Zhu and Yang *EURASIP Journal on Wireless Communications and Networking* (2015) 2015:176
DOI 10.1186/s13638-015-0398-y

EURASIP Journal on Wireless Communications and Networking
a SpringerOpen Journal

RESEARCH Open Access

Buffer-aware adaptive resource allocation scheme in LTE transmission systems

Ruiyi Zhu† and Jian Yang*†

Abstract

Dynamic resource allocation scheme is a key component of 3GPP long-term evolution (LTE) for satisfying quality-of-service (QoS) requirement as well as improving the system throughput. In this paper, a buffer-aware adaptive resource allocation scheme for LTE downlink transmission is proposed for improving the overall system throughput while guaranteeing the statistic QoS and keeping certain fairness among users. Specifically, the priorities of the users' data queues in the base station are ranked by their remaining life time or their queue overflow probability which is estimated by applying large deviation principle. An online measurement based algorithm which requires no statistical knowledge of the network conditions uses the queue priorities to dynamically allocate the resource blocks (RBs) for avoiding buffer overflow and providing statistic QoS guarantee. The simulation results show that the proposed algorithm improves the throughput and fairness while considerably reducing the average bit loss rate.

Keywords: Dynamic resource allocation; LTE; QoS; Large deviation principle

Introduction

Mobile communication technologies have been developed rapidly, and switched from the third generation (3G) of mobile communication systems to the long-term evolution (LTE) systems, which aims to provide high-data-rate, low-latency, packet-optimized radio-access, and flexible bandwidth deployments [1]. LTE system allows high flexibility in the resource allocation, which enables dynamic resource blocks (RBs) allocation among the potential users [2] and [3].

Conventional resource allocation schemes in wireless system are generally based on user's priority [4, 5]. They are designed according to user's channel status and QoS guarantee to maximize overall system throughput. However, providing fairness among users is another essential design consideration, although it usually sacrifices the system throughput and/or violates QoS requirements. Some resource allocation schemes based on buffer-aware can improve some of these performance metrics [6]. The resource allocation problem in wireless system has been widely addressed in some literatures, but it is still challenging in LTE system to design a buffer-aware resource allocation scheme for improving the performance including increasing the system throughput as large as possible, guaranteeing QoS requirements, and achieving fairness.

In this paper, we propose a buffer-aware adaptive resource allocation scheme by jointly considering the user scheduling and RBs allocation to provide QoS guarantee in LTE transmission systems. In the aspect of user scheduling, considering that the finite buffer maximum size, each user's queue priority is ranked according to its remaining life time or its queue overflow probability which is estimated by applying large deviation principle. For RBs allocation, an online measurement based algorithm for dynamically allocating RBs is proposed for adjusting the service rates of the user queues in order to provide QoS guarantee. The goal is improving the total system throughput as large as possible while subjecting to provide QoS guarantee for different users and to guarantee certain fairness.

Related work

In this paper, we consider multiuser resource allocation for the downlink in LTE systems. The scheduler at the base station is responsible for allocating resources to the different users as a function of the users' queue priority

*Correspondence: jianyang@ustc.edu.cn
†Equal contributors
University of Science and Technology of China, Hefei, China

as well as the current channel conditions. There are many prior works on this problem. The classic scheduling algorithms include Round Robin (RR) algorithm [7], Max C/I algorithm [8], and proportional fair (PF) algorithm [9]. Although many works [10–13] apply multiuser diversity in user scheduling for maximizing system throughput, the system buffer size in these schemes is assumed to be infinite, that is to say, any arriving bit can be buffered and any bit loss due to buffer overflow will not happen. This assumption may not be reasonable since the buffer size is limited in the transceivers.

Resource allocation for finite buffer space has been discussed in the literature related to the wireless network. The authors in [14] design a new LTE buffer aware scheduler to opportunistically assign RBs for video streaming applications in order to maximize the average video quality. In [15], the buffer occupancy based approach is presented to achieve video rate adaptation, while in [16], a dynamic programming framework is applied to study the buffer *vs*. QoS tradeoff for wireless media streaming in a single user scenario. These papers cited above mainly focus on video traffic. But the eNB in the practical situation schedules and transmits general data traffic besides video traffic.

There are several related works for packet scheduling and resource allocation in wireless data systems. In [17], M. Andrews et al. focus on how to adapt MaxWeight algorithm to the multicarrier wireless data systems, and a simple variant was introduced into the objective for reducing resource wastage. In [18], M. Realp et al. propose a resource allocation algorithm in multiuser OFDMA by considering queue and channel state information. However, these methods focus on maximizing the overall throughput by improving spectral efficiency, which may lead to unfair resource sharing among users. In fact, fairness is necessary to guarantee minimum performance of the users experiencing bad channel conditions. The buffer-aware adaptive resource allocation proposed for LTE system in this paper will consider the problem of keeping certain fairness while improving the total system throughput.

Due to the limited available resource, RBs allocation aims to efficiently use the shared resource and allocate the resource in a fair manner. Naturally, there is a tradeoff between fairness and system throughput. PF algorithm has emerged as a prominent candidate since it balances between fairness and throughput. In [19], S. Lee proposes a sub-optimal method, i.e., PF metric(2), which introduces the status of queues into PF metric(1). However, it is pointed out in [19] that although PF metric(2) is more responsive to the queues than PF metric(1), it incurs a reduced system throughput because its isolated RB assignment strategy may assign the RB to a user having low channel quality. Similar work related to PF scheduling in LTE systems can also be found in [20, 21] and [22]. By considering both fairness and the constraint of finite buffer space, a channel-adapted and buffer-aware (CABA) packet scheduling algorithm is proposed in [23]. This method defines and applies the user priority in the resource allocation for avoiding buffer overflow. However, the empirical parameters in the priority function are hard to appropriately choose. Inappropriate parameters will lead to an inaccurate user priority, which induces excessive resource allocated to the users and reduce the utilization of the system resource.

The eNodeB may have large capacity to cache traffic such as audio and video streams, but it substantially increases the delay and reduces QoS. Hence, we consider the finite buffer size and queue overflow probability in this paper. We will jointly exploit the priorities of user queues and the RBs capacity for controlling the service rate of each user data queue in the base station, instead of solely relying on any one of them. Under the constraint of finite buffer space, the proposed buffer-aware adaptive resource scheduling algorithm aims at achieving three objectives: (1) keep bit loss rate as low as possible by means of taking buffer status into account, (2) improve the total system throughput as large as possible, and (3) keep certain fairness among users by means of adjusting the overflow probability.

Contributions

In this paper, we proposed a buffer-aware adaptive resource allocation scheme for LTE downlink transmission by jointly exploiting the priorities of user queues and RBs capacity. The proposed problem is formulated as improving the total system throughput subject to providing QoS guarantee for different users while keeping certain fairness. Specifically, our main contributions are listed as follows:

- User scheduling: Firstly, the user scheduling scheme is considering the finite buffer size. Secondly, the scheme is depending on the users' queue priority which is calculated by their remaining life time or their queue overflow probability. The overflow probability estimation model is derived by applying the large deviation principle [24], which incorporates both the queue fullness and its variation.
- RBs allocation: An online measurement-based algorithm is further presented to adjust the service rate of the user queues, which requires no statistical knowledge of the network conditions. According to the user queues' priorities, we control the service rate of each user queue by dynamically allocating the RBs, in order to avoid queue overflow and provide statistic QoS guarantee.

- We present experimental results to show that the proposed algorithm is able to improve the total system throughput while guaranteeing certain fairness among users and providing QoS guarantee.

The rest of the paper is organized as follows: Section 'System model and problem statement' describes a system and channel model for resource allocation. In section 'User priority determination scheme', we present user priority determination algorithm, including calculating the remaining life time or queue overflow probability. Section 'Online measurement-based algorithm for dynamic RBs allocation' is devoted to describing the online measurement-based algorithm for dynamic resource allocation. Section 'Performance evaluation' provides the experimental results and performance comparisons. Finally, conclusions are drawn in Section 'Conclusion'.

System model and problem statement

System model

We consider LTE system architecture with downlink RBs allocation as shown in Fig. 1. The eNode B (eNB) controls the bit service rate through dynamically allocating RBs to users. The total number of the data bits within a RB is referred to as RB capacity. The better channel condition of an RB implies a higher achievable RB capacity. Different RBs may have distinct channel conditions [20]. The smallest resource unit that can be allocated to a user is a scheduling block (SB), which consists of two consecutive RBs [25, 26]. In each time slot, several SBs may be allocated to a single user, but each SB is uniquely assigned to a user.

We focus on single-cell downlink resource allocation in eNB of LTE system employing OFDMA. The implementation of adaptive resource scheduling in eNB relies on the following factors: buffer status (e.g., unoccupied buffer space and current queue length), traffic characteristics (e.g., bit arrival rate) and channel quality. We assume that eNB has perfect and instant channel information for all downlink transmissions via the feedback channel, while the channel quality is assumed stationary for the duration of each subframe, but may vary from subframe to subframe. Since the data queue of each user locates in eNB, it is natural that eNB knows the amount of each user's data in the transmission-side buffer without additional signaling to report.

Let K and N, respectively, denote the number of the users and the number of all SBs. In the practical situation, eNB does not differentiate the transmitting data types. Hence, one user is assumed to have a single queue. Then, the kth queue length can be updated as

$$Q_k(t+1) = max\{Q_k(t) - V_k(t), 0\} + A_k(t), \tag{1}$$

where $A_k(t) \in A = \{0, 1, \ldots, m_A\}$ denotes the bit arrival number of the kth user queue during the slot t, and m_A is the maximum number of bits arriving in a single slot.

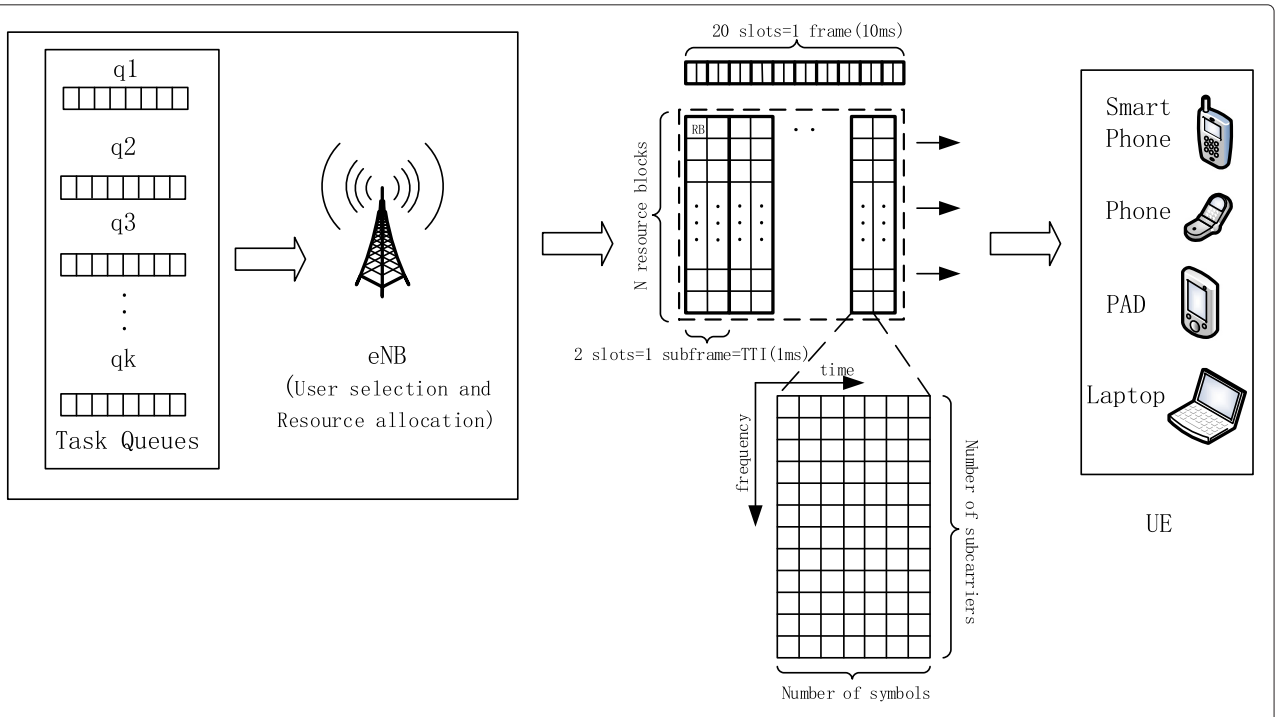


Fig. 1 System architecture

Here, we assume that the bit arrival process of the kth user queue $A_k(t)$ to be an *i.i.d* sequence. $V_k(t) \in V = \{0, 1, \ldots, m_V\}$ represents the bit number transmitted during the slot t, where m_V is the maximum number of bits served in a single slot. Define $Q_k(t)$ as the length of the kth queue in terms of bits at the beginning of the slot t. Here, we consider the practical scenario of finite buffer space and integrate the buffer status into the scheduling decision to decrease the bit loss rate. It is noted that buffer overflow implies the resource is not enough for transmitting the data in the queue, and the data has to be dropped when the buffer is full, while buffer underflow means that the resource is sufficient for conveying the data in the queue, and data loss will not occur. Hence, we only consider the buffer overflow in the transmitter-side (eNB-side). Let us define a threshold Q_k^{max} as the the maximum length of the kth user queue. If the kth queue length is higher than Q_k^{max}, it implies that an excessive number of bits is buffered, and bit loss may be likely to occur. Naturally, any queue length exceeding Q_k^{max} is undesirable. Hence, the problem may be described as that of selecting an appropriate service rate to keep each queue length lower than Q_k^{max}. The kth user queueing system model is shown in Fig. 2. When the arrival rate is larger than the service rate, bit loss may occur due to the queue overflow. In order to reduce the average bit loss rate (BLR), we plan to apply the large deviation algorithm to calculate the queue overflow probability. Accordingly, we define overflow probability of the kth queue as

$$P_{k_{overflow}} = P\left(Q_k(t) > Q_k^{max}\right). \tag{2}$$

For a given slot t, the increase of the kth queue length is characterized by

$$I_k(t) = A_k(t) - V_k(t). \tag{3}$$

We define remaining life time $R_k(t)$ to denote the remaining time of the kth user queue to be fullness. Then the remaining life time of the kth user at the slot t can be calculated by

$$R_k(t) = \frac{Q_k^{max} - Q_k(t)}{E[I_k(t)]}. \tag{4}$$

However, the distribution of $I_k(t)$ is not available. Hence, we use the sample mean of queue variations in the last N slots to estimate $E[I_k(t)]$, i.e., $\frac{1}{N}\sum_{t_0=t-N-1}^{t-1}(A_k(t_0) - V_k(t_0))$. Then, the remaining life time of the kth user at the slot t can be calculated by

$$R_k(t) = \frac{Q_k^{max} - Q_k(t)}{\frac{1}{N}\sum_{t_0=t-N-1}^{t-1}(A_k(t_0) - V_k(t_0))}. \tag{5}$$

Problem statement

This paper aims for maximizing the throughput while reducing BLR and keeping certain fairness among users. In order to achieve this, we jointly consider the users' queue priority and RBs capacity for controlling the service rate of each data queue. The users' queue priority is based on the remaining life time or the queue overflow probability which is calculated by applying the large deviation principle. Then, according to the queue priority, we adjust the service rate of each user queue through dynamically allocating the RBs, in turn providing different transmission rate to achieve statistic QoS guarantee.

Channel quality indicator (CQI) reporting procedure is a fundamental feature of LTE networks since it enables the estimation of the downlink channel quality at the eNB [27]. UE reports a CQI value of each RB to the eNB, and the eNB uses CQI for the resource allocation [28]. Let $r_k^n(t)$ denote instantaneous data transmission rate when the nth SB is assigned to the kth user queue at the slot t. According to CQI information, $r_k^n(t)$ can be calculated using the AMC module or simply estimated via the well-known Shannon's formula for the channel capacity [27], i.e.,

$$r_k^n(t) = log_2\left(1 + \gamma_{k,n}\right). \tag{6}$$

where $\gamma_{k,n}$ is the signal-to-interference-plus-noise-ratio (SINR) for the kth user on the nth SB.

Let us define $x_k^n(t)$ to indicate whether the nth SB is assigned to the kth user at the slot t. If the nth SB is assigned to the kth user at the slot t, we have $x_k^n(t)$=1. Otherwise $x_k^n(t)$=0. Then, the resource allocation problem can be defined as improving the system throughput as large as possible, i.e.,

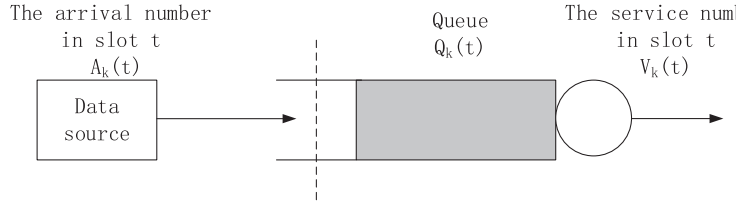


Fig. 2 Queueing system model

$$\max \sum_{k=1}^{K} \sum_{n=1}^{N} x_k^n(t) r_k^n(t), \tag{7}$$

In this paper, we apply *Jain's* fairness index [29], $F(t)$, to indicate the fairness to denote the system fairness at time t. The formula of *Jain's* fairness index are given as (33) and (34) in the section of Performance evaluation. The constraints are listed as follows:

$$\sum_{k=1}^{K} x_k^n(t) = 1, n \in \{1, 2, \ldots, N\}. \tag{8}$$

$$1 - F(t) \leq \xi. \tag{9}$$

where ξ is a given threshold value of the fairness deviation. Equation (8) indicates that each SB can be only assigned to one user during the slot t. Equation (9) indicates that the difference between 1 and the system fairness should be kept less than ξ.

The resource allocation problem (7–9) is complicated and intractable to obtain the optimal solution by exhaustive search. Here, we propose a buffer-aware adaptive resource allocation scheme which considers the different priorities of user queues and RB capacity, in order to achieve a better performance tradeoff of throughput, fairness and average BLR.

User priority determination scheme

In order to improve the total system throughput and QoS for different users while keeping certain fairness, we need to determine the users' queue priority for deriving an online measurement based resource allocation. The overflow probability estimation model is derived by applying large deviation principle, which requires no statistical knowledge of the network conditions. Then, we rank the users' queue priority by their remaining life time or their queue overflow probability.

Estimation model for the queue overflow probability

The arrival rate of the incoming bits depends on the service type, while the service rate depends on the resource allocation policy as well as the wireless channel conditions which are time-varying in nature. Hence, the arrival process and the service process are independent of each other. Our aim is to control all user queues in such a way that the service demands of the data in each queue could be satisfied. Moreover, the resulted scheme should be robust to the variations of the arrival and service processes.

Let $I_k(t) = A_k(t) - V_k(t)$, where $I_k(t) \in \{-m_V, \cdots, 0, 1, \cdots, m_A\}$, and let $\pi_i^k = P(I_k(t) = i)$ denote the corresponding kth user queue-length variation probability distribution. Since $A_k(t)$ is determined by the bit arrival number during the slot t and $V_k(t)$ is determined by the bit number served during the slot t, their difference $I_k(t)$ characterizes the mismatch between the bit service rate and the bit arrival rate of the kth user queue during the slot t. $I_k(t) < 0$ implies that the bit service rate is higher than the arrival rate in the tth slot, while $I_k(t) > 0$ implies that the bit service rate cannot satisfy the bit arrival. Due to the time-varying number of bit arrivals and the state of SBs, the polarity of the sequence $I_k(t)(t = 1, 2, \ldots)$ may change frequently between negative and positive.

The kth user queue length increment during the period spanning from the tth slot to the $(t + T)$th slot can be formulated as

$$I_k^{t+T} = \sum_{i=1}^{T} I_k(t + i), \tag{10}$$

where T is called prediction interval.

Then, the length of the kth user queue at the beginning of the $(t + T)$th slot can be expressed as

$$Q_k(t + T) = Q_k(t) + I_k^{t+T}. \tag{11}$$

Let $P_{k_{overflow}}^{t+T}$ denote the overflow probability of the kth user queue during the slot $(t + T)$, which is defined as

$$P_{k_{overflow}}^{t+T} = P\left(Q_k(t + T) > Q_k^{max}\right). \tag{12}$$

The above expression can be rewritten as

$$P_{k_{overflow}}^{t+T} = P\left(Q_k(t) + I_k^{t+T} > Q_k^{max}\right). \tag{13}$$

Define the achievable average queue growth of the kth user queue during the future T slots as

$$g_k = \frac{Q_k^{max} - Q_k(t)}{T}, \tag{14}$$

and the expected average queue growth of the kth user queue in each slot during the T slots as

$$c_k = E\left[\frac{\sum_{i=1}^{T} I_k(t + i)}{T}\right]. \tag{15}$$

where $E[\cdot]$ denotes expectation operator. $c_k > g_k$ implies that there is high overflow possibility of the kth user queue after T slots. Equation (12) can be further written as

$$\begin{aligned} P_{k_{overflow}}^{t+T} &= P\left(Q_k(t) + I_k^{t+T} > Q_k^{max}\right) \\ &= P\left(I_k^{t+T}/T > (Q_k^{max} - Q_k(t))/T\right) \\ &= p\left(\frac{\sum_{i=1}^{T} I_k(t + i)}{T} > g_k\right). \end{aligned} \tag{16}$$

The term $\frac{\sum_{i=1}^{T} I_k(t+i)}{T}$ in (16) is determined by the bits departure or the resource allocation, while g_k is determined by the current length of the kth user queue. Since the queue overflow probability indicates the mismatch between the resource and the traffic, we can dynamically rank the users' queue priority based on the value of $P_{k_{overflow}}^{t+T}$. The larger value of $P_{k_{overflow}}^{t+T}$ means that queue overflow is more likely to occur and the corresponding

user queue should have the higher priority of resource allocation, thus reducing the bit loss rate and satisfying QoS requirement. This is why the proposed method jointly considers RB capacity and the queue priority.

Cramér's Theorem in the context of large deviation principle can be applied to estimate the overflow probability in [30]. Since $A_k(t)$ is an i.i.d process, $I_k(t)(t = 1, 2, \ldots)$ are also i.i.d random variables with a finite moment generating function $G(\theta) = E\{e^{\theta I_k(t)}\}$. According to *Cramér's* Theorem [31], if $c_k < g_k$, the sequence $I_k(t)(t = 1, 2, \ldots)$ obeys the large deviation principle, and we have

$$\lim_{T\to\infty} \frac{1}{T} logP\left(\frac{\sum_{i=1}^{T} I_k(t+i)}{T} > g_k\right) = -l(g_k). \quad (17)$$

where

$$l(g_k) = sup_{\theta>0}\{g_k\theta - logG(\theta)\}. \quad (18)$$

and

$$logG(\theta) = log\left\{\sum_{i=-m_V}^{m_A} \pi_i^k e^{i\theta}\right\}. \quad (19)$$

Note that $logG(\theta)$ is a convex function, and the rate function $l(g_k)$ is also convex [31]. For a sufficiently large value of T, according to (17) the overflow probability can be approximated by

$$P_{k_{overflow}}^{t+T} \approx e^{-Tl(g_k)}. \quad (20)$$

In theory, the overflow probability estimate becomes more accurate as T increases. Hence, the value of T should be sufficiently large. However, owing to the rapid exponential decay of the overflow probability estimate with T, we can set T to a moderate value for the sake of acquiring an accurate overflow probability estimate. The experimental results in the section of 'Performance evaluation' demonstrate that $T \geq 60$ is appropriate.

In the next section, we show how to online estimate the overflow probability based on (20).

Online estimation of the queue overflow probability

According to (20), estimating the overflow probability requires the values of g_k, c_k, and π_i^k. It is easy to calculate g_k according to (14). However, we have to estimate c_k and π_i^k because there is no prior knowledge about $I_k(t)$. Therefore, the historical observations are utilized to estimate these parameters by applying a sliding window-based method.

Suppose the observed sequence is given by $\{I_1, I_2, I_3, \cdots\}$. The sliding window covers the T_s most recent entries in this sequence, which is slid over this sequence. For the nth window, the observation vector is denoted by $W_n = [I_n, I_{n-1}, I_{n-2}, \cdots I_{n-T_s+1}]$.

For the parameter c_k, we use the sample mean as its estimate, i.e.,

$$\hat{c}_k = \frac{\sum_{i=n-T_s+1}^{n} I_k(i)}{T_s}. \quad (21)$$

Following the similar steps in [32], we can apply the large deviation principle to analyze the confidence interval of c_k.

Below, we will estimate $\pi_i^k (i \in \{-m_V, \ldots, 0, 1, \ldots, m_A\})$. Let T_i^k denote the number of $I_k(t) = i$ events during the T_s slots, which can be calculated by

$$T_i^k = \sum_{t=n-T_s+1}^{n} 1_i(I_k(t)). \quad (22)$$

where $1_i(\cdot)$ is a indicator function. When $I_k(t) = i$, it has a value of 1, otherwise 0. Then, the frequency of $I_k(t) = i$ can be estimated as

$$\hat{u_i^k}(t) = \frac{T_i^k}{T_s}. \quad (23)$$

If the value of T_s is too small, it may result in a large estimate error of $\hat{u_i^k}(t)$, while too large, it may reduce the sensitivity to queue variations. Hence, T_s should be set to a moderate value. We set it to 60 in our experiments. We apply an exponential smoothing method to smoothen the estimated value, which is written as

$$\hat{\pi_i^k}(t) = \rho\hat{\pi_i^k}(t-1) + (1-\rho)\hat{u_i^k}(t). \quad (24)$$

where the parameter $\rho \in [0, 1]$. If ρ approaches to 1, the value of $\hat{\pi_i^k}(t)$ largely depends on the past estimation, while if ρ=0, $\hat{\pi_i^k}(t)$ totally depends on the current estimate $\hat{u_i^k}(t)$. According to Gardner's report [33], $\rho \in [0.7, 0.9]$ is usually recommended.

The above steps assist us to derive the online measurement-based method to estimate the queue overflow probability $P_{k_{overflow}}^{t+T}$ based on (20) in the $(t + T)$ slot, by setting T to a moderate value in a practical application. The experimental results show that $T \geq 60$ is appropriate.

User priority determination algorithm

In the case of $\hat{c}_k \geq g_k$, the average growth length of the kth user queue in each slot, $\hat{c}_k$, is higher than the achievable average growth length of the kth user queue per slot, g_k, in the forthcoming T slots. This implies that if keeping the current queue configuration unchanged with the bit service rate $V_k(t)$, after T slots, the queue will be more likely to have an overflow situation. Therefore, in this scenario, we should improve the bit service rate to prolong the remaining life time. In this paper, remaining life time $R_k(t)$ can be calculated by (5) to rank the queue priority of resource allocation.

However, in the case of $\hat{c}_k < g_k$, the average growth length of the kth user queue in each slot, $\hat{c}_k$, is lower than the achievable average growth length of the kth user queue, g_k, in the forthcoming T slots. But this does not necessarily imply that no overflow will happen in the future T slots, since $\hat{c}_k$ is the average growth per slot which cannot characterize the specific queue length growth in a single time slot. Hence, the queue overflow might still occur. Since we have $g_k > \hat{c}_k$, the queue overflow remains a rare event, and the queue overflow probability in the $(t + T)$ slot, $P^{t+T}_{k_{overflow}}$, can be approximated by (20).

The buffer-aware priority determination algorithm determines a priority value for each user, where the user in the case of $\hat{c}_k \geq g_k$ is more emergent than in the case of $\hat{c}_k < g_k$. The smallest value of $R_k(t)$ indicates the highest priority of the kth user. The value in ascending order represents that the users' priority is from high to low. The smaller value is $P^{t+T}_{k_{overflow}}$, the lower priority is the kth user. The value in descending order indicates that the users' priority is from high to low. According to the user queues' different priorities, in the next section, we show how to dynamically allocate the RBs to adjust the service rate for each user queue for improving the system throughput subject to providing QoS guarantee while keeping a certain fairness.

Online measurement-based algorithm for dynamic RBs allocation

In this section, we will present the proposed online estimation based dynamic service rate control algorithm, which relies on a strategy of mitigating the overflow probability or extending the remaining life time.

Suppose there are K user queues indexed by the set $\Phi = \{1, 2, \ldots K\}$ and N SBs indexed by the set $\Omega = \{1, 2, \ldots N\}$. For any $k \in \Phi$, we calculate the value of $R_k(t)$ and $P^{t+T}_{k_{overflow}}$ in the case of $\hat{c}_k \geq g_k$ and $\hat{c}_k < g_k$, respectively. Then, the resource allocation strategy operates as follows:

1. In the proposed buffer-aware resource allocation scheme, at the slot t we seek the user
$$k_1 = \arg\min_{k\in\Phi}\{R_k(t)\}. \tag{25}$$
2. The SB having the maximum SINR can be obtained by
$$n_1 = \arg\max_{n\in\Omega}\{\gamma_{n,k_1}\}. \tag{26}$$
Then, we can calculate the transmission rate $r^{n_1}_{k_1}(t)$ according to (6).
3. If $A_{k_1}(t) > r^{n_1}_{k_1}(t)$, it means that allocating SB is not enough to transmit the bits in the buffer for the most emergent user queue. Let $\Omega = \Omega\backslash\{n_1\}$ (which means removing the element n_1 from the set Ω), then we choose the SB $n_2 = \arg\max_{n\in\Omega}\{\gamma_{n,k_1}\}$ and calculate the transmission rate $r^{n_2}_{k_1}(t)$. Compare the value of $A_{k_1}(t)$ with the value of $r^{n_1}_{k_1}(t) + r^{n_2}_{k_1}(t)$. If $A_{k_1}(t) \leq r^{n_1}_{k_1}(t) + r^{n_2}_{k_1}(t)$, execute the step 4. Otherwise, let $\Omega = \Omega\backslash\{n_2\}$. Choose the SB $n_3 = \arg\max_{n\in\Omega}\{\gamma_{n,k_1}\}$ and calculate the transmission rate $r^{n_3}_{k_1}(t)$. Compare the value of $A_{k_1}(t)$ with the value of $r^{n_1}_{k_1}(t) + r^{n_2}_{k_1}(t) + r^{n_3}_{k_1}(t)$, and repeat the above procedure until $A_{k_1}(t) \leq r^{n_1}_{k_1}(t) + r^{n_2}_{k_1}(t) + \cdots + r^{n_m}_{k_1}(t)$.
4. If $r^{n_1}_{k_1}(t) + r^{n_2}_{k_1}(t) + \cdots + r^{n_m}_{k_1}(t) \leq Q_{k_1}(t) + A_{k_1}(t)$, the bit number transmitted in the slot t can be calculated as
$$V_{k_1}(t) = r^{n_1}_{k_1}(t) + r^{n_2}_{k_1}(t) + \cdots + r^{n_m}_{k_1}(t). \tag{27}$$
Otherwise $r^{n_1}_{k_1}(t) + r^{n_2}_{k_1}(t) + \cdots + r^{n_m}_{k_1}(t) > Q_{k_1}(t) + A_{k_1}(t)$, the bit number transmitted in the slot t can be calculated as
$$V_{k_1}(t) = Q_{k_1}(t) + A_{k_1}(t). \tag{28}$$
5. Then, let $\Phi = \Phi\backslash\{k_1\}$, $\Omega = \Omega\backslash\{n_m\}$, we seek the user $k_2 = \arg\min_{k\in\Phi}\{R_k(t)\}$, by repeating the procedures 2, 3, and 4, and allocate several SBs for transmitting the data of the k_2th user. Repeat the procedure 5 until all the users which have the value of $R_k(t)$ have been allocated with the SBs.
6. After that, we further allocate SBs to the users who have the value of $P^{t+T}_{k_{overflow}}$. We choose the user
$$k_l = \arg\max_{k\in\Phi}\left\{P^{t+T}_{k_{overflow}}\right\}. \tag{29}$$
and repeat the similar procedures 2, 3, 4, and 5 to allocate the resource and schedule users until $\Omega = \emptyset$.
7. If $\Omega \neq \emptyset$ and $\Phi = \emptyset$, it means that there are RBs which have not be used. In order to make the best utilization of RBs, we choose the users who have $V_k(t) < Q_k(t) + A_k(t)$ and constitute a new user set $\bar{\Phi}$. We seek the user
$$k_w = \arg\max_{k\in\bar{\Phi}}\{Q_k(t) + A_k(t) - V_k(t)\}. \tag{30}$$
8. Then the remaining SB with the maximum SINR can be obtained via
$$n_w = \arg\max_{n\in\Omega}\{\gamma_{n,k_w}\}. \tag{31}$$
According to (6), we can calculate the transmission rate $r^{n_w}_{k_w}(t)$.
9. If $Q_{k_w}(t) + A_{k_w}(t) - V_{k_w}(t) > r^{n_w}_{k_w}(t)$, it means that the number of allocated SB is not enough to transmit

the remaining bits in the buffer. Let $\Omega = \Omega \backslash \{n_w\}$, we choose the SB $n_{w_1} = \arg\max_{n\in\Omega}\{\gamma_{n,k_w}\}$ and calculate the transmission rate $r_{k_w}^{n_{w_1}}(t)$. Compare the value of $Q_{k_w}(t) + A_{k_w}(t) - V_{k_w}(t)$ with the value of $r_{k_w}^{n_w}(t) + r_{k_w}^{n_{w_1}}(t)$, if $Q_{k_w}(t) + A_{k_w}(t) - V_{k_w}(t) \leq r_{k_w}^{n_w}(t) + r_{k_w}^{n_{w_1}}(t)$, we allocate the SBs n_w, n_{w_1} to the k_wth user, if not, let $\Omega = \Omega \backslash \{n_2\}$, choose the SB $n_{w_3} = \arg\max_{n\in\Omega}\{\gamma_{n,k_w}\}$ and calculate the transmission rate $r_{k_w}^{n_{w_3}}(t)$. Repeat the above procedure, choose the SBs by using the same method until $Q_{k_w}(t) + A_{k_w}(t) - V_{k_w}(t) \leq r_{k_w}^{n_w}(t) + r_{k_w}^{n_{w_1}}(t) + \cdots + r_{k_w}^{n_{w_w}}(t)$. Accordingly, we allocate the SBs $n_w, n_{w_1}, \cdots, n_{w_w}$ to the k_wth user.

10. Let $\bar{\Phi} = \bar{\Phi} \backslash \{k_w\}$, $\Omega = \Omega \backslash \{n_w\}$, and seek the user $k_{w_1} = \arg\max_{k\in\bar{\Phi}}\{Q_k(t) + A_k(t) - V_k(t)\}$. Repeat the procedures 8, 9, and 10 until $\Omega = \emptyset$ or $\bar{\Phi} = \emptyset$.
11. After allocating SBs to the users in the slot t, we update the values of $\hat{c}_k$ and g_k corresponding to all the users in the slot $t+1$. Apply the user priority determination algorithm in the section III to rank the users' queue priority again, and then repeat the above all procedure.

The workflow of the proposed method is illustrated in Fig. 3. The proposed method will use the observations of buffer fullness, data arrival rate $A_k(t)$, and CQI feedback from user equipments (UEs) to calculate the users' queue priority based on their remaining life time $R_k(t)$ or their queue overflow probability $P_{k_{overflow}}^{t+T}$. Once the users' queue priority has been determined, the dynamic RBs allocation algorithm based on online measurement is applied to adjust the service rate $V_k(t)$ by making decisions $x_k^n(t)$. Then, the decisions $x_k^n(t)$ are forwarded to eNB Scheduler to execute the resource scheduling.

The algorithm operates at every beginning of the scheduling interval. The detail of the strategy is presented in Algorithm 1.

Algorithm 1 Buffer-aware adaptive resource allocation algorithm

Transmitted bit number of each user queue in slot t is denoted by $V_k(t)$, for $k = 1, \cdots, K$
for $k = 1; k \leq K; k++$ **do**
  if $Q_k(t) \geq Q_{max}$ **then**
    Calculate $\bar{C}_k$
  else
    Calculate $\hat{c}_k, g_k$
    if $\hat{c}_k \geq g_k$ **then**
      Calculate $R_k(t)$
    else
      Calculate $P_{k_{overflow}}^{t+T}$
    end if
  end if
end for
while $\Phi \neq \emptyset$ or $\Omega \neq \emptyset$ **do**
  if the value of $R_k(t)$ exists **then**
    Choose the user $k_1 = \arg\min_{k\in\Phi}\{R_k(t)\}$
  else
    Choose the user $k_1 = \arg\max_{k\in\Phi}\{P_{k_{overflow}}^{t+T}\}$
  end if
  Seek the SB $n_1 = \arg\max_{n\in\Omega}\{\gamma_{n,k_1}\}$ for the k_1th user
  Calculate $r_{k_1}^{n_1}(t)$
  $Sum_{k_1}(t) = 0$
  while $Sum_{k_1}(t) \leq A_{k_1}(t)$ **do**
    $Sum_{k_1}(t) = Sum_{k_1}(t) + r_{k_1}^{n_1}(t)$
    $\Omega = \Omega \backslash \{n_1\}$
    Seek the SB $n_1 = \arg\max_{n\in\Omega}\{\gamma_{n,k_1}\}$
  end while
  if $Sum_{k_1}(t) \leq Q_{k_1}(t) + A_{k_1}(t)$ **then**
    $V_{k_1}(t) = Sum_{k_1}(t)$
  else
    $V_{k_1}(t) = Q_{k_1}(t) + A_{k_1}(t)$
  end if
  $\Phi = \Phi \backslash \{k_1\}$
end while

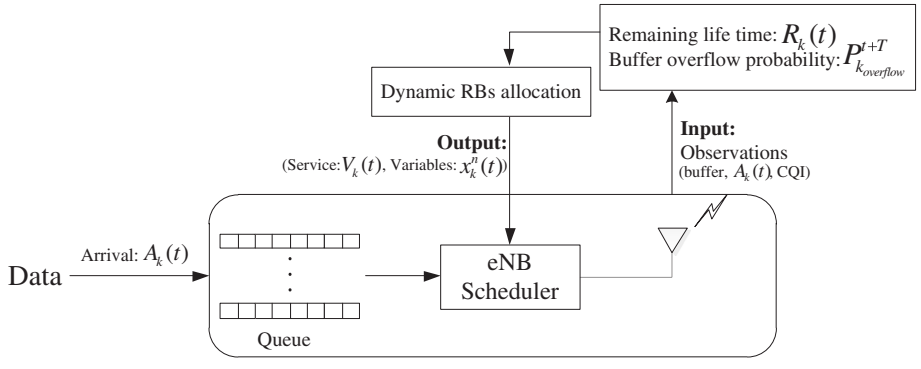


Fig. 3 A buffer-aware adaptive resource allocation algorithm for LTE downlink transmission

Performance evaluation

In this section, we characterize the performance of our online measurement-based adaptive resource allocation algorithm, and provide performance comparisons with other five algorithms, namely CABA algorithm [23], PF metric(1) algorithm, PF metric(2) algorithm [19], MaxWeight-Alg(3) [17], and IHRR algorithm [18]. We first describe the simulation setup, and then the metrics used for performance evaluation are presented.

Experiment setup

We simulated a multiuser scenario, where the maximum number of communicating users was set to $K = 10, 30, 50$. Here, the bit arrival rate for each user is assumed to obey the Poisson distribution with $\lambda = 50\ kbit/ms$.

CQI is discretized into 15 levels which correspond to 15 different pairs of modulation choice and code rate. This implies that there are 15 possible transmission rates. A mapping between SINR ranges and CQIs is presented in [34]. The CQI values are used together with the number of allocated RBs to determine the transmission rates.

Performance metrics

To evaluate the performance of the proposed dynamic resource allocation, we define three metrics as follows:

- Average bit loss rate: This metric indicates QoS of K users. It is defined as time average bit loss rate during a period of Δ, i.e.,

$$\bar{C}_k = \frac{1}{\Delta + 1} \sum_{t=T_0}^{T_0+\Delta} \frac{D_k(t)}{A_k(t)}. \tag{32}$$

  where $D_k(t)$ denotes the number of bit loss during the slot t for the kth user. Obviously, smaller $\bar{C}$ is preferred.
- Fairness: This metric is measured using *Jain's* fairness index [29], which is widely applied for evaluating the system fairness. It is described as follows

$$F(t) = \frac{\left(\sum_{k=1}^{K} V_k(t)\right)^2}{K \sum_{k=1}^{K} V_k^2(t)}, \tag{33}$$

  where $F(t)$ denotes the fairness at time t. Then, the system fairness can be calculated according to

$$F = \frac{1}{\Delta + 1} \sum_{t=T_0}^{T_0+\Delta} F(t). \tag{34}$$

- Average throughput: Our aim is improving the system throughput subject to providing QoS guarantee for different users. The larger average system throughput implies better performance.

All the simulation results were averaged over 50 independent runs.

Experimental results

Performance comparison for different user index

We used Matlab for implementing the simulations. The simulation model is based on the 3GPP LTE system model and it has a single cell with downlink transmission, where the number of RBs is 50, the carrier frequency is 2 GHz, and the system bandwidth is 10 MHz. Following the similar steps in [32], we can applying the large deviation principle to analyze the confidence interval of $\hat{c}_k$.

The corresponding simulation parameters are listed in Table 1. The prediction interval $T = 60$, the sliding window length $T_s = 60$, the forgetting factor $\rho = 0.7$, and the average channel SINR of 15 dB. In order to simplify the calculation, we set $Q_k^{max} = Q^{max} = 3 \times 10^4\ bit$.

In Fig. 4, we plotted the average BLR of ten users for all the resource allocation schemes. The X axis denotes the user index. It can be seen that the proposed algorithm achieves the best performance with average BLR of 2.12×10^{-3}, which is lower than those of PF metric(1) (about 2.57×10^{-3}), CABA (about 2.29×10^{-3}), IHRR (about 2.14×10^{-3}), and PF metric(2) (about 2.13×10^{-3}). MaxWeight-Alg(3) may perform unfair resource sharing among users. Hence, the curve of the average BLR for MaxWeight-Alg(3) is unstable compared with other algorithms. While for the proposed algorithm, we calculate the priority for each user queue by the remaining life time or queue overflow probability, which is applied to allocate RBs. As a result, it helps to reduce the overflow probability of the queue with highest priority. Thus, it achieves a lower value of the average BLR.

In Fig. 5, we show the average throughput corresponding to ten users for different resource allocation algorithms. It can been seen that the average throughput for each user in the proposed algorithm significantly outperforms CABA, PF metric(1), PF metric(2), and IHRR. The reason for this is that adaptive resource allocation with the queue priority considers both the buffer status and the RBs capacity, thus improving all users' transmission rate and keeping a high fairness among all users. By contrast,

Table 1 Simulation parameters

Parameter	Setting
Carrier frequency	2 GHz
System bandwidth	10 MHz
Transmission time interval	1 ms
Subcarriers per resource block	12
Resource block bandwidth	180 KHz
Number of resource blocks	50
Type of system	Single cell
Channel model	Urban
Simulation time	1000 TTIs

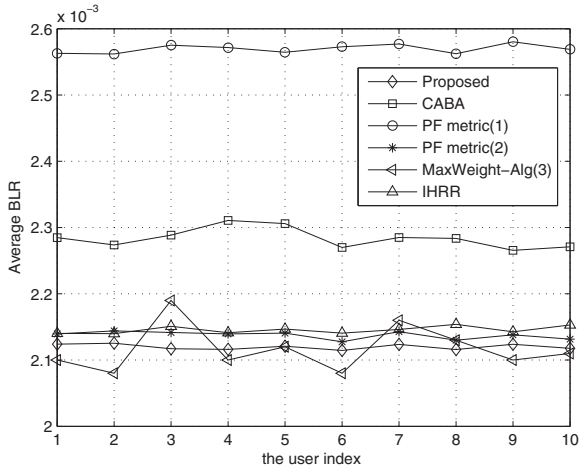


Fig. 4 The average BLR for different user

PF metric(1) does not consider the queue length at all, which lead to the lowest performance. PF metric(2) suffers from the isolated RB assignment strategy, and thus it fails to improve the system throughput. For CABA, the weighted factor in the priority function may influence the performance. Considering that the user priority determination in IHRR has some limitation for resource allocation. MaxWeight-Alg(3) performs better than the other algorithms, but it does not consider the fairness.

Performance at different SINRs

This section investigates the performance of the proposed algorithm and other compared algorithms under different channel SINR conditions. In the simulation, the average channel SINR recorded varies from 11 to 20 *dB* with a step-size of 1 *dB*. The other parameters and simulation settings were the same as those in Section 'Performance comparison for different user index'. The average BLR for all users is calculated by

$$\bar{C} = \frac{1}{K}\sum_{k=1}^{k=K} \bar{C}_k. \tag{35}$$

The average BLR versus the average channel SINR for these resource allocation schemes with ten users were plotted in Fig. 6. As shown in Fig. 6, the average BLR of the proposed method decreases as the value of SINR increases. The reason is that, at low SINR region, the bit service rate is not sufficient, and the current queue may have a shorter remaining life time or a larger overflow probability, which induces a large number of bits lost. As the average SINR increases, the bit service rate is increasing. Thus, the remaining life time is prolonged as well as the overflow probability decreases, which may reduce the average BLR. Compared with other algorithms, the proposed algorithm achieves the lowest BLR. The reason is that other algorithms fail to consider the priorities of user queues based on the buffer status and the RBs capacity.

Figure 7 shows the fairness of the proposed algorithm, CABA, PF metric(1), PF metric(2), MaxWeight-Alg(3), and IHRR. We applied *Jain's* fairness index in the simulation. It is shown that the fairness index of the proposed algorithm is the highest among these algorithms, which is approximately 0.998. This indicates that the queue priority assists the proposed algorithm to balance the resource allocation among the users, thereby achieving certain throughput fairness. From Fig. 7, we can see that the proposed algorithm may be insensitive to the value of the average SINR, but other algorithms undergo a relatively large variation for the different average SINRs. This also implies that their performance is subject to the channel quality.

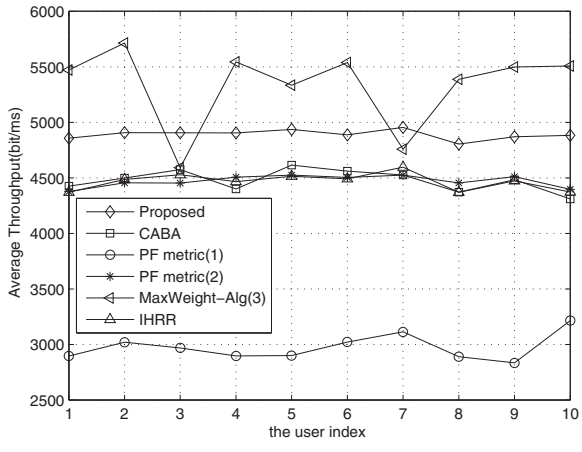


Fig. 5 The average throughput for different user

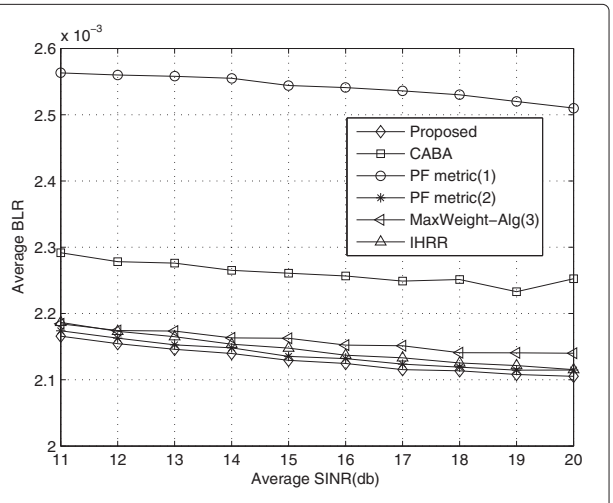


Fig. 6 The average BLR for different average channel SINR

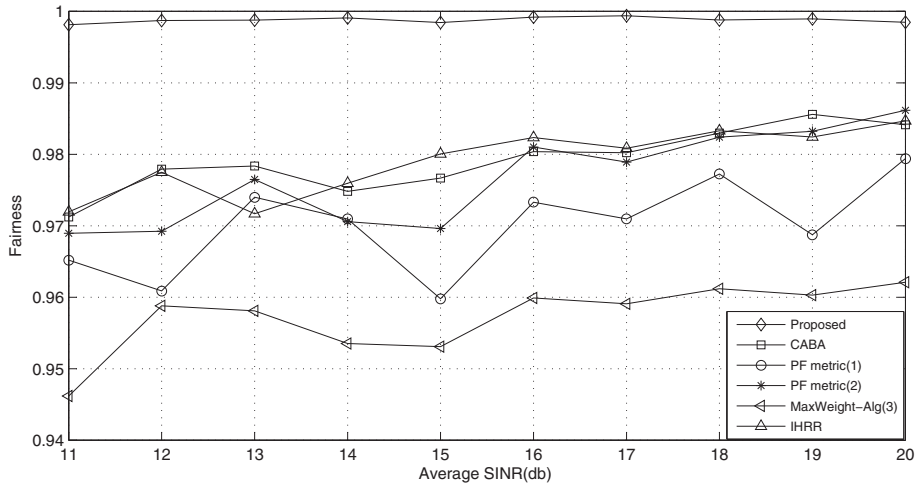


Fig. 7 The fairness for different average channel SINR

Figure 8 shows the average system throughput for different average channel SINR for the different resource allocation schemes with ten users. As shown in Fig. 8, the average system throughput increases upon increasing the average SINR. The results demonstrate that MaxWeight-Alg(3) performs better than the other strategies in terms of the overall throughput, but it has a lowest fairness level as shown in Fig. 7. For the rest algorithms, the proposed algorithm performs the better. The reason is that by choosing the appropriate RBs for the user queues according to their priority, the system throughput is improved. Combining the results of Figs. 7 and 8, it can be concluded that compared to the other methods, the proposed method both improves the fairness and the system throughput. This essentially benefits from the technique of queue priority applied in the proposed method.

Performance for different number of users

This section investigates the performance of the proposed algorithm and other benchmark algorithms for different number of users. In the simulation, the number of users K were chosen in the range [10, 50]. The other parameters and simulation settings were the same as those in Section 'Performance comparison for different user index'.

Figure 9 shows that the average BLR decreases as the number of users increases for the different resource allocation schemes. This reason is that the same amount of resources has to be shared among a higher number of

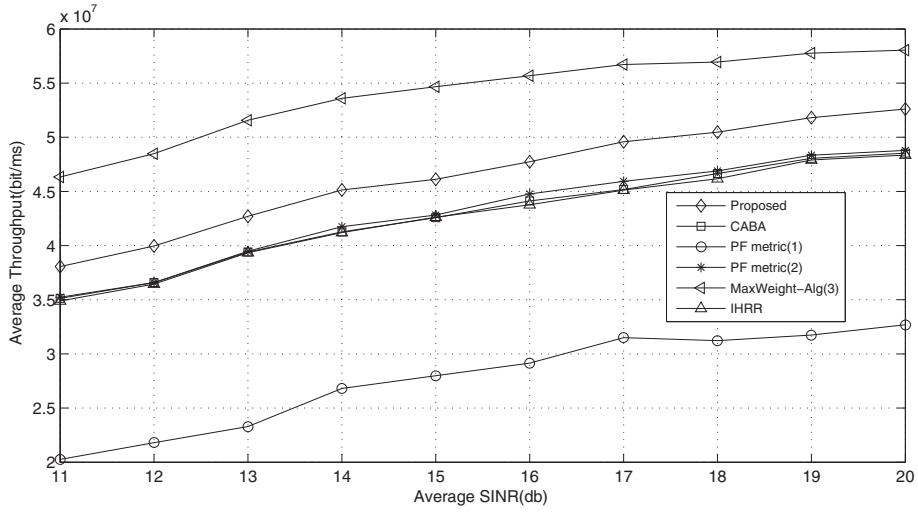
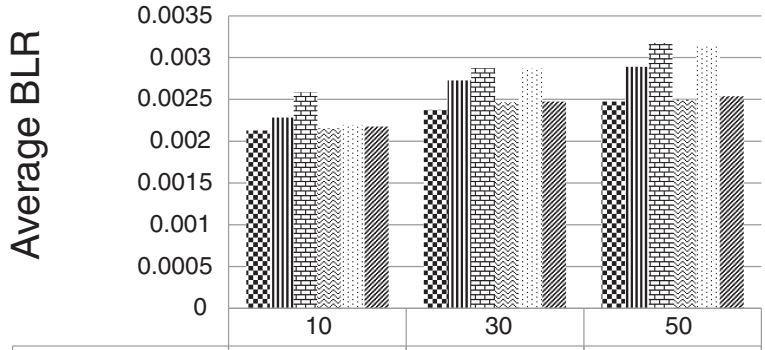


Fig. 8 The average throughput for different average channel SINR



	10	30	50
proposed	0.002125	0.002375	0.002475
CABA	0.002282	0.002727	0.002889
PF metric(1)	0.002586	0.002875	0.003175
PF metric(2)	0.002153	0.002465	0.002496
MaxWeight-Alg(3)	0.002193	0.002873	0.003132
IHRR	0.002175	0.002474	0.002537

Fig. 9 The average BLR for different number of users

candidates, which implies that with the increasing number of users, there is a much higher probability of bit loss. From Fig. 9, it can be observed that the average BLR of the proposed algorithm is the lowest among these algorithms, which maintains a small and steady growth trend with the increasing number of users in the cell. Since PF metric(1) does not consider buffer fullness, and PF metric(2), MaxWeight-Alg(3), IHRR as well as CABA fail to characterize the priorities of user data queues based on the buffer status and the RBs capacity, they have a higher average BLR than the proposed algorithm.

Figure 10 shows that the fairness index for the different resource allocation schemes decreases as the number of users increases. The fairness index of the proposed algorithm is the highest among these algorithms, which means that it provides high fairness regardless of the user in the cell. The reason for this is that the queue priority based on the buffer status and the RBs capacity assists the proposed algorithm to balance the resource allocation among the users. The algorithm having the worst fairness index is MaxWeight-Alg(3); the reason is that it aims to maximize the overall system throughput, rather than the throughput of a single user.

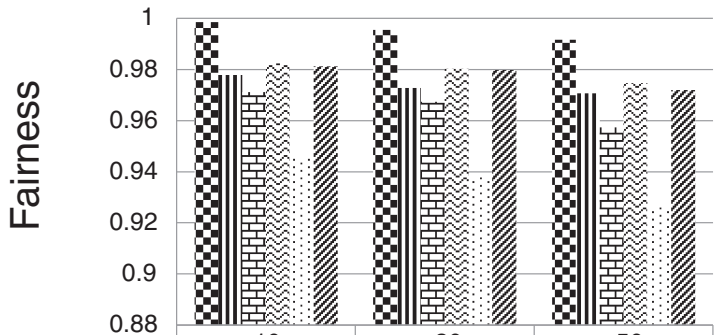
Figure 11 shows that the average user throughput for all strategies decreases as the number of users increases. This result is natural because a higher number of candidates are sharing the same amount of resources. MaxWeight-Alg(3) results in the highest throughput, followed by the proposed, PF metric(2), CABA, IHRR, and PF metric(1). From Figs. 9, 10, and 11, it can be concluded that compared to the other methods, the proposed method both improves the fairness and the average user throughput, at the same time reduces the average BLR.

Effect of prediction interval (T)

In this section, we carried out an experiment in order to investigate the effect of the prediction interval by setting $T = 20, 40, 60, 80, 100$. The other parameters and the simulation settings were the same as those in Section 'Performance comparison for different user index'. Figures 12 and 13 show the simulation results for the different prediction intervals. Observe in Fig. 12 that as the prediction interval duration increases, the average BLR decays rapidly. Although the prediction interval, T, should be sufficiently large according to the large deviation approximation in (20), the simulation results show that a choice $T \geq 60$ allows the proposed algorithm to achieve a reduced bit loss rate. Figure 13 shows that the average throughput increases upon increasing the interval T. This is because as T increases, the queue overflow probability estimate becomes more accurate, which results in more accurate resource allocation for achieving a higher average throughput.

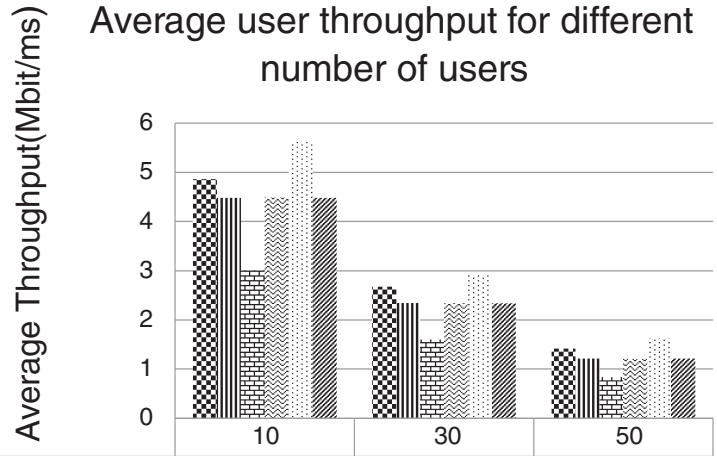
Effect of buffer size (Q^{max})

In this section, we carried out an experiment in order to analyze the different performance obtained by changing the buffer size Q^{max} from (0.5×10^4) bit to (5×10^4)



	10	30	50
proposed	0.9985	0.9954	0.9916
CABA	0.9778	0.9727	0.9706
PF metric(1)	0.9711	0.9673	0.9573
PF metric(2)	0.9823	0.9802	0.9746
MaxWeight-Alg(3)	0.9457	0.9382	0.9265
Max-Min Fair	0.9811	0.9798	0.9719

Fig. 10 The fairness for different number of users



	10	30	50
proposed	4.8599	2.6723	1.4183
CABA	4.4786	2.3458	1.2098
PF metric(1)	3.0083	1.5988	0.8374
PF metric(2)	4.4832	2.3377	1.2085
MaxWeight-Alg(3)	5.6312	2.9293	1.6244
IHRR	4.4761	2.3399	1.2096

Fig. 11 The average throughput for different number of users

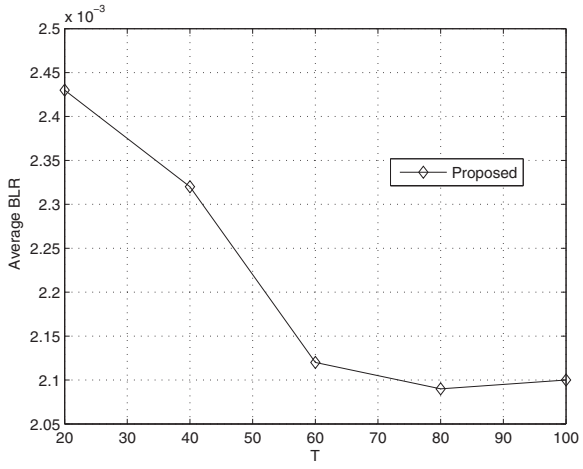


Fig. 12 The average BLR for different prediction intervals (T)

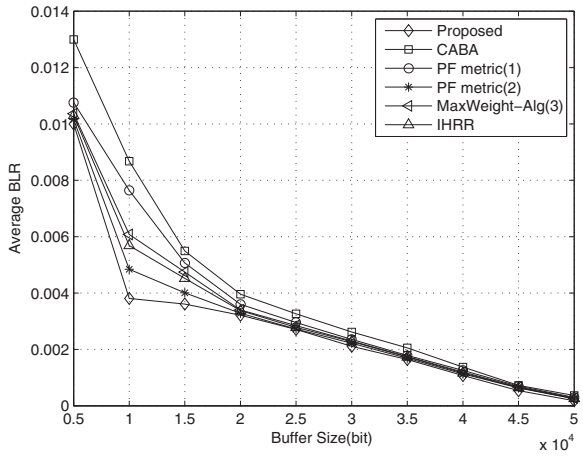


Fig. 14 The average BLR for different buffer size (Q^{max})

bit with a step-size of (0.5×10^4) bit. The other parameters and the simulation settings were the same as those in Section 'Performance comparison for different user index'. The simulation results were plotted in Figs. 14 and 15.

From Fig. 14, we can see that the average BLR decreased rapidly as the buffer size increased from (0.5×10^4) bit to (2×10^4) bit. This mean that too small buffer size is more likely to incur queue overflow and bit loss. As Q^{max} continues to increase, the average BLR reduces slowly. The reason is that larger capacity of the buffer has a lower probability of buffer overflow. Figure 14 shows that the proposed algorithm outperform the other methods in terms of average BLR. This benefits from the application of the user queue's priority calculated by the remaining life time or queue overflow probability.

We also observe from Fig. 15 that the average system throughput for all strategies is improved by increasing the buffer size. The proposed algorithm performs better than other algorithms except for MaxWeight-Alg(3). The reason is that increasing the buffer size decreases the queue overflow probabilities. The proposed algorithm chooses the appropriate RBs for the user queues according to their remaining life time or queue overflow probability, and the system throughput is improved. From Figs. 14 and 15, we can concluded that compared with other algorithms, the proposed method reduces the average BLR and improves the average system throughput as increasing the buffer size.

Conclusion

In this paper, we jointly consider user queue priority and the RBs capacity to develop a buffer-aware adaptive

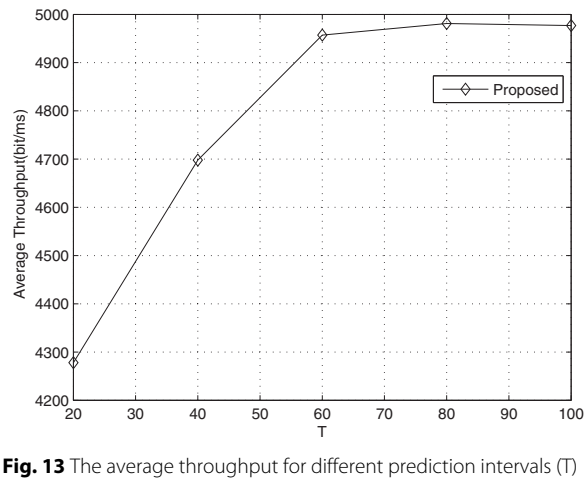


Fig. 13 The average throughput for different prediction intervals (T)

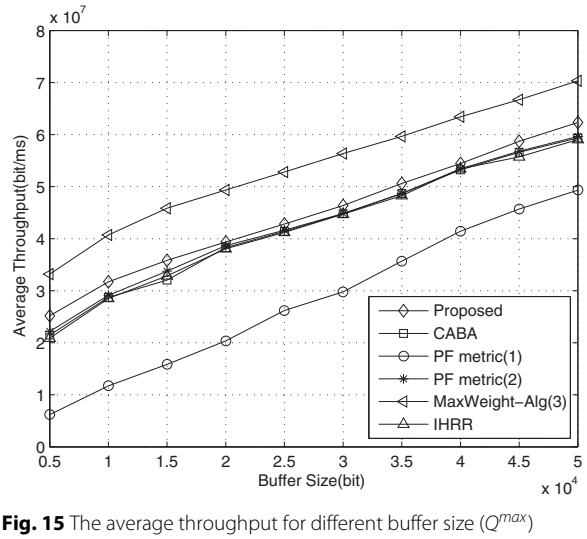


Fig. 15 The average throughput for different buffer size (Q^{max})

resource allocation scheme in LTE transmission systems. Under the constraint of finite buffer space, the proposed scheme aims for improving both the overall system throughput and the statistic QoS while keeping certain fairness among users. We derived an analytical formula based on the large deviation principle invoked for estimating the overflow probability as a function of the buffer variance. Also, the remaining life time of a queue was defined, and its estimation model was presented. Both the queue overflow probability and remaining life time were applied to determine the queue priority. According to the queue priority, an online measurement based algorithm was proposed to schedule RBs for adjusting the service rate of the user queues. The proposed algorithm does not rely on any prior knowledge about network conditions. Numerical results show that compared to traditional scheduling schemes, the proposed algorithm has a better tradeoff among throughput, fairness, and QoS. It improves the average system throughput and keeps a better fairness among users, while reducing the average BLR. It should be pointed out that this paper considered all the traffic at the eNodeB. However, the emerging technology of SDN and middle deep packet inspection (DPI) boxes can be applied to identify the traffic. Hence, we will consider the application aware scheduling in our future work with the aid of SDN and DPI.

Competing interests
The authors declare that they have no competing interests.

Received: 9 October 2014 Accepted: 26 May 2015
Published online: 20 June 2015

References

1. A Toskala, H Holma, K Pajukoski, E Tiirola, in *Proceedings of IEEE 17th International Symposium on Personal, Indoor and Mobile Radio Communications*. UTRAN long term evolution in 3gpp (Helsinki, 2006), pp. 1–5
2. WY Yeo, SH Moon, JH Kim, Uplink scheduling and adjacent-channel coupling loss analysis for TD-LTE deployment. Sci. World J. **2014**, 1–15 (2014)
3. P Phunchongharn, E Hossain, DI Kim, Resource allocation for device-to-device communications underlaying LTE-advanced networks. IEEE Wirel. Commun. **20**(4), 91–100 (2013)
4. Y Peng, SM Armour, JP McGeehan, An investigation of dynamic subcarrier allocation in mimo–ofdma systems. IEEE Trans. Veh. Technol. **56**(5), 2990–3005 (2007)
5. M Katoozian, K Navaie, H Yanikomeroglu, Utility-based adaptive radio resource allocation in ofdm wireless networks with traffic prioritization. IEEE Trans. Wireless Commun. **8**, 66–71 (2009)
6. J Huang, Z Niu, in *Proceedings of IEEE Wireless Communication and Networking Conferenc (WCNC)*. Buffer-aware and traffic-dependent packet scheduling in wireless ofdm networks (Hong Kong, 2007), pp. 1554–1558
7. IC Wong, O Oteri, W McCoy, Optimal resource allocation in uplink SC-FDMA systems. IEEE Trans. Wirel. Commun. **8**(5), 2161–2165 (2009)
8. HAM Ramli, R Basukala, K Sandrasegaran, R Patachaianand, in *Proceedings of IEEE Malaysia International Conference on Communications (MICC)*. Performance of well known packet scheduling algorithms in the downlink 3GPP LTE system (Kuala Lumpur, 2009), pp. 815–820
9. I Bisio, M Marchese, The concept of fairness: definitions and use in bandwidth allocation applied to satellite environment. IEEE Aerosp. Electron. Syst. Mag. **29**(3), 8–14 (2014)
10. Z Zhang, Y He, EK Chong, in *Proceedings of IEEE Wireless Communication and Networking Conferenc(WCNC)*. Opportunistic downlink scheduling for multiuser OFDM systems, vol. 2 (New Orleans, 2005), pp. 1206–1212
11. Z Diao, D Shen, VO Li, in *Proceedings of IEEE Global Communicatios Conference(GLOBECOM)*. CPLD-PGPS scheduling algorithm in wireless OFDM systems, vol. 6 (Dallas, 2004), pp. 3732–3736
12. G Song, Y Li, Utility-based resource allocation and scheduling in OFDM-based wireless broadband networks. IEEE Commun. Mag. **43**(12), 127–134 (2005)
13. S Ryu, B Ryu, H Seo, M Shin, in *Proceedings of IEEE International Conference on Communications (ICC)*. Urgency and efficiency based packet scheduling algorithm for OFDMA wireless system, vol. 4 (Seoul, 2005), pp. 2779–2785
14. A Ahmedin, K Pandit, D Ghosal, A Ghosh, in *Proceedings of the Conference on Emerging Networking EXperiments and Technologies (CoNEXT) student workshop*. Content and buffer aware scheduling for video delivery over lte (Santa Barbara, 2013), pp. 43–46
15. T-Y Huang, R Johari, N McKeown, M Trunnell, M Watson, in *Proceedings of the ACM Conference on SIGCOMM*. A buffer-based approach to rate adaptation: evidence from a large video streaming service (Chicago, 2014), pp. 187–198
16. A Dua, N Bambos, in *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM)*. Buffer management for wireless media streaming (Washington, 2007), pp. 5226–5230
17. M Andrews, L Zhang, Scheduling algorithms for multicarrier wireless data systems. IEEE Trans. Netw. **19**(2), 447–455 (2011)
18. M Realp, R Knopp, AI Perez-Neira, in *Proceedings of IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. Resource allocation in wideband wireless systems, vol. 2 (Berlin, 2005), pp. 852–856
19. S Lee, Swap-based frequency-domain packet scheduling algorithm for small-queue condition in OFDMA. IEEE Commun. Lett. **17**, 1028–1031 (2013)
20. IF Chao, CS Chiou, in *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC)*. An enhanced proportional fair scheduling algorithm to maximize QoS traffic in downlink OFDMA systems (Shanghai, 2013), pp. 239–243
21. B Yang, K Niu, Z He, W Xu, Y Huang, in *Proceedings of IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. Improved proportional fair scheduling algorithm in LTE uplink with single-user MIMO transmission (London, 2013), pp. 1789–1793
22. MR Sabagh, M Dianati, MA Imran, R Tafazolli, in *Proceedings of IEEE International Conference on Communications (ICC)*. A heuristic energy efficient scheduling scheme for VoIP in 3GPP LTE networks (Budapest, 2013), pp. 413–418
23. L Yan, GY Yue, in *Proceedings of IEEE International Conference on Wireless Communications, Networking and Mobile Computing (WICOM)*. Channel-adapted and buffer-aware packet scheduling in LTE wireless communication system (Dalian, 2008), pp. 1–4
24. JA Bucklew, Large Deviation Techniques in Decision, Simulation, and Estimation (2007)
25. KI Pedersen, TE Kolding, F Frederiksen, IZ Kovács, D Laselva, PE Mogensen, An overview of downlink radio resource management for UTRAN long-term evolution. IEEE Commun. Mag. **47**(7), 86–93 (2009)
26. 3GPP, *Further advancements for E-UTRA physical layer aspects (release 9)*. 3GPP TR 36.814, (2010)
27. F Capozzi, G Piro, LA Grieco, G Boggia, P Camarda, Downlink packet scheduling in LTE cellular networks: Key design issues and a survey. IEEE Commun. Surv. Tutorials. **15**(2), 678–700 (2013)
28. Y Timner, J Pettersson, H Hannu, M Wang, I Johansson, in *Proceedings of the 2014 ACM SIGCOMM Workshop on Capacity Sharing Workshop*. Network assisted rate adaptation for conversational video over LTE, concept and performance evaluation (Chicago, 2014), pp. 45–50
29. A Bin Sediq, RH Gohary, H Yanikomeroglu, in *Proceedings of IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. Optimal tradeoff between efficiency and jain's fairness index in resource allocation (Sydney, 2012), pp. 577–583
30. JS Chase, DC Anderson, PN Thakar, AM Vahdat, RP Doyle, in *ACM SIGOPS Operating Systems Review*. Managing energy and server resources in hosting centers, vol. 35, (2001), pp. 678–700

31. M Mandjes, *Large Deviations for Gaussian Queues: Modelling Communication Networks*. (John Wiley & Sons, West Sussex, 2007)
32. C Budianu, L Tong, in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal (ICASSP)*.
Good-turing estimation of the number of operating sensors: a large deviations analysis, vol. 2 (Quebec, 2004), pp. ii–1029
33. ES Gardner, Exponential smoothing: the state of the art. J. forecasting. **4**(1), 1–28 (1985)
34. 3GPP, *Physical layer procedures (release 9). 3GPP TS 36.214, 6.2.0*, (2010)