Resource Allocation Achieving High System Throughput with QoS Support in OFDMA-Based System

Tsern-Huei Lee, *Senior Member, IEEE*, and Yu-Wen Huang, *Student Member, IEEE*

***Abstract*—In this paper, we present a resource allocation algorithm for OFDMA-based systems which handles both real-time and non-real-time traffic. For real-time traffic, the QoS requirements are specified with delay bound and loss probability. The resource allocation problem is formulated as one which maximizes system throughput subject to the constraint that the bandwidth allocated to a flow is no less than its minimum requested bandwidth, a value computed based on loss probability requirement and running loss probability. A user-level proportional-loss scheduler is adopted to determine the resource share for flows attached to the same subscriber station (SS). In case the available resource is not sufficient to provide every flow its minimum requested bandwidth, we maximize the amount of real-time traffic transmitted subject to the constraint that the bandwidth allocated to an SS is no greater than the sum of minimum requested bandwidths of all flows attached to it. Moreover, a pre-processor is added to maximize the number of real-time flows attached to each SS that meet their QoS requirements. We show that, in any frame, the proposed proportional-loss scheduler guarantees QoS if there is any scheduler which guarantees QoS. Simulation results reveal that our proposed algorithm performs better than previous works.**

***Index Terms*—OFDMA, QoS, delay bound, loss probability, proportional-loss.**

I. Introduction

RESOURCE allocation is an important component of OFDMA-based wireless systems, such as IEEE 802.16 [1] and the Long Term Evolution (LTE) [2], where channel access is partitioned into frames in the time domain and sub-channels in the frequency domain to achieve multi-user and frequency diversities. One obvious performance metric to evaluate resource allocation schemes is system throughput. A simple strategy to achieve high system throughput is to allocate more resources to users with better channel qualities. This strategy, unfortunately, may lead to starvation and cause QoS violation to real-time applications attached to users who have poor channel qualities. A well-designed resource allocation scheme should, therefore, take QoS support into consideration while maximizing system throughput.

Several previous works, say, [3], [4], adopted the concept of proportional fairness (PF) to eliminate starvation while maintaining acceptable system throughput. These schemes, although achieve a kind of fairness among users, are not suitable for QoS support. In [5] and [6], the ideas of PF and static minimum bandwidth guarantee were combined to support multiple service classes. This enhanced algorithm, however, does not take delay bound and loss probability requirements of real-time flows into consideration and thus is unlikely to provide QoS support well.

In [7], a power and sub-carrier allocation policy was proposed for system throughput optimization with the constraint that the average delay of each traffic flow is controlled to be lower than its pre-defined level. Guaranteeing average delay, however, is in general not sufficient for real-time applications. The results presented in [8] reveal that dynamic power allocation can only give a small improvement over fixed power allocation with an effective adaptive modulation and coding (AMC) scheme. As a result, to reduce the complexity, it is reasonable to design resource allocation schemes under the assumption that equal power is allocated to each sub-channel.

Some resource allocation algorithms were proposed, assuming equal-power allocation, to assign a user a higher priority for channel access if the deadline of its head-of-line (HOL) packet is smaller [9]-[12]. A simple scheme, called modified largest weighted delay first (M-LWDF), which uses a kind of utility function that is sensitive to loss probability and delay bound requirements as well as delay of HOL packets, was presented in [10]. Obviously, considering only the deadlines of HOL packets is not optimal. A QoS scheduling and resource allocation algorithm which considers deadlines of all packets was presented in [13]. This scheme requires high computational complexity and thus may not be practical for real systems. To reduce computational complexity, a matrix-based scheduling algorithm was proposed in [4]-[6]. The M-LWDF, the scheme proposed in [13] and the matrix-based scheduling algorithm are related to our work and will be reviewed in Section III.

The purpose of this paper is to present a resource allocation algorithm which tries to maximize system throughput with QoS support for real-time traffic flows. Our contributions include: 1) define and derive the minimum requested bandwidth of each real-time flow based on the loss probability requirement and the running loss probability, 2) formulate the resource allocation problem as one which maximizes system throughput subject to the constraint that the bandwidth allocated to a flow is greater than or equal to its minimum

Paper approved by A. MacKenzie, the Editor for Game Theory and Cognitive Networking of the IEEE Communications Society. Manuscript received October 15, 2010; revised May 9, 2011 and October 18, 2011.

The authors are with the Institute of Communication Engineering, National Chiao Tung University, Hsinchu, 30010 Taiwan (e-mail: {tlee, vicent}@banyan.cm.nctu.edu.tw).

Digital Object Identifier 10.1109/TCOMM.2012.020912.100632

requested value, 3) propose a user-level proportional-loss (PL) scheduler for multiple real-time traffic flows attached to the same subscriber station (SS) to share the allocated resource, and 4) modify the resource allocation problem to maximize the amount of real-time traffic transmitted and add a pre-processor in front of the PL scheduler to maximize the number of real-time flows attached to each SS that meet their QoS requirements, when the available resource is not sufficient to provide each flow its minimum requested bandwidth. We show that, in any frame, the proposed PL scheduler guarantees QoS if there is any scheduler which guarantees QoS. Simulation results reveal that our proposed algorithm performs better than previous works.

The rest of this paper is organized as follows. In Section II, we describe the investigated system model. Related works are reviewed in Section III. Section IV contains our proposed scheme. Simulation results are presented in Section V. Finally, we draw conclusion in Section VI.

II. System Model

We consider a single-cell OFDMA-based system which consists of one base station (BS) and multiple users or subscriber stations (SSs). Time is divided into frames, and the duration of a frame is equal to T_{frame}. In a frame, there are M sub-channels and S time slots. We assume that the sub-channel statuses of different SSs are independent. Moreover, for a given SS, its statuses on the M sub-channels are also independent. The channel quality for a given SS on a specific sub-channel is fixed during one frame. Transmission power is equally allocated to each sub-channel. To improve reliable transmission rate, an effective AMC scheme is adopted to choose a transmission mode based on the reported signal-to-noise ratio (SNR). We only consider downlink transmission.

For ease of description, we assume that no SS is attached with both real-time and non-real-time traffic flows. Let Γ_{RT} and Γ_{NRT} represent, respectively, the sets of SSs that are attached with real-time and non-real-time traffic flows. Further, let Γ=$\Gamma_{RT}\cup\Gamma_{NRT}$. We shall use K_n to denote the number of traffic flows attached to SS n. All non-real-time flows attached to the same SS are aggregated into one so that $K_n = 1$ if SS $n\in\Gamma_{NRT}$. The QoS requirements of real-time traffic flows are specified by delay bound and loss probability. The k^{th} flow attached to SS n is denoted by $f_{n,k}$. If SS $n\in\Gamma_{RT}$, then the delay bound and loss probability requirements of $f_{n,k}$ are represented by $D_{n,k}\cdot T_{frame}$ and $P_{n,k}$, respectively. Data are assumed to arrive at the beginning of frames.

In the BS, a separate queue is maintained for each real-time traffic flow while non-real-time data are stored per SS. Assume that SS $n\in\Gamma_{RT}$. The data of flow $f_{n,k}$ are buffered in $Queue_{n,k}$, which can be partitioned into $D_{n,k}$ disjoint virtual sub-queues, denoted by $Queue^d_{n,k}$, $1 \le d \le D_{n,k}$, where $Queue^d_{n,k}$ contains the data in $Queue_{n,k}$ that can be buffered up to $d\cdot T_{frame}$ without violating their delay bounds. We shall use $Q^d_{n,k}[t]$ to represent the size of $Queue^d_{n,k}$ at the beginning of the t^{th} frame (including the newly arrived), $Q_{n,k}[t] = \sum_{d=1}^{D_{n,k}} Q^d_{n,k}[t]$, and $Q_n[t] = \sum_{k=1}^{K_n} Q_{n,k}[t]$. Data which violate their delay bounds are dropped. It is assumed that the size of each queue is sufficiently large so that no data will be dropped due to buffer overflow. To simplify notation, the queue for storing data of SS $n \in \Gamma_{NRT}$ is denoted by $Queue_n$.

III. Related Works

In all the reviewed related works, resource allocation is performed at the beginning of each frame and, therefore, it suffices to consider one specific frame, say the t^{th} frame. For SS n, we denote its maximum achievable transmission rate on the m^{th} sub-channel in the t^{th} frame and its long-term average throughput up to the t^{th} frame by $r_{n,m}[t]$ and $\overline{r}_n[t]$, respectively.

A. Scheme of [13]

In [13], resource allocation is formulated as an optimization problem which maximizes some utility function subject to QoS guarantee. It consists of two stages. In the first stage, resources are allocated to real-time traffic flows only. If there are un-allocated resources after the first stage, the second stage is performed to allocate the remaining resources to non-real-time traffic.

In the first stage, called real-time QoS scheduling, the minimum requested bandwidth of each real-time traffic flow is calculated by R_n^{min}=$\sum_{k=1}^{K_n}\sum_{d=1}^{D_{n,k}} \frac{Q^d_{n,k}[t]}{d^\beta}$. Note that substituting β with 0, 1, or ∞ corresponds, respectively, to strict priority [14], average QoS provisioning [15], or urgent [16] scheduling policy. With the assumption that sub-channel is the smallest resource granularity, the first stage aims to minimize the total number of sub-channels used to serve the sum of calculated minimum requested bandwidths of all real-time flows. This problem can be modeled as maximum weighted bipartite matching (MWBM) and solved by the famous On Kuhn's Hungarian method, whose complexity is $O(M|\Gamma_{RT}|(\min(M,|\Gamma_{RT}|))^2)$ [17], where $|\Gamma_{RT}|$ is the size of Γ_{RT}.

In the second stage, the m^{th} sub-channel, if still available, is allocated to the SS which satisfies $n^* = \arg\max_{n\in\Gamma_{NRT}} U'_n(\overline{r}_n[t])r_{n,m}[t]$, where $U'_n(x)$, called marginal utility function, is the first derivative of the utility function. For every SS, the utility function, defined by α-proportional fairness [18], is given by

$$U^\alpha(x) = \begin{cases} (1-\alpha)^{-1}x^{(1-\alpha)}, & \text{if } \alpha \neq 1 \\ \log(x), & \text{otherwise,} \end{cases} \quad (1)$$

where x represents the average throughput. Note that the policy corresponds to maximum throughput, proportional fairness, or max-min fairness if α is chosen to be 0, 1, or ∞, respectively.

It was shown in [13] that the above scheme with $\beta = 1$ makes a reasonable trade-off between QoS support and maximization of system utility. However, it has some drawbacks. Firstly, assuming the granularity of resource to be sub-channels can result in waste of bandwidth. In current standards such as IEEE 802.16 and LTE, a sub-channel can be shared by multiple SSs. Secondly, although the number of sub-channels used to serve real-time traffic is minimized in the first stage, the remaining service capability for non-real-time traffic may

not be maximized. This is because the qualities of remaining sub-channels could be poor for SSs attached with non-real-time traffic flows. Thirdly, calculation of the minimum requested bandwidth for each real-time traffic flow does not take its loss probability requirement into consideration. Real-time traffic usually can tolerate data loss to certain degree. System throughput can be improved significantly if one takes advantage of this feature in resource allocation. Finally, the complexity of the Hungarian method could make this scheme infeasible for a real system.

B. Matrix-based Scheduling Algorithm [4]

A matrix-based scheduling algorithm which tries to maximize the utility sum of all users with acceptable computational complexity was proposed in [4]. In this scheme, a matrix $U = [u_{n,m}]$ of dimension $|\Gamma| \times M$ is defined for resource allocation, where $u_{n,m} = \frac{r_{n,m}[t]}{\overline{r}_n[t]}$ represents the marginal utility of user n on sub-channel m. For sub-channel m, let s_m represent the number of slots that have not been allocated and $x_{n,m}$ the number of slots allocated to SS n. Initially, we have $s_m = S$ and $x_{n,m} = 0, n \in \Gamma, 1 \le m \le M$. The matrix-based scheduling algorithm consists of three steps: 1) Find an (n^*, m^*) which satisfies $u_{n^*,m^*} = \max_{n\in\Gamma, 1\le m\le M}\{u_{n,m}\}$. 2) Set $x_{n^*,m^*} = \min(s_{m^*}, \lceil \frac{Q_{n^*}[t]}{r_{n^*,m^*}[t]} \rceil)$ (allocate $\lceil \frac{Q_{n^*}[t]}{r_{n^*,m^*}[t]} \rceil$ or all the remaining slots of sub-channel m^*, whichever is smaller, to user n^*), $Q_{n^*}[t] = \max(0, Q_{n^*}[t] - x_{n^*,m^*} \cdot r_{n^*,m^*}[t])$(update queue status of user n^*), and $s_{m^*} = s_{m^*} - x_{n^*,m^*}$(update the remaining number of slots of sub-channel m^*). Replace the $(n^*)^{th}$ row of U by an all-zero row if $Q_{n^*}[t] = 0$ (user n^* does not need any more resource) and the $(m^*)^{th}$ column of U by an all-zero column if $s_{m^*} = 0$ (all slots of sub-channel m^* are allocated). 3) Update $\overline{r}_{n^*}[t]$. If $Q_{n^*}[t] > 0$, then re-calculate $u_{n^*,m} = \frac{r_{n^*,m}[t]}{\overline{r}^*_{n^*}[t]}$ for all $m \ne m^*$ (update the marginal utilities of user n^* on various sub-channels before allocating the remaining resources). The above three steps are repeatedly executed until all elements of U are replaced with zeroes. The resulting values of $x_{n,m}$, $n \in \Gamma$, $1 \le m \le M$, are the solutions. Assuming that $M \ge |\Gamma|$, the computational complexity of the matrix-based scheduling algorithm in the worst case is $O(M^2|\Gamma| + |\Gamma|^2)$, which happens when $M-1$ columns of U are replaced by all-zero columns one by one, followed by replacing the rows by all-zero rows one by one. Its complexity is $O(|\Gamma|^2 M + M^2)$ if $M < |\Gamma|$.

Note that the matrix-based scheduling algorithm takes queue occupancy into consideration. However, it does not consider QoS support. The same authors combined the idea of PF with static minimum bandwidth guarantee to support multiple service classes [5], [6]. A user whose channel quality is better than some threshold is guaranteed a pre-defined minimum bandwidth. This enhanced version, still, cannot provide QoS support well because it does not consider delay bound and loss probability requirements of real-time flows.

C. Modified-largest Weighted Delay First (M-LWDF) [10]

The goal of the M-LWDF scheme is to achieve $P(W_{n,k} > D_{n,k}) \le P_{n,k}$ for all $n \in \Gamma_{RT}$, $1 \le k \le K_n$. In M-LWDF, the marginal utility of flow $f_{n,k}$ on sub-channel m is $\gamma_{n,k} \cdot W_{n,k}[t] \cdot r_{n,m}[t]$, where $W_{n,k}[t] \cdot T_{frame}$ is the delay of the HOL packet of $Queue_{n,k}$ at the beginning of frame t and $\gamma_{n,k}$ is an arbitrary positive constant. To transmit data, the flow with the largest marginal utility on some available sub-channel is selected for service. It was shown that M-LWDF is throughput-optimal in the sense that it is able to keep all queues stable if this is at all feasible to do with any scheduling algorithm. Moreover, it was reported that $\gamma_{n,k} = \frac{a_{n,k}}{\overline{r}_n[t]}$, where $a_{n,k} = -\frac{(\log P_{n,k})}{D_{n,k}}$, performs very well. Clearly, for such a selection of $\gamma_{n,k}$, the marginal utility is sensitive to loss probability and delay bound requirements as well as delay of the HOL packet. When combined with a token bucket control, M-LWDF can provide QoS support to flows with minimum bandwidth requirements. However, how to serve non-real-time flows with zero minimum bandwidth requirements was not studied. To compare its performance with that of our proposed scheme, we shall assume that the operation of M-LWDF is divided into two stages. In the first stage, only real-time traffic flows are considered. As a consequence, the first stage of M-LWDF is the same as that of the matrix-based scheduling, except for a different marginal utility function. The complexity of the first stage is $\max\{O(M^2|\Gamma_{RT}|+|\Gamma_{RT}|^2), O(|\Gamma_{RT}|^2 M+M^2)\}$. If there are un-allocated resources after the first stage, then the remainig resources are allocated in the second stage to non-real-time flows with zero minimum resource requirements. The goal of the second stage is to maximize system throughput. Assume that the matrix-based scheduling algorithm is adopted in the second stage. As a result, the complexity of the second stage is $\max\{O(M^2|\Gamma_{NRT}| + |\Gamma_{NRT}|^2), O(|\Gamma_{NRT}|^2 M + M^2)\}$.

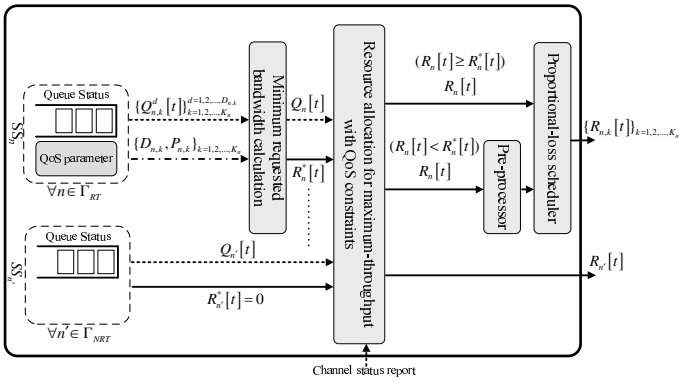

Fig. 1. Architecture of the proposed scheme.

IV. The Proposed Scheme

In this section, we present a resource allocation scheme which considers both delay bound and loss probability requirements requested by real-time traffic flows. As shown in Fig. 1, the minimum requested bandwidths of real-time flows are computed, summed for each SS, and then used together with queue occupancy as constraints in resource allocation. After the solution is obtained, a PL scheduler is adopted to determine how multiple real-time traffic flows attached to the same SS share the allocated bandwidth. In case the available resource is not sufficient to provide each flow its minimum requested bandwidth, a pre-processor is required to maximize the number of real-time flows attached to each SS that meet

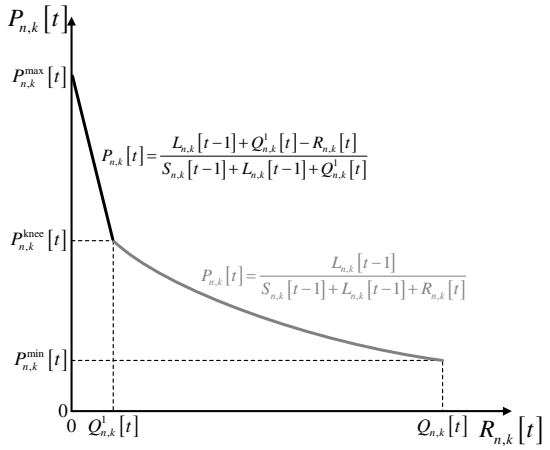

Fig. 2. The relationship between $P_{n,k}[t]$ and $R_{n,k}[t]$.

their QoS requirements. We describe calculation of minimum requested bandwidth, resource allocation, PL scheduler, and pre-processor separately below.

A. The Minimum Requested Bandwidth

For flow $f_{n,k}$ attached to SS $n \in \Gamma_{RT}$, define $P_{n,k}[x]$, the running loss probability up to frame x, as $P_{n,k}[x] = \frac{L_{n,k}[x]}{S_{n,k}[x]+L_{n,k}[x]}$, where $S_{n,k}[x]$ and $L_{n,k}[x]$ represent, respectively, the accumulated amount of data served and lost up to the end of the x^{th} frame. Consider the t^{th} frame. Let $R_{n,k}[t]$ be the bandwidth allocated to flow $f_{n,k}$. For convenience, $R_{n,k}[t]$ is expressed in terms of the amount of data served. As a result, we have $0 \le R_{n,k}[t] \le Q_{n,k}[t]$. Let $x^+ = \max(0, x)$. Since data are lost only due to violation of their delay bounds, we have

$$P_{n,k}[t] = \frac{L_{n,k}[t-1] + (Q_{n,k}^{1}[t] - R_{n,k}[t])^+}{S_{n,k}[t-1] + L_{n,k}[t-1] + \max(R_{n,k}[t], Q_{n,k}^{1}[t])}. \quad (2)$$

It is not hard to see that $P_{n,k}[t]$ is a continuous, strictly decreasing function of $R_{n,k}[t]$ in the range $0 \le R_{n,k}[t] \le Q_{n,k}[t]$. The curve of $P_{n,k}[t]$ as a function of $R_{n,k}[t]$ is illustrated in Fig. 2. In this figure, there are three special points on the y-axis, namely, $P_{n,k}^{\max}[t]$, $P_{n,k}^{\text{knee}}[t]$, and $P_{n,k}^{\min}[t]$, which can be obtained by substituting $R_{n,k}[t]$ with 0, $Q_{n,k}^{1}[t]$, and $Q_{n,k}[t]$ into equation (2), respectively. Note that if $Q_{n,k}[t] = 0$, we have $P_{n,k}[t] = P_{n,k}[t-1] = P_{n,k}^{\max}[t] = P_{n,k}^{\text{knee}}[t] = P_{n,k}^{\min}[t]$.

The minimum requested bandwidth of $f_{n,k}$, denoted by $R_{n,k}^{*}[t]$, is determined as follows. If $P_{n,k} \ge P_{n,k}^{\max}[t]$, then we set $R_{n,k}^{*}[t] = 0$ because there is no loss probability violation even if zero resource is allocated to $f_{n,k}$. Assume that $P_{n,k}^{\max}[t] > P_{n,k} > P_{n,k}^{\min}[t]$. In this case, $R_{n,k}^{*}[t]$ is obtained by solving $P_{n,k} = P_{n,k}[t]$, where $P_{n,k}[t]$ is described by equation (2). Finally, if $P_{n,k} \le P_{n,k}^{\min}[t]$, then the running loss probability is still greater than or equal to the pre-defined level $P_{n,k}$ even if all buffered data of $f_{n,k}$ are served. Therefore, we assign $R_{n,k}^{*}[t] = Q_{n,k}[t]$ to minimize the difference between $P_{n,k}[t]$ and $P_{n,k}$. For convenience, we use $P_{n,k}^{*}[t]$ to denote the running loss probability of $f_{n,k}$ at the end of the t^{th} frame if the bandwidth allocated to $f_{n,k}$ is $R_{n,k}^{*}[t]$. Clearly, $P_{n,k}^{*}[t]$ equals $P_{n,k}^{\max}[t]$ if $P_{n,k} > P_{n,k}^{\max}[t]$ or $P_{n,k}^{\min}[t]$ if $P_{n,k} < P_{n,k}^{\min}[t]$. The following lemma states that $P_{n,k}^{*}[t]$ is closer to $P_{n,k}$ than any other $P_{n,k}[t]$.

Lemma 1. *It holds that*

$$\min_{0 \le R_{n,k}[t] \le Q_{n,k}[t]} |P_{n,k}[t] - P_{n,k}| = |P_{n,k}^{*}[t] - P_{n,k}|.$$

Proofs of lemmas and theorems are provided in Appendix A. The minimum requested bandwidth for all cases is summarized in Table I. Note that the actual allocated bandwidth could be different from $R_{n,k}^{*}[t]$. After obtaining $R_{n,k}^{*}[t]$ for all k, $1 \le k \le K_n$, one can compute $R_{n}^{*}[t]$, the aggregate minimum requested bandwidth for SS n, as $\sum_{k=1}^{K_n} R_{n,k}^{*}[t]$. The values of $R_{n}^{*}[t]$, $n \in \Gamma_{RT}$ are used in the resource allocation algorithm described in the next sub-section.

B. Resource Allocation for Maximum-throughput With QoS Constraints

As described in Problem **P1**, the proposed resource allocation algorithm maximizes system throughput while providing QoS guarantee to real-time traffic flows. In problem **P1**, we let $R_{n}^{*}[t] = 0$ for all SS $n \in \Gamma_{NRT}$. As in previous section, we use $r_{n,m}[t]$ to denote the maximum achievable transmission rate on the m^{th} sub-channel for SS n in the t^{th} frame. The variable $x_{n,m}[t]$ represents the number of time slots allocated to SS n on the m^{th} sub-channel, in the t^{th} frame.

P1

$$\max \sum_{n \in \Gamma} \sum_{m=1}^{M} x_{n,m}[t] \cdot r_{n,m}[t], \quad (3)$$

subject to

$$\sum_{n \in \Gamma} x_{n,m}[t] \le S, \forall m, 1 \le m \le M, \quad (4)$$

$$R_{n}^{*}[t] \le \sum_{m=1}^{M} x_{n,m}[t] \cdot r_{n,m}[t] \le Q_{n}[t], \forall n \in \Gamma, \quad (5)$$

and

$$x_{n,m}[t] \in \{0, 1, 2, ..., S\}, \forall n \in \Gamma, 1 \le m \le M. \quad (6)$$

Problem **P1** can be solved by some integer linear programming algorithm [19]. If there is no feasible solution, meaning that the available resource is smaller than the summation of all minimum requested bandwidths, we set $x_{n,m}[t] = 0$, for all $n \in \Gamma_{NRT}$, $1 \le m \le M$, and solve a modified problem, called problem **P2**, which is basically the same as problem **P1** except that the constraint shown in equation (5) is replaced by $0 \le \sum_{m=1}^{M} x_{n,m}[t] \cdot r_{n,m}[t] \le R_{n}^{*}[t]$, $\forall n \in \Gamma$. Note that the solution of Problem **P2** always exists because $x_{n,m}[t] = 0$, for all $n \in \Gamma$, $1 \le m \le M$, is one feasible solution. Unfortunately, the complexity of integer linear programming is NP-complete [20]. One possible strategy to mitigate the computational complexity is to set $u_{n,m} = r_{n,m}[t]$ for all $n \in \Gamma$, $1 \le m \le M$, and conduct the matrix-based scheduling algorithm for one or two rounds. In the first round, we only consider SSs contained in Γ_{RT}, assuming that the queue occupancy of SS n is equal to $R_{n}^{*}[t]$. The algorithm ends if the resource is exhausted in the first round. Otherwise, the second round is performed to

TABLE I
CALCULATION OF $R^*_{n,k}[t]$ AND THE RESULTING $P^*_{n,k}[t]$ FOR FOUR CONDITIONS

Condition	$R^*_{n,k}[t]$	$P^*_{n,k}[t]$
$P_{n,k} \geq P^{\max}_{n,k}[t]$	0	$P^{\max}_{n,k}[t]$
$P^{\max}_{n,k} > P_{n,k} \geq P^{\text{knee}}_{n,k}[t]$	$(1-P_{n,k})(L_{n,k}[t-1]+Q^1_{n,k}[t])$ $-P_{n,k}\cdot S_{n,k}[t-1]$	$P_{n,k}$
$P^{\text{knee}}_{n,k} > P_{n,k} > P^{\min}_{n,k}[t]$	$\frac{L_{n,k}[t-1]}{P_{n,k}}$ $-(S_{n,k}[t-1]+L_{n,k}[t-1])$	$P_{n,k}$
$P_{n,k} \leq P^{\min}_{n,k}[t]$	$Q_{n,k}[t]$	$P^{\min}_{n,k}[t]$

allocate the remaining resource to all SSs, assuming the queue occupancy of SS n is equal to $Q_n[t] - R^*_n[t]$. According to the analysis provided in the last section, the computational complexity of the modified matrix-based scheduling algorithm is $O(\max(M^2|\Gamma| + |\Gamma|^2, |\Gamma|^2 M + M^2))$.

Let $y_{n,m}[t]$ be the solution obtained either from integer linear programming or matrix-based scheduling algorithm. We have $R_n[t] = \sum_{m=1}^{M} y_{n,m}[t] \cdot r_{n,m}[t]$. If $R_n[t] = R^*_n[t]$, then the bandwidth allocated to the k^{th} attached flow, i.e., $R_{n,k}[t]$, is equal to $R^*_{n,k}[t]$. Assume that $R_n[t] \neq R^*_n[t]$. In this case, we need a user-level resource allocation algorithm for the attached flows to share the allocated bandwidth. In the following sub-section, we define the PL scheduler to solve this problem.

C. Proportional-loss (PL) Scheduler

Consider SS n and assume that it is attached with multiple real-time traffic flows. Define three disjoint sets U_{Z}, U_{P}, and U_{A} such that flow $f_{n,k}$ is contained in U_{Z}, U_{P}, or U_{A} iff $R_{n,k}[t] = 0$, $0 < R_{n,k}[t] < Q_{n,k}[t]$, or $R_{n,k}[t] = Q_{n,k}[t]$, respectively. Given $R_{n,k}[t]$, the proposed PL scheduler is a scheduler which achieves, for any $f_{n,z} \in U_{\text{Z}}$, $f_{n,p}, f_{n,p'} \in U_{\text{P}}$, and $f_{n,a} \in U_{\text{A}}$,

$$\frac{P_{n,z}[t]}{P_{n,z}} \leq \frac{P_{n,p}[t]}{P_{n,p}} = \frac{P_{n,p'}[t]}{P_{n,p'}} \leq \frac{P_{n,a}[t]}{P_{n,a}}, \tag{7}$$

subject to

$$R_n[t] = \sum_{k=1}^{K_n} R_{n,k}[t]. \tag{8}$$

Define $\frac{P_{n,k}[t]}{P_{n,k}}$ as the normalized running loss probability of $f_{n,k}$ up to frame t. The proposed PL scheduler achieves min-max optimality, as stated in Lemma 2. In Theorem 3, we show that if there exists a scheduler which guarantees the loss probability requirements, so does the PL scheduler.

Lemma 2. *Given $R_n[t] > 0$, $S_{n,k}[t-1]$, $L_{n,k}[t-1]$ and $\{Q^d_{n,k}[t]\}_{d=1}^{D_{n,k}}$, $1 \leq k \leq K_n$, the proposed PL scheduler minimizes the maximum normalized running loss probability of all the traffic flows attached to SS n.*

Theorem 3. *Given $R_n[t] > 0$, $S_{n,k}[t-1]$, $L_{n,k}[t-1]$ and $\{Q^d_{n,k}[t]\}_{d=1}^{D_{n,k}}$, $1 \leq k \leq K_n$, if there exists a scheduler which can guarantee the loss probability requirements of all the K_n traffic flows, so can the PL scheduler.*

Theorem 3 provides the answer why the PL scheduler is proposed as the user-level resource allocation algorithm. Define $[R_n[t], S_{n,k}[t-1], L_{n,k}[t-1]$, and $\{Q^d_{n,k}[t]\}_{d=1}^{D_{n,k}} (1 \leq k \leq K_n)]$ as the state of SS n at the beginning of the t^{th} frame. Given the state at the beginning of the first frame, the PL scheduler is preferred over other schedulers in the first frame, according to Theorem 3. Assume that the PL scheduler is adopted in the first frame. The state at the beginning of the second frame is determined once traffic arrivals at the beginning of the second frame is known and $R_n[2]$ is provided. Based on Theorem 3 again, the PL scheduler is still the preferred scheduler in the second frame. The arguments can be applied to all frames.

In the rest of this sub-section, we present a realization of the PL scheduler. Again, consider SS n in the t^{th} frame and assume that $R_n[t]$ is given. We need to determine $R_{n,k}[t]$, $1 \leq k \leq K_n$, so that equations (7) and (8) are satisfied.

Lemma 4. *If $R_n[t] = R^*_n[t]$, equations* (7) *and* (8) *are satisfied for $R_{n,k}[t] = R^*_{n,k}[t]$, $1 \leq k \leq K_n$.*

Assume that $R_n[t] \neq R^*_n[t]$. We have the following Theorem 5.

Theorem 5. *Define $\Delta R_n[t] = R_n[t] - R^*_n[t]$ and $\Delta R_{n,k}[t] = R_{n,k}[t] - R^*_{n,k}[t]$, $1 \leq k \leq K_n$. Under the PL scheduler, it holds that $\Delta R_{n,k}[t] \geq 0$ $(1 \leq k \leq K_n)$ if $\Delta R_n[t] \geq 0$ or $\Delta R_{n,k}[t] \leq 0$ otherwise.*

A consequence of Theorem 5 is that $R^*_{n,k}[t] = Q_{n,k}[t]$ implies $R_{n,k}[t] = Q_{n,k}[t]$ if $R_n[t] \geq R^*_n[t]$; and $R^*_{n,k}[t] = 0$ implies $R_{n,k}[t] = 0$ if $R_n[t] \leq R^*_n[t]$. To realize the PL scheduler, we start with $R_{n,k}[t] = R^*_{n,k}[t]$, $1 \leq k \leq K_n$. If $R_n[t] = R^*_n[t]$, then the solution is found. Adjustment is necessary if $R_n[t] \neq R^*_n[t]$. To do the adjustment, flows are classified into four sets U_{Z}, U_{P1}, U_{P2}, and U_{A} such that $f_{n,k}$ is in U_{Z}, U_{P1}, U_{P2}, or U_{A} iff $R^*_{n,k}[t] = 0$, $0 < R^*_{n,k}[t] \leq Q^1_{n,k}[t]$, $Q^1_{n,k}[t] < R^*_{n,k}[t] < Q_{n,k}[t]$, or $R^*_{n,k}[t] = Q_{n,k}[t]$, respectively. Two cases are considered separately.

Case 1 $R_n[t] > R^*_n[t]$

According to Theorem 5, $R_n[t] > R^*_n[t]$ implies $R_{n,k}[t] \geq R^*_{n,k}[t]$. Therefore, we should increase the value of $R_{n,k}[t]$ for $f_{n,k} \in U_{\text{P1}} \cup U_{\text{P2}} \cup U_{\text{Z}}$. Our idea is to increase $R_{n,k}[t]$ gradually, keeping equations (7) satisfied, until $R_n[t] = \sum_{k=1}^{K_n} R_{n,k}[t]$ is true. During the process of increasing $R_{n,k}[t]$, we shall either find a solution or have to move a flow from U_{Z} to U_{P1}, from U_{P1} to U_{P2}, or from U_{P2} to U_{A}. For example, assume that $f_{n,i} \in U_{\text{P1}}$ and the first event, called Event 1, we encountered is to move $f_{n,i}$ from U_{P1} to U_{P2}. For Event 1 to happen, the conditions to be met are 1) $\frac{P^{\text{knee}}_{n,i}[t]}{P_{n,i}} = \max_{f_{n,k} \in U_{\text{P1}}} \frac{P^{\text{knee}}_{n,k}[t]}{P_{n,k}}$ (no flow is moved from U_{P1} to U_{P2} earlier than Event 1), 2) $\frac{P^{\text{knee}}_{n,i}[t]}{P_{n,i}} \geq \max_{f_{n,k} \in U_{\text{P2}}} \frac{P^{\min}_{n,k}[t]}{P_{n,k}}$ (no flow is moved from U_{P2} to U_{A} earlier than Event 1), 3) $\frac{P^{\text{knee}}_{n,i}[t]}{P_{n,i}} \geq \max_{f_{n,k} \in U_{\text{Z}}} \frac{P^{\max}_{n,k}[t]}{P_{n,k}}$ (no

$$h_{n,k}(x;t)=\begin{cases}\frac{1}{x}\cdot L_{n,k}[t-1]-S_{n,k}[t-1]-L_{n,k}[t-1], & \text{if } P_{n,k}^{\min}[t]\le x<P_{n,k}^{\text{knee}}[t]\\ L_{n,k}[t-1]+Q_{n,k}^{1}[t]-x\cdot(S_{n,k}[t-1]+L_{n,k}[t-1]+Q_{n,k}^{1}[t]), & \text{if } P_{n,k}^{\text{knee}}[t]\le x\le P_{n,k}^{\max}[t]\end{cases} \quad (9)$$

flow is moved from U_{Z} to U_{P1} earlier than Event 1), and 4) $\sum_{f_{n,k}\in U_{\text{P1}}\cup U_{\text{P2}}} h_{n,k}((\frac{P_{n,i}^{\text{knee}}[t]}{P_{n,i}})\cdot P_{n,k};t)+\sum_{f_{n,k}\in U_{\text{A}}} Q_{n,k}[t] < R_n[t]$ (no solution is found earlier than Event 1), where the definition of $h_{n,k}(x;t)$ is shown in equation (9). Note that $h_{n,k}(x;t)$ is the inverse function of $P_{n,k}[t]$ shown in equation (2). The conditions for other events to happen can be similarly determined. After all flows are placed in the correct sets, the solution can be obtained by solving equations (7) and (8). To summarize, we repeatedly check the inequality shown in equation (10). If it holds, flow f_{n,k^*} is moved from one set to another.

$$\sum_{f_{n,k}\in U_{\text{P1}}\cup U_{\text{P2}}} h_{n,k}(p\cdot P_{n,k};t)+\sum_{f_{n,k}\in U_{\text{A}}} Q_{n,k}[t]<R_n[t], \quad (10)$$

where

$$p=\max(\max_{f_{n,k}\in U_{\text{Z}}}\frac{P_{n,k}^{\max}[t]}{P_{n,k}},\max_{f_{n,k}\in U_{\text{P1}}}\frac{P_{n,k}^{\text{knee}}[t]}{P_{n,k}},\max_{f_{n,k}\in U_{\text{P2}}}\frac{P_{n,k}^{\min}[t]}{P_{n,k}}), \quad (11)$$

and

$$k^*=\arg\max(\max_{f_{n,k}\in U_{\text{Z}}}\frac{P_{n,k}^{\max}[t]}{P_{n,k}},\max_{f_{n,k}\in U_{\text{P1}}}\frac{P_{n,k}^{\text{knee}}[t]}{P_{n,k}},\max_{f_{n,k}\in U_{\text{P2}}}\frac{P_{n,k}^{\min}[t]}{P_{n,k}}). \quad (12)$$

All flows are placed in their correct sets once the inequality shown in equation (10) becomes false. The solution can then be obtained as follows. Set $R_{n,k}[t]=0$ if $f_{n,k}\in U_{\text{Z}}$ or $Q_{n,k}[t]$ if $f_{n,k}\in U_{\text{A}}$. For $f_{n,k}\in U_{\text{P1}}\cup U_{\text{P2}}$, $R_{n,k}[t]$ can be obtained by $R_{n,k}[t]=h_{n,k}(P_n^{\text{F}}[t]\cdot P_{n,k};t)$, where $P_n^{\text{F}}[t]$ represents the normalized running loss probability for any $f_{n,k}\in U_{\text{P1}}\cup U_{\text{P2}}$ at the end of the t^{th} frame and is derived in Appendix B.

Case 2 $R_n[t]<R_n^*[t]$

Case 2 is similar to Case 1, except that we need to decrease $R_{n,k}[t]$ for $f_{n,k}\in U_{\text{P1}}\cup U_{\text{P2}}\cup U_{\text{A}}$. For this case, we repeatedly check the inequality shown in equation (13) until it becomes false. If it is true, flow f_{n,k^*} is moved from U_{A} to U_{P2}, from U_{P2} to U_{P1}, or from U_{P1} to U_{Z}.

$$\sum_{f_{n,k}\in U_{\text{P1}}\cup U_{\text{P2}}} h_{n,k}(p\cdot P_{n,k};t)+\sum_{f_{n,k}\in U_{\text{A}}} Q_{n,k}[t]>R_n[t], \quad (13)$$

where

$$p=\min(\min_{f_{n,k}\in U_{\text{P1}}}\frac{P_{n,k}^{\max}[t]}{P_{n,k}},\min_{f_{n,k}\in U_{\text{P2}}}\frac{P_{n,k}^{\text{knee}}[t]}{P_{n,k}},\min_{f_{n,k}\in U_{\text{A}}}\frac{P_{n,k}^{\min}[t]}{P_{n,k}}), \quad (14)$$

and

$$k^*=\arg\min(\min_{f_{n,k}\in U_{\text{P1}}}\frac{P_{n,k}^{\max}[t]}{P_{n,k}},\min_{f_{n,k}\in U_{\text{P2}}}\frac{P_{n,k}^{\text{knee}}[t]}{P_{n,k}},\min_{f_{n,k}\in U_{\text{A}}}\frac{P_{n,k}^{\min}[t]}{P_{n,k}}). \quad (15)$$

After the inequality shown in equation (13) becomes false, the solution can be obtained as follows. Set $R_{n,k}[t]=0$ if $f_{n,k}\in U_{\text{Z}}$ or $Q_{n,k}[t]$ if $f_{n,k}\in U_{\text{A}}$. For $f_{n,k}\in U_{\text{P1}}\cup U_{\text{P2}}$, $R_{n,k}[t]$ can be obtained by $R_{n,k}[t]=h_{n,k}(P_n^{\text{F}}[t]\cdot P_{n,k};t)$. The pseudo code of the above realization of the PL scheduler is provided below.

Algorithm 1: PL scheduler

Data:
1) $U_{\text{Z}}=\{f_{n,k}: R_{n,k}^*[t]=0\}$
2) $U_{\text{P1}}=\{f_{n,k}: 0<R_{n,k}^*[t]\le Q_{n,k}^1[t]\}$
3) $U_{\text{P2}}=\{f_{n,k}: Q_{n,k}^1[t]<R_{n,k}^*[t]<Q_{n,k}[t]\}$
4) $U_{\text{A}}=\{f_{n,k}: R_{n,k}^*[t]=Q_{n,k}[t]\}$

Result: $R_{n,k}[t]$ for all $f_{n,k}$ with $Q_{n,k}[t]>0$, $1\le k\le K_n$

begin
- **if** $R_n[t]=R_n^*[t]$ **then**
  - $R_{n,k}[t]=R_{n,k}^*[t]$,$1\le k\le K_n$
- **else if** $R_n[t]>R_n^*[t]$ **then**
  - **while** (1) **do**
    - calculate p according to equation (11)
    - **if** equation (10) *is false* **then**
      - $R_{n,k}[t]=0$ for all $f_{n,k}\in U_{\text{Z}}$
      - $R_{n,k}[t]=Q_{n,k}[t]$ for all $f_{n,k}\in U_{\text{A}}$
      - $R_{n,k}[t]=h_{n,k}(P_n^{\text{F}}[t]\cdot P_{n,k};t)$ for all $f_{n,k}\in U_{\text{P1}}\cup U_{\text{P2}}$
      - (Flow $f_{n,k}$ is moved from U_{P2} to U_{A} if $R_{n,k}[t]=Q_{n,k}[t]$.)
      - **exit**
    - **else**
      - determine k^* according to equation (12)
      - **if** $f_{n,k^*}\in U_{\text{Z}}$ **then**
        - $U_{\text{Z}}=U_{\text{Z}}-f_{n,k^*}$,$U_{\text{P1}}=U_{\text{P1}}\cup f_{n,k^*}$
      - **else if** $f_{n,k^*}\in U_{\text{P1}}$ **then**
        - $U_{\text{P1}}=U_{\text{P1}}-f_{n,k^*}$,$U_{\text{P2}}=U_{\text{P2}}\cup f_{n,k^*}$
      - **else**
        - $U_{\text{P2}}=U_{\text{P2}}-f_{n,k^*}$,$U_{\text{A}}=U_{\text{A}}\cup f_{n,k^*}$
      - **end**
    - **end**
  - **end**
- **else**
  - **while** (1) **do**
    - calculate p according to equation (14)
    - **if** equation (13) *is false* **then**
      - $R_{n,k}[t]=0$ for all $f_{n,k}\in U_{\text{Z}}$
      - $R_{n,k}[t]=Q_{n,k}[t]$ for all $f_{n,k}\in U_{\text{A}}$
      - $R_{n,k}[t]=h_{n,k}(P_n^{\text{F}}[t]\cdot P_{n,k};t)$ for all $f_{n,k}\in U_{\text{P1}}\cup U_{\text{P2}}$
      - (Flow $f_{n,k}$ is moved from U_{P2} to U_{P1} if $R_{n,k}[t]=Q_{n,k}^1[t]$ or from U_{P1} to U_{Z} if $R_{n,k}[t]=0$.)
      - **exit**
    - **else**
      - determine k^* according to equation (15)
      - **if** $f_{n,k^*}\in U_{\text{P1}}$ **then**
        - $U_{\text{P1}}=U_{\text{P1}}-f_{n,k^*}$,$U_{\text{Z}}=U_{\text{Z}}\cup f_{n,k^*}$
      - **else if** $f_{n,k^*}\in U_{\text{P2}}$ **then**
        - $U_{\text{P2}}=U_{\text{P2}}-f_{n,k^*}$,$U_{\text{P1}}=U_{\text{P1}}\cup f_{n,k^*}$
      - **else**
        - $U_{\text{A}}=U_{\text{A}}-f_{n,k^*}$,$U_{\text{P2}}=U_{\text{P2}}\cup f_{n,k^*}$
      - **end**
    - **end**
  - **end**
- **end**

end

Note that, for Case 1, the maximum number of iterations needed for the PL scheduler is $3K_n$, which happens when each flow is moved from $U_{\rm Z}$ to $U_{\rm P1}$, from $U_{\rm P1}$ to $U_{\rm P2}$, and then from $U_{\rm P2}$ to $U_{\rm A}$. In each iteration, the computational complexity is $O(K_n)$. Therefore, the total computational complexity is $O(K_n^2)$. Obviously, the complexity for Case 2 is the same.

D. Pre-processor

Assume that $R_n[t] < R_n^*[t]$ (i.e., Case 2 occurs) and $R_{n,k}^*[t] > 0$. In this case, flow $f_{n,k}$ will violate its loss probability requirement if the PL scheduler is adopted. As a consequence, all flows attached to SS n violate their loss probability requirements if $R_{n,k}^*[t] > 0$ for all k. This is clearly not desirable. One possible remedy is to place a pre-processor in front of the PL scheduler to maximize the number of flows which meet their loss probability requirements. Let $\Omega = U_{\rm P1} \cup U_{\rm P2} \cup \{f_{n,k} | f_{n,k} \in U_{\rm A}, P_{n,k}^*[t] = P_{n,k}\}$. The operation of the pre-processor is as follows. 1) Select flow $f_{n,k}$ which satisfies $R_{n,k}^*[t] = \min_{f_{n,i} \in \Omega}\{R_{n,i}^*[t]\}$, 2) End the pre-processor operation if $R_{n,k}^*[t] > R_n[t]$. Otherwise, set $R_{n,k}[t] = R_{n,k}^*[t]$ and remove $f_{n,k}$ from the set it originally belongs to, 3) Update $R_n[t] = R_n[t] - R_{n,k}^*[t]$ and $\Omega = \Omega - \{f_{n,k}\}$, 4) End the pre-processor operation if $\Omega = \emptyset$. Otherwise, repeat the process. After the operation of the pre-process ends, the remaining resource is allocated to the remaining flows belonging to $U_{\rm P1} \cup U_{\rm P2} \cup U_{\rm A}$ by the PL scheduler. Clearly, the computational complexity of the pre-processor is $O(K_n' \log K_n')$, where $K_n' = |U_{\rm P1} \cup U_{\rm P2} \cup \{f_{n,k} | f_{n,k} \in U_{\rm A}, P_{n,k}^*[t] = P_{n,k}\}| \leq K_n$. As will be seen in the next section, adoption of the pre-processor can significantly increase the number of real-time flows which meet their QoS requirements.

V. Simulation Results

In our simulations, SSs are uniformly distributed in a circular area of radius 2Km and the BS is located at the center. Two types of real-time traffic flows are studied. Parameters of the simulation environment, AMC schemes, traffic specifications and QoS requirements of real-time flows are summarized in Table II. A frame is decomposed into downlink and uplink sub-frame. We only consider downlink transmission, which is assumed to occupy 30 time slots in a frame. The other time slots are used for uplink transmission and signaling overhead. For non-real-time traffic, we assume that its queue is always non-empty. Two scenarios are investigated. In both scenarios, we assume that $|\Gamma_{NRT}| = 40$ and the minimum requested bandwidth of every non-real-time flow is zero.

In the first scenario, in addition to the 40 non-real-time flows, there are various number of SSs each attached with one Type I real-time flow. The second scenario has 13 SSs each attached with two real-time flows, one of Type I and another of Type II. Simulations are performed for 10,000 frames using Matlab on a PC with an Intel Core 2 Quad CPU operated at 2.83GHz with 3072 MB of RAM.

For the first scenario, we compare our proposed scheme with the pure maximum-throughput algorithm, the three scheduling polices proposed in [13], and the M-LWDF scheme. To maximize system throughput, the minimum requested bandwidth of any real-time traffic flow is zero for the pure maximum-throughput algorithm. For fair comparison, we change the resource granularity from sub-channel to time slot for the three policies proposed in [13]. With such a change, their performances are better than the original versions. We label our proposed scheme by "proposed:ILP" or "proposed:Matrix" if the resource allocation problem is solved by integer linear programming or matrix-based scheduling algorithm, respectively. Both the PL scheduler and the pre-processor are adopted in Scenario 2 for all investigated schemes, except the M-LWDF scheme.

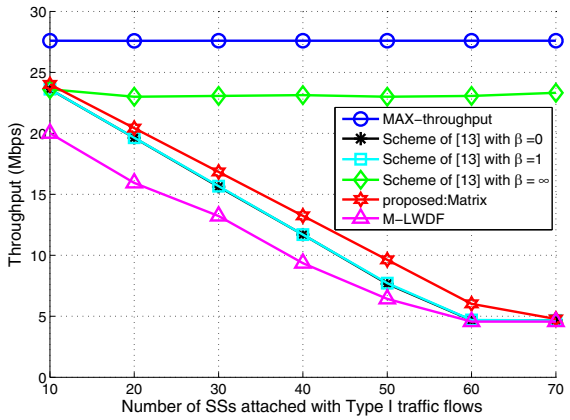

Fig. 3. Throughputs of various schemes in the first scenario.

In Fig. 3 and Fig. 4, we compare, respectively, total system throughput and loss probability of the investigated schemes for SSs attached with Type I real-time traffic flows in the first scenario. Compared with the schemes presented in [13] for $\beta = 0$ and $\beta = 1$, our proposed scheme achieves better system throughput. The maximum improvement is about 28% (6.018Mbps versus 4.696Mbps), which occurs when $|\Gamma_{NRT}| = 60$. Although the pure maximum-throughput algorithm and the scheme presented in [13] for $\beta = \infty$ have better throughput performance than our proposed scheme, their loss probabilities are higher than the specified value. In fact, a large proportion (about 80%) of real-time data is lost for the pure maximum-throughput algorithm. The reason is that there are many SSs attached with non-real-time traffic flows that are assumed to always have data for transmission. The improvement of our proposed scheme stops when $|\Gamma_{RT}| \geq 70$. The reason is that, for $|\Gamma_{RT}| \geq 70$, the average running loss probability is greater than the loss probability requirement and, therefore, the resource is allocated to users with good channel qualities by our proposed scheme and the scheme presented in [13] for $\beta = 0$ and $\beta = 1$. Compared with the M-LWDF scheme, our proposed algorithm achieves higher throughput without sacrificing QoS guarantee.

In Fig. 5 and Fig. 6, we compare the performances of our proposed:ILP and proposed:Matrix schemes. Results show that the difference is not significant. For $|\Gamma_{RT}| = 30$, the execution time of the proposed:Matrix scheme is 0.9 ms, which is much smaller than 47.4 ms, the execution time of the proposed:ILP scheme.

TABLE II
PARAMETERS OF SIMULATION ENVIRONMENT, TRAFFIC CHARACTERISTICS, QOS REQUIREMENTS AND ADOPTED MODULATION AND CODING SCHEME.

Simulation environment			
Radius of cell	2 km		
User distribution	Uniform		
Bandwidth	10 MHz		
Channel model	Rayleigh fading channel		
Doppler frequency	4.6 Hz (speed:2 km/hr)		
Pass loss exponent	4		
Frame duration	5ms		
Time slot duration	0.1ms		
Number of sub-channels	16		
Number of sub-carriers	64 (per sub-channel)		
Traffic characteristics and QoS requirements			
Traffic Type	Type I	Type II [21]	
Content	Voice	video streaming (Star War II)	
Codec format	G.711	MPEG 4	
Mean inter-arrival time	20ms	40ms	
Mean packet size	200 bytes	267bytes	
Delay bound	80ms	160ms	
Loss probability requirement	10(%)	5, 10, 15, 20, 25(%)	
The adopted modulation and coding scheme [12]			
Mode	Modulation	Coding rate	Receiver SNR (dB)
1	QPSK	1/2	5
2	QPSK	3/4	8
3	16QAM	1/2	10.5
4	16QAM	3/4	14
5	64QAM	1/2	16
6	64QAM	2/3	18
7	64QAM	3/4	20

TABLE III
LOSS PROBABILITIES FOR USERS ATTACHED WITH ONE TYPE I AND ONE TYPE II REAL-TIME FLOWS.

Loss probability requirement	M-LWDF		Scheme of [13] with $\beta = 0$		Scheme of [13] with $\beta = 1$		proposed: Matrix	
	$P_{L,\mathrm{I}}$	$P_{L,\mathrm{II}}$	$P_{L,\mathrm{I}}$	$P_{L,\mathrm{II}}$	$P_{L,\mathrm{I}}$	$P_{L,\mathrm{II}}$	$P_{L,\mathrm{I}}$	$P_{L,\mathrm{II}}$
5%	0.0025	0.0013	0.0182	0.0091	0.0671	0.0336	0.1000	0.0502
10%	0	0.0035	0.0122	0.0122	0.0448	0.0448	0.1000	0.1000
15%	0	0.0036	0.0094	0.0141	0.0342	0.0513	0.1002	0.1505
20%	0	0.0037	0.0079	0.0158	0.0280	0.0561	0.1000	0.2000
25%	0	0.0039	0.0066	0.0165	0.0238	0.0594	0.1001	0.2503

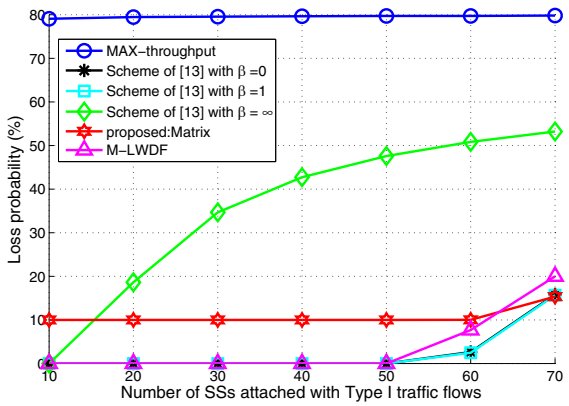

Fig. 4. Loss probabilities of SSs attached with real-time traffic flows in the first scenario.

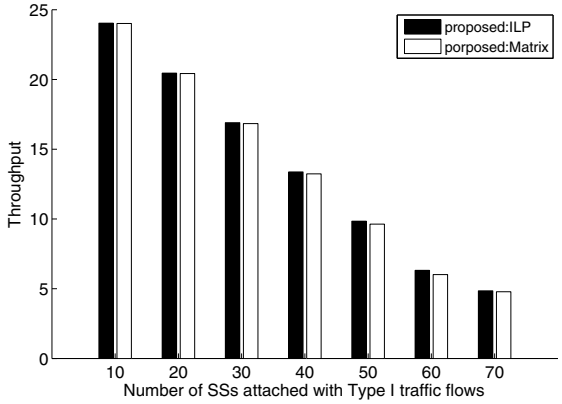

Fig. 5. Throughput comparison between proposed:ILP and proposed:Matrix schemes.

Fig. 7 shows the comparison of throughput performances of the investigated schemes which guarantee QoS of all the real-time flows in the second scenario. As one can see, our proposed:Matrix scheme outperforms M-LWDF and the scheme of [13] with $\beta = 0$ or 1. The improvement increases as the loss probability requirement increases. The reason is simply because our proposed:Matrix scheme takes loss probability requirements into consideration in calculating the minimum requested bandwidth of every real-time flow. As shown in Table III, both M-LWDF and the scheme of [13] (with $\beta = 0$ or 1) do not take full advantage of the tolerance of data loss feature of real-time flows. By controlling the actual loss probabilities close to requirements, our proposed scheme improves system throughput.

TABLE IV
NUMBER OF TYPE I AND TYPE II FLOWS WHICH MEET THEIR QOS REQUIREMENTS IN THE SECOND SCENARIO.

Number of SSs	proposed: Matrix		proposed: Matrix without pre-processor		M-LWDF	
	Type I	Type II	Type I	Type II	Type I	Type II
10	10	10	10	10	10	10
20	20	20	20	20	19	13
30	12	30	12	12	28	14
40	16	40	16	16	30	16
50	20	50	20	20	32	20

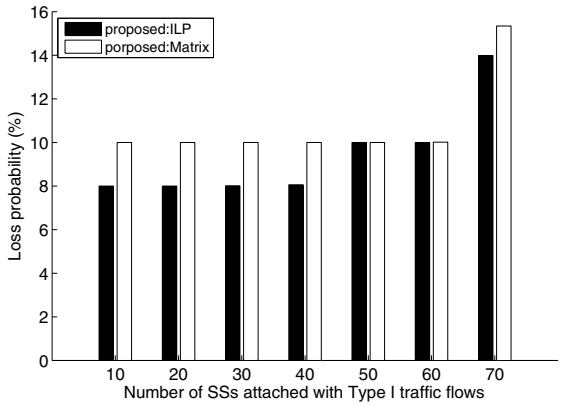

Fig. 6. Loss probability comparison between proposed:ILP and proposed:Matrix schemes.

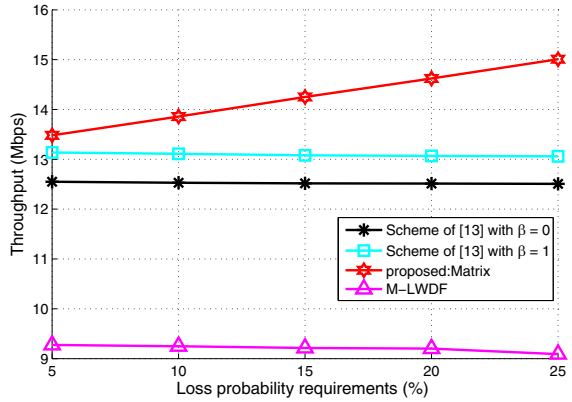

Fig. 7. Throughputs of various schemes in the second scenario.

To study the effect of pre-processor, we conduct simulations for our proposed:Matrix scheme with and without pre-processor. The results are shown in Table IV. For comparison, we also include simulation results of the M-LWDF scheme. In this table, the loss probability requirement of Type II real-time flows is chosen to be 10%. As one can see, the number of Type II flows which meet their QoS requirements with pre-processor is much larger than that without pre-processor when $|\Gamma_{RT}|$ is large. The reason is that, under the PL scheduler, the denominator of the running loss probability, i.e, $S_{n,k}[t] + L_{n,k}[t]$, is often smaller for a real-time flow with a smaller data arrival rate. As a result, a flow with a smaller data arrival rate tends to have a smaller minimum requested bandwidth and is more likely to be selected by the pre-processor. In our simulations, a flow of Type II has a smaller data arrival rate than a flow of Type I. When compared with M-LWDF, the proposed:Matrix scheme with pre-processor yields more flows which meet their QoS requirements. One interesting observation is that M-LWDF favors Type I flows. This is because Type I flows require more stringent delay bounds than Type II flows, which implies Type I flows are assigned higher priority than Type II flows when loss probability requirements are identical. We also conducted simulations for a scenario where all SSs are attached with two Type II flows. The loss probability requirement is 10% for one flow and 20% for the other. Results show that the pre-processor favors flows with 20% loss probability requirement. This is intuitively true because, under the same data arrival distribution, a flow with a larger loss probability requirement tends to have a smaller minimum requested bandwidth than one which has a smaller loss probability requirement. Owing to space limitation, we do not show these results.

VI. CONCLUSION

We have presented in this paper an efficient resource allocation scheme which tries to maximize system throughput while providing QoS support to real-time traffic flows. The basic idea of our proposed scheme is to calculate a dynamic minimum requested bandwidth for each traffic flow and use it as a constraint in an optimization problem which maximizes system throughput. The minimum requested bandwidth is a function of the pre-defined loss probability and the running loss probability. In addition, a user-level PL scheduler is proposed to determine the bandwidth share for multiple real-time flows attached to the same SS. A pre-processor is adopted to maximize the number of real-time flows attached to each SS which meet their QoS requirements, when the resource is not sufficient to provide every flow its minimum requested bandwidth. Computer simulations were conducted to evaluate the performance of our proposed scheme. Results show that the running loss probabilities of traffic flows attached to the same SS are effectively controlled to be proportional to their loss probability requirements. Besides, compared with previous designs, our proposed scheme achieves higher throughput while providing QoS support. Although we present our designs for long time average of loss probabilities, the idea can be applied to other measurements such as exponentially weighted moving average. How to design a pre-processor which meets user's need is an interesting topic which can be further studied. Evaluation of the impact to user perception of satisfaction for various performance measurements is another potential further research topic.

APPENDIX A
PROOFS OF LEMMAS AND THEOREMS

Proof of Lemma 1: Lemma 1 is obviously true for $P_{n,k}^{\min}[t] \leq P_{n,k} \leq P_{n,k}^{\max}[t]$ because, in this case, we have $P_{n,k}^{*}[t]-P_{n,k}=0$. For $P_{n,k} > P_{n,k}^{\max}[t]$, it holds that

$$|P_{n,k}^{*}[t]-P_{n,k}| = P_{n,k} - \frac{L_{n,k}[t-1]+Q_{n,k}^{1}[t]}{S_{n,k}[t-1]+L_{n,k}[t-1]+Q_{n,k}^{1}[t]} \leq P_{n,k} - \frac{L_{n,k}[t-1]+(Q_{n,k}^{1}[t]-R_{n,k}[t])^{+}}{S_{n,k}[t-1]+L_{n,k}[t-1]+\max(R_{n,k}[t],Q_{n,k}^{1}[t])}.$$

since $R_{n,k}[t] \geq 0$. Therefore, Lemma 1 is true for $P_{n,k} > P_{n,k}^{\max}[t]$. For $P_{n,k} < P_{n,k}^{\min}[t]$, we have

$$|P_{n,k}^{*}[t]-P_{n,k}| = \frac{L_{n,k}[t-1]}{S_{n,k}[t-1]+L_{n,k}[t-1]+Q_{n,k}[t]} - P_{n,k} \leq \frac{L_{n,k}[t-1]+(Q_{n,k}^{1}[t]-R_{n,k}[t])^{+}}{S_{n,k}[t-1]+L_{n,k}[t-1]+\max(R_{n,k}[t],Q_{n,k}^{1}[t])} - P_{n,k}.$$

since $R_{n,k}[t] \leq Q_{n,k}[t]$. This completes the proof of Lemma 1.

Proof of Lemma 2: Let $R_{n,k}[t]$ and $P_{n,k}[t]$ be, respectively, the bandwidth allocated to and the resulting running loss probability of $f_{n,k}$ under our proposed PL scheduler. Further, let $R'_{n,k}[t]$ and $P'_{n,k}[t]$ be the same variables under some other scheduler. Assume that $\phi = \arg\max_{1\leq k\leq K_n} \frac{P_{n,k}[t]}{P_{n,k}}$. We shall prove $\frac{P_{n,\phi}[t]}{P_{n,\phi}} \leq \max_{1\leq k\leq K_n} \frac{P'_{n,k}[t]}{P_{n,k}}$.

Let U_{Z}, U_{P}, and U_{A} be the three sets such that flow $f_{n,k}$ is contained in U_{Z}, U_{P}, or U_{A} iff $R_{n,k}[t]=0$, $0<R_{n,k}[t]<Q_{n,k}[t]$, or $R_{n,k}[t]=Q_{n,k}[t]$, under the proposed PL scheduler. Assume that $U_{\mathrm{A}}=\emptyset$. Since $R_n[t]>0$, it must hold that $\phi \in U_{\mathrm{P}}$. If $\frac{P_{n,\phi}[t]}{P_{n,\phi}} > \frac{P'_{n,\phi}[t]}{P_{n,\phi}}$, meaning that $R_{n,\phi}[t] < R'_{n,\phi}[t]$, there must exist $f_{n,k} \in U_{\mathrm{P}}$ such that $R_{n,k}[t] > R'_{n,k}[t]$. Otherwise, equation (8) is violated. Since $\frac{P'_{n,k}[t]}{P_{n,k}} > \frac{P_{n,k}[t]}{P_{n,k}} = \frac{P_{n,\phi}[t]}{P_{n,\phi}}$, Lemma 2 is true for this case. Consider the case $U_{\mathrm{A}} \neq \emptyset$. The proposed PL scheduler allocates $R_{n,i}[t]=Q_{n,i}[t]$ to all $f_{n,i} \in U_{\mathrm{A}}$, which implies $f_{n,\phi}$ is in U_{A} or can be selected from U_{A}, according to equation (7). Consequently, Lemma 2 is true because $R_{n,\phi}[t] \geq R'_{n,\phi}[t]$, which implies $\frac{P_{n,\phi}[t]}{P_{n,\phi}} \leq \frac{P'_{n,\phi}[t]}{P_{n,\phi}}$.

Proof of Theorem 3: Assume that there exists a scheduler which can guarantee the loss probability requirements of all the K_n traffic flows. In other words, it holds that $\frac{P'_{n,k}[t]}{P_{n,k}} \leq 1$, $1 \leq k \leq K_n$, where $P'_{n,k}[t]$ is the loss probability of flow $f_{n,k}$ at the end of the t^{th} frame, under the considered scheduler. Let $P_{n,k}[t]$ be the loss probability of flow $f_{n,k}$ at the end of the t^{th} frame, under the PL scheduler. According to Lemma 2, we have $\frac{P_{n,k}[t]}{P_{n,k}} \leq \max_{1\leq i\leq K_n} \frac{P'_{n,i}[t]}{P_{n,i}} \leq 1$, $1 \leq k \leq K_n$, and, therefore, Theorem 3 is true.

Proof of Lemma 4: Lemma 4 can be easily verified with the calculation results shown in Table I.

Proof of Theorem 5: We prove Theorem 5 for $\Delta R_n[t] \geq 0$. The other case can be proved similarly. Let V_{Z}, V_{P} and V_{A} be three sets such that $f_{n,k}$ is in V_{Z}, V_{P}, or V_{A} iff $R_{n,k}^{*}[t]=0$, $0<R_{n,k}^{*}[t]<Q_{n,k}[t]$, or $R_{n,k}^{*}[t]=Q_{n,k}[t]$, respectively. Similarly, $f_{n,k}$ is in U_{Z}, U_{P}, or U_{A} iff $R_{n,k}[t]=0$, $0<R_{n,k}[t]<Q_{n,k}[t]$, or $R_{n,k}[t]=Q_{n,k}[t]$, respectively. Recall that equations (7) and (8) are satisfied under the PL scheduler.

Assume that $\Delta R_{n,i}[t] < 0$ for some flow $f_{n,i}$. Since $\Delta R_n[t] \geq 0$, there must be some other $f_{n,j}$ with $\Delta R_{n,j}[t] > 0$. The assumption $\Delta R_{n,i}[t] < 0$ implies $f_{n,i} \in V_{\mathrm{P}} \cup V_{\mathrm{A}}$ and $\Delta R_{n,j}[t] > 0$ implies $f_{n,j} \in V_{\mathrm{Z}} \cup V_{\mathrm{P}}$. From Lemma 4, we have $\frac{P_{n,i}^{*}[t]}{P_{n,i}} \geq \frac{P_{n,j}^{*}[t]}{P_{n,j}}$. The assumption $\Delta R_{n,i}[t] < 0$ also implies $f_{n,i} \in U_{\mathrm{Z}} \cup U_{\mathrm{P}}$ and $\Delta R_{n,j}[t] > 0$ implies $f_{n,j} \in U_{\mathrm{P}} \cup U_{\mathrm{A}}$. According to equation (7), we have $\frac{P_{n,i}[t]}{P_{n,i}} \leq \frac{P_{n,j}[t]}{P_{n,j}}$, a contradiction, because $P_{n,k}[t]$ is a strictly decreasing function of $R_{n,k}[t]$ for $0 \leq R_{n,k}[t] \leq Q_{n,k}[t]$, which together with $\frac{P_{n,i}^{*}[t]}{P_{n,i}} \geq \frac{P_{n,j}^{*}[t]}{P_{n,j}}$, $\Delta R_{n,i}[t] < 0$, and $\Delta R_{n,j}[t] > 0$ imply $\frac{P_{n,i}[t]}{P_{n,i}} > \frac{P_{n,j}[t]}{P_{n,j}}$. This proves Theorem 5.

APPENDIX B
DERIVATION OF $P_n^{\mathrm{F}}[t]$

Given $P_n^{\mathrm{F}}[t]$, one can compute $h_{n,k}(P_n^{\mathrm{F}}[t]\cdot P_{n,k};t)$ based on equation (9) for any $f_{n,k} \in U_{\mathrm{P1}} \cup U_{\mathrm{P2}}$. Substituting $h_{n,k}(P_n^{\mathrm{F}}[t]\cdot P_{n,k};t)$ into $\sum_{f_{n,k}\in U_{\mathrm{P1}}\cup U_{\mathrm{P2}}} h_{n,k}(P_n^{\mathrm{F}}[t]\cdot P_{n,k};t) = R_n[t] - \sum_{f_{n,k}\in U_{\mathrm{A}}} Q_{n,k}[t]$, we get $A\cdot(P_n^{\mathrm{F}}[t])^2 + B\cdot(P_n^{\mathrm{F}}[t]) + C = 0$, where $A = \sum_{f_{n,k}\in U_{\mathrm{P1}}} P_{n,k}\cdot(S_{n,k}[t-1]+L_{n,k}[t-1]+Q_{n,k}^{1}[t])$, $B = R_n[t] + \sum_{f_{n,k}\in U_{\mathrm{P2}}}(S_{n,k}[t-1]+L_{n,k}[t-1]) - \sum_{f_{n,k}\in U_{\mathrm{A}}} Q_{n,k}[t] - \sum_{f_{n,k}\in U_{\mathrm{P1}}}(L_{n,k}[t-1]+Q_{n,k}^{1}[t])$ and $C = -\sum_{f_{n,k}\in U_{\mathrm{P2}}} \frac{L_{n,k}[t-1]}{P_{n,k}}$. If $U_{\mathrm{P1}}=\emptyset$, which implies $A=0$, $P_n^{\mathrm{F}}[t]$ can be obtained by $P_n^{\mathrm{F}}[t] = -\frac{C}{B}$. Assume that $A \neq 0$. In this case, we have $P_n^{\mathrm{F}}[t] = \frac{-B+\sqrt{B^2-4AC}}{2A}$ because $B^2-4AC \geq B^2$ and $P_n^{\mathrm{F}}[t]$ must be non-negative.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable comments which lead to improvement of the paper.

REFERENCES

[1] IEEE Standard for Local and Metropolitan Area Networks-Part 16: Air Interface for Fixed Broadband Wireless Access Systems, IEEE Std. 802.16-2009, May 2009.

[2] E. Dahlman, S. Parkvall, J. Skold, and P. Beming, *3G HSPA and LTE for Mobile Broadband*. Academic, 2007.

[3] M. Kaneko, P. Popovski, and J. Dahl, "Proportional fairness in multi-carrier system with multi-slot frames: upper bound and user multiplexing algorithms," *IEEE Trans. Wireless Commun.*, vol. 7, no. 1, pp. 22–26, Jan. 2008.

[4] N. Ruangchaijatupon and Y. Ji, "Simple proportional fairness scheduling for OFDMA-based wireless systems," in *Proc. 2008 IEEE WCNC*, pp. 1593–1597.

[5] N. Ruangchaijatupon and Y. Ji, "OFDMA resource allocation based on traffic class-oriented optimization," *IEICE Trans. Commun.*, vol. E92-B, no.1, pp. 93–101, Jan. 2009.

[6] N. Ruangchaijatupon and Y. Ji, "Integrated approach to proportional fair resource allocation for multiclass services in an OFDMA system," in *Proc. 2009 IEEE GLOBECOM*.

[7] D. S. W. Hui, V. K. N. Lau, and W. H. Lam, "Cross-layer design for OFDMA wireless systems with heterogeneous delay requirements," *IEEE Trans. Wireless Commun.*, vol. 6, no. 8, pp. 2872–2880, Aug. 2007.

[8] J. Jang and K. B. Lee, "Transmit power adaptation for multiuser OFDM system," *IEEE J. Sel. Areas Commun.*, vol. 21, no. 12, pp. 171–178, Feb. 2003.

[9] S. Shakkottai and A. L. Stolyar, "A study of scheduling algorithms for a mixture of real and non-real time data in HDR," Bell Labs Tech. Memo., Aug. 2000.

[10] M. Andrews, K. Kumaran, K. Ramanan, A. L. Stolyar, P. Whiting, and R. Vijayakumar, "Providing quality of service over a shared wireless link," *IEEE Commun. Mag.*, vol. 39, no. 2, pp. 150–154, Feb. 2001.

[11] A. K. F. Khattab and K. M. F. Elsayed, "Opportunistic scheduling of delay sensitive traffic in OFDMA-based networks," in *Proc. 2006 IEEE WOWMOM*, pp. 109–114.

[12] X. Zhu, J. Huo, C. Xu, and W. Ding, "QoS-guaranteed scheduling and resource allocation algorithm for IEEE 802.16 OFDMA system," in *Proc. 2008 IEEE ICC*, pp. 3463–3468.

[13] Y. Kim, K. Son, and S. Chong, "QoS scheduling for heterogeneous traffic in OFDMA-based wireless systems," in *Proc. 2009 IEEE GLOBECOM*.
[14] R. Chipalkatti, J. Jurose, and D. Towsley, "Scheduling policies for real-time and non-real-time traffic in a statistical multiplexer," in *Proc. 1989 IEEE INFOCOM*, pp. 774–783.
[15] R. Yang, C. Yuan, and K. Yang, "Cross layer resource allocation of delay sensitive service in OFDMA wireless systems," in *Proc. 2008 IEEE ICCSC*, pp. 862–866.
[16] V. Huang and W. Zhuang, "QoS-oriented packet scheduling for wireless multimedia CDMA communications," *IEEE Trans. Mobile Comput.*, vol. 3, no. 1, pp. 73–85, Jan. 2004.
[17] A. Frank, "On Kuhn's Hungarian method—a tribute from Hungary," *Naval Research Logistics*, vol. 52, no. 1, pp. 2–5, Dec. 2005.
[18] J. Mo and J. Walrand, "Fair end-to-end window-based congestion control," *IEEE/ACM Trans. Netw.*, vol. 8, no. 5, pp. 556–567, Oct. 2000.
[19] J. E. Beasley, *Advances in Linear and Integer Programming*. Oxford Science, 1996.
[20] A. Schrijver, *Theory of Linear and Integer Programming*. Wiley, 1986.
[21] "MPEG-4 and H.263 video traces for network performance evaluation," Oct. 2006. Available: http://www.tkn.tu-berlin.de/research/trace/trace.html

Tsern-Huei Lee (S'86-M'87-SM'98) received the B.S. degree from National Taiwan University, Taipei, Taiwan, the M.S. degree from the University of California, Santa Barbara, and the Ph.D. degree from the University of Southern California, Los Angeles, in 1981, 1983, and 1987, respectively, all in electrical engineering.

Since 1987, he has been a member of the faculty of National Chiao Tung University, Hsinchu, Taiwan, where he is a professor in the Department of Electrical Engineering. He received an Outstanding Paper Award from the Institute of Chinese Engineers in 1991. During the past years, he has served as a consultant to various companies to develop large scale QoS-enabled frame-based switches/routers, integrated access devices, and unified threat management Internet appliances. His current research interests are in communication protocols, broadband switching systems, traffic management, wireless communications, and network security.

Yu-Wen Huang (S'07) was born in Gangshan District, Kaohsiung City, Taiwan, in 1982. He received the B.S. and M.S. degrees in communication engineering from National Chiao Tung University, Hsinchu, Taiwan, in 2004 and 2006, respectively. Currently, he is pursing his Ph.D. degree at the same university. His current research interests include resource allocation, power management, and communication protocols in wireless networks.