MLWDF Scheduling and Resource Allocation in 802.11ax

Reshan Faraz
Engineer
rfaraz@qti.qualcomm.com

I. Abstract

802.11ax introduces OFDMA in WiFi. It enables multiplexing users in frequency domain.The motivation of this project is to understand the OFDMA performance in a dense WLAN. We come up with scheduling and Resource Allocation strategy to understand how clients are scheduled . We will simulate a dense WLAN with large number of clients. In addition, the clients will run different applications i.e., browsing, long download, voice call etc. Eg, how a real life network would look like. The task is to analyse such network scenarios in terms of various network metrics such as throughput, delay, rtt, jitter, upload time.

Keywords: MLWDF Scheduler, Resource Allocation, OFDMA, NS-3, Wireshark

II. Simulation Scenario and setup

For OFDMA simulations, we have considered a different scenario with four different applications: 1) Voice (using on off traffic), 2) Bulk, 3) HTTP and 4) On-off. We have installed two applications installed on each station i.e. 'Packet sink' and 'HTTP Client' application. Simulation Time: 10 sec.

Parameter	Value
Channel Bandwidth	20 MHz
MCS	$[1, 12]$
No. of STAs	32
Guard Interval	3.2 us
TXOP limit	10016 ms
Simulation Time	10 sec

The following is the outline of the article: In Section III we provide the trace analysis, how the trace files were used to compute the results. In Section IV, the decription of MLWDF is provided. In Section V, a psuedo implementation of scheduler is provided. Finally in Section VI, the results are provided. This section is subdivided into two parts: a) Case-I and b) Case-II.

III. Trace Analysis

While analysis the PCAP traces the following key points were observed.

- For Round Robin scheduler the number of clients are going upto 9 (Maximum RUs). But we can observe no OFDMA frames with 5,6,7,8 clients. It is because RR always divide the RUs into equal size.
- For dense network scenarios, as in our case we can see the maximum number of frames having clients equal to the maximum number of allocated RU(i.e 9).
- We can observe that number of re-transmission for RR scheduler is 3004 packets. While for MLWDF the number of re-transmissions account to almost half of the RR i.e. 1682 packets which account to 3.2% of the total transmissions.
- We can observe that increase in the simulation time increases the broadcast packets and not the transmitted packets. Thus, if we increase the simulation time from 10 sec to 15 sec. the observations remain same.

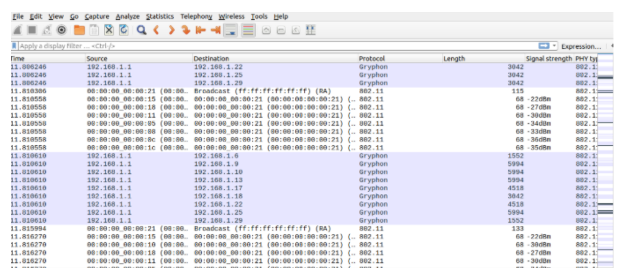

Fig. 1. Snap of Pcap trace file

IV. Scheduler Design

The scheduler works on the queues and for each time slot 'i' it serves the queue with maximum value of utility function U = γ_jW$_j$(i)r$_j$(i) Where, γ_j is a constant, W$_j$(i) is the head of line packet delay (HoL) for the queue and r$_j$(i) is the channel capacity for the flow. MLWDF does not only consider delay but it is also throughput optimal. The current channel conditions and state of the queues are taken into consideration. The parameter γ is the parameter which helps to assign different delays to different users. Increasing the value of γ for a user reduces its delay at the cost of delay of other users. Hence, we need to select the optimal value of this parameter. It is given that keeping $\gamma_j = \alpha_j\sqrt{r}_j$ = -log$_2\delta_j$/T$_j\bar{r}_j$. Here, $\bar{r}_j$ is the average rate of the channel, T$_j$ is the desired delay threshold and δ_j is the maximum violation probability.
802.11ax introduces OFDMA where the AP divides the entire frequency band into multiple subsets of orthogonal sub-

carriers,termed as Resource Units (RU). And these RU can be assigned to different users simultaneously. The main objective of 802.11ax is to serve in a dense environment without any collisions and to increase throughput density. To accomplish the above features of 802.11ax we are implementing the M-LWDF.

V. Implementation of Scheduler

A. Pseudo Code

Pseudo Code to implement our Scheduling and Resource allocation Algorithm.

Algorithm 1 Pseudo Code:

Begin
W = utility function, C = set of Clients, Bandwidth=20 Mhz.
For each client C_i calculates the Utility Function say W_i and stores the result in the list.
Sort the clients in non-increasing order based on their Utility Function.
if If number of client are greater than 8 Then 5. select the first 9 clients from the list. **then**
    Allocate 26 tones RU to each Client
else
    Based on the number of clients, pick corresponding RU sets and allocate in non increasing order.
end if
End

B. Files Modified

For analysis we have modified three files:(find the attachment for the same)

*1) **rr-ofdma-manager.h**:* It contains the function declaration and macro definition used in rr-ofdma-manager.cc

*2) **rr-ofdma-manager.cc**:* We added the cost metric function and allocated RUs based on cost metric.

*3) **wifi-dl-ofdma.cc**:* We changed the traffic parameters of the users and assigned the traffic to different access categories.

C. Names of the function created/ modified:

1) type_of_App(): In this function we are allocating the values of α_i for each client according to traffic. If the traffic is voice traffic then α_i = 10, for video it is 5.4927, for BE it is 2.3299 and background it is 1.50515. $\{P(\hat{w}_i > T_i) \leq \delta_i\}$

2) returnRate(): In this we are returning the rate of the system according to the mcs selected. For ex. For BPSK with $\frac{1}{2}$ coding rate the rate is 8.1 Mbps for 20 MHz channel and 16.3 for 40 Mhz.

3) timereq(): In this we are calculating the time required for given values of mcs and datasize. First we map mcs to the coding scheme i.e. which QAM it is using and what is the encoding rate. Then we are calculating data rate as:
Data rate = $\log_2$(QAM type) $\times$ encoding Rate $\times$ 242
After calculating the data we calculate the number of symbols transmitted. Which is then used to calculate the time required for all the symbols to be transmitted. Time required = symbols $\times$ ((guard interval + symbol duration) = 0.0000136sec)

4) averageChannelcapacity(): Here, based on mcs we first calculate the average throughput, i.e. data size/time required(calculated through timereq). We then use this throughput to calculate the capacity of channel $\log_2$(throughput/data size).

5) RUAlloc(): In this function we are allocating the RUs according to number of clients and bandwidth. For eg: If the number of clients are 5 and bandwidth is 20 MHz then it will allocate $\{$106,52,26,26,26$\}$ RU combinations. If the bandwidth is 40 MHZ then it will allocate $\{$242,106,52,52,26$\}$.

6) mlwdf: In this we are calculating the cost metric for users in order to allocate RUs. For each to which the RUs needs to be assigned we calculate the cost metric as:
final_cost=(Station List size $\times$ α_i $\times$ rate) / average throughput
Where, α_i is calculated from calculate a_i function rate is through returnRate() function and average throughput is from averageChannelcapacity() function. We then sort the station list and assign RUs to stations.

7) DlOfdmaInfo: In this function we first calculate the number of stations to which RUs are to be allocated. Then we call the function mlwdf() and assign RUs to those stations. While assigning RUs we modify mapping to RU function to allocate 484, 242, 106, 52 and 26 tones RU.
We also set the Random MCS for simulation.

VI. Results

A. Number of clients in one OFDMA transmission

Fig. 2 and Fig. 3 show the average number of unique clients in one OFDMA frame averaged over time for MLWDF and RR respectively. We can see that the average number of clients decreases for RR. It is thus evident that in MLWDF the clients are allocated RU more closer to maximum RU assigned.

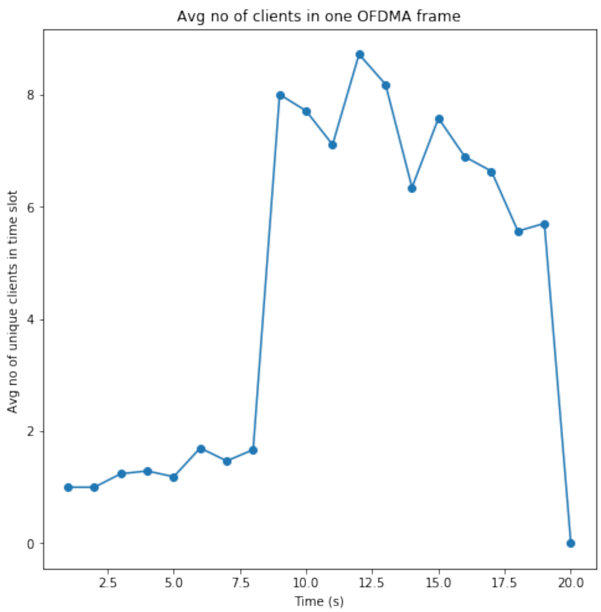

Fig. 2. Average No. of clients in OFDMA frame for MLWDF

Fig. 4. OFDMA frame transmission time for MLDWF

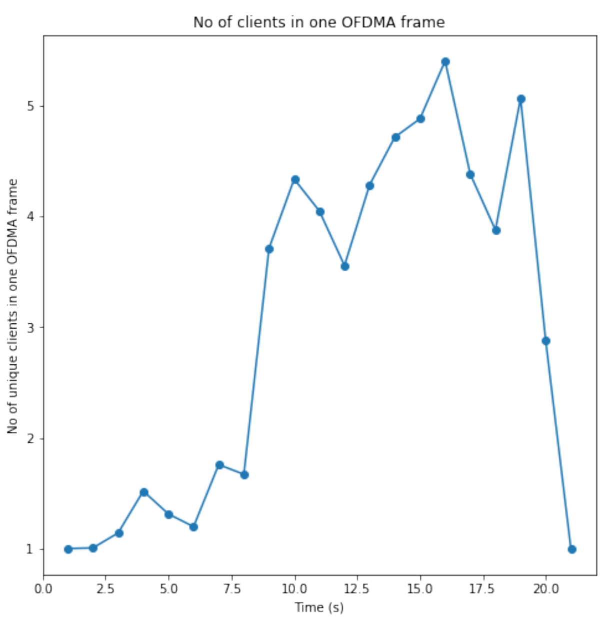

Fig. 3. Average No. of clients in OFDMA frame for RR

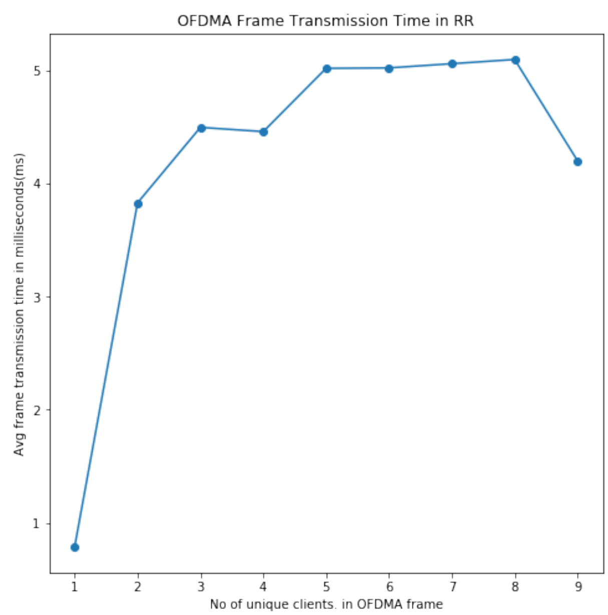

Fig. 5. OFDMA frame transmission time for RR

B. OFDMA frame transmission duration

Fig. 4 and 5 show the OFDMA frame transmission duration. We can see that as the number of unique clients in OFDMA increases the frame transmission time for MLWDF also increases because as the number of client increases they will get smaller tone RU, Our algorithm always pick the clients which has large packet in their queue and have higher priority in terms of type service.

C. Single user transmissions

Single user transmissions are observed here. This is also evident from Fig 3 and 2 where for the first 2s, an OFDMA frame contains one user only.

D. Throughput (Bulk clients)

The throughput for both the RR and MLWDF is almost performing the same way. The maximum through can be seen at lower instance of time. This is due to the fact that for such cases the delay performance is also low.

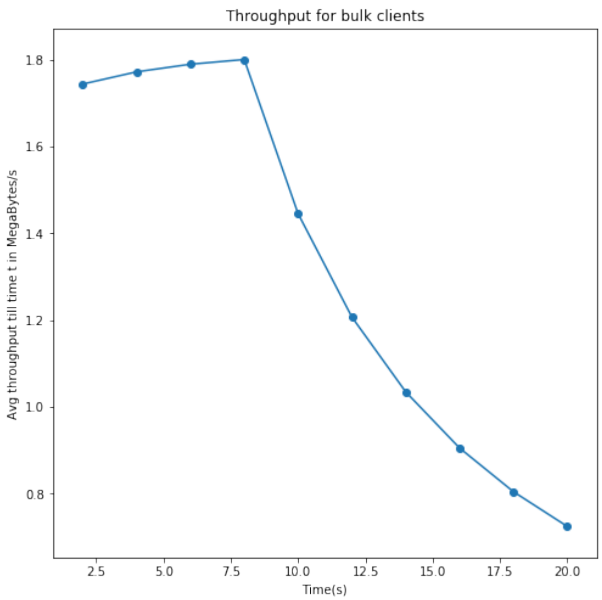

Fig. 6. Throughput for MLDWF

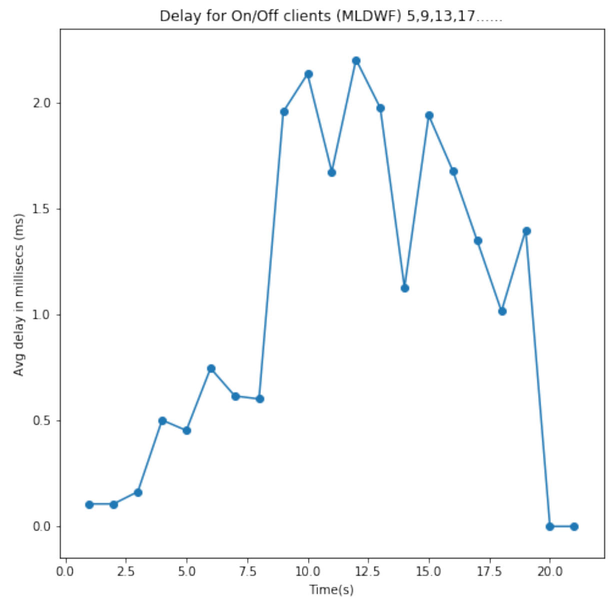

Fig. 8. Average Delay for On-Off Clients for MLDWF

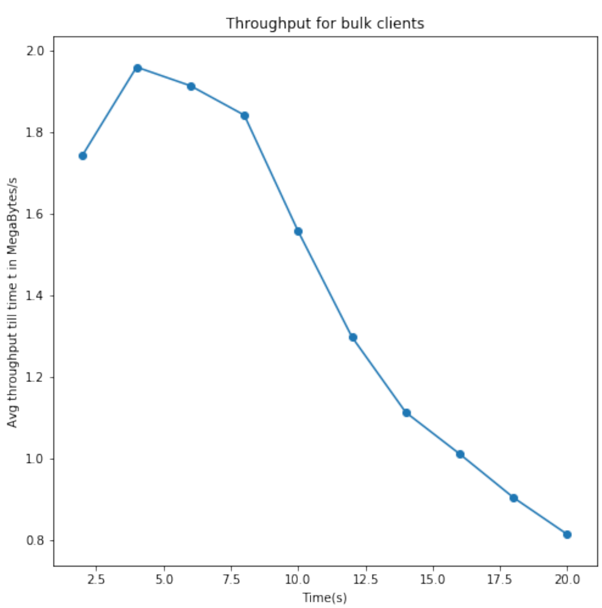

Fig. 7. Throughput for RR

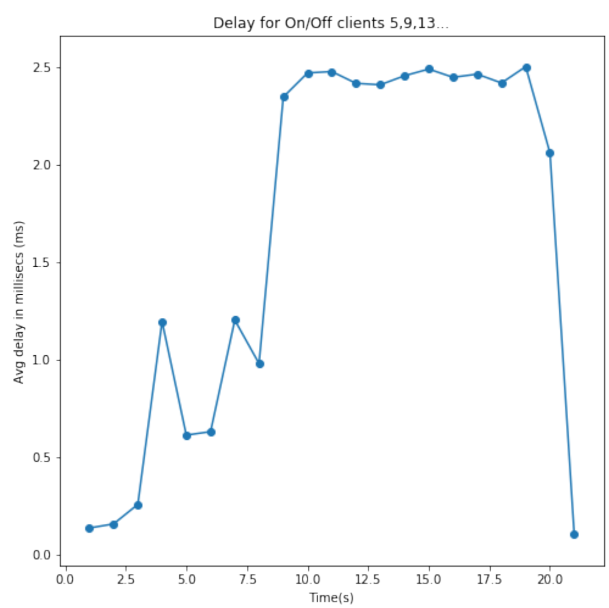

Fig. 9. Average Delay for On-Off Clients for RR

E. Delay (On-off clients)

The delay performance of on-off clients for OFDMA packets is shown in 8 and 9. We can observe that the delay for RR is smaller as compared to the delay performance of MLWDF. This is because of the fact that no. of unique clients in OFDMA frame is more for MLWDF compared to RR.

F. Scenarios where OFDMA could provide benefit

OFDMA will perform better when there is dense environment when the data from number of users can be combined into single OFDMA frame. By using OFDMA, we can improve the average throughput performance of the users.

G. Delay (Bulk clients)

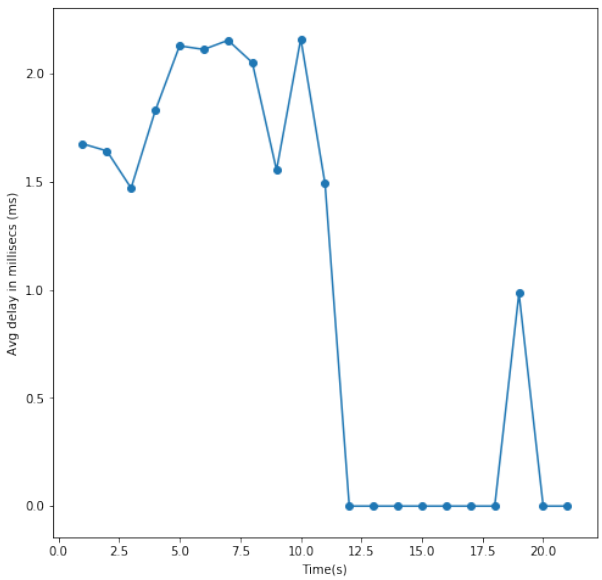

Fig. 10. Average Delay for Bulk Clients for MLDWF

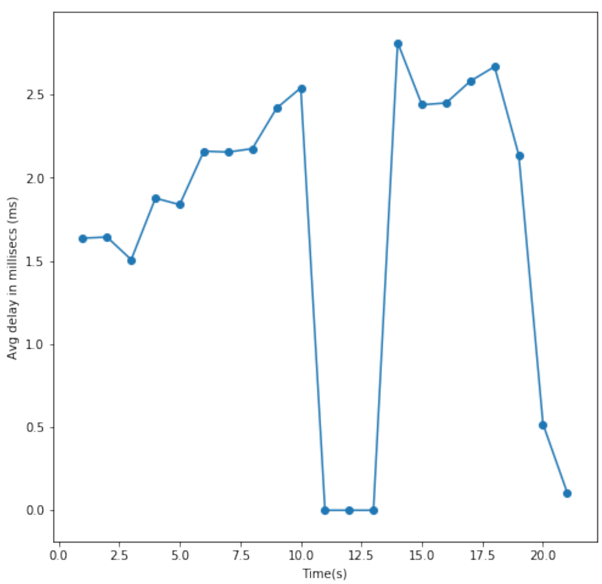

Fig. 11. Average Delay for Bulk Clients for RR

H. Jitter (On-off clients)

Fig. 12 and 13 shows the jitter performnce for MLWDF and RR respectively. From the figure it can be seen that jitter is lesser for MLWDF compared to RR. Which accounts to performance improvement in MLWDF compared to RR.

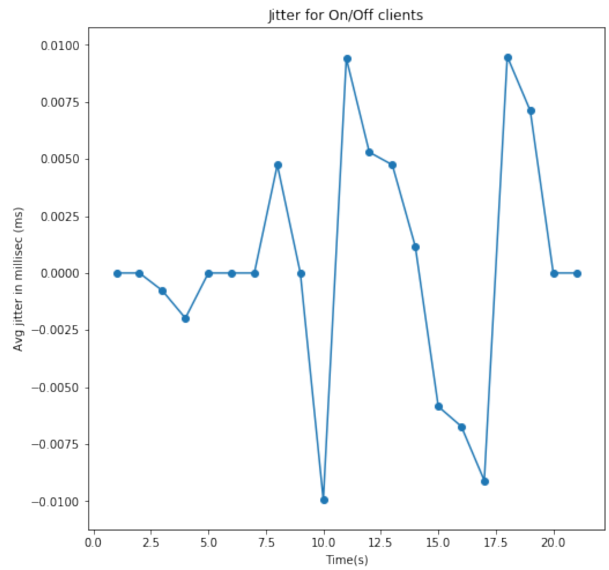

Fig. 12. Average Jitter for MLDWF

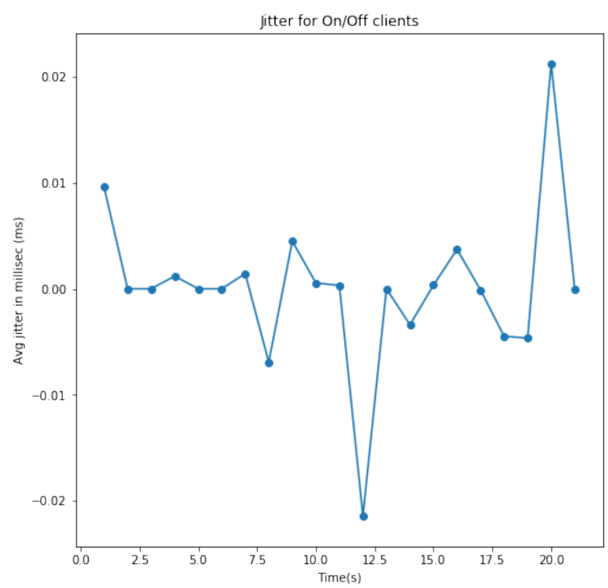

Fig. 13. Average Jitter for RR

VII. Reference

1) Afroz, Farhana, Barua, Shouman and Sandrasegaran, Kumbesan. (2014). *Performance Analysis of FLS, EXP, LOG and M-LWDF Packet Scheduling Algorithms in Downlink 3GPP LTE System,* International Journal of Wireless and Mobile Networks. 6. 77-91. 10.5121/ijwmn.2014.6507.
2) Andrews, Matthew, Kumaran, K., Ramanan, Kavita, Stolyar, Alexander, Whiting, Phil and Vijayakumar, Rajiv. (2001). *Providing Quality Of Service over a Shared Wireless Link,* Communications Magazine, IEEE. 39. 150 - 154. 10.1109/35.900644.

3) Vora, A.; Kang, K.-D. *Effective 5G Wireless Downlink Scheduling and Resource Allocation,* Cyber-Physical Systems Technologies 2018, 6, 105.
4) Ali S, Zeeshan M (2011) *A delay-scheduler coupled game theoretic resource allocation scheme for LTE networks,* Frontiers of Information Technology (FIT)
5) I. Angri, M. Mahfoudi, A. Najid, and M. El Bekkali, *"Exponential MLWDF (EXP-MLWDF) Downlink Scheduling Algorithm Evaluated in LTE for High Mobility and Dense Area Scenario,"* Int. J. Electr. Comput. Eng (IJECE)., vol. 8, no. 3, pp. 1618–1628, 2018.