# Performance of Gradient-Based OFDMA Subcarrier Schedulers with TCP Traffic

Theerat Sakdejayont, Nataporn Jindapornsopit, and Somsak Kittipiyakul
School of Information, Computer, and Communication Technology
Sirindhorn International Institute of Technology, Thammasat University, Pathumthani, Thailand
E-mail: theerat.sakdejayont@siit.tu.ac.th, somsak@siit.tu.ac.th

***Abstract*—We study interaction between gradient-based subcarrier allocation algorithms and TCP traffic sources in a downlink single-hop OFDMA wireless system. Specifically, we are interested in evaluating the maximum-rate (Max-SNR), Proportional Fair (PF), and queue-based Max-Weight (QBMW) subcarrier schedulers, in both non-iterative and iterative versions, with long-lived TCP fluid traffic. Using system throughput and the Jain's fairness index as benchmarks, our simulation results show that even when the users are homogeneous, QBMW leads to suboptimal throughput and extreme unfairness in the presence of TCP traffic. On the other hand, when the users are heterogeneous, Max-SNR is also unfair and gives suboptimal system throughput. However, in both cases, the iterative PF gives the best throughput as well as the best fairness among all considered schedulers, due to its slow variation in the subcarrier allocation, which is preferred by TCP.**

## I. Introduction

The next generation of OFDMA-based wireless systems such as WiMAX is designed to serve Internet data transmission which usually uses the Transmission Control Protocol (TCP). TCP is designed to achieve reliable data transfer and also provide an end-to-end congestion control to avoid packet loss due to link congestion and buffer overflow in intermediate routers. A TCP-controlled source is a reactive source that generates traffic according to the amount of ACKs received.

Both TCP congestion control and a wireless channel scheduler at a base station are designed to give high performance when they are considered independently [1]. They share the same common goals as regulating the traffic in order to maximize channel utilization, minimize packet loss, and provide QoS which is basically considered as fairness among users. However, they employ different methods from different point of view. That is, the scheduler perceives a view from a base station which has queue length and channel gain of each user as basic information. To achieve the maximum total throughput without knowing the statistics of the arriving processes, some wireless schedulers such as QBMW regulate traffic by allocating more subcarriers to longer queues and fewer subcarriers to shorter queues. On the other hand, the TCP congestion control works from the view of users or hosts. The congestion control of a host defines its throughput according to the experienced packet losses. The data rate is decreased when transmitted packets are lost and increased when they are received successfully.

The interaction between these two traffic regulation algorithms may conflict with each other and result in poor performance. This has been shown in some recent works (e.g., [1], [2], [3]). TCP is negatively affected by rapid bandwidth fluctuations which are common in adaptive wireless systems with dynamic schedulers. Such fluctuations are expected to be further amplified in OFDMA systems, which are usually composed of many subcarriers.

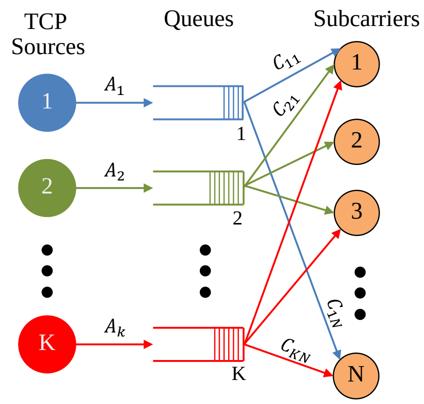


Figure 1. System Overview.

Many OFDMA subcarrier scheduling algorithms have been proposed (e.g., [4], [5], [6]) and analyzed but few were evaluated with TCP traffic sources, while the studies in [1], [3] involve both schedulers and TCP but the schedulers are considered as non-iterative schedulers which lead to overassignment of subcarriers. In this work, we extend the concept of iterative schedulers presented in [6] and evaluate their performance in the system involving TCP traffic sources. The simulation results show that iterative schedulers which tend to allocate the subcarriers more evenly across users give better throughput and better utilization of the assigned subcarriers. Other recent work that studied iterative OFDMA schedulers includes [5].

The rest of the paper is organized as follows. The system model is presented in Section II. The subsequent sections elaborate two main components: the scheduling algorithms in Section III and the TCP fluid model in Section IV. Section V gives the simulation results. Section VI concludes this paper.

## II. System overview

Similar to the model used in [3], we consider a mixed wired-wireless system shown in Figure 1, with $K$ TCP connections from wired sources to wireless users connected to a single Base Station (BS). We associate a source to each user. The wireless part which is the connection bottleneck is a downlink single-hop $N$-subcarrier OFDMA system. The system bandwidth $BW$ is divided into $N$ OFDM subcarriers. At the beginning of each *allocation frame*, the subcarriers are allocated to the TCP sources, according to some scheduling algorithm described in Section III. The centralized subcarrier scheduler at the BS has perfect knowledge of both the queue backlogs and the channel states. During each allocation frame, each subcarrier can be allocated to at most one user.

At the BS, there are $K$ first-come-first-served queues, one for each TCP source, to buffer the data that cannot be transmitted immediately. Each queue can store up to $B$ packets. Packets have a fixed size and for simplicity are assumed to be served in any fraction. Packets that arrive when the buffer is full are discarded. The TCP sources are assumed to be long-lived and in saturation. TCP SACK Reno is assumed. The TCP sources are modeled by the *TCP fluid model* described in Section IV. By discrete-time approximation, the continuous time in the fluid model is sampled into discrete timeslots. The TCP dynamics happen at the start of each timeslot. We assume that the allocation frame contains $T_a$ timeslots.

The channel gains are assumed constant during an allocation frame but changes at the beginning of the allocation frames. In other words, at the beginning of the $k$-th allocation frame (i.e., at time $kT_a$), the channel gain of user $i$ at subcarrier $j$, $H_{ij}[t]$, is equal to $H_{ij}[kT_a]$ for all timeslot $t \in [kT_a, (k+1)T_a)$. In addition, $H_{ij}[kT_a]$, for $i = 1, \ldots, K, j = 1, \ldots, N, k = 0, 1, \ldots$, are i.i.d. Rayleigh random variables with parameter $\sigma_i$. With perfect channel estimation at the receiver and equal power allocation over all OFDM subcarriers, the feasible number of packets that could be served from queue $i$ by subcarrier $j$ during timeslot $t$ is related to the channel gain as

$$C_{ij}[t] = \gamma \frac{BW}{N} \log_2 \left( 1 + 0.56 \frac{P}{N} \left| H_{ij}[t] \right|^2 \right), \quad (1)$$

where $P$ is the total transmit power of the base station, the value 0.56 takes into account the non-ideal modulation and coding [4], and $\gamma$ is the ratio of the number of OFDM symbols/timeslot over the packet size (in bits).

## III. Subcarrier allocation algorithms

The function of the schedulers is to assign at the beginning of allocation frame $k$, the assignment matrix $[s_{ij}[t]]_{K\times N}$, where $s_{ij}[t] = 1$ means subcarrier $j$ is allocated to user $i$ at any timeslot $t$ during that allocation frame. From (1), the total feasible number of packets of user $i$ served during timeslot $t$ is given as $C_i[t] = \sum_{j=1}^{N} s_{ij}[t] \, C_{ij}[t]$.

To extend the work in [4], [6] to TCP sources, we study the same class of subcarrier allocation schedulers, called gradient-based schedulers. The gradient-based schedulers are defined such that, at timeslot $t_k = kT_a$, the beginning of allocation frame $k$, the scheduler selects an assignment matrix maximizing a weighted sum of the rates at allocation frame $k$, i.e.,

$$\max_{[s_{ij}[t_k]]_{K\times N}} \sum_{i=1}^{K} \mu_i[t_k] C_i[t_k], \quad (2)$$

where $\mu_i[t_k]$ is the weight which will define the degree of dependence to running-average throughput or queue length. Specifically, we consider

$$\mu_i[t] = \begin{cases} (W_i[t])^{\alpha-1}, \alpha \in [0,1], \\ (Q_i[t])^{\alpha-1}, \alpha > 1, \end{cases} \quad (3)$$

where $W_i[t]$ and $Q_i[t]$ are the running-average throughput and the queue length of user $i$, respectively, at the beginning of the allocation frame.

The dynamics of the running-average throughput are

$$W_i[(k+1)T_a] = \theta W_i[kT_a] + (1-\theta) a C_i[kT_a], \quad (4)$$

and the queue dynamics are

$$Q_i[t+1] = \min\left(B, Q_i[t] + A_i[t] - U_i[t]\right), \quad (5)$$

where $\theta \in [0,1]$ indicates the degree of dependence to the current running average and $A_i[t]$ and $U_i[t]$ denote the number of packets arriving to and departing from queue $i$ of TCP source $i$ during timeslot $t$, respectively. The values of $A_i$ are dictated by TCP source $i$ and described further in Section IV.

Note that, setting $\alpha = 1$ in (3) gives a Max-SNR rule which maximizes the total feasible throughput for each timeslot, while setting $\alpha = 0$ results in the Proportional Fair (PF) scheduler which provides fairness among users. The value of $\alpha = 2$ provides a typical QBMW scheduler.

For queue-based schedulers like QBMW, within the same allocation frame, the weight $\mu_i$ in (2) which depends on $Q_i$ should decrease as subcarriers are assigned to user $i$. In addition, as subcarriers are assigned to queues, the running-average-throughput schedulers should also take into account the reduced queue lengths, although they do not update the weights $W_i[t]$, to avoid further assignment of subcarriers to queues that are already served up. We call the schedulers that iteratively take into account the gradual reducing queues, *iterative schedulers*, while those that do not, *non-iterative schedulers*.

### A. Non-iterative schedulers

The non-iterative schedulers do not update the weights $\mu_i[t]$ in (3) as subcarriers are being assigned to users. Each subcarrier $j$ is allocated one by one to the user with the largest value of $\mu_i[t]C_{ij}[t]$. At the beginning of each allocation frame, any user who has no data in the queue cannot be assigned any subcarrier. Since the weight $\mu_i[t]$ is constant during the assignment process, the allocation can be implemented in any order of subcarriers. In addition, we assume that non-iterative schedulers do not iteratively reduce the queue lengths as subcarriers are sequentially assigned.

### B. Iterative schedulers

Since the non-iterative schedulers cause an overassignment and hence degrade the system performance, a heuristic iterative version of QBMW was proposed in [6]. The iterative schedulers allocate subcarriers one by one and the queue lengths are updated as subcarriers are being assigned. For QBMW, the weight $\mu_i[t]$ is also updated. In this work, we also consider an iterative version of PF and Max-SNR rules which reduce the queue lengths as subcarriers are being assigned and do not assign further subcarriers to any user with empty queue.

Note that the iterative schedulers incur a negligible extra computational cost, compared to the non-iterative version because, at each allocation frame, it requires at most $N$ extra updates of the queue lengths.

## IV. TCP FLUID MODEL

We simulate the TCP connection behavior with a fluid model, similar to [2], [3]. This model is shown to give good accuracy with low computational cost. Here we assume that packet loss is only due to buffer overflow and each TCP source has an infinite amount of data packets to transmit.

For simplicity we look at a TCP connection to describe its fluid model. We assume the wireless link as a bottleneck link which has channel capacity $C[t]$ (packets) during timeslot $t$. At the beginning of timeslot $t$, the congestion window is denoted by $cwnd[t]$, the queue length is $Q[t]$. The amount of fluid entering and leaving the bottleneck queue (at the base station) in the interval $[t, t+1)$ are denoted by $A[t]$ and $U[t]$. The round trip time ($RTT$) is denoted by $T_m$ in terms of timeslots. Generally, we have $Q\left[t+1\right] = \min\{B, Q\left[t\right] + A\left[t\right] - U\left[t\right]\}$, and $U\left[t\right] = \min\{C\left[t\right], Q\left[t\right] + A\left[t\right]\}$.

The TCP model can be represented by the state diagram depicted in Figure 2. A TCP connection begins with Slow Start (SS) state, where $cwnd\left[t+1\right] = cwnd\left[t\right] + U[t - T_m]$, and $A\left[t\right] = 2U[t - T_m]$, until $cwnd[t] \geq ssthresh$. Then the TCP state is moved to Congestion Avoidance (CA) state where $cwnd\left[t+1\right] = cwnd\left[t\right] + \frac{U[t-T_m]}{cwnd[t]}$, and $A\left[t\right] = U\left[t-m\right] + \frac{U[t-T_m]}{cwnd[t]}$.

In the CA state, the queue grows and eventually a loss event occurs when $Q[t+1] > B$. Let $t^*$ denote the timeslot where the loss event occurs. Loss Unaware (LU) is the state where the TCP source does not realize a loss of packet yet. That is, the LU state is from $t^*$ until $t^* + m_{LU}$ where $m_{LU}$ is the smallest index satisfying $\sum_{\tau=t^*}^{t^*+m_{LU}} U[\tau] \geq Q[t^*]$. Let $cwnd_f$ be $cwnd[\cdot]$ at the end of the LU state.

Then, the Fast Retransmit (FRx) state which has $A[t] = 0$ lasts until the amount of fluid drained out is $b{\cdot}cwnd_f$ where $b \in (0, 1)$. Finally,the Fast Recovery (FRc) state which has $A[t] = U[t - T_m]$ ends when the rest $(1-b)cwnd_f$ fluid got out of the queue. At that point, $cwnd[t] = (1-b)cwnd_f$ and $ssthresh \leftarrow (1-b)ssthresh$ and CA is resumed. If the loss recovery states (FRx and FRc) take longer than $5RTT$ or $5T_m$, the *Timeout* event is activated and TCP state is back to SS where $cwnd = 1$ and $A = 1$.

## V. SIMULATION RESULTS

The simulation is performed using MATLAB, with $N = 16$ subcarriers, the average SNR of 10 dB per subcarrier, and the buffer size $B = 1,000$ packets. The $RTT$ is assumed to be constant for each TCP connection at 100 timeslots where the allocation frame duration $T_a$ is 10 timeslots. Each simulation run is done over $T = 30,000$ timeslots. The simulation begins with the SS state of the TCP which results in a low capacity utilization and, by observation, the SS state approximately takes $3,000$ timeslots. After a loss occurs, the $ssthresh$ is set to half of $cwnd_f$ ($b = 0.5$). To determine the running-average throughput $W_i(t)$ as in (4), $\theta$ is set to 0.9.

### A. Homogeneous Users with Different Starting Time

In this scenario, we let $K = 16$ users. The users are divided into two groups that start connection in the different time, i.e., Group 1 (user 1 to 12) starts at timeslot $t = 1$ while Group 2 (user 13 to 16) starts late at timeslot $t = 5001$. Figure 3 shows the average throughput per timeslot per users in the same group for all schedulers. Both non-iterative and iterative QBMW rules show extremely unfair allocation between groups. This confirms the observation in [1], [3] that TCP sources in Group 1 which start earlier than those in Group 2 tend to have longer queue lengths due to larger congestion windows and hence are assigned more subcarriers, making them more advantageous than those sources in Group 2 who are starved of bandwidth.

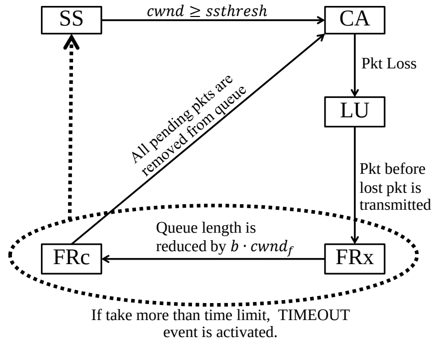


Figure 2. TCP state diagram.

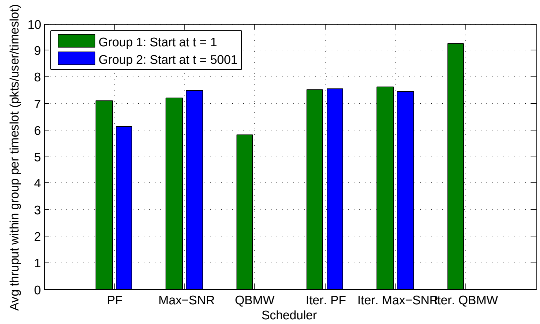


Figure 3. Homogeneous users with different starting time: Average total throughput in each group.

### B. Heterogeneous Users

Again we consider $K = 16$ users but now every TCP source starts at timeslot 1. The users have heterogeneous channels, i.e., Group 1 (user 1 to 8) has lower average channel gain, which is Rayleigh distributed with $\sigma = 1/2$, while Group 2 has better average channel with $\sigma = 1/\sqrt{2}$. As shown in Figure 4, PF and QBMW provide better fairness among the two groups than Max-SNR. Although the channel SNR of group 2 is better than that of group 1 by 40%, the non-iterative Max-SNR gives Group 2 about 5 times more throughput, while the iterative Max-SNR can improves the fairness but still rather unfair.

### C. Varying Number of Heterogeneous Users

Here we vary the number of users in the system from $K = 2$ to 32 users, while keeping the number of subcarriers at $N = 16$. The users have heterogeneous channels where the users are uniformly located in the

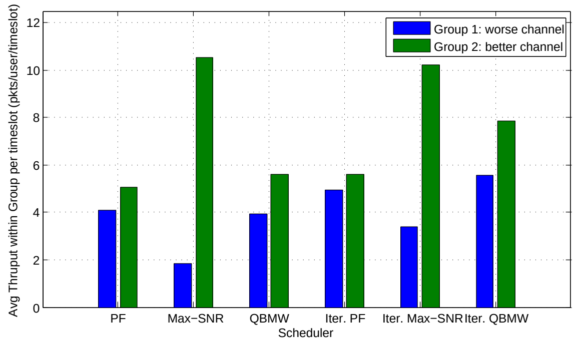


Figure 4. Heterogeneous users with same starting time: average total throughput in each group.

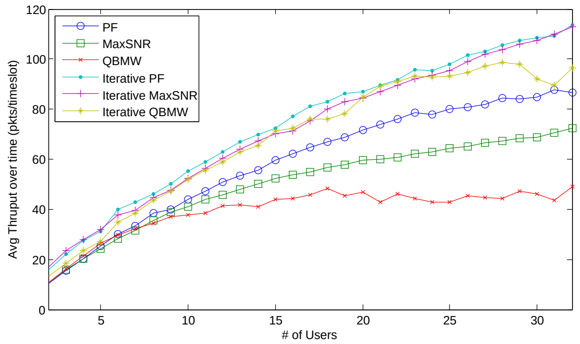


Figure 5. Varying number of heterogeneous users: average total throughput over time.

circle with the BS at the center. All TCP sources start connection at the same time. We perform 10 independent simulation runs. Figures 5, 6, and 7 show the average total throughput per timeslot, Jain's fairness index, and the average utilization which is the ratio of the actual throughput over the feasible (assigned) capacity, respectively. The Jain's fairness index is a fairness benchmark defined as $FI = \left(\sum_{i=1}^{K} R_i\right)^2 / \left(K \cdot \sum_{i=1}^{K} R_i^2\right)$, where $R_i$ is the total throughput of user $i$.

In Figure 7 the iterative version of any scheduler as expected gives better utilization than the non-iterative version of that same scheduler. Interestingly the iterative QBMW gives the best utilization due to the combination of iterative scheme and the fact that QBMW prefers serving longer queues and hence overassignment occur less often. With respect to the total system throughput and the fairness index, the iterative PF is the best scheduler. The PF scheduler is suitable to TCP traffic because, by maintaining the throughput to be near the running-average, the PF scheduler tends to give small variation in the capacity allocation. This is appreciated by TCP which adjusts the transmission rate slowly.

## VI. Conclusion

In this paper, we apply a TCP fluid model to study the interaction with PF, Max-SNR, and QBMW scheduling algorithms, in both non-iterative and iterative versions. The iterative schedulers can achieve higher throughput, better fairness, improved capacity efficiency. Specifically, the

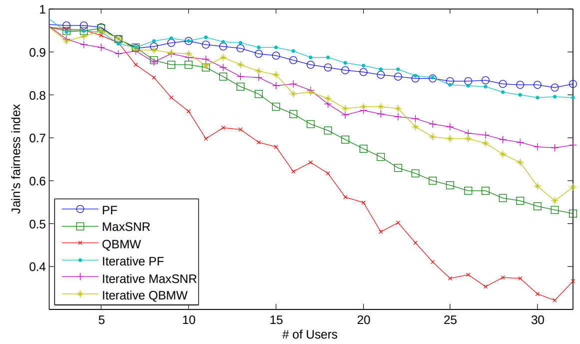


Figure 6. Varying number of heterogeneous users: Jain's fairness index.

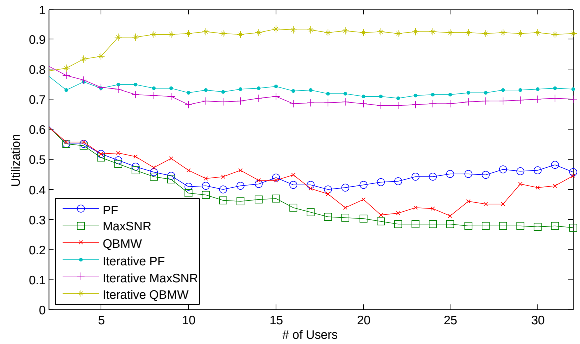


Figure 7. Varying number of heterogeneous users: utilization.

QBMW and its iterative version are not suitable to work with TCP traffic because of the extreme unfairness. The Max-SNR utilizes the multiuser diversity gain but cannot achieve good total throughput, unlike when it operates with non-responsive traffic. The iterative PF provides high throughput along with fairness, independent of the multi-user diversity and connection starting time.

## Acknowledgement

This work is supported in part by the Young Researcher Fund, Thammasat University, Thailand.

## References

[1] A. Shpiner and I. Keslassy, "Modeling the interactions of congestion control and switch scheduling," in *17th International Workshop on Quality of Service (IWQOS 2009)*, 2009, pp. 1–9.

[2] A. Baiocchi and F. Vacirca, "TCP fluid modeling with a variable capacity bottleneck link," in *26th IEEE International Conference on Computer Communications (INFOCOM 2007)*, 2007, pp. 1046–1054.

[3] A. Todini, A. Baiocchi, and D. Venturi, "Inadequacy of the queue-based max-weight optimal scheduler on wireless links with TCP sources," in *IEEE International Conference on Communications (ICC '09)*, 2009, pp. 1–6.

[4] J. Huang, V. G. Subramanian, R. Agrawal, and R. Berry., "Joint scheduling and resource allocation in downlink of OFDM systems." *IEEE Trans. Wireless Commun.*, vol. 8, no. 1, pp. 288–296, Jan. 2009.

[5] S. Bodas, S. Shakkottai, L. Ying, and R. Srikant, "Scheduling in multi-channel wireless networks: rate function optimality in the small-buffer regime," in *11th International Joint Conference on Measurement and Modeling of Computer Systems*. Seattle, WA, USA: ACM, 2009, pp. 121–132.

[6] S. Kittipiyakul, A. Boonkajay, and T. Aphichartsuphapkhajorn, "Performance of low-feedback-rate, gradient-based OFDMA subcarrier allocation with partial channel information," in *5th IEEE International Symposium on Wireless Pervasive Computing (ISWPC 2010)*, Italy, May 2010, pp. 511–516.