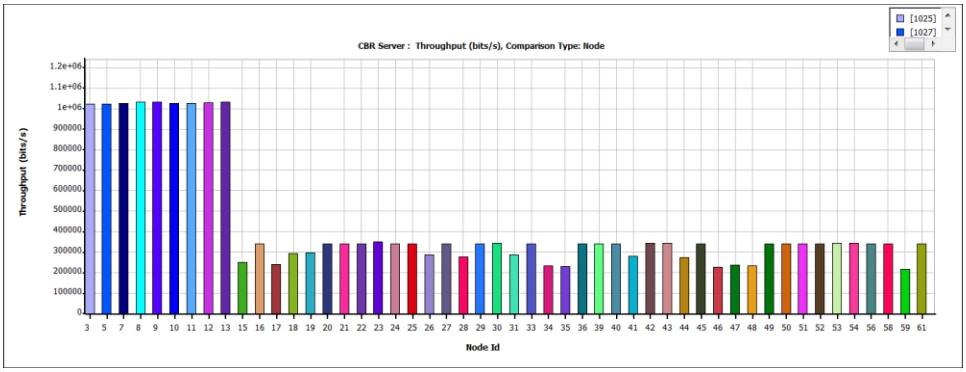

Figure 2: Throughput plot for reference [1] for 60 nodes

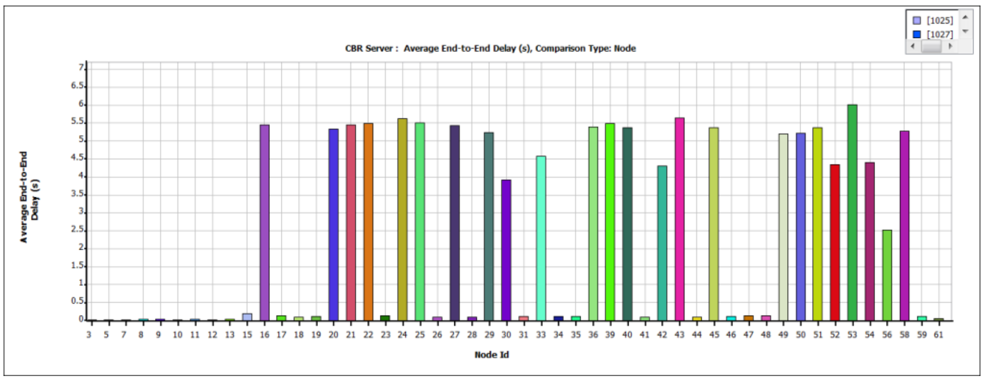

Figure 3: Average end-to-end delay plot for reference [1] for 60 nodes

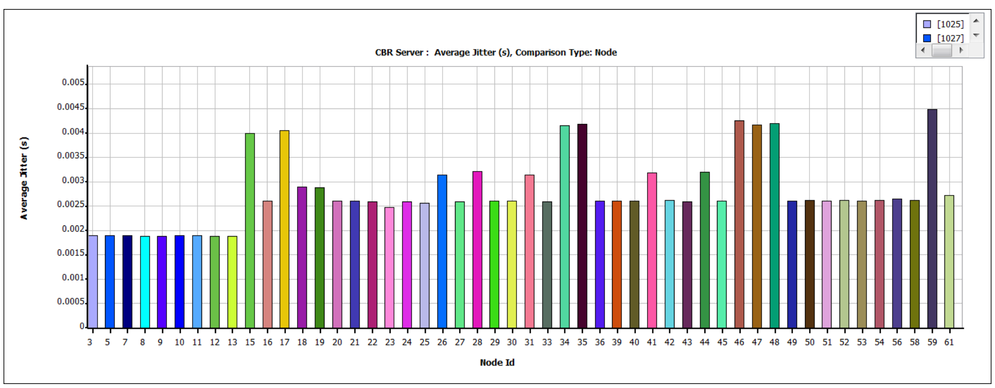

Figure 4: Average jitter for reference [1] for 60 nodes

2. Using AORAA

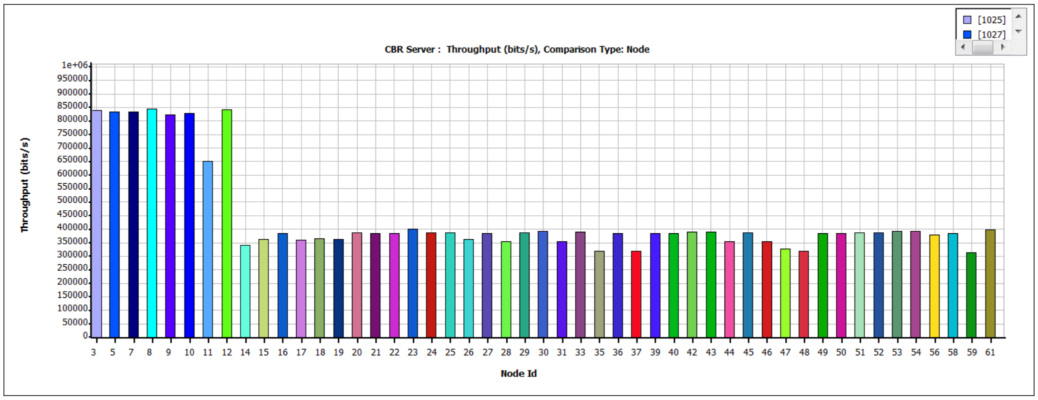

Figure 5: Throughput performance using AORAA for 60 users

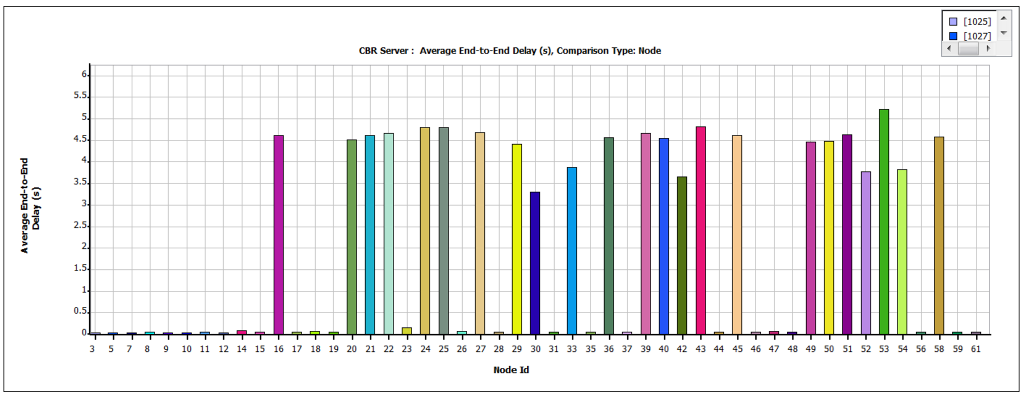

Figure 6: Delay performance with AORAA for 60 users

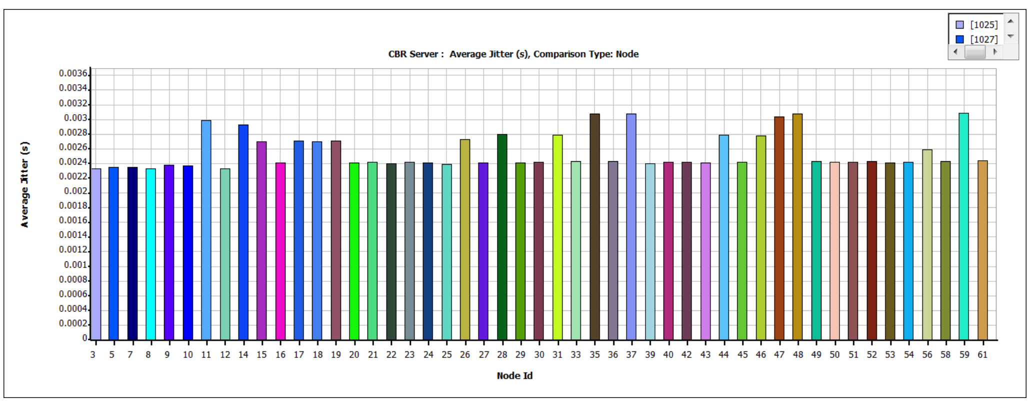

Figure 7: Jitter with AORAA for 60 users

► **Throughput and delay analysis of 60 nodes case**

In case of throughput for reference [1], it is seen that the maximum of 20% nodes with higher priority and requirements have the same throughput which is around 1Mbps whereas the throughput for rest of the 80% nodes are having around 250 Kbps. The delay of the 80 % nodes with lower priority is having average of around 5s. Some of the nodes have delays in millisecond scale. When AORAA uses the throughput of the higher priorities are reduced to around 850Kbps rather than keeping it 1Mbps. The other 80% of the nodes experience throughput of around 350Kbps compared to 250 Kbps in the case of [1]. The delay of

AORAA is also distorted but with the range of 4-4.50 s whereas it was near 5 s in case of reference [1]. This stemmed from the fact that the most disadvantaged nodes would be provided with the share of the resources from the best and highest resource-endowed nodes until the threshold (T_h) value is reached.

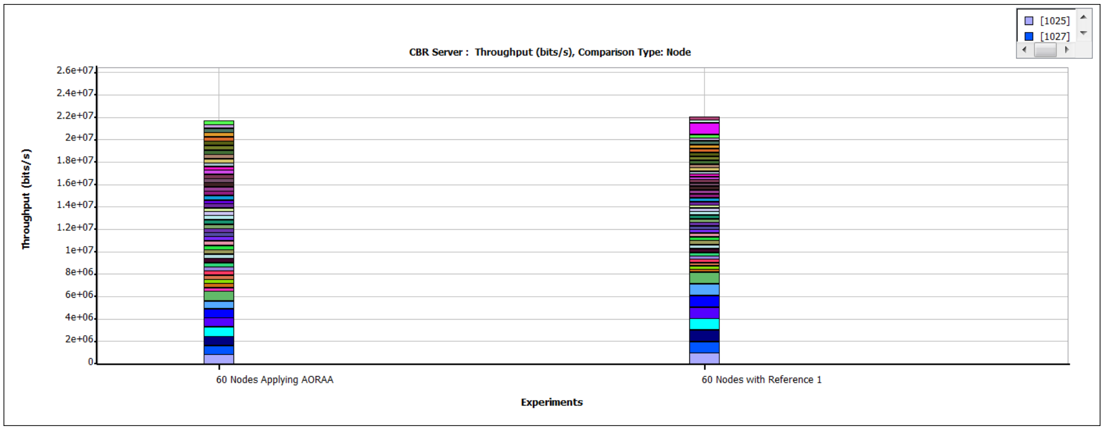

Figure 8: Throughput performance using AORAA and reference [1] for maximum of 60 nodes

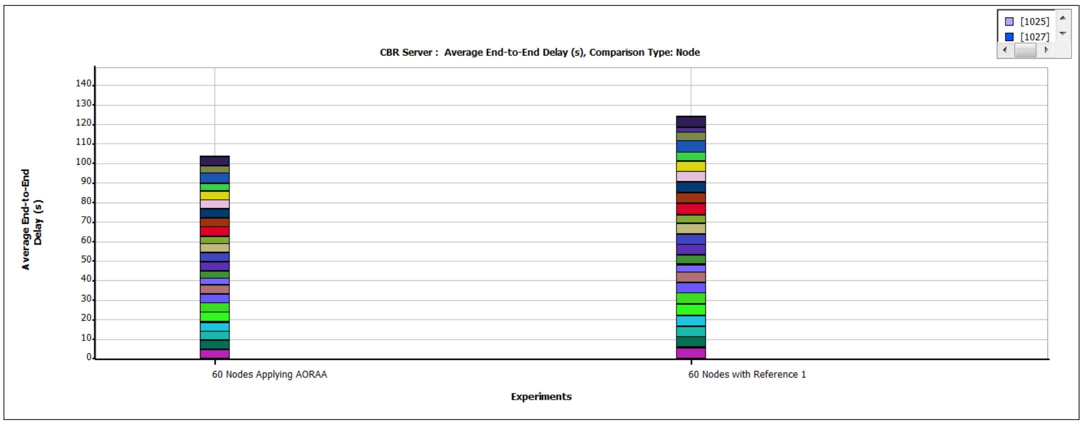

Figure 9: Delay performance using AORAA and reference[1] for maximum of 60 nodes

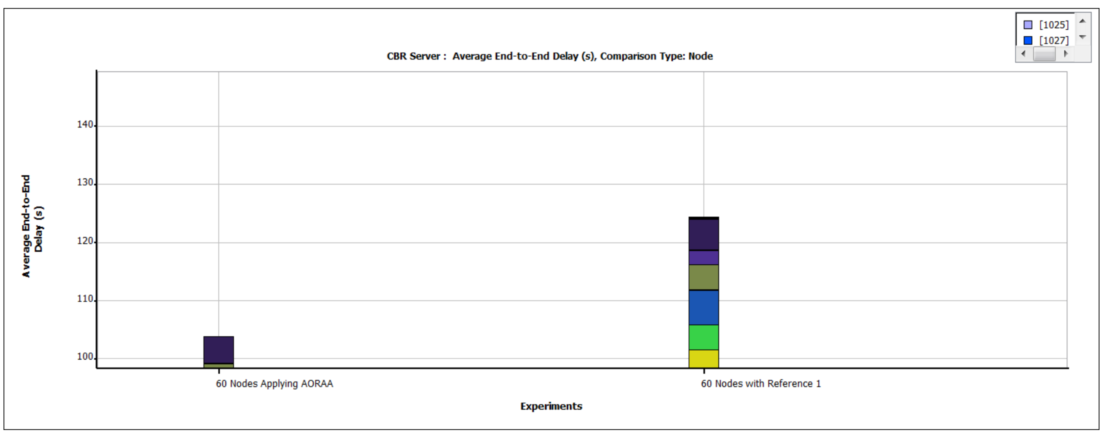

Figure 10: Delay performance using AORAA and reference [1] for maximum of 60 nodes (zoomed)

B. Scenario with 80 nodes

There are a maximum of 80 nodes for this simulation. Like previous scenarios the 20-80 division of priority is followed as before. The following figures show the output from applying reference [1] and AORAA. The comparison is given after few figures below.

1. Using reference [1]

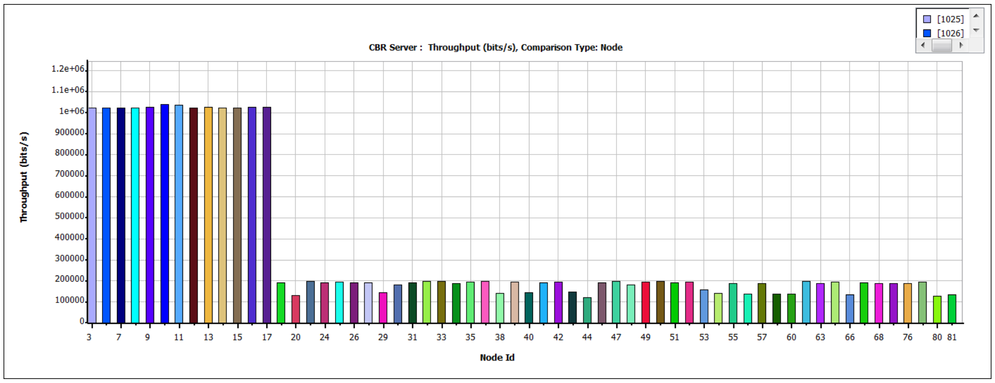

Figure 11: Throughput performance of reference [1] for maximum of 80 users

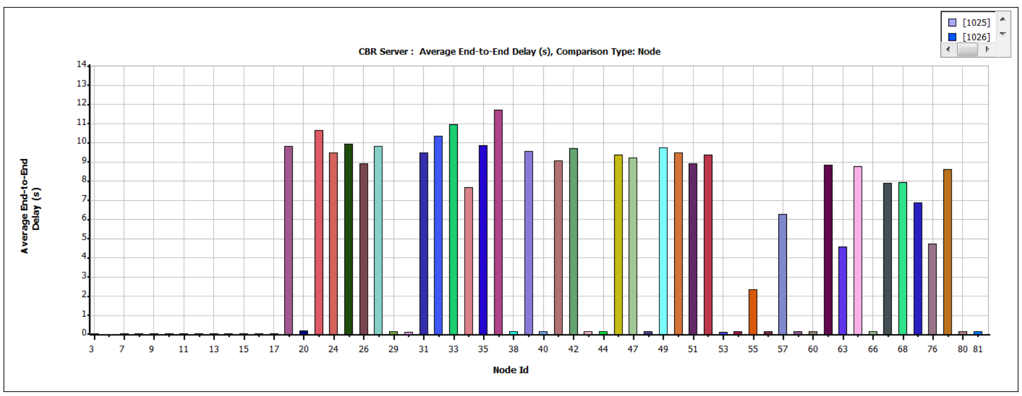

Figure 12: Delay performance of reference [1] for maximum of 80 users

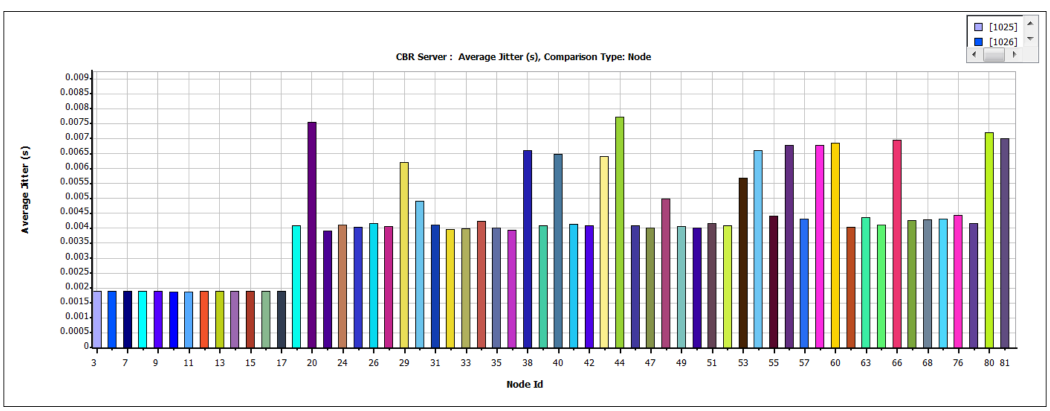

Figure 13: Jitter performance of reference [1] for maximum of 80 users

2. Using AORAA

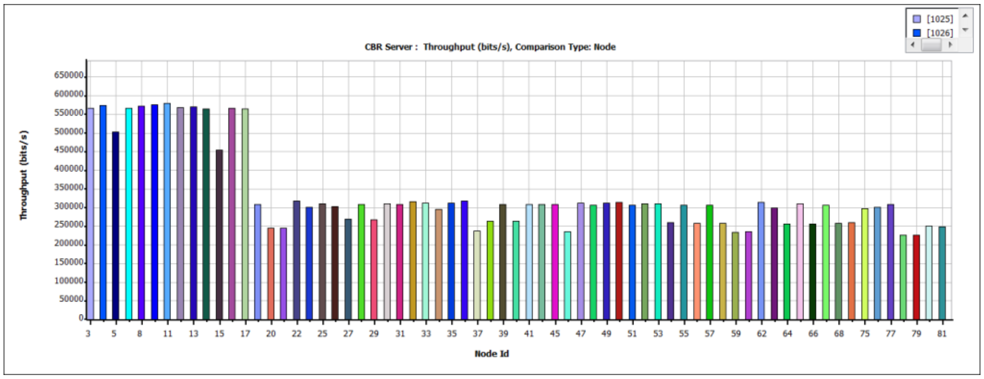

Figure 14: Throughput performance using AORAA on 80 users

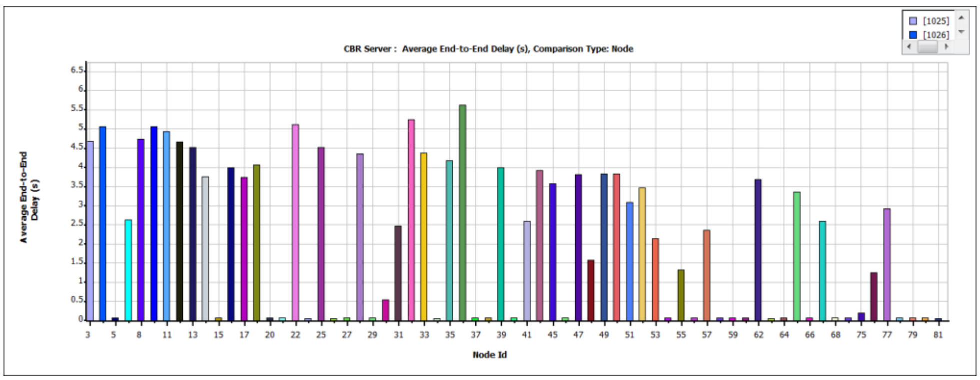

Figure 15: Delay performance using AORAA on 80 users

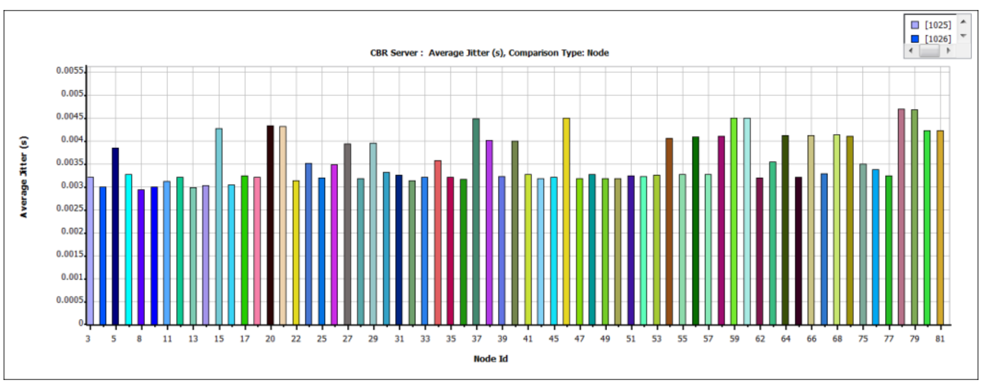

Figure 16: Jitter performance using AORAA on 80 users

► **Throughput and delay analysis of 80 user case:**

In this scenario, the 20% nodes with higher priority get what they should get which is around 1Mbps; and the rest of the 80% of lower priority nodes get the throughput of around 150 Kbps (where each of them should have 512Kbps at the best case). With AORAA, 80% of the nodes with lower priority get around 300Kbps bandwidth which is around double that of reference [1]. Then, for the average end-to-end delay, both cases of AORAA and reference [1], the delay performance is not uniform. But in case of AORAA the 20% high priority nodes suffer some distorted delay as well. However, for the average performance of the whole system as shown in figures 19, 20. AORAA shows better delay and throughput performance than reference [1].

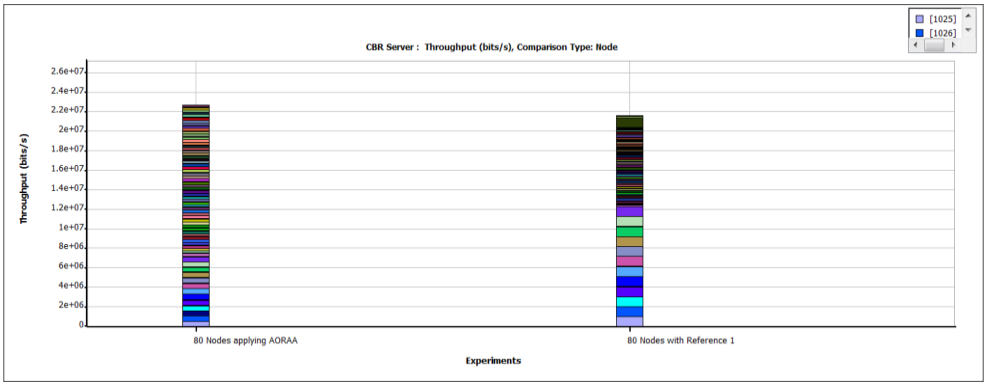

Figure 19: Throughput performance comparison of AORAA and reference [1] for maximum of 80 nodes

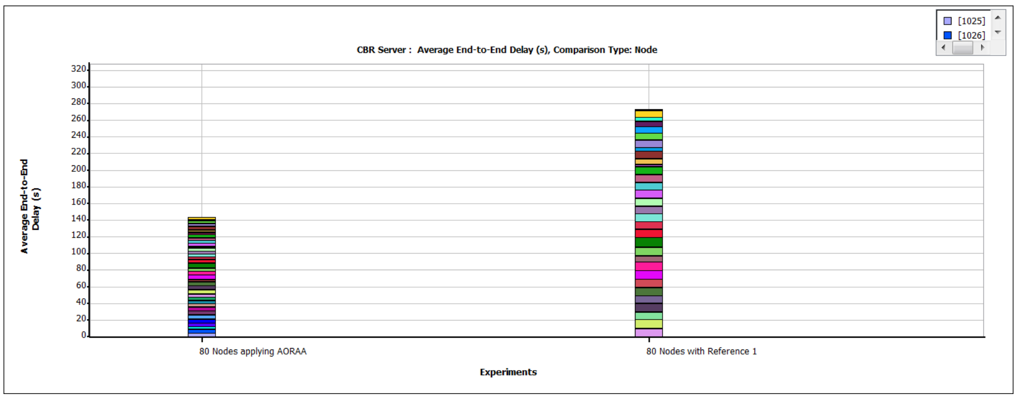

Figure 20: Delay performance comparison of AORAA and reference [1] for maximum of 80 nodes

C. Scenario with 100 nodes

In this scenario, there are a maximum of 100 users considered. This has quite a big effect on the overall system performance, however using the proposed algorithm the enhancements are proved here using the plots below comparing it to reference [1]. The discussion will follow next.

1. Using reference [1]:

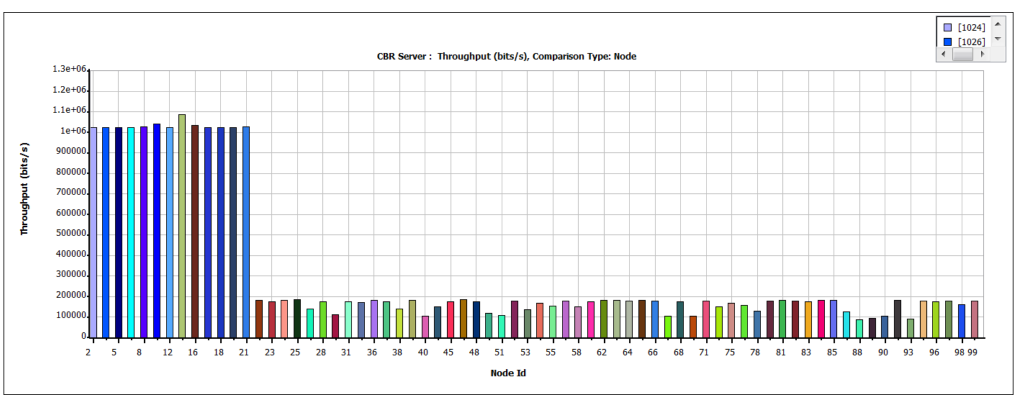

Figure 21: Throughput performance of reference [1] for maximum of 100 users

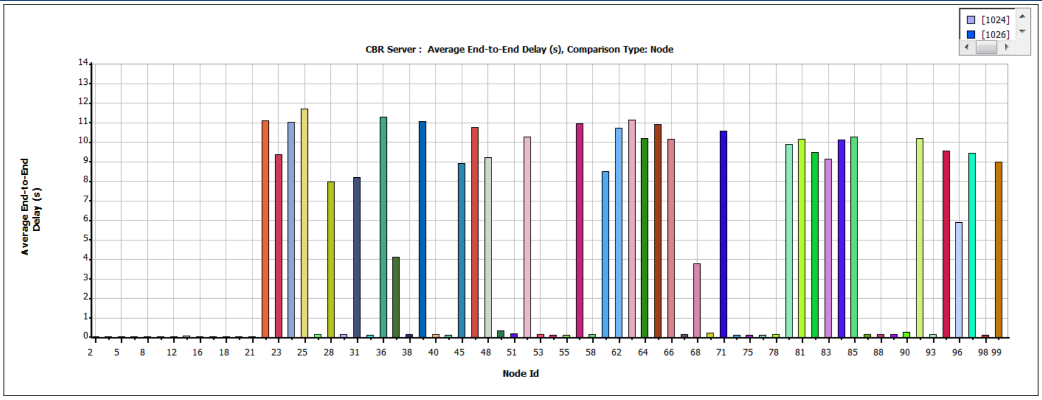

Figure 22: End to end delay performance of reference[1] for maximum of 100 users

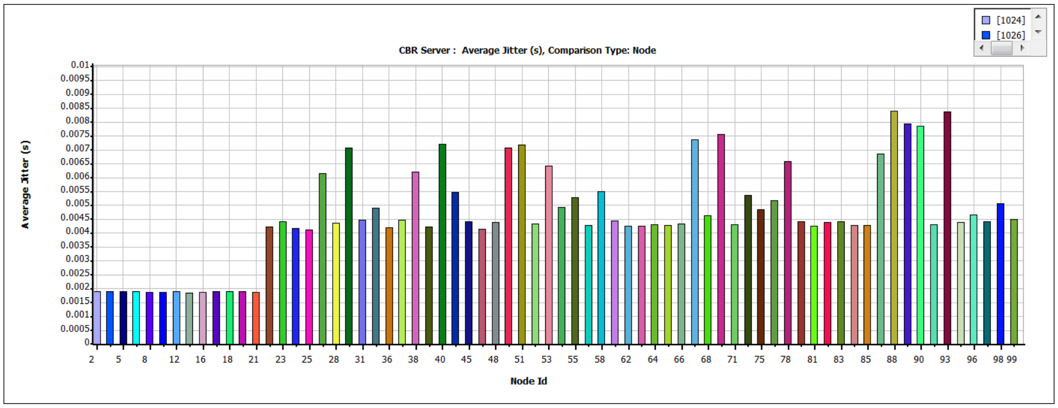

Figure 23: Jitter performance of reference [1] for maximum of 100 users

2. Using AORAA:

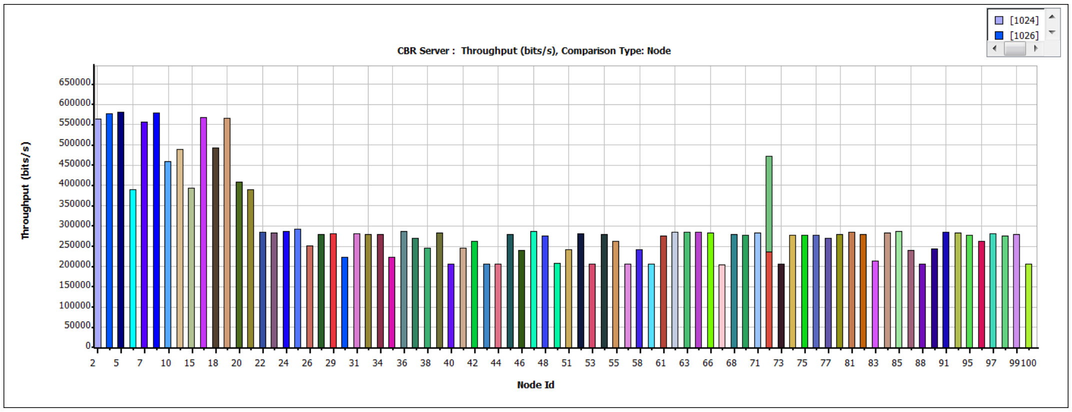

Figure 24: Throughput performance of maximum of 100 nodes using AORAA

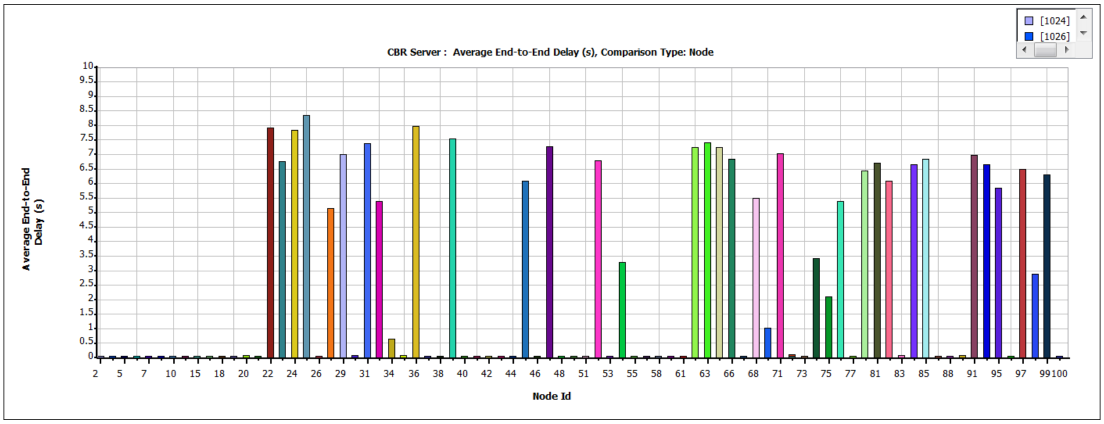

Figure 25: Delay performance of maximum of 100 nodes using AORAA

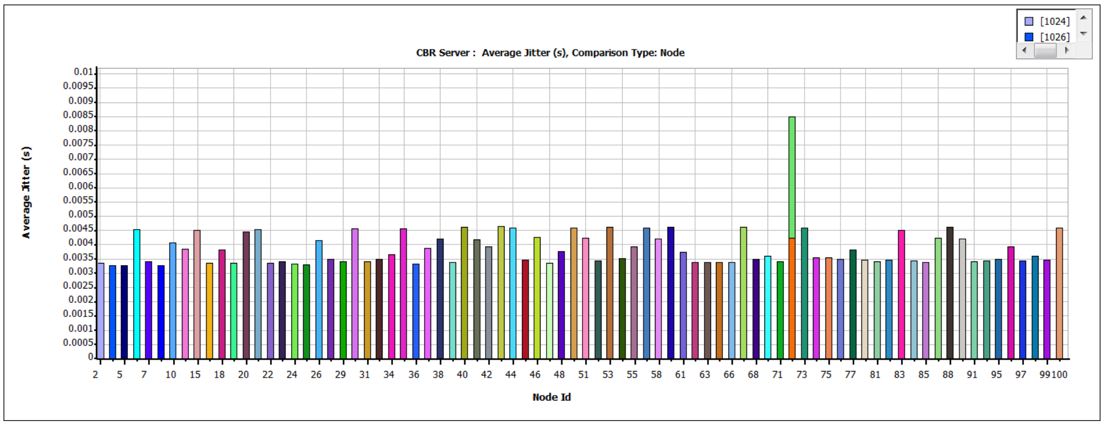

Figure 26: Jitter performance of maximum of 100 nodes using AORAA

► **Analysis of throughput and average end to end delay for 100 nodes:**

The compared values are given in a tabular format next. It is observed that for maximum of 100 nodes, the throughput of AORAA for whole downlink system is around 24.5 Mbps whereas reference [1] has around 22.5Mbps. For the case of delay it is observed that AORAA's average delay is 220s whereas for reference [1] delay is 320s. So it is clear that in both the cases the performance is enhanced.

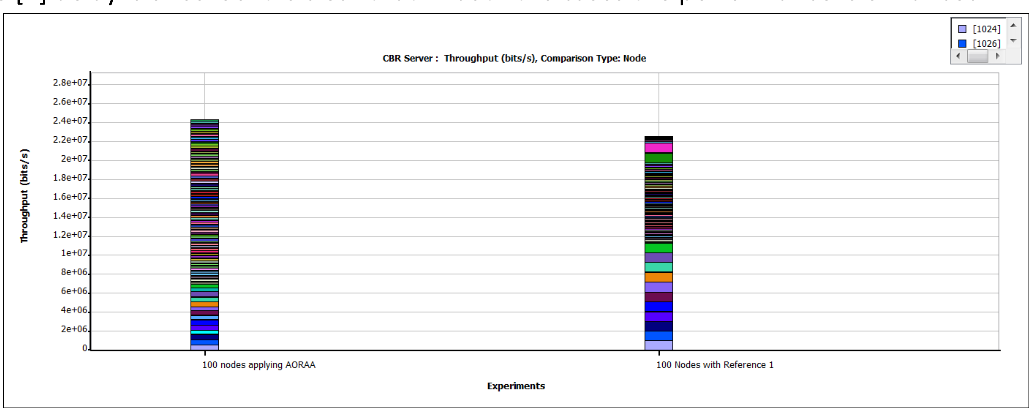

Figure 27: Throughput comparison of AORAA and reference [1] for maximum of 100 nodes

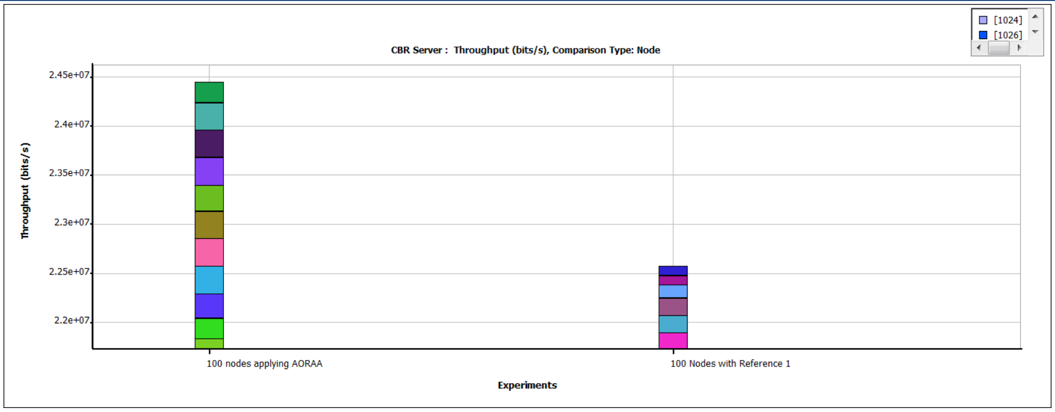

Figure 28: Throughput comparison of AORAA and reference [1] for maximum of 100 nodes (zoomed version of Fig. 27)

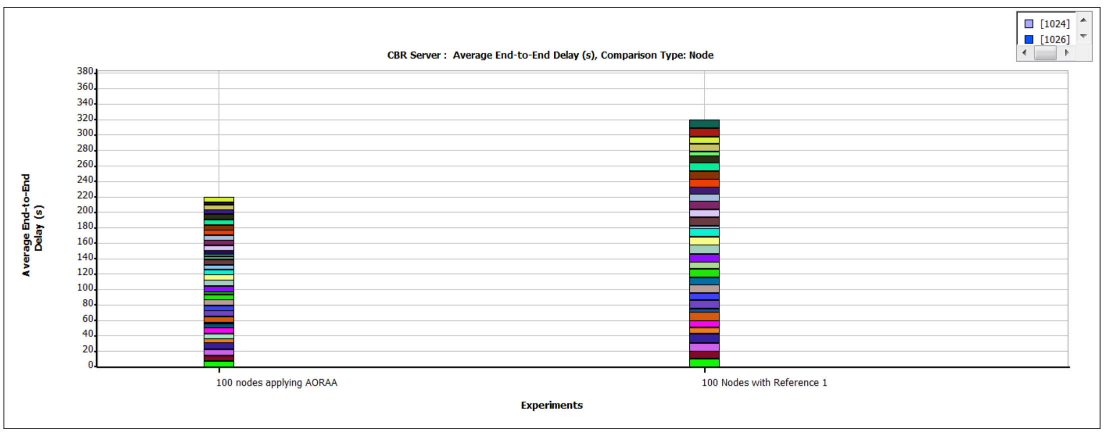

Figure 29: Delay comparison of AORAA and reference [1] for maximum of 100 nodes (total delay of nodes)

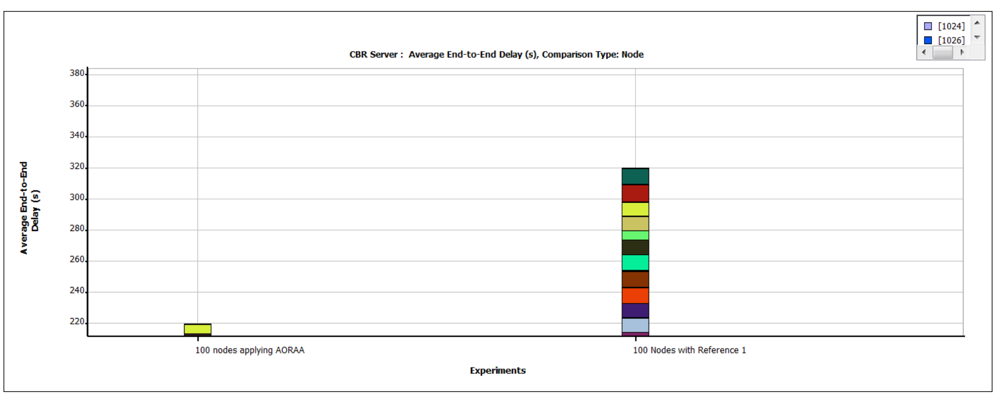

Figure 30: Delay comparison of AORAA and reference [1] for maximum of 100 nodes (zoomed)

D. Comparison of Global Trends of AORAA And Reference [1] for Threshold Based RA

From the plots below it is observed that the throughput increases as long as resources are available and the number of users increase. It happens for both the compared algorithms. Delay also increases the same way for both algorithms. However, by comparison it is shown that AORAA enhances the overall system performance in terms of throughput and delay over reference [1]. Also it could be seen that this happens mostly when the number of users are increasing, this is when AORAA acts to balance the system.

1) AORAA

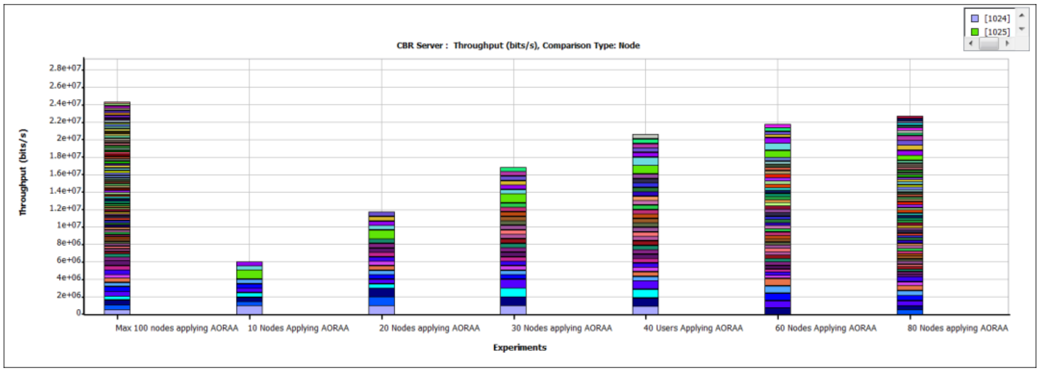

Figure 31: Trend of throughput for AORAA for different load

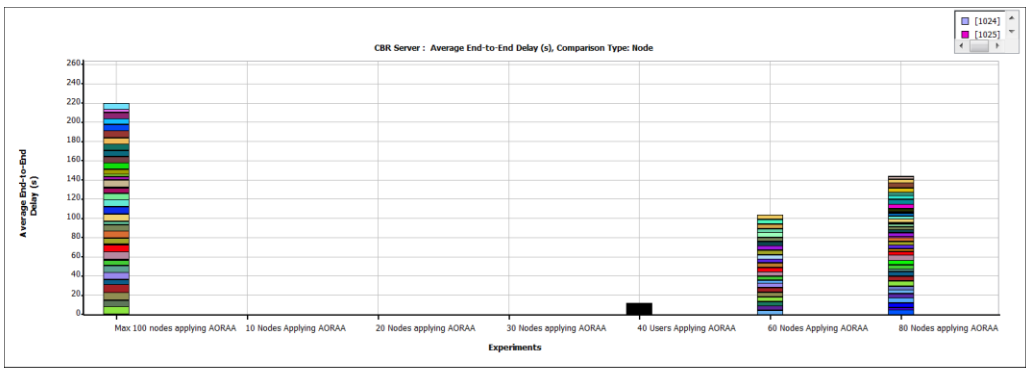

Figure 32: Trend for average end to end delay for AORAA (100 nodes at left most)

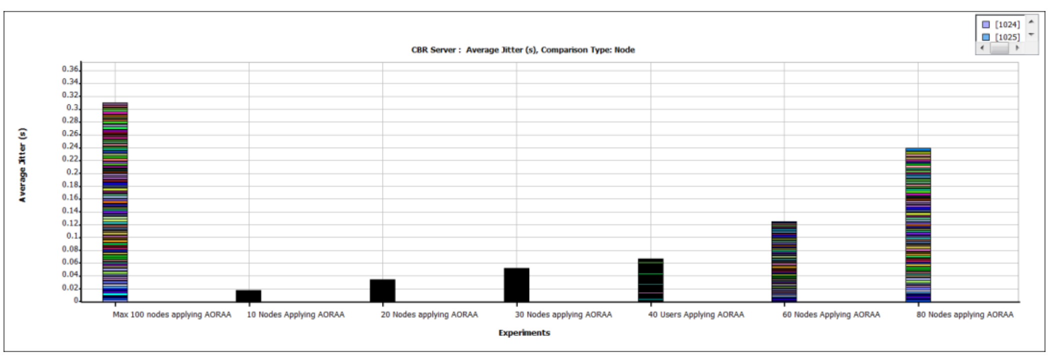

Figure 33: Jitter trend for AORAA (left most is the 100 nodes)

2) REFERENCE [1]

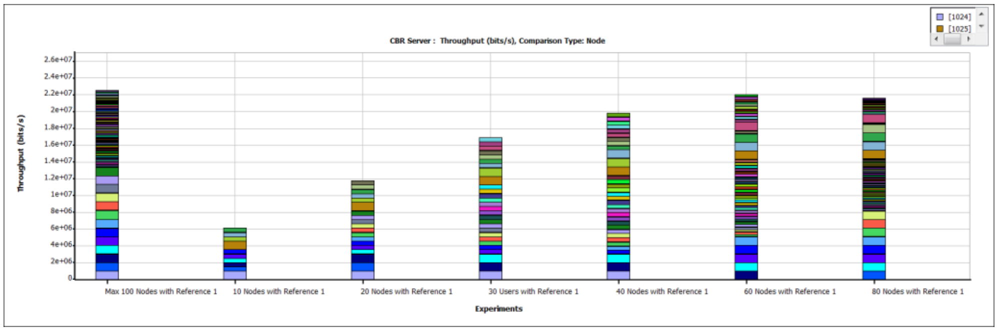

Figure 34: Throughput performance trend for reference [1]

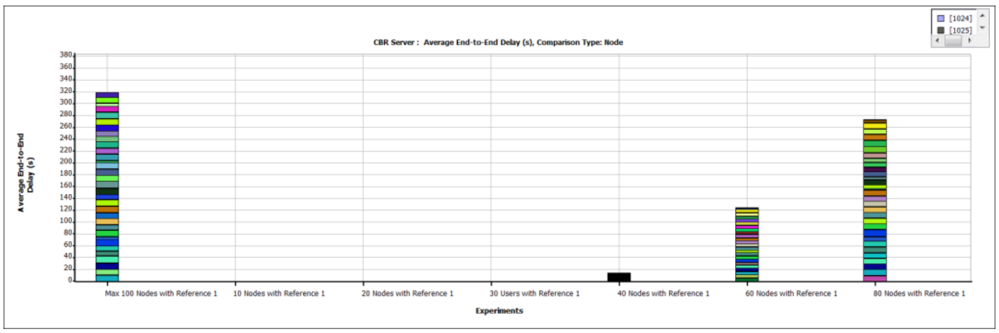

Figure 35: Delay performance trend for reference [1]

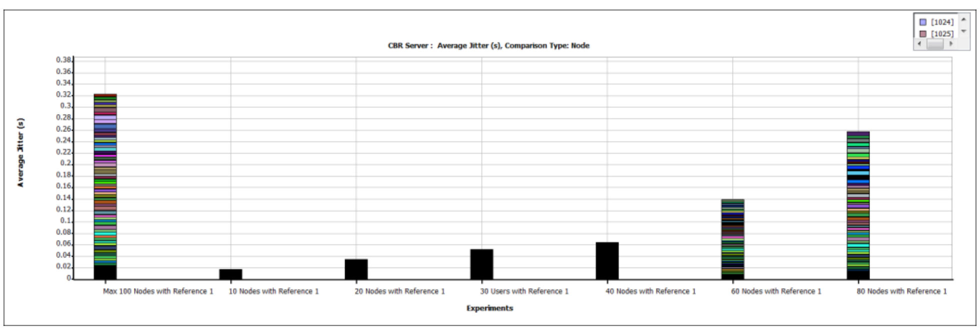

Figure 36: Jitter performance trend for reference [1]

6 Fairness Comparison and Algorithm Complexity

We need to compare fairness of the algorithm as well because this is an important performance parameter in a scheduling algorithm. In this section, we evaluate Jain's fairness index. On a whole fairness is reduced as more load is added to the same existing resources which has happened to all the comparing algorithms in figure 37. However, with the same condition, we will observe which one is performing better. It could be observed that threshold based AORAA achieves higher fairness value in the form of Jain's fairness index till around drop zone as marked in figure 37. Then it drops for some time and then increases than before for higher number of user. This is attributed to the threshold-based logic of the proposed solution. At that point it behaves a bit unfairly because it redistributes a portion of the high resource-consuming nodes to the starving nodes due to higher load on the system. Then it raises again in the raising zone shown in the same figure. Whereas, fairness of ref[1] drops almost linearly and the fairness index is most of the time lower than the proposed solution. The algorithm complexity is at most $O(n^3)$. It provides optimal solution for subcarrier matching and guarantees to find matching in every cycle which results in higher throughput. In contrast Z. Shen et al. [1], cannot guarantee optimality and thus have performance gap, to guarantee same as proposed solutions, it requires O(n!) which is much higher comparing to our solution

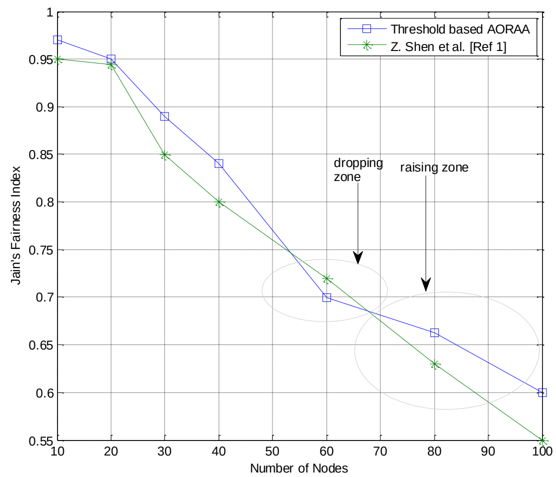

Figure 37: Fairness comparison using Jain's fairness index.

7 Conclusion

This paper described the proposed system AORAA for subcarrier allocation, threshold based optimized reallocation and re-distribution of user resource granting. This is to enhance network performance focusing mainly on system sum data rate or throughput and QoS metrics, especially delay and jitter. This has been simulated and compared to reference work [1]. After calculating the average performance metrics, we show that the proposed solution performs 31.9% higher for throughput and 23% better for delay performance. However we intend to implement the system on 4G and forthcoming 5G networks and it is left as near future work.

