

Research Article

A New Swap-Based Frequency-Domain Packet Scheduling Algorithm in OFDMA System with Data Queue Size Constraints

Lin Shao,¹ Yunxiao Zu,¹ and Hanyong Shao²

¹*School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China*

²*School of Engineering, Qufu Normal University, Rizhao 276826, China*

Correspondence should be addressed to Lin Shao; linshao2001@163.com

Received 6 January 2016; Revised 11 July 2016; Accepted 11 July 2016

Academic Editor: Ruben Specogna

Copyright © 2016 Lin Shao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper aims at the frequency-domain packet scheduling (FDPS) problem in orthogonal frequency division multiple access (OFDMA) system. Under users' data queue size constraints, a new swap-based FDPS algorithm is proposed to achieve further improvement in system throughput. In this algorithm, the swap of physical resource blocks (PRBs) between different users is introduced to give a comprehensive view of the overall scheduling process. Moreover, the proposed algorithm optimizes the choosing method of swap candidates and always tries to select the user who can maximize the throughput improvement. Simulation results demonstrate that this new algorithm can improve the system throughput significantly as well as reduce the resource waste effectively.

1. Introduction

Due to the explosive growth of users' demand, 3GPP Long Term Evolution (LTE) has been broadly acknowledged as the most promising standard for next generation cellular systems [1]. In LTE downlink, orthogonal frequency division multiple access (OFDMA) is used as the radio access technology for its robustness to multipath fading and higher spectral efficiency [2, 3]. Multiple access is achieved by assigning different frequency portions of the total bandwidth to individual users in OFDMA [4]. The basic radio resource unit of the system bandwidth is called physical resource block (PRB). Before the assignment of PRBs, the instantaneous channel state information (CSI) needs to be collected from users. Such assignment of PRB based on the quality of channels is called frequency-domain packet scheduling (FDPS).

References [5–8] on FDPS have shown potential gains of system capacity with the assumption of the infinitely backlogged queue model. Reference [9–11] focused on PRB allocation to maximize the system throughput regardless of users' data queue size (i.e., data prepared to be transmitted per unit time). There are other existing works which had a

different optimization objective; however, these works did not consider users' data queue size as well. For example, in [12–14], PRBs are allocated to achieve proportional fairness (PF) among users using the infinitely backlogged queue model. However, this model is not suitable for every situation. Recent studies [15–17] on traffic measurement have shown that part of users have a small amount of data available for transmission during off-peak hours. In this case, if users' data queue size is neglected, the channel rate of scheduled users may exceed the amount of data prepared to be transmitted per unit time. It may reduce the system throughput, and PRB resources will not be utilized effectively. So the finite data queue size of users should be taken into account.

More attention has been paid to users' data queue size constraints in recent works. For example, [18] performed optimal resource assignment by taking into account the channel quality as well as queue backlogs of each user. Reference [19] presented a cross-layer design optimization for subchannel and power allocation in OFDMA system with users' queue status being considered. Reference [20] proposed a new type of queue-aware channel-adapted scheduling technique which can be integrated into different types of

scheduling schemes in the LTE network. Users' data queue size was discussed in [4] and a swap-based FDPS algorithm was put forward to improve the system throughput. In the algorithm, the isolated resource-assignment strategy was replaced in order to approach the optimal outcome. However, more effective swap candidate selection can achieve higher throughput improvement. In this paper, a new swap-based FDPS algorithm is proposed and the choosing method of swap candidates is optimized. Simulation results demonstrate the superior performance of the proposed algorithm in both improving the system throughput and reducing the resource waste.

The rest of this paper is organized as follows. Swap-based FDPS algorithm in [4] is analyzed in Section 2. Our new swap-based FDPS algorithm is proposed in Section 3. Complexity analysis is in Section 4. Simulation results and discussions are in Section 5 and Section 6 gives the conclusions.

2. Analysis of Swap-Based FDPS Algorithm

Existing PF FDPS algorithm schedules PRBs one by one by using PF algorithm on every PRB in turn. The original PF metric is shown in (1) and for each PRB, the user \hat{i} with the maximum metric value will be selected [4]:

$$\hat{i} = \arg \max_i \frac{r_{i,c}(t)}{R_i(t)}, \quad (1)$$

where $r_{i,c}(t)$ is the supportable instantaneous data rate of user i on PRB c (c is the PRB index) at time t and $R_i(t)$ is the average data rate of user i until time t .

PF metric (1) does not involve the data queue size, so it is likely to provide more service than users' demand. In order to overcome this shortcoming, PF metric (1) is modified in [4] as follows:

$$\hat{i} = \arg \max_i \frac{\min\{r_{i,c}(t), q_i^c(t)\}}{R_i(t)}, \quad (2)$$

where $q_i^c(t)$ is the data queue size of user i when the scheduling procedure is on PRB c . $q_i^c(t)$ is updated using the following formula:

$$q_i^{c+1}(t) = \begin{cases} \max\{0, q_i^c(t) - r_{i,c}(t)\} & \text{if } i = \hat{i} \\ q_i^c(t) & \text{otherwise.} \end{cases} \quad (3)$$

Remark 1. The main progress made by PF metric (2) is the introduction of the data queue size. If a user's channel rate overtakes (or is equal to) its data queue size, the user's metric value on every unscheduled PRB will be set to zero. This means that the user will not be given PRBs any longer, which by contrast is inevitable when PF metric (1) is used. Extra PRBs given to the user make no contribution to the system throughput and waste the resource as well; hence, PF metric (2) is one step forward of PF metric (1) in improving the system throughput and reducing the resource waste.

TABLE 1: The comparison of PF metric (1) and PF metric (2).

For PRB 1					
i	Q_i	$r_{i,1}(t)$	$R_i(t)$	PF metric (1) value	PF metric (2) value
1	100	120	60	2	1.667
2	80	80	80	1	1
For PRB 2					
i	Q_i	$r_{i,2}(t)$	$R_i(t)$	PF metric (1) value	PF metric (2) value
1	100	60	60	1	0
2	80	70	80	0.875	0.875

TABLE 2: The scheduling result.

PF metric	Scheduling result	f_1	f_2	$f = \sum_i f_i$
PF metric (1)	PRB 1—user 1	80	0	80
	PRB 2—user 1			
PF metric (2)	PRB 1—user 1	20	0	20
	PRB 2—user 2			

In order to illustrate how much resource has been wasted, a new criterion is introduced as follows:

$$f_i = \begin{cases} r_i - Q_i & \text{if } r_i > Q_i \\ 0 & \text{otherwise,} \end{cases} \quad (4)$$

where r_i is the supportable data rate of user i when a scheduling process of all PRBs is finished and Q_i is the data queue size of user i before all PRBs are scheduled. Obviously, f_i shows the useless part of r_i and the more f_i is, the more resource waste an algorithm has produced on user i . We can use $f = \sum_i f_i$ to illustrate how much resource waste an algorithm has produced on all users. Consider a simple example with two users and two PRBs in Table 1. The scheduling result comparison between PF metric (1) and PF metric (2) is shown in Table 2. Suppose that the assignment of PRB 2 is after the assignment of PRB 1.

According to PF metric (1), both the PRBs will be assigned to user 1. Therefore, $f = \sum_i f_i = (120 + 60 - 100) + 0 = 80$. However, according to PF metric (2), PRB 1 is scheduled to user 1 and PRB 2 is scheduled to user 2. Therefore, $f = \sum_i f_i = (120 - 100) + 0 = 20$. Obviously, PF metric (2) has a better performance than PF metric (1) in reducing the resource waste.

Although PF metric (2) is more responsible for finite data queue size than PF metric (1), it can still suffer from the isolated PRB assignment strategy. The shortcoming of this strategy is allocating PRBs one by one regardless of other PRB assignment status. More parameters should be taken into account to address this problem. Therefore, a swap-based FDPS algorithm has been introduced in [4]. The algorithm is shown in detail in Algorithm 1.

The swap-based FDPS algorithm can be briefly described as follows: (1) pick the best user \hat{i} who maximizes PF metric

- (1) Let λ_i^c be the PF metric (2) value of user i on PRB c and M is the total number of PRBs.
- (2) **for** PRB $c = 1$ to M **do**
- (3) select the best user \hat{i} with largest value λ_i^c on PRB c
- (4) **if** there is a user j satisfying $r_{j,c}(t) > q_j^c(t)$, who would rank first in the scheduling priority of PRB c if enough data in the queue **then**
- (5) Let k be max-rate PRB already assigned to user j
- (6) **if** swapping PRB c for k can bring higher throughput **then**
- (7) re-assign PRB k to user \hat{i}
- (8) assign PRB c to user j
- (9) update the data queue size of user \hat{i} and user j
- (10) **end if**
- (11) **end if**
- (12) **end for**

ALGORITHM 1: Swap-based FDPS algorithm.

(2) on PRB c ; (2) seek the potential candidate user ready for a PRB swap according to the established rules; (3) carry on a swap if higher throughput can be obtained.

The three steps above are implemented on all PRBs one by one. Obviously, the third step is the most important, because every swap is bound to improve the system throughput. The more times swap happens, the more throughput Algorithm 1 improves. However, a swap needs several conditions. Only if every condition is satisfied will a swap happen. The worst case is that no swap takes place and Algorithm 1 has the same performance as PF metric (2) in the system throughput. On the other hand, how much throughput improvement a swap brings is also a key factor besides swap occurrence.

In conclusion, we hope that swap happens more frequently and each swap can result in higher system throughput improvement. Based on these two points, Algorithm 1 is analyzed as follows.

Remark 2. Algorithm 1 only chooses the user who maximizes PF metric (1) on PRB c as the potential candidate user, which can be deduced from step 4 in Algorithm 1. More users need to be added to candidates in order to increase the possibility of swap occurrence.

Algorithm 1 sets four conditions for a swap: the potential candidate user must be the user j who maximizes PF metric (1) on PRB c ; user j must satisfy $r_{j,c}(t) > q_j^c(t)$; user j has been given at least one PRB already; $T_{\hat{i}} + T_j > t_{\hat{i}} + t_j$, where $T_{\hat{i}}, T_j$ are the throughput obtained by user \hat{i} and user j after a swap and $t_{\hat{i}}, t_j$ are the throughput obtained by user \hat{i} and user j before the swap. Only if the four conditions are satisfied will a swap happen. How much throughput is improved by a swap is determined by $\{(T_{\hat{i}} + T_j) - (t_{\hat{i}} + t_j)\}$. $\{(T_{\hat{i}} + T_j) - (t_{\hat{i}} + t_j)\}$ can be further expanded as follows:

$$\begin{aligned} (T_{\hat{i}} + T_j) - (t_{\hat{i}} + t_j) &= (T_{\hat{i}} - t_{\hat{i}}) + (T_j - t_j) \\ &= \left(\min \{r_{\hat{i},k}(t), q_{\hat{i}}^c(t)\} - \min \{r_{\hat{i},c}(t), q_{\hat{i}}^c(t)\} \right) \end{aligned}$$

$$\begin{aligned} &+ \left(\min \left\{ \sum_{\omega \in \Omega} r_{j,\omega}(t) + r_{j,c}(t), Q_j \right\} \right. \\ &\left. - \min \left\{ \sum_{\omega \in \Omega} r_{j,\omega}(t) + r_{j,k}(t), Q_j \right\} \right), \end{aligned} \quad (5)$$

where Ω is the set of PRBs which have already been allocated to user j except PRB k . Formula (5) demonstrates that $r_{\hat{i},k}(t)$, $r_{j,c}(t)$, $\sum_{\omega \in \Omega} r_{j,\omega}(t)$, $r_{j,k}(t)$, and Q_j influence the throughput improvement together, so swapping PRBs with user j may not result in the highest throughput improvement.

3. New Swap-Based FDPS Algorithm

Based on Remark 2, a new swap-based FDPS algorithm is put forward to improve the system throughput. The details of the new algorithm are described in Algorithm 2.

Remark 2 demonstrates that Algorithm 1 needs to choose candidates more reasonably to achieve further improvement in the system throughput. As is mentioned in Remark 2, more users ought to be added to candidates. However, the amount of candidates should not be so large that too many calculations are needed. Choosing all users (except user \hat{i}) as candidates is optimal but too complex to be implemented in practice. For this reason, user i satisfying $r_{i,c}(t) > q_i^c(t) \& K_i \neq \Phi$ will be chosen as a candidate. As there may be more than one qualified user, the possibility of swap occurrence will increase. At the same time, the condition $r_{i,c}(t) > q_i^c(t) \& K_i \neq \Phi$ picks up only part of all users in order to control the amount of candidates as well. It is reasonable to set this condition because qualified users have the potential to improve the system throughput if they are given PRB c . Users satisfying the condition may not get a high rank of scheduling priority on PRB c due to their lack of data for transmission. However, these users may obtain high throughput on PRB c if there is enough data to be transmitted, so it is reasonable to choose them as swap candidates.

```

(1) Let  $\lambda_i^c$  be the PF metric (2) value of user  $i$  on PRB  $c$  and  $M$  is the total number of PRBs
(2) for PRB  $c = 1$  to  $M$  do
(3) pick the best user  $\hat{i}$  with largest value  $\lambda_i^c$  on PRB  $c$ 
(4) Let  $U$  be the set of users and  $U = \{i \neq \hat{i} \mid r_{i,c}(t) > q_i^c(t) \ \& \ K_i \neq \Phi\}$  where  $K_i$  is the set of PRBs
    have already been assigned to user  $i$ 
(5) if  $U \neq \Phi$ 
(6)   select the user  $i_0 = \arg \max_{i \in U} \{(T_{\hat{i}} + T_i) - (t_{\hat{i}} + t_i)\}$ 
(7)   if  $\{(T_{\hat{i}} + T_{i_0}) - (t_{\hat{i}} + t_{i_0})\} > 0$ 
(8)     re-assign PRB  $d$  to user  $\hat{i}$  where PRB  $d$  is the max-rate PRB already assigned to user  $i_0$ 
(9)     assign PRB  $c$  to user  $i_0$ 
(10)    update the data queue size of user  $\hat{i}$  and user  $i_0$ 
(11)  else
(12)    assign PRB  $c$  to user  $\hat{i}$ 
(13)    update the data queue size of user  $\hat{i}$ 
(14)  end if
(15) else
(16)  assign PRB  $c$  to user  $\hat{i}$ 
(17)  update the data queue size of user  $\hat{i}$ 
(18) end if
(19) end for

```

ALGORITHM 2: New swap-based FDPS algorithm.

On the other hand, user i_0 is finally chosen for the swap in order to achieve the highest throughput improvement among all the candidates. As the swap candidate set of Algorithm 1 must be a subset of that of Algorithm 2, Algorithm 2 certainly will find a user who will bring no lower throughput improvement than user j for each swap.

Remark 3. For Algorithm 2, the number of users has little effect on the throughput as long as all PRBs are allocated, since the throughput mainly depends on the bandwidth assigned. As for f , it will also change little, in general, when the number of users increases or decreases. The reason is that Algorithm 2 prevents users from continuing to receive PRBs when the users' data requirements are satisfied, and extra PRBs will be given to the users with full buffer. Therefore, unless there is no user with full buffer, the wasted data f can be controlled.

The only exceptional case is when all the users have finite data queue size. In this case, as the number of users increases, the throughput will increase, and f will decrease. But both will tend to be steady when the number of users increases to a definite value.

The main progress made by Algorithm 2 is as follows:

- (1) The algorithm selects more users as candidates to increase swap occurrence.
- (2) It sets reasonable conditions to control the candidate amount.
- (3) It chooses the user who maximizes the throughput improvement from all the candidates for each swap.

4. Complexity Analysis

PF FDPS algorithm using PF metric (2) goes through all M PRBs one by one and selects the best user from PF metric among N users (N is the amount of users) for each PRB, so the computation complexity is $O(N \cdot M)$. In comparison, Algorithm 1 contains extra swap process, so its complexity is $O(M) \cdot (O(N) + O(N)) = O(N \cdot M)$. Algorithm 2 chooses more swap candidates than Algorithm 1 according to different filter condition and its complexity becomes $O(M) \cdot (O(N) + O(X_c)) = O(M) \cdot O(\max(N, X_c))$, where X_c is the number of swap candidates for each PRB and must be less than N . Hence the complexity of Algorithm 2 is $O(N \cdot M)$. So Algorithms 2 and 1 and PF FDPS algorithm using PF metric (2) are in the same order of magnitude.

5. Simulation Results and Discussions

In this section, PF metric (2) FDPS algorithm and Algorithms 1 and 2 are evaluated by LTE downlink simulations on system level. In the simulation, part of users have data queue size constraints, while the others have infinite data for transmission. The performances of the three algorithms are shown in different situations where the number of users with finite data queue size is changed. The three algorithms are compared in both the system throughput and f value. The simulation parameters are listed in Table 3.

In Table 3, d is the distance between base station and the user and its unit is km. The simulation results are shown in Figures 1, 2, and 3.

Figure 1 illustrates the system throughput comparison among PF metric (2) FDPS algorithm and Algorithms 1 and 2. Algorithm 2 results in the highest throughput in

TABLE 3: Simulation parameters.

Parameter	Setting
System bandwidth	10 MHz
Subcarriers per PRB	12
PRB bandwidth	180 kHz
Number of PRBs	50
Cell-level user distribution	Uniform
Number of active users in cell	20
Path-loss model	$128.1 + 37.6 * \lg(d)$
Channel model	Rayleigh
Transmit power of the base station	43 dBm

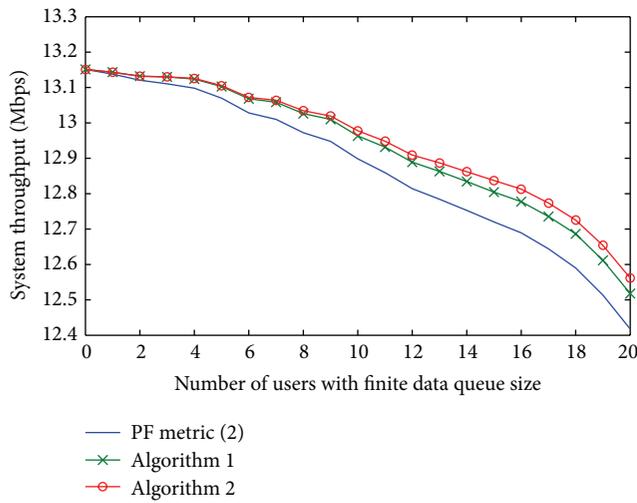


FIGURE 1: System throughput comparison for PF metric (2) FDPS algorithm and Algorithms 1 and 2.

most scenarios, because every swap in Algorithm 2 brings about the highest throughput improvement among the three algorithms. Moreover, Algorithm 2 performs better than Algorithm 1, mainly because Algorithm 2 has more choices for swap candidates and always tries to select the user who maximizes the throughput improvement. The three algorithms have the same performance when the number of users with finite data queue size is zero. The reason is that as the channel rate of every user on each PRB cannot overtake the user's data queue size at any time, no swap will happen in both Algorithms 1 and 2. Therefore, the three algorithms will definitely select the same user for each PRB.

Table 4 shows the total required data for transmission corresponding to different number of users with finite data queue size. Note that the required data is only for the users with finite data queue size, as the other users all have full buffers. Figures 2 and 3 illustrate the value of f (i.e., the wasted data) comparison among PF metric (2) FDPS algorithm and Algorithms 1 and 2. The result shows that Algorithm 2 has the lowest value of f corresponding to the same horizontal ordinates among the three algorithms.

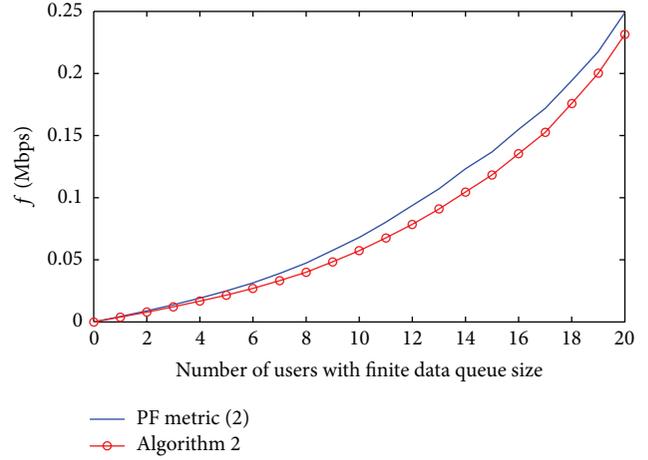


FIGURE 2: Value of f comparison between PF metric (2) FDPS algorithm and Algorithm 2.

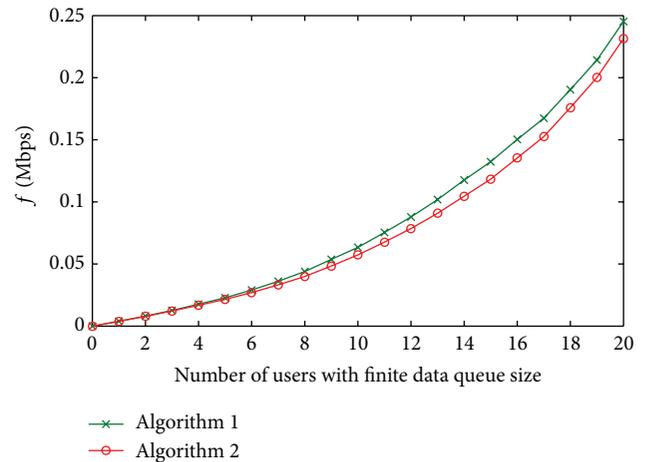


FIGURE 3: Value of f comparison between Algorithms 1 and 2.

Therefore, Algorithm 2 performs better than the other two algorithms in reducing the resource waste. Overall, Figures 1–3 demonstrate that Algorithm 2 has the best performance in both improving the system throughput and reducing the resource waste.

6. Conclusion

In this paper, a new swap-based algorithm was put forward to address the FDPS problem in OFDMA system. The proposed algorithm aims at improving the system throughput under data queue size constraints. In order to maximize the throughput improvement, we designed reasonable rules to select swap candidates effectively. It was demonstrated by simulation that the algorithm had a superior performance in

TABLE 4: Total required data for transmission.

Number of users with finite data queue size	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Total required data for transmission (Mbps)	0	0.7	1.4	2.1	2.8	3.5	4.2	4.9	5.6	6.3	7	7.7	8.4	9.1	9.8	10.5	11.2	11.9	12.6	13.3	14

improving the system throughput and reducing the resource waste.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work is supported by “Beijing Key Laboratory of Work Safety Intelligent Monitoring (Beijing University of Posts and Telecommunications)” and National Natural Science Foundation of China under Grant no. 61374090.

References

- [1] X. Xiao, X. Tao, and J. Lu, “Energy-efficient resource allocation in LTE-Based MIMO-OFDMA systems with user rate constraints,” *IEEE Transactions on Vehicular Technology*, vol. 64, no. 1, pp. 185–197, 2015.
- [2] C. Turyagyenda, T. O’Farrell, and W. Guo, “Energy efficient coordinated radio resource management: a two player sequential game modelling for the long-term evolution downlink,” *IET Communications*, vol. 6, no. 14, pp. 2239–2249, 2012.
- [3] M. M. Matalgah, O. M. Hammouri, and B. Paudel, “Cross-layer capacity optimisation in WiMAX orthogonal frequency division multiple access systems with multi-class quality of services and users queue status,” *IET Communications*, vol. 8, no. 14, pp. 2500–2508, 2014.
- [4] S.-B. Lee, “Swap-based frequency-domain packet scheduling algorithm for small-queue condition in OFDMA,” *IEEE Communications Letters*, vol. 17, no. 5, pp. 1028–1031, 2013.
- [5] A. Pokhariyal, T. E. Kolding, and P. E. Mogensen, “Performance of downlink frequency domain packet scheduling for the utran long term evolution,” in *Proceedings of the IEEE 17th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC ’06)*, pp. 1–5, Helsinki, Finland, September 2006.
- [6] A. Pokhariyal, K. I. Pedersen, G. Monghal et al., “HARQ aware frequency domain packet scheduler with different degrees of fairness for the UTRAN long term evolution,” in *Proceedings of the IEEE 65th Vehicular Technology Conference (VTC-Spring ’07)*, pp. 2761–2765, Dublin, Ireland, April 2007.
- [7] C. Wengertter, J. Ohlhorst, and A. G. E. Von Elbwart, “Fairness and throughput analysis for generalized proportional fair frequency scheduling in OFDMA,” in *Proceedings of the IEEE 61st Vehicular Technology Conference (VTC ’05)*, pp. 1903–1907, Stockholm, Sweden, June 2005.
- [8] S.-B. Lee, S. Choudhury, A. Khoshnevis, S. Xu, and S. Lu, “Downlink MIMO with frequency-domain packet scheduling for 3GPP LTE,” in *Proceedings of the 28th Conference on Computer Communications (IEEE INFOCOM ’09)*, pp. 1269–1277, Rio de Janeiro, Brazil, April 2009.
- [9] H.-L. Liu and Q. Wang, “A hybrid evolutionary algorithm for OFDM resource allocation,” in *Proceedings of the IEEE Symposium on Computational Intelligence for Engineering Solutions (CIES ’13)*, pp. 43–47, Singapore, April 2013.
- [10] J. Shi, G. Yu, Z. Zhang et al., “Resource allocation in OFDM based multihop wireless networks,” in *Proceedings of the Vehicular Technology Conference (VTC ’06)*, pp. 319–323, Melbourne, Australia, May 2006.
- [11] N. Zhou, X. Zhu, Y. Huang, and H. Lin, “Adaptive resource allocation for multi-destination relay systems based on OFDM modulation,” in *Proceedings of the IEEE International Conference on Communications (ICC ’09)*, pp. 1–5, IEEE, Dresden, Germany, June 2009.
- [12] S.-B. Lee, I. Pefkianakis, A. Meyerson, S. Xu, and S. Lu, “Proportional fair frequency-domain packet scheduling for 3GPP LTE uplink,” in *Proceedings of the 28th International Conference on Computer Communications (IEEE INFOCOM ’09)*, pp. 2611–2615, Rio de Janeiro, Brazil, April 2009.
- [13] L. Xiao and L. Cuthbert, “A two-hop proportional fairness scheduling algorithm for relay based OFDMA systems,” in *Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM ’08)*, pp. 1–4, Dalian, China, October 2008.
- [14] T. Zhang, Z. Zeng, and Y. Qiu, “A subcarrier allocation algorithm for utility proportional fairness in OFDM systems,” in *Proceedings of the IEEE 67th Vehicular Technology Conference (VTC ’08)*, pp. 1901–1905, Singapore, May 2008.
- [15] S.-B. Lee, D. Pei, M. Hajiaghayati et al., “Threshold compression for 3G scalable monitoring,” in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM ’12)*, pp. 1350–1358, Orlando, Fla, USA, March 2012.
- [16] C. Peng, S.-B. Lee, S. Lu, H. Luo, and H. Li, “Traffic-driven power saving in operational 3G cellular networks,” in *Proceedings of the 17th Annual International Conference on Mobile Computing and Networking (MobiCom ’11)*, pp. 121–132, Las Vegas, Nev, USA, September 2011.
- [17] Q. Xu, J. Erman, A. Gerber, Z. Mao, J. Pang, and S. Venkataraman, “Identifying diverse usage behaviors of smartphone apps,” in *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC ’11)*, pp. 329–344, Berlin, Germany, November 2011.
- [18] H. Ahmed, K. Jagannathan, and S. Bhashyam, “Queue-aware optimal resource allocation for the LTE downlink,” in *Proceedings of the IEEE Global Communications Conference (GLOBECOM ’13)*, pp. 4122–4128, Atlanta, Ga, USA, December 2013.
- [19] M. M. Matalgah, B. Paudel, and O. M. Hammouri, “Cross-layer resource allocation approach in OFDMA systems with multi-class QoS services and users queue status,” in *Proceedings of the*

IEEE Global Communications Conference (GLOBECOM '13), pp. 1385–1390, Atlanta, Ga, USA, December 2013.

- [20] A. Zolfaghari and H. Taheri, "Queue-aware channel-adapted scheduling and congestion control for best-effort services in LTE networks," *Canadian Journal of Electrical and Computer Engineering*, vol. 38, no. 2, Article ID 7120215, pp. 170–182, 2015.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

