User Grouping in Multi-User Uplink Protocols for Wi-Fi Networks

WeiChong Chen
Department of Electrical Engineering, Rice University

Nate Raymondi
Department of Electrical Engineering, Rice University

I. Introduction

Due to size and power limitations, mobile users typically have a very small number of antennas compared to their respective access points (APs). When the AP is only able to serve one user at a time, this antenna asymmetry leads to a significant decrease in capacity. The AP is effectively bottlenecked by the lower number of antennas at the mobile user. In [1] it is shown that the AP can achieve full-rank downlink transmission by grouping a number of mobile clients together to form a virtual antenna array with the same number of antennas as the AP. This approach has been standardized and implemented in commercial Wi-Fi systems.

Despite these advances in multi-user downlink schemes, the uplink case has remained unchanged since the original Wi-Fi standard introduced in 1997 [3]. This is because the mirror solution for the uplink case would require a separate control channel for channel estimation, user selection, and time synchronization. In [2], a solution to this problem is presented. MUSE (Multi-User Scalable Uplink) has been designed to achieve scalable full-rank multi-user uplink without a control channel.

Presented in [2] is a combined physical layer and medium access protocol that allows for multi-user uplink when the users' channels are sufficiently orthogonal. Specifically in the medium access protocol (MUSE-MAC), a single user wins the channel via the same random-access backoff countdown as in traditional Wi-Fi. Once a single user wins the channel, a group of other users are attached by a predefined function in order to achieve a full-rank link. The mobile users IDs and the function by which the users are grouped are all predefined, so there is no need for a control channel.

II. Previous Work

The purpose of this paper is to study the performance of different user grouping strategies as compared to the method presented in [2]. In order to do this, we will first describe the MUSE MAC protocol to understand how the original user grouping strategy comes into play. Next, we will discuss scenarios where the MUSE MAC user grouping strategy does not perform well. Finally, we will present and analyze alternative user grouping strategies for MUSE MAC and compare their performance with the original.

A. MUSE MAC Protocol

The MUSE MAC protocol provides a method by which a group of users, all assumed to have a single transmitting antenna and where the total number of users is equal to the antenna rank of the AP, can be granted access to the medium. This full-rank communication link allows us to take full advantage of the diversity offered by the MIMO AP.

First we note that when a network is established, all users wishing to associate with the AP will be given a unique identification number. This process is part of the IEEE 802.11 standard and is referred to as the Reassociation Request and Response procedure. Each user knows its own ID number and they are assigned sequentially.

The method by which a single user wins control of the medium is the same as conventional WiFi. All backlogged users compete for the medium. However, once a user wins the medium (by its backoff counter reaching 0 before all others), it will broadcast a triggering message to all other users and the AP. This message will notify a predefined set of other users that they may also access the medium during the next data transmission slot (timing information and synchronization is also contained in the triggering message). This predefined of users set is simply a grouping of users with IDs that successively increase from the the winning user's ID. The size of this group is equal to the rank of the MIMO AP.

For example, if user K wins the medium and wishes to establish a full-rank link with an AP of rank 4, its trigger message will notify users $\{K+1, K+2, K+3\}$ that they may transmit data at the same time. It is also worth noting that the set of all association IDs is circularly wrapped. That is, if a triggered association ID number exceeds the actual number of users, we will wrap back to the first user. So if there are N users in our network and users $\{K, K+1, K+2, K+3\}$ have been triggered where $\{K, K+1 \leq N\}$ but $\{K+2, K+3 > N\}$, then we will wrap and allow users $\{K, K+1, 1, 2\}$ to transmit.

A visual representation of the circular user grouping and MUSE MAC time line can be seen in Figures 1 and 2. Here we see that out of a set of 7 users, user 6 wins the medium. The triggering message beacon is sent to all users in the network and informs users 7, 1, and 2 that they may access the medium as well. After the data transmission period, individual ACKs are sent from the AP to each user that sent data.

One final thing to note about the MUSE MAC protocol is

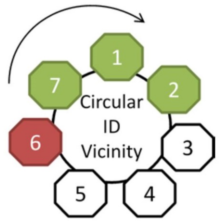

Fig. 1. Circular User Association ID Set[2]

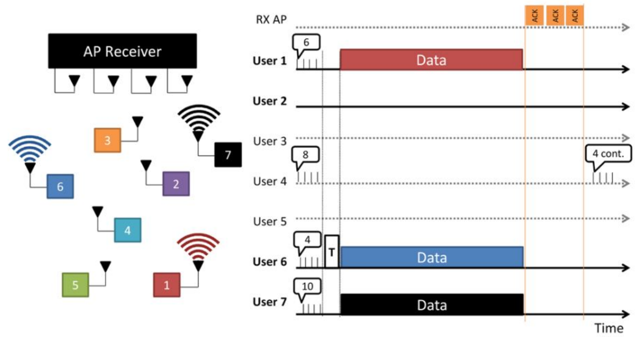

Fig. 2. A 4x3 MUSE-MAC Timeline[2]

that it retains the fairness of conventional WiFi. Each user has the same chance to win the medium when it has a packet it wishes to send. Furthermore, under the assumption that all users are infinitely backlogged, on average each user will be called upon to access the medium the same amount of times.

B. Shortcomings of MUSE MAC User Grouping

Much of the benefit of the MUSE MAC protocol stems from the assumption that all users in a network are infinitely backlogged. In terms of PHY throughput ratio (compared to the MISO uplink case), this is clearly the best case scenario as all triggered users within a group will have data ready to send. This means we are taking full advantage of the full-rank communication link for every data transmission. In terms of fairness, all users being backlogged averages out to an equal number of data transmissions for each user, due to the contention-based nature of the MAC protocol.

However, both of these benefits break down when the assumption of infinite backlog at all users is removed. We can easily imagine the case where a user might have very sparse data generation (perhaps loading a webpage and then taking the time to read it). If this user is triggered and given access to the medium without having data to send, we are no longer taking full advantage of our full-rank link to the AP. This case also has implications to the network fairness. Since this user with no traffic is no longer contending for the medium, it will never be the case that the users within this user's triggering group will be granted access. Namely, if user K is never winning the medium, users $\{K+1, K+2, K+3\}$ will never be triggered together and thus these users are indirectly penalized for use K's lack of traffic.

These shortcomings are shown in Figure 3. This plot shows the average PHY throughput ratio to a rank 4 MIMO AP (over 10000 trials) where the 10 users are given a probability that they will have data to transmit it they are called upon. These probabilities are chosen uniformly over the range [0,x]. For example, the data point located at $x = 0.5$ represents the average throughput for each trial where 4 users are grouped out of 10 (in the same manner as MUSE MAC) where each of these users has a probability of being backlogged of $U[0, 0.5]$. The lower bound of this curve will be at PHY throughput ratio 100%, which is where all triggered users are not backlogged (only the user winning contention will have data to send). The upper bound is along 400%, or more generally AP Rank*100%, which is the case where all triggered users have data to send.

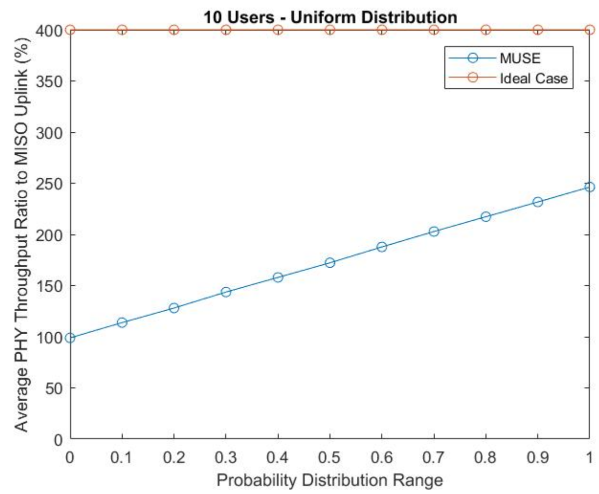

Fig. 3. MUSE MAC User Grouping Performance for Non-Infinite Traffic Backlog

We recall that we are operating under the assumption that the channels are sufficiently orthogonal for multi-user transmission. This is performed by the MUSE-PHY protocol [2]. Due to this assumption, channel conditions are not taken into consideration. While this may be a crude simulation, it effectively shows the potential shortcomings of the MUSE MAC user grouping strategy for situations where users have sparse traffic. In the next section, we will propose alternative user grouping strategies to address the shortcomings mentioned here.

III. Alternative User Grouping Strategies

In this section we present alternative user grouping strategies to increase the average PHY throughput ratio. As an obvious class of solutions, we could always achieve full-rank transmission links by performing polling to fill up the user groups. This would provide assurance that the users that are grouped would have data to transmit. However, this would require far too much overhead to be a feasible solution. Another approach is to use previous data rate measurements to predict future values [4].

A. *Moving Average History Based Data Rate Predictor*

History-based data rate prediction is similar to a standard time-series prediction, wherein a random process can be predicted for a future instance based off of past measurements. In the same way, we can look at past behavior of a user (i.e. their backlog status) to make a prediction about the possibility of them being backlogged when next called upon. The moving average (MA) predictor is a simple linear predictor model that we can use for this purpose. The MA predictor is defined generally for a time series X as

$$\hat{X}_{i+1} = \frac{1}{n} \sum_{k=i-n+1}^{i} X_k \tag{1}$$

where $\hat{X}_{i+1}$ is the predicted value and X_i is the sampled value at instance i. We can use the predicted backlog to create an optimal group consisting of the user that wins the medium and the three other users with the highest predicted probability of being backlogged.

The first problem we run into considering this solution is the problem of how to compute and notify users of this new grouping. Since the AP is the central node in the network (i.e. all traffic runs through it), it has the information necessary to make these predictions for each user. Therefore we can perform these predictions at the AP. This still does not solve the problem of notifying users of the new group. We consider an alternative time line based off of the MUSE standard from Figure 2.

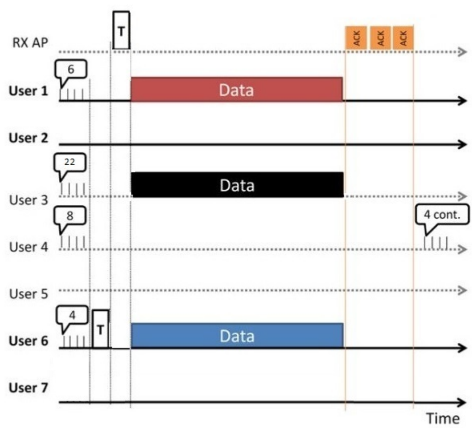

Fig. 4. A 4x3 Timeline with Predictive User Grouping

In the same way as with MUSE, a user wins access to the medium through a contention-based backoff countdown. This winning user broadcasts the same trigger message to all other users and the AP. At this point if the AP determines, through the previously mentioned MA data rate prediction, that a different grouping of users might have a better chance of achieving full-rank, it will also broadcast a message to all users notifying them of the new grouping. This provides all users with the new user grouping scheme.

This is modeled in Figure 4. As previously, user 6 wins the medium after backoff and sends out a message to trigger users 7, 1, and 2. However at this point the AP overrides this grouping, deciding that user 3 is more likely to help achieve a full-rank link than perhaps user 2 or 7 (which are both shown not have have any backlogged data). Therefore, using this strategy we should have a higher rank communication link on average than with standard MUSE user grouping.

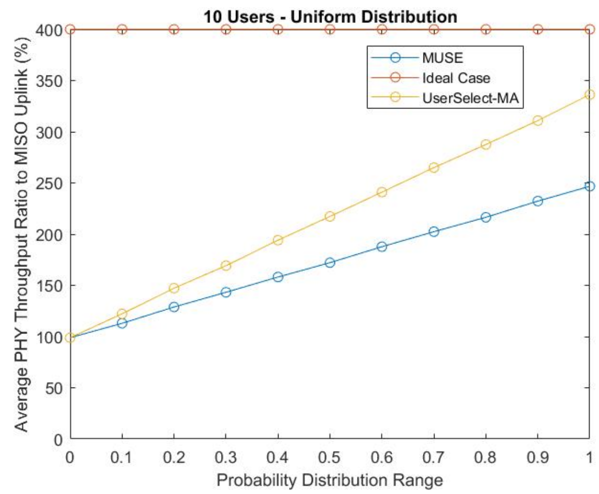

Fig. 5. Moving Average Predictive User Grouping with Non-Infinite Traffic Backlog

Figure 5 shows the PHY throughput ratio performance of the MA predictive user grouping compared to the MUSE scheme under the same conditions discussed for Figure 3. We can see that the predictive grouping gives an improvement in performance over the MUSE grouping that increases along the x-axis. This improvement can be interpreted as such: For situations where users have a larger range of possible backlog probabilities (further along the x-axis), this predictive model is more easily able to determine users with higher backlog probabilities (users that are most likely to have data to send when called upon). We can think of this in terms of user diversity - we are able to pick the ”best” users from the whole network, so where the ”best” users’ performance is much better than the average performance, we are getting a higher reward for being able to pick them.

This idea of user diversity is further illustrated in Figure 6. This simulation considers the same test conditions as Figure 5, but with 50 users in the network, instead of 10. While the MUSE user grouping achieves the same performance in terms of PHY throughput ratio, we can see that slope of the predictive grouping curve has increased, offering better performance. This is because we are now able to choose from a much larger group of users, which gives us a higher chance to create a full-rank link.

B. *Holt-Winters History Based Data Rate Predictor*

A slightly more advanced history based predictor is the Holt-Winters (HW) predictor [4]. HW is a variant of an exponentially weighted MA predictor. It is used to capture

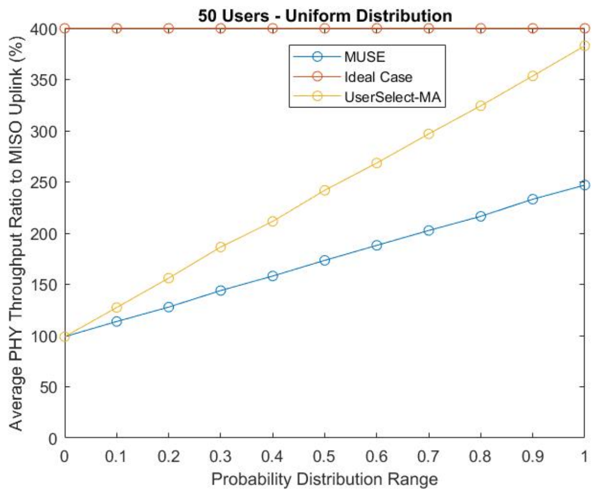

Fig. 6. Moving Average Predictive User Grouping with Non-Infinite Traffic Backlog - 50 Users

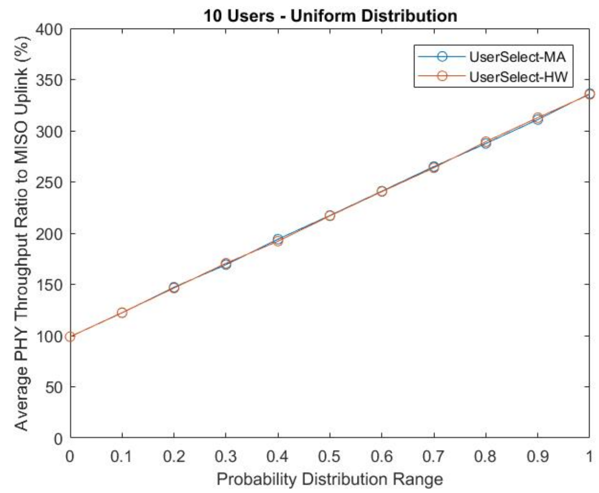

Fig. 7. Comparison of Holt-Winters and Moving Average Predictive User Grouping Performance

a trend in the underlying data, if there is one to find, thus we have

$$\hat{X}_i = \hat{X}_i^s + \hat{X}_i^t \tag{2}$$

where

$$\hat{X}_{i+1}^s = \alpha X_i + (1-\alpha)\hat{X}_i \tag{3}$$

and

$$\hat{X}_{i+1}^t = \beta(\hat{X}_i^s - \hat{X}_{i-1}^s) + (1-\beta)\hat{X}_{i-1}^t. \tag{4}$$

Here $\hat{X}^s$ is a smoothing component and $\hat{X}^t$ is a trend component.

In Figure 7 we perform the same simulation as in Figure 5 with just 10 users. From this we can see that using the HW predictor yields extremely similar performance to the simple MA predictor. In a general case, we can infer from this that there is not any type of identifiable trend to the data being analyzed. In our case, we performed the simulation with a simple random data generation scheme, so it makes sense that the HW predictor would not offer an improvement to our prediction accuracy.

An astute observation at this point would be that the network fairness of these predictive grouping strategies needs to be taken into consideration. After all, the expressed goal of the predictive user grouping has been to simply achieve the maximum average PHY throughput ratio. If we think of the case where 3 users in the system are continuously backlogged and trying to send their data, we can imagine and infer from the plots that eventually our predictor will lock on to these 3 users and group them every time. This would satisfy the full-rank communication link constraint. However this would lead to other users, perhaps with a lower packet generation rate, to be starved of the medium as they are never going to be a better option than any of our 3 continuously backlogged users. This leaves winning the medium through backoff countdown as the only option for medium access. Clearly there is some trade-off between blindly maximizing the predicted average PHY throughput ratio and the overall network fairness.

To analyze the different user grouping strategies, we will use Jain's network fairness measure [5] as

$$\jmath(x_1, x_2, ..., x_n) = \frac{(\sum_{i=1}^n x_i)^2}{n \cdot \sum_{i=1}^n x_i^2}, \tag{5}$$

where n is the number of users and x_i is the throughput of the i^{th} connection. This metric returns a maximum value of 1 when all users receive the same allocation, or in this case, transmission opportunities. We will use this metric later to analyze the fairness of our simulated systems.

IV. Simulations

In this section we describe the simulations performed to evaluate the proposed user-selection improvements. We performed simulations using both Matlab and NS3, both experimental setups and results will be described below.

A. Matlab Simulations

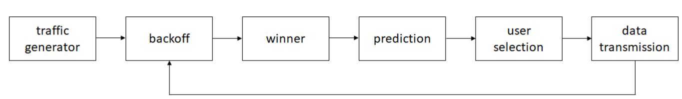

Fig. 8. Matlab Simulation Flow-Chart

For the simulations taking place in Matlab, we expand upon and improve the simulation setup used previously for preliminary evaluation of the user-grouping strategies. A rough flow chart diagram can be seen in Figure 8.

The main difference between this simulation and the previous simulations is how traffic is modeled and generated. Here we employ a traffic generator using the geometric distribution as

$$\Pr(X = x) = (1-p)^x p. \tag{6}$$

A few sample probability mass functions of the geometric distribution can be seen in Figure 9. The geometric distribution is completely defined by the parameter p where $0 < p \leq 1$.

We utilize the geometric distribution as a packet generator by assigning each user a p value. This value directly corresponds to how busy that user will be. Each time the system resets, or each time a new grouping is being created after some user wins the medium, packets are generated at each user based on a realization of the geometric PMF of that user. To model sparse traffic, we give a user a p parameter close to 1. This means that most of the time, there will be 0 packets generated by that user. If we want a user to be very busy, we can assign a low p value. This would mean that a packet (or even multiple packets) will be generated by this user each time. There is no limit to the queue length for each user.

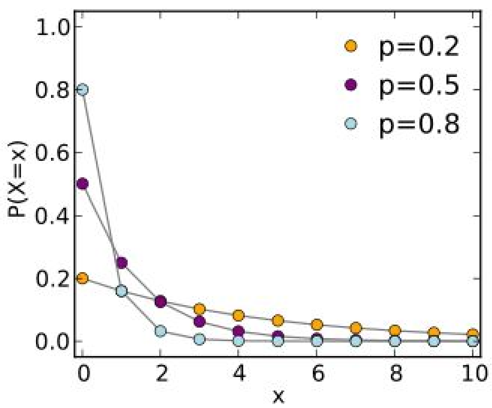

Fig. 9. Sample PMF of a Geometric Distribution

We now perform our simulation using this setup. First we choose our traffic generation to be defined by all users having parameter p=0.5. This means that half of the time, 1 or more packets will be generated by each user. We see in Figure 10 that we do not get any throughput gains in this case. Due to the traffic generation parameter p, it seems most users will typically be backlogged when grouped. This means that the MUSE will perform well, as it utilizes less overhead when grouping users.

The normalized throughput for MUSE is calculated as

$$\text{Throughput} = \frac{T_{data}}{T_{overhead} + T_{data} + T_{collision} + T_{backoff}} \tag{7}$$

where the renewal time is from the valid data transmission to the next data transmission. And for each transmission, we will have a instant throughput, then after 100,000 trials, we can have the average throughput afterwards.

We also study the fairness of the system in Figure 11. First we reiterate that all users have the same traffic generation model. Due to this, it makes sense that the MUSE user grouping strategy will have maximum user fairness. Next we note that the moving average user grouping strategy performs sub-optimally in terms of fairness. This makes intuitive sense.

The moving average predictor will simply group users with high chance of being backlogged. Since we are considering a case where the traffic is not very sparsely generated, we expect

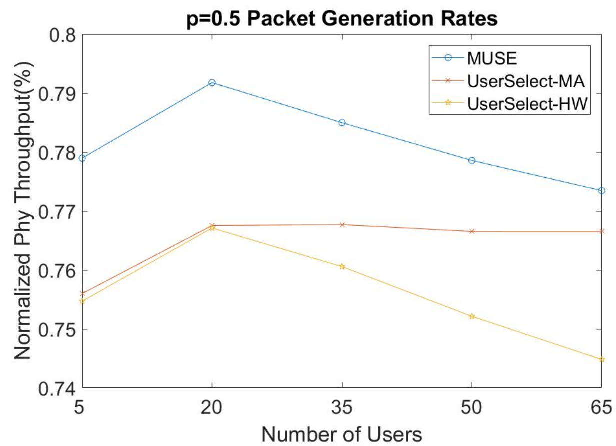

Fig. 10. Normalized Throughput for p=0.5 Packet Generation

the MA predictor to latch onto a couple users that are often backlogged and not let any others in. Since Jain's Fairness index is a per-user fairness rating, it makes sense that the value will decrease as more users are added to the system.

Finally we see that the HW predictor actually has the same fairness measure as the MUSE case. This is because we are still dealing with a independent traffic generation at each of the users. There is no trend in the data for the HW predictor to take advantage of, so there is no real change between the HW predictor and MUSE. This is also reflected in the throughput analysis in Figures 10, 12, and 14. The only difference here is that the HW predictor is adding a slightly increased amount of overhead into the grouping process, but is not giving us any performance gains due to the traffic generator.

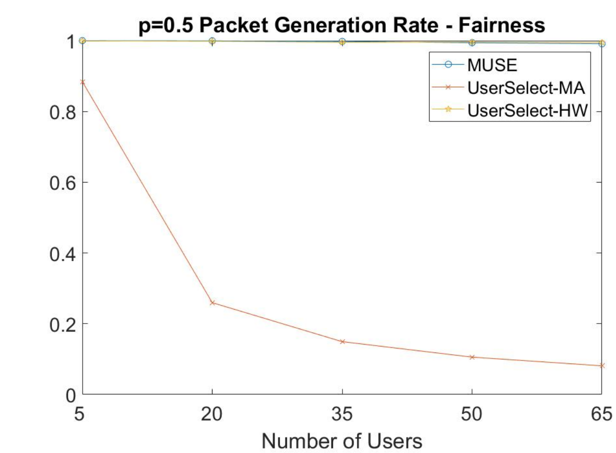

Fig. 11. Jain's Fairness for p=0.5 Packet Generation

We next performed the same simulation using geometric distribution parameter p=0.9. This effectively gives us a more sparse traffic generator, as 90% of the time there will be no packet generated at a user. The results of these simulations can be seen below in figures 12 and 13.

Finally, we also performed the simulation using the geometric distribution traffic generator parameter p=0.95. This

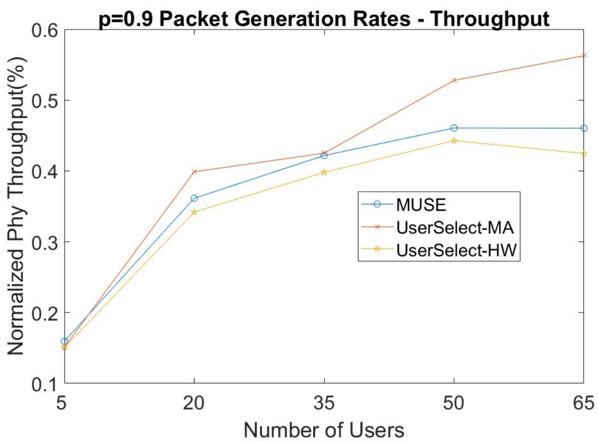

Fig. 12. Normalized Throughput for p=0.9 Packet Generation

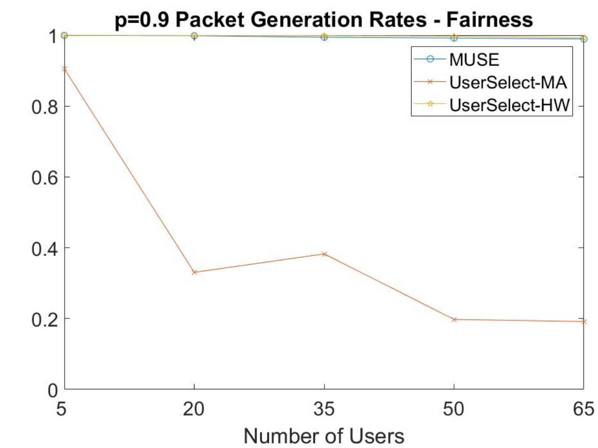

Fig. 13. Jain's Fairness for p=0.9 Packet Generation

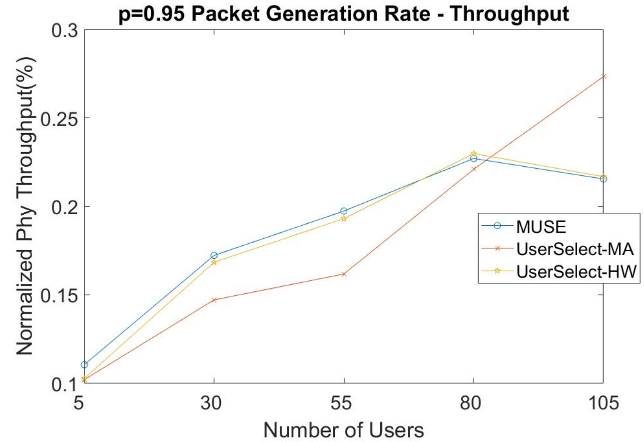

Fig. 14. Normalized Throughput for p=0.95 Packet Generation

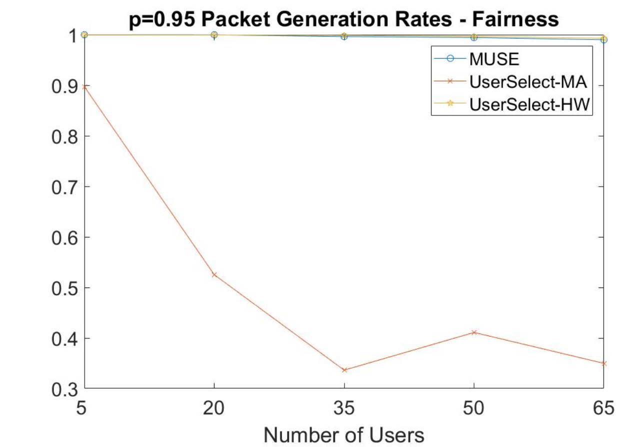

Fig. 15. Jain's Fairness for p=0.95 Packet Generation

provided an even more sparsely generated traffic pattern at each user. The results of these simulations can be seen in Figures 14 and 15. We also provide the time constants used for calculations in Table I.

TABLE I
TIME CONSTANTS USED

T_{DIFS}	=	34μs
T_{SIFS}	=	16μs
$T_{Backoff}$	=	9μs
T_{ACK}	=	24μs
$T_{Trigger}$	=	24μs
T_{Data}	=	248ms

B. NS3 Simulations

In ns3, we first try to understand what Peshal provided us, the ns3.21 code for downlink grouping. However, the previous version does not provide multi-user package, the provided codes are specified by the author resulting in some constraints. After some understanding and researches, we eventually find that ns3.26 now provides us the ability to analyze multi-user uplink scenario. And therefore we get the chance to utilize the package provided by ns3.26 to realize our uplink user grouping research.

The system setting is under 802.11, choosing MCS 0, bandwidth is 20 MHz, and the channel is set as default YansWifiChannel provided by ns. The data rate from users are set as 50 Mbps, for each packet size is 1472 bytes. And for the system simulation time is set as 10 seconds with 100 trials, and the number of users from 5 to 125 with interval 20 where the users are randomly deployed within 10 meters range from AP. However, for the time sake (it took hours to run a single case), we only have the throughput comparison between MUSE and MA with dense traffic. We plotted the results of this simulation using Matlab.

We can see that the throughput is decreasing as the number of users increases. That is because the AP needs to spend more time to communicate with users, inducing higher overhead. And the low throughput comes from the chosen of MCS, since for MCS 0, the system can only support 32 Mbps while we are transmitting as 50 Mbps. We can see that MA performs almost the same as MUSE under dense traffic because they can always find some users having data to transmit. However, as we discussed earlier, the fairness for MA is relatively low compared to MUSE.

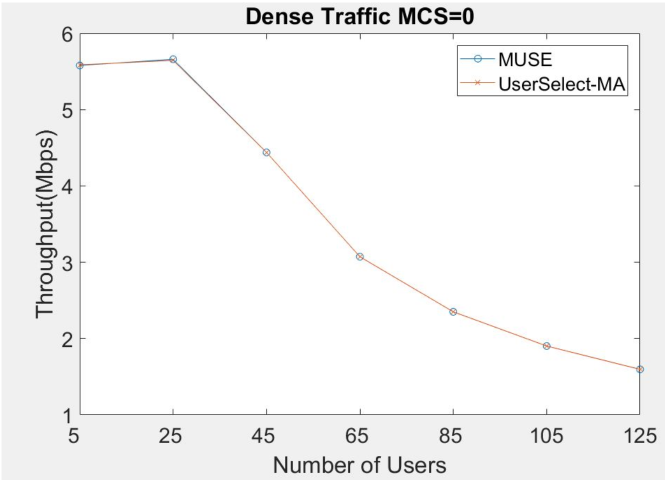

Fig. 16. Comparison of Holt-Winters and Moving Average Predictive User Grouping Performance

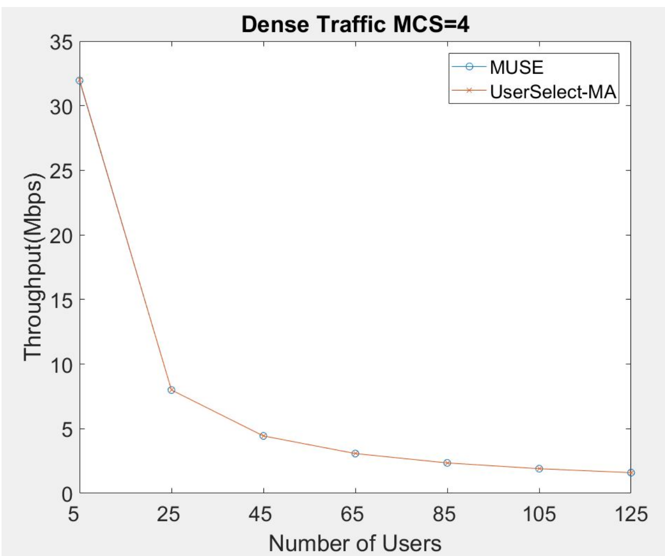

Fig. 17. Comparison of Holt-Winters and Moving Average Predictive User Grouping Performance

V. Conclusion

We can see that MUSE can perform very well in most of the time. The biggest problem of it is when operating under sparse traffic without any further information. Because sparse traffic could cause non full-rank problem to MUSE, we come up with MA and HW, trying to predict the potential users from historical data. We find out that MA can increase the throughput when traffic is sparse while sacrificing fairness; HW performs similar to MUSE with same level of fairness. One thing that would have made our simulations more accurate and helpful would be a better way to model data traffic at each of the users. In real life, traffic patterns for a user can be drastically changing depending on the nature of the traffic. For example, if a user is trying to watch a video, there will be a period of very high activity as the user loads the video, followed by a long period of inactivity while the pre-loaded video plays. This type of pattern information would be abundantly useful in application to these user grouping strategies because it would help us make better informed decisions. Some work has been done on TCP prediction using Machine Learning techniques to increase accuracy [6].

References

[1] C. Shepard, H. Yu, N. Anand, E. Li, T. Marzetta, R. Yang, and L. Zhong, ”Argos: Practical many-antenna base stations”, in *Proc. ACM MobiCom*, 2012.

[2] A. Flores, S. Quadri, and E. Knightly, ”A Scalable Multi-User Uplink for Wi-Fi”, in *USENIX Symposium on Networked Systems Design and Implementation*, 2016.

[3] B. Crow, I. Widjaja, J.G. Kim, and P. Sakai, ”IEEE 820.11 Wireless Local Area Networks”, *IEEE Communications Magazine*, September 1997.

[4] Q. He, C. Dovrolis, and M. Ammar, ”Prediction of TCP throughput: Model-based and history-based methods”, In *Sigmetrics*, 2005.

[5] R. Jain, D. Chiu, and W. Hawe, ”A quantitative measure of fairness and discrimination for resource allocation in shared systems,” tech. rep., *Digital Equipment Corporation*, DEC-TR-301, 1984.

[6] M. Mirza, J. Sommers, P. Barford and X. Zhu, ”A Machine Learning Approach to TCP Throughput Prediction,” in *IEEE/ACM Transactions on Networking*, vol. 18, no. 4, pp. 1026-1039, Aug. 2010.