Performance Analysis of Wireless Opportunistic Schedulers using Stochastic Petri Nets

Lei Lei, Chuang Lin, *Senior Member, IEEE*, Jun Cai, *Member, IEEE*, and Xuemin (Sherman) Shen, *Fellow, IEEE*

***Abstract*—In this paper, performance of wireless opportunistic schedulers in multiuser systems is studied under a dynamic data arrival setting. Different from the previous studies which mostly focus on the network stability and the worst case scenarios, we emphasize on the average performance of wireless opportunistic schedulers. We first develop a framework based on Markov queueing model and then analyze it by applying decomposition and iteration techniques in the stochastic Petri nets (SPN). Since the size of the state space in our analytical model is small, the proposed framework shows an improved efficiency in computational complexity. Based on the established analytical model, performance of both opportunistic and non-opportunistic schedulers are studied and compared in terms of average queue length, mean throughput, average delay and dropping probability. Analytical results demonstrate that the multiuser diversity effect as observed in the infinite backlog scenario is only valid in the heavy traffic regime. The performance of the opportunistic schedulers in the light traffic regime is worse than that of the non-opportunistic round-robin scheduler, and becomes worse especially with the increase of the number of users. Simulations are also performed to verify the accuracy of the analytical results.**

***Index Terms*—Opportunistic scheduling, MMDP, stochastic Petri nets.**

I. Introduction

In wireless systems, channel conditions are inherently time-varying due to the existence of fading and shadowing effects. Moreover, since different wireless users may experience independent channel variations, the event that there exist users with strong channel gains at any time instant occurs with high probability, which is referred to as *multiuser diversity*. In order to exploit such channel fluctuation and multiuser diversity for throughput improvement in wireless systems, opportunistic scheduling (OS) has been appeared. Here, the "opportunistic" means the mechanism can take advantages of the favorable channel conditions in resource allocation. So far, the concept of OS has been widely applied in the third-generation (3G) wireless systems such as Code Division Multiple Access (CDMA) 2000 1xEV-DO [1] and Universal Mobile Telecommunications System (UMTS) High Speed Downlink Packet Access (HSPDA) [2]. While all OS algorithms take into account the channel state information, some of them may also consider the queueing status of users. In this paper, we use "channel/queue-aware" and "channel-aware" to indicate respectively whether queue state information is considered or not in OS algorithms.

Investigation of OS algorithms in wireless systems has been carried out at both packet [3]–[9] and flow levels [10], [11]. The packet level analysis focuses on the evaluation of OS algorithms with saturated (i.e., each user always has data to transmit) or dynamic packet arrivals and time variant channel conditions, while the flow level one insists on the systems with dynamic user populations. Compared to the flow level analysis, the packet level analysis shows advantages in involving more realistic channel models, which include the simplest memoryless on-off channel, the two-state Gilbert-Elliot channel [12], [13], and the more complicated and accurate finite state Markov channel (FSMC) [14]. In this paper, our emphasis will be on the packet level analysis.

Previous work on packet-level analysis considering dynamic packet arrivals focused on network stability, i.e., the queue occupancy can be bounded whenever feasible [4], [5], with both channel-aware and channel/queue-aware OS algorithms. The typical observations are that most channel-aware OS algorithms, e.g., the proportional fair (PF) algorithm, are unstable [4] and a quadratic Lyapunov function argument can be used to prove the stability for channel/queue-aware algorithms [5]. Recently, several research work also studied the statistical worst case performance of OS algorithms using effective bandwidth and its related concepts [6]–[8]. In [7], a formula is provided to approximate the tail distribution of packet delay for the greedy channel-aware and round-robin algorithms under the FSMC model. In [8], the maximum throughput for the channel-aware and the channel/queue-aware algorithms are estimated under the constraints that the tail distribution of the queue length cannot exceed a certain threshold, where the wireless channel condition or variation is assumed to be a memoryless on-off process.

While all these work on the network stability and statistical worst case performance provide important insights into the queueing behavior of the OS algorithms, the discussions on some average performance, such as average delay and average throughput, are missed because of the difficulties in deriving the steady-state distribution of the queue states. This

Manuscript received April 15, 2008; revised July 20, 2008; accepted November 9, 2008. The associate editor coordinating the review of this paper and approving it for publication was W. Liao.

This work was supported by the National Nature Science Foundation of China (No. 60702009), and the National Grand Fundamental Research Program of China(973) (No.2009CB320504).

L. Lei and C. Lin are with the Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China (e-mail: {leilei, clin}@csnet1.cs.tsinghua.edu.cn).

J. Cai is with the Department of Electrical and Computer Engineering, University of Manitoba, Winnipeg, Manitoba, Canada R3T 5V6 (e-mail: jcai@ee.umanitoba.ca).

X. Shen is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1 (e-mail: xshen@bbcr.uwaterloo.ca).

Digital Object Identifier 10.1109/TWC.2009.080523

situation becomes even harder in wireless systems due to the time-varying channel conditions. In [15], a two-dimensional Markov model for computing the steady-state distribution in single user systems is proposed, where two dimensions represent channel and queue states, respectively. However, such analytical model cannot be directly extended to multiuser systems since the state space of the Markov model grows exponentially as the number of users increases. Thus, designing a practically implementable analysis model is critical for analyzing the performance of OS algorithms in multiuser wireless systems.

In this paper, a new analytical framework is proposed for multiuser systems, which can be used for studying the performance of different wireless schedulers in terms of average queue length, mean throughput, average delay and dropping probability. The wireless schedulers under consideration include not only opportunistic schedulers using channel-aware and channel/queue-aware algorithms, but also non-opportunistic ones using round-robin and queue-aware algorithms. Specifically, by characterizing the service process from the finite state Markov channel (FSMC) as a Markov modulated deterministic process (MMDP), the wireless downlink for the multiuser system is first modeled as an M/MMDP/1/K queueing system. Then, a deterministic & stochastic Petri nets (DSPNs) model is constructed, where different scheduling algorithms can be expressed by model parameters, such as the enabling predicates and random switches. By applying the model decomposition technique in stochastic Petri nets (SPNs), the multiuser system is decomposed into multiple single user subsystems with inter-correlated service rates. To facilitate the analysis for each subsystem separately, some approximation methods including the replacement of the instantaneous service rates by steady-state average ones and a fixed-point iteration method, are introduced. The proposed analytical framework significantly reduces the state space of the Markovian system model in analysis and shows good performance in scalability. Numerical results show that 1) the channel-aware algorithm performs better than the round robin algorithm only in heavy traffic regime; 2) the scheduling gain of the channel-aware algorithm increases with the number of users only when the traffic load per user is heavy; and 3) the channel/queue-aware algorithm outperforms the channel-aware algorithm in light traffic regime and converges to the channel-aware algorithm in heavy traffic regime.

Notice that the selection of stochastic Petri nets approach results from the facts that 1) it provides an intuitive and efficient way in describing the multiuser system, especially facilitating the inclusion of different scheduling strategies; and 2) there exist a set of well-developed techniques for stochastic Petri nets, which can decompose the original complex model into simple subsystems and provide iteration methods for performance approximation.

The remainder of the paper is organized as follows. In Section II, a DSPN model for the multiuser system is formulated. In Section III, the single user system is first analyzed, and then the model decomposition and iteration method in SPN are discussed to approximate the multiuser system performance. In Section IV, both analytical and simulation results are presented to compare the performance of different scheduling algorithms. Section V concludes this paper.

II. The DSPN Model Formulation

In this section, a general framework is introduced for modelling and analyzing a multiuser system using deterministic and stochastic Petri nets (DSPNs). For basic concepts of stochastic Petri nets (SPNs), please refer to Appendix A.

A. The M/MMDP/1/K Queueing System

Consider the downlink of a cellular wireless network, where a base station (BS) transmits data to N $(N \geq 1)$ mobile users. The BS maintains a separate data buffer for each mobile user and each buffer has a finite capacity of $K < \infty$ bits. For each data buffer n, the data arrives according to Poisson distribution with average rate λ_n bits/sec. The transmission in the time is slot-by-slot based and each slot has an equal length ΔT. In each time slot the BS can transmit data to one user only. It is assumed that all channel conditions are available at the BS so that the adaptive modulation and coding (AMC) algorithms can be applied. The wireless channel for each mobile user is modeled as an independent finite-state Markov channel (FSMC) with total L states. Each state of FSMC corresponds to one non-overlapping consecutive SNR region and a fixed transmission rate determined by the AMC algorithm. During each time slot, the channel stays at the same state.

The described system model can be formulated by an M/MMDP/1/K queueing system as follows. The defined fluid queueing system consists of a finite number, N, of input flows indexed by $n = 1, 2, \ldots, N$, and one server. For any n, there is a finite, irreducible, and continuous-time Markov chain $\mathcal{H}_n(\tau)$ with total L server states, which corresponds to the L channel states of the FSMC model. The transition rates of the Markov chain depends on its channel fading speed and is not necessarily identical for all input flows, but the transitions of different Markov chains are independent. Associated with the l-th ($l \in \{1, \ldots, L\}$) state of $\mathcal{H}_n(\tau)$ is a fixed service rate $R_{n,l}$ bits/sec, which is a non-negative integer. If at time τ the server is allocated to flow n with $\mathcal{H}_n(\tau)$ in the l-th state, the queue n is served at a deterministic rate $R_{n,l}$, i.e., the user is served according to an L-state MMDP.

B. The DSPN Model

The M/MMDP/1/K queueing system designed for the multiuser wireless downlink can be equivalently modeled as a DSPN by following the similar procedure as shown in [16]. The modeled DSPN consists of a SPN for representing service processes and a DSPN for representing resource sharing. The SPN, as shown in Fig. 1(a), is further composed of N subnets and each subnet n corresponds to the L-state Markov modulated service process of user n. Each subnet is described by places ($\{H_{nl}\}_{l=1}^{L}$) and transitions ($\{tu_{nl}\}_{l=1}^{L-1}$ and $\{td_{nl}\}_{l=1}^{L-1}$). The DSPN, as shown in Fig. 1(b), models the resource sharing relationship of the multiuser system and can be characterized by places ($\{Q_n\}_{n=1}^{N}$, $\{w_n\}_{n=1}^{N}$ and r) and transitions ($\{c_n\}_{n=1}^{N}$, $\{d_n\}_{n=1}^{N}$ and $\{s_n\}_{n=1}^{N}$). The meanings of all the places and transitions are described as follows.

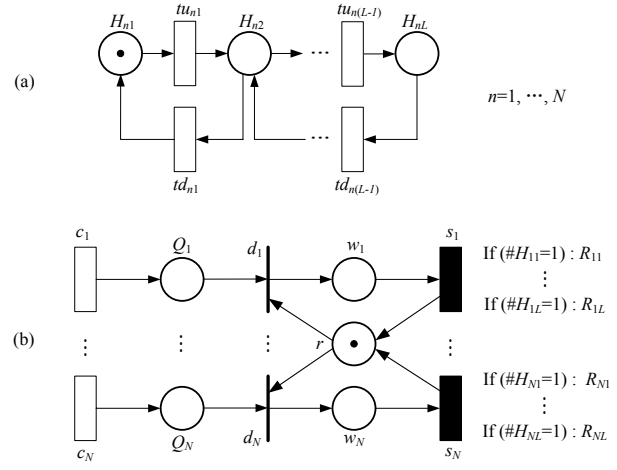


Fig. 1. The DSPN model for multiuser wireless downlink.

Q_n: a place for the queue state of user n.

H_{nl}: a place for the l-th server state of user n.

c_n: an exponentially-distributed timed transition denoting new bit arrivals from user n, with firing rate λ_n. When it fires, one bit of data arrives at the queue place Q_n.

d_n: an immediate transition denoting the execution of scheduling strategy. Different scheduling strategies are expressed using different enabling predicates and random switches. The details will be discussed in the next subsection.

s_n: a deterministic timed transition for service process. When it fires, one bit of data is transmitted from the queue place Q_n. Its firing rate μ_n depends on the marking of the places $\{H_{nl}\}_{l=1}^{L}$, i.e., if $M(H_{nl}) = 1$, then

$$\mu_n = R_{n,l}, \quad l = 1, \ldots, L$$

where $M(\cdot)$ is a mapping function from a place to the number of tokens assigned to it, and $M(H_{nl})$ is either 1 or 0, which represents whether user n is in its l-th server state or not.

tu_{nl}, td_{nl}: exponentially-distributed timed transitions for the server state transitions of user n. The firing rates of tu_{nl} and td_{nl} equal $p^n_{l,l+1}/\Delta T$ and $p^n_{l+1,l}/\Delta T$, which are determined by (35) and (36) in Appendix B, respectively. When tu_{nl} (td_{nl}) fires, the server state transits from l $(l+1)$ to $l+1$ (l).

C. Scheduling Strategies

The performance of the multiuser system depends on the scheduling strategies applied. In the defined DSPN model, different scheduling strategies can be described by different enabling predicates and random switches of the immediate transition d_n. The enabling predicate specifies the condition under which user n is an eligible candidate for data transmission, while the random switch indicates the probability that user n will be selected for service.

1) Round-robin (RR) algorithm: In round-robin algorithm, the scheduler polls queues for service in a cyclic order independent of the wireless channel conditions. Therefore, for the RR algorithm, the enabling predicate y_n of d_n is

$$y_n : (M(Q_n) > 0) \tag{1}$$

and the random switch $g_n(\mathbf{M})$ of d_n is

$$g_n(\mathbf{M}) = \begin{cases} 1/\|\mathrm{RR}(\mathbf{M})\|, & \text{if } n \in \mathrm{RR}(\mathbf{M}) \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where $\mathbf{M}$ is a vector representing the number of tokens in each place of the DSPN model, which include $\{Q_n, \{H_{nl}\}_{l=1}^{L}\}_{n=1}^{N}$ and $\mathrm{RR}(\mathbf{M}) = \{i \mid M(Q_i) > 0\}$.

2) Channel-aware (CA) algorithms: The channel-aware algorithms aim at improving the scheduling performance by incorporating channel state information. Two typical CA algorithms are greedy algorithm and PF algorithm.

Greedy algorithm is also referred to as the Max-SNR algorithm. The algorithm always picks the user with the best SNR for transmission, or equivalently, the best transmission data rate is guaranteed at every scheduling instant. For the greedy algorithm, the enabling predicates and random switches can be found as

$$y_n : (M(Q_n) > 0) \wedge (\forall i \neq n, \mu_i \leq \mu_n) \vee (\forall i \neq n, M(Q_i) = 0), \tag{3}$$

$$g_n(\mathbf{M}) = \begin{cases} 1/\|\mathrm{CA}(\mathbf{M})\|, & \text{if } n \in \mathrm{CA}(\mathbf{M}) \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where $\mathrm{CA}(\mathbf{M}) = \{i \mid \mu_i = \max(\mu_1, \ldots, \mu_N), M(Q_i) > 0\}$, and the operators $\wedge$ and $\vee$ represent "logical and" and "logical or", respectively.

PF algorithm, on the other hand, picks the user in each time slot among all backlogged users in the system which has the best transmission data rate normalized by the average throughput it has already received so far. Obviously, if all users experience statistically identical channels, there is no difference between the greedy algorithm and the PF algorithm in the long run, i.e., equations (3) and (4) can also be used to describe the behavior of the PF algorithm. In fact, the enabling predication and random switches defined in (3) and (4) can be easily extended to describe the PF algorithm in more general scenarios by replacing the actual transmission rate μ_i, $i = 1, \ldots N$, with the normalized transmission rate $\mu_i/\overline{\mu}_i$, $i = 1, \ldots N$, where $\overline{\mu}_i$, $i = 1, \ldots N$, is the average transmission rate of user i .

3) Queue-aware (QA) algorithms: The queue-aware algorithms are actually not 'opportunistic' in the sense that they do not consider the channel state information in scheduling. Although they are more commonly used in wireline networks, in this paper, for the purpose of comparison with the channel/queue-aware algorithms, a simple QA algorithm, which always selects the user with the largest queue length, is studied. For the QA algorithm, we have

$$y_n : (M(Q_n) > 0) \wedge (\forall i \neq n, M(Q_i) \leq M(Q_n)), \tag{5}$$

$$g_n(\mathbf{M}) = \begin{cases} 1/\|\mathrm{QA}(\mathbf{M})\|, & \text{if } n \in \mathrm{QA}(\mathbf{M}) \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

where $\mathrm{QA}(\mathbf{M}) = \{i \mid M(Q_i) = \max(M(Q_1), \ldots, M(Q_N)), M(Q_i) > 0\}$.

$$\mathbf{P}_n = \begin{pmatrix} p^n_{1,1}\boldsymbol{\nu}_{n,1} & p^n_{1,2}\boldsymbol{\nu}_{n,1} & & & \\ p^n_{2,1}\boldsymbol{\nu}_{n,2} & p^n_{2,2}\boldsymbol{\nu}_{n,2} & p^n_{2,3}\boldsymbol{\nu}_{n,2} & & \\ & p^n_{3,2}\boldsymbol{\nu}_{n,3} & p^n_{3,3}\boldsymbol{\nu}_{n,3} & p^n_{3,4}\boldsymbol{\nu}_{n,3} & \\ & \ddots & \ddots & \ddots & \\ & & p^n_{L-1,L-2}\boldsymbol{\nu}_{n,L-1} & p^n_{L-1,L-1}\boldsymbol{\nu}_{n,L-1} & p^n_{L-1,L}\boldsymbol{\nu}_{n,L-1} \\ & & & p^n_{L-1,L}\boldsymbol{\nu}_{n,L} & p^n_{L,L}\boldsymbol{\nu}_{n,L} \end{pmatrix}. \quad (12)$$

4) Channel/queue-aware (CQA) algorithms: The systems with CA algorithms are ordinarily not stable, since the CA algorithms do not take into account the queue length information, and therefore do not know how to react when one queue starts getting too large. On the contrary, the CQA algorithms are a class of schedulers, which consider the information about both channel and queue occupancy. A simple CQA algorithm is called the Max-Weight algorithm, where the user with maximum product of queue length and transmission data rate is served. For Max-Weight algorithm, we have

$$y_n : (M(Q_n) > 0) \wedge (\forall i \neq n, M(Q_i)\mu_i \leq M(Q_n)\mu_n), \quad (7)$$

$$g_n(\mathbf{M}) = \begin{cases} 1/\|\mathrm{CQA}(\mathbf{M})\|, & \text{if } n \in \mathrm{CQA}(\mathbf{M}) \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

where $\mathrm{CQA}(\mathbf{M}) = \{i \mid \mu_i M(Q_i) = \max(\mu_1 M(Q_1), \ldots, \mu_N M(Q_N)), M(Q_i) > 0\}$.

III. Model Solution and Performance Analysis

In this section, analysis methods are introduced to find the solutions of the system derived in Section II.

A. Single User System

We first consider the simplest case where there is only one mobile user, e.g., user n, in the system. Since previously defined M/MMDP/1/K queue underlying the DSPN model does not hold Markovian property, the direct analysis for such a single user system is still difficult. In order to simplify the analysis, we introduce the following embedded Markov chain [17]. Let $\mathcal{H}_n(\tau)$ be the Markovian server state process and $\mathcal{Q}_n(\tau)$ be the length of the queue occupancy at any time instant τ. Since the channel state keeps unchanged during each time slot, we can discretize the random processes at every ΔT interval and define $\mathcal{H}_{n,t} := \mathcal{H}_n(t \times \Delta T)$ and $\mathcal{Q}_{n,t} := \mathcal{Q}_n(t \times \Delta T)$. After discretization, the server and queue states are assumed to change only at the sample instant. Obviously, the two-dimensional embedded Markov chain $\{(\mathcal{H}_{n,t}, \mathcal{Q}_{n,t}), t = 0, 1, ...\}$ can accurately represent the system behavior. A similar discrete-time Markov chain has also been introduced in [15] for a single user wireless LAN (WLAN) system.

Let $p^n_{(l,k),(m,h)}$ be the transition probability from state (l,k) to state (m,h) of the embedded Markov chain. Then,

$$\begin{aligned} p^n_{(l,k),(m,h)} &= \mathbb{P}\{\mathcal{Q}_{n,t+1} = h | \mathcal{H}_{n,t} = l, \mathcal{Q}_{n,t} = k\} p^n_{l,m} \\ &= \nu^{n,l}_{k,h} p^n_{l,m} \end{aligned} \quad (9)$$

where $\nu^{n,l}_{k,h} = \mathbb{P}\{\mathcal{Q}_{n,t+1} = h | \mathcal{H}_{n,t} = l, \mathcal{Q}_{n,t} = k\}$ and $p^n_{l,m}$ denotes the transition probability of the FSMC from state l to m. The determination of $p^n_{l,m}$ under Rayleigh fading channel is given in Appendix B.

Let $A_{n,t}$ denote the number of bits arrived during the t-th time slot. Since the queueing process evolves following

$$\mathcal{Q}_{n,t+1} = \min[K, \max[0, \mathcal{Q}_{n,t} - R_{n,l}\Delta T] + A_{n,t}], \quad (10)$$

we have

$$\nu^{n,l}_{k,h} = \begin{cases} \mathbb{P}(A_{n,t} = h - k + R_{n,l}\Delta T) & k \geq R_{n,l}\Delta T, h \neq K \\ \mathbb{P}(A_{n,t} = h) & k < R_{n,l}\Delta T, h \neq K \\ \mathbb{P}(A_{n,t} \geq K - k + R_{n,l}\Delta T) & k \geq R_{n,l}\Delta T, h = K \\ \mathbb{P}(A_{n,t} \geq K) & k < R_{n,l}\Delta T, h = K \end{cases} \quad (11)$$

where $\mathbb{P}(A_{n,t} = u) = \frac{(\lambda_n \Delta T)^u}{u!} e^{-\lambda_n \Delta T}$ due to Poisson assumptions.

Define matrices $\boldsymbol{\nu}_{n,l} = [\nu^{n,l}_{k,h}]$ and $\mathbf{P}_n = [p^n_{(l,k),(m,h)}]$. We can partition $\mathbf{P}_n$ into blocks, each of which is a $(K+1) \times (K+1)$ matrix as shown in (12) at the top of the page. Note that except for the main, upper and lower diagonals, all other blocks are zeros.

Define the steady-state probability $\pi^n_{l,h} \equiv \lim_{t\to\infty} \mathbb{P}\{\mathcal{H}_{n,t} = l, \mathcal{Q}_{n,t} = h\}$ and the vector $\boldsymbol{\pi}_n = (\pi^n_{1,0}, \pi^n_{1,1}, \ldots, \pi^n_{1,K}, \ldots, \pi^n_{L,0}, \pi^n_{L,1}, \ldots, \pi^n_{L,K})$. Then, the stationary distribution of the ergodic process $\{(\mathcal{H}_{n,t}, \mathcal{Q}_{n,t})\}$ can be uniquely determined from the balance equations

$$\boldsymbol{\pi}_n = \boldsymbol{\pi}_n \mathbf{P}_n, \ \boldsymbol{\pi}_n \mathbf{e} = 1 \quad (13)$$

where $\mathbf{e}$ is the unity vector of dimension $L \times (K+1)$ and $\boldsymbol{\pi}_n$ can be derived as the normalized left eigenvector of $\mathbf{P}_n$ corresponding to eigenvalue 1. Given $\boldsymbol{\pi}_n$, the performance metrics such as the average queue length, the mean throughput, the average delay and the dropping probability can be derived.

- The average queue length equals

$$\overline{Q}_n = \sum_{k=0}^{K} \sum_{l=1}^{L} \pi^n_{l,k} k. \quad (14)$$

- The mean throughput can be expressed as

$$\overline{T}_n = \sum_{l=1}^{L} \sum_{k=1}^{K} T^n_{l,k} \pi^n_{l,k} \quad (15)$$

where

$$T^n_{l,k} = \begin{cases} R_{n,l} & \text{if } k \geq R_{n,l}\Delta T \\ \frac{k}{\Delta T} & \text{if } k < R_{n,l}\Delta T \end{cases}. \quad (16)$$

This is the sum of the product between the service rate in state l and the probability that the server is in state l given there are data in the system.

- The average delay then can be calculated as

$$\overline{D}_n = \overline{Q}_n / \overline{T}_n. \quad (17)$$

- Let $B_{l,k}^n$ be the random variable which represents the amount of dropped bits when $\mathcal{H}_{n,t} = l$ and $\mathcal{Q}_{n,t} = k$. Since $K + b = A_{n,t} + \max[0, k - R_{n,l}\Delta T]$, where b is the number of bits dropped during the t-th slot,

$$\mathbb{P}(B_{l,k}^n = b) = \mathbb{P}(A_{n,t} = K + b - \max[0, k - R_{n,l}\Delta T]). \tag{18}$$

Then, the dropping probability p_d^n can be estimated as

$$\begin{aligned} p_d^n &= \frac{\text{Average \# of bits dropped in a time slot}}{\text{Average \# of bits arrived in a time slot}} \\ &= \frac{\sum_{l=1}^{L}\sum_{k=0}^{K}\sum_{b=0}^{\infty} b\mathbb{P}(B_{l,k}^n = b)\pi_{l,k}^n}{\lambda_n \Delta T}. \end{aligned} \tag{19}$$

B. Model Decomposition and Iteration

Although the analytical method in previous section for the single user system can be applied to the multiuser scenario by constructing an embedded Markov chain $\{\mathcal{H}_{1,t}, \mathcal{Q}_{1,t}, \ldots, \mathcal{H}_{N,t}, \mathcal{Q}_{N,t}\}$ with appropriately defined transition probabilities, the exponentially enlarged state space makes it unacceptable for a large user population. Since directly solving the DSPN suffers the high computational complexity, in this subsection, model decomposition and an iteration procedure are introduced to simplify the analysis.

1) Model decomposition: According to [19], the original DSPN can be decomposed into a set of subnets, as shown in Fig. 2. The subnet Fig. 2(a) remains the same structure as that in Fig. 1(a), while the DSPN in Fig. 1(b) is decomposed into N DSPN subnets in Fig. 2(b). The n-th DSPN subnet consists of the exponentially-distributed timed transition c_n, the deterministic timed transition s_n, and the queue place Q_n. Note that the places r, w_n and the immediate transitions d_n in Fig. 1 are all deleted in the decomposed model for simple model description. The transition c_n and the place Q_n remain the same as that in the original DSPN model, while the firing rate of transition s_n is associated with the random switch $g_n(\mathbf{M})$ of the immediate transition d_n in the original DSPN. The resource sharing relationship of the original DSPN model is implicitly expressed in the marking-dependent firing rate of s_n as

$$\mu_n' = \mu_n g_n(\mathbf{M}). \tag{20}$$

By decomposition, the original multiuser system is represented by N subsystems, each of which consists of one SPN in Fig. 2(a) and one DSPN in Fig. 2(b). Obviously, if each subsystem can be analyzed separately, the model decomposition can significantly reduce the size of the state space in the analysis and achieve better performance in computational complexity. Since each subsystem n in the decomposed DSPN model is almost the same as that defined for the single user system, a similar two-dimensional Markov model $(\mathcal{H}_{n,t}, \mathcal{Q}_{n,t})$ as described in Section III-A can be constructed for each subsystem. The only difference is that the firing rate of s_n becomes μ_n' instead of μ_n.

However, unfortunately, such model decomposition is not 'clean', i.e., there exist interactions among subsystems due to the marking-dependent firing rate μ_n'. This is because the random switch $g_n(\mathbf{M})$ at the t-th time slot depends not only on its own marking, but also on the markings of all other subsystems. Thus, in order to solve the n-th subsystem, the markings of all other subsystems have to be available at the same time. There are two difficulties arising from this situation: 1) the marking of a subsystem is equivalent to the states of a Markov chain, which is time-dependent and cannot be used/derived as the input/output parameters of any other subsystems; 2) the subsystems cannot be ordered, giving rise to cycles in the model solution process. Considering a two-user system as an example, each subsystem 1 or 2 needs the output of the other as input and neither can be first solved correctly. In order to solve these difficulties, two methods are introduced in the following two subsections. The first difficulty will be solved by using the steady-state probabilities of the markings instead of the instant markings as the subsystem input, and the second difficulty will be solved by the method, called fixed point iteration [20]. Simulation results shown in Section IV indicate that such approximation is reasonable.

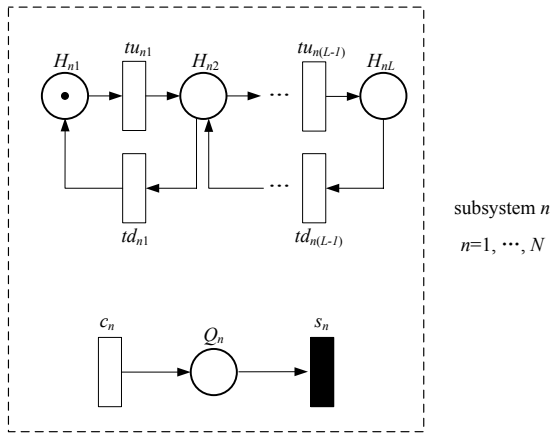


Fig. 2. Decomposed DSPN model.

2) Subsystem solution: Let $\boldsymbol{\pi}_n$ denote the steady-state probabilities of the embedded Markov chain $\{\mathcal{H}_{n,t}, \mathcal{Q}_{n,t}\}$ (i.e., the steady-state probabilities of the markings) for the n-th subsystem. Due to the interactions between the subsystems, the solution $\boldsymbol{\pi}_n$ for the n-th subsystem can be obtained only when the solutions of all the other subsystems $\{\boldsymbol{\pi}_i\}_{i=1,i\neq n}^N$ are known. Obviously, if the N subsystems are solved sequentially by index, the above requirement cannot be satisfied since only $\{\boldsymbol{\pi}_i\}_{i=1}^{n-1}$ are known when solving the n-th subsystem. In this subsection, we will derive the solution for the n-th subsystem under the assumption that $\{\boldsymbol{\pi}_i\}_{i=1,i\neq n}^N$ are given. We will leave the discussion of how to satisfy this assumption using the fixed point iteration method in the next subsection.

Given $\{\boldsymbol{\pi}_i\}_{i=1,i\neq n}^N$, we can approximate the random switch $g_n(\mathbf{M})$ with $\widetilde{g}_n(\mathbf{M})$, which is a function of the steady-state probabilities of the markings. Notice that $g_n(\mathbf{M})$ indicates the probability that user n will be selected given a certain marking $\mathbf{M}$, while $\widetilde{g}_n(\mathbf{M})$ indicates the long-run selection probability for user n. Based on $\widetilde{g}_n(\mathbf{M})$, the firing rate of s_n can be determined as

$$\widetilde{\mu}_n' = \begin{cases} \mu_n, & \text{with probability } \widetilde{g}_n(\mathbf{M}) \\ 0, & \text{with probability } 1 - \widetilde{g}_n(\mathbf{M}). \end{cases} \tag{21}$$

In the follows, the derivation of $\widetilde{g}_n(\mathbf{M})$ for different scheduling algorithms is discussed.

- For the RR algorithm, the selection probability of user n can be expressed as

$$\widetilde{g}_n(\mathbf{M})=\sum_{U\in\mathcal{U}}\frac{1}{\|U\|}\prod_{i\in U,n\neq i}\mathbb{P}[M(Q_i)>0]\prod_{j\in\overline{U}}\mathbb{P}[M(Q_j)=0] \quad (22)$$

where $U=\{n,\ldots\}$ is a set of users that *must* include user n and *may* include any other users, and $\mathcal{U}$ is the set that contains all possible sets of U.

- For the CA algorithm, the selection probability of user n depends on the server state. If the server is in the l-th state for user n, i.e. $M(H_{nl})=1$, we have $\mu_n=R_{n,l}$, and $\widetilde{g}_n(\mathbf{M})$ can be expressed as

$$\widetilde{g}_n(\mathbf{M})=\sum_{U\in\mathcal{U}}\prod_{i\in U,n\neq i}\mathbb{P}[M(Q_i)>0]\prod_{j\in\overline{U}}\mathbb{P}[M(Q_j)=0]p_U(\mathbf{M}) \quad (23)$$

where

$$p_U(\mathbf{M})=\sum_{V\in\mathcal{V}}\frac{1}{\|V\|}\prod_{i\in V,n\neq i}\mathbb{P}[M(H_{il})=1]\prod_{j\in U,j\in\overline{V}}\sum_{m=1}^{l-1}\mathbb{P}[M(H_{jm})=1], \quad (24)$$

$V\subseteq U$ and $\mathcal{V}$ is the set that contains all possible subsets of U.

- For the QA algorithm, the selection probability of user n depends on its queue length. If the queue length of user n is k, i.e. $M(Q_n)=k$, $\widetilde{g}_n(\mathbf{M})$ can be expressed as

$$\widetilde{g}_n(\mathbf{M})=\sum_{U\in\mathcal{U}}\prod_{i\in U,n\neq i}\mathbb{P}[M(Q_i)>0]\prod_{j\in\overline{U}}\mathbb{P}[M(Q_j)=0]p_U(\mathbf{M}) \quad (25)$$

where

$$p_U(\mathbf{M})=\sum_{V\in\mathcal{V}}\frac{1}{\|V\|}\prod_{i\in V,n\neq i}\mathbb{P}[M(Q_i)=k]\prod_{j\in U,j\in\overline{V}}\mathbb{P}[M(Q_j)<k]. \quad (26)$$

- For the CQA algorithm, the selection probability of user n depends on both the server state and its queue length. If the server is in the l-th state and the queue length is k for user n, i.e. $M(H_{nl})=1$ and $M(Q_n)=k$, $\widetilde{g}_n(\mathbf{M})$ can be expressed

$$\widetilde{g}_n(\mathbf{M})=\sum_{U\in\mathcal{U}}\prod_{i\in U,n\neq i}\mathbb{P}[M(Q_i)>0]\prod_{j\in\overline{U}}\mathbb{P}[M(Q_j)=0]p_U(\mathbf{M}) \quad (27)$$

where

$$p_U(\mathbf{M})=\sum_{V\in\mathcal{V}}\frac{1}{\|V\|}\prod_{i\in V,n\neq i}\mathbb{P}[\mu_i M(Q_i)=\mu_{nl}k]\prod_{j\in U,j\in\overline{V}}\mathbb{P}[\mu_j M(Q_j)<\mu_{nl}k]. \quad (28)$$

With $\widetilde{g}_n(\mathbf{M})$, each subsystem n can be solved by following the similar way as in Section III-A with the following revisions for some system parameters.

- The computation of $p^n_{(l,k),(m,h)}$ should be changed to

$$p^n_{(l,k),(m,h)}=(\nu^{n,l}_{k,h}\widetilde{g}_n(\mathbf{M})+\nu^{n,0}_{k,h}(1-\widetilde{g}_n(\mathbf{M})))p^n_{l,m} \quad (29)$$

where $\nu^{n,l}_{k,h}$ and $\nu^{n,0}_{k,h}$ are determined according to (11) with $R_{n,0}=0$.

- The mean throughput of user n becomes

$$\overline{T}_n=\sum_{l=1}^{L}\sum_{k=1}^{K}T^n_{l,k}\widetilde{g}_n(\mathbf{M})\pi^n_{l,k} \quad (30)$$

where $T^n_{l,k}$ is derived according to (16). The total mean throughput of the multiuser system is

$$\overline{T}=\sum_{n=1}^{N}\overline{T}_n. \quad (31)$$

- The dropping probability can be computed as (19). However, the value of $\mathbb{P}(B^n_{l,k}=b)$ equals

$$\begin{aligned}&\mathbb{P}(B^n_{l,k}=b)\\&=\mathbb{P}(A_{n,t}=K+b-\max[0,k-R_{n,l}\Delta T])\widetilde{g}_n(M)\\&+\mathbb{P}(A_{n,t}=K+b-k)(1-\widetilde{g}_n(\mathbf{M})).\end{aligned} \quad (32)$$

3) Fixed point iteration: In the previous subsection, we assume that the solutions of all the other subsystems $\{\boldsymbol{\pi}_i\}_{i=1,i\neq n}^{N}$ are already derived and can be used as input parameters when solving the n-th subsystem. However, this assumption is not true if the N subsystems are solved sequentially by index. In this subsection, fixed point iteration is used to deal with this problem.

Let $\{\boldsymbol{\pi}_1,\ldots,\boldsymbol{\pi}_N\}$ be the vector of *iteration variables* of the *fixed point equation*

$$\{\boldsymbol{\pi}_1,\ldots,\boldsymbol{\pi}_N\}=f(\{\boldsymbol{\pi}_1,\ldots,\boldsymbol{\pi}_N\}) \quad (33)$$

where the function f is realized by solving the N subsystems successively with the subsystem solution method as described in the previous subsection. That is, the function f can be decomposed into N independent functions f_n, $n=1,\ldots,N$, with f_n representing the solution of the n-th subsystem

$$\boldsymbol{\pi}_n=f_n(\{\boldsymbol{\pi}_1,\ldots,\boldsymbol{\pi}_N\}).$$

Obviously, the vector of steady-state distributions of the N subsystems $\{\boldsymbol{\pi}_1,\ldots,\boldsymbol{\pi}_N\}$ satisfies (33), which is referred to as the *fixed point* of this equation.

The fixed point can be derived by *successive substitution* [20]. Let the initial vector of iteration variables be $\{\boldsymbol{\pi}^0_1,\ldots,\boldsymbol{\pi}^0_N\}$. Each element of $\boldsymbol{\pi}^0_n$ $(n=1,\ldots,N)$ can be set to an arbitrary value between 0 and 1. In the z-th iteration, we have

$$\{\boldsymbol{\pi}^z_1,\ldots,\boldsymbol{\pi}^z_N\}=f(\{\boldsymbol{\pi}^{z-1}_1,\ldots,\boldsymbol{\pi}^{z-1}_N\}) \quad (34)$$

where the iteration variables are determined by the function f based on the values of the last iteration, and the function f is realized by solving the N subsystems successively using the solution method as described above. Specifically, in solving the n-th subsystem in the z-th iteration,

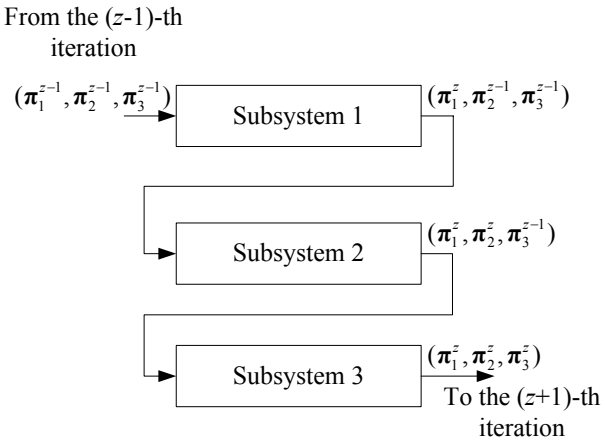


Fig. 3. The z-th iteration for the three-user scenario.

$\{\boldsymbol{\pi}_1^z, \ldots, \boldsymbol{\pi}_{n-1}^z, \boldsymbol{\pi}_n^{z-1}, \ldots, \boldsymbol{\pi}_N^{z-1}\}$ is used as the input to derive the value of $\boldsymbol{\pi}_n^z$. After that, $\boldsymbol{\pi}_n^{z-1}$ is replaced by $\boldsymbol{\pi}_n^z$ as input in solving the rest of the subsystems (from the $(n+1)$-th to the N-th subsystems) during the z-th iteration. An example for the z-th iteration of a three-user system is illustrated in Fig. 3.

The iteration is terminated when the differences between the iteration variables of two successive iterations are less than a certain threshold value. The convergence of the fixed point iteration is proved in Appendix C.

The computational complexity of the proposed analytical approach can be obtained as follows. Let Z represent the number of iterations for analysis to converge. Since in each iteration, the steady-state probabilities of the N subsystems have to be derived, the total time for the approach termination is $T = Z \times N \times T_{\text{sub}}$, where T_{sub} denotes the amount of time to solve each subsystem. T_{sub} depends on the state number of the embedded Markov chain for each subsystem, which equals $(K+1) \times L$. Note that the state number of the embedded Markov chain before decomposition equals $\big((K+1) \times L\big)^N$. Therefore, the proposed analytical approach with decomposition and iteration techniques greatly reduces the computational complexity.

IV. Simulation and Discussions

In this section, both analytical and simulation results are presented to compare the performance of different scheduling algorithms. In the simulation, all users are assumed to have statistically identical wireless channels. The SNR thresholds and the corresponding transmission rates for each service process are defined in Table I [22]. If the instantaneous SNR is below -12.5dB, the transmission rate is set to be 0. Accordingly, the FSMC model has 12 states in total. The carrier frequency f and the time slot duration ΔT are set to 1.9GHz and 1.67ms, respectively. The velocity v of the mobile users is assumed to be 3km/h so that the Doppler frequency f_d^n becomes 5Hz. We also let the mean SNR $\overline{\gamma}_n$ be 0dB and the buffer size $K = 2500$bits. Notice that for other values of SNR, similar observations can be obtained.

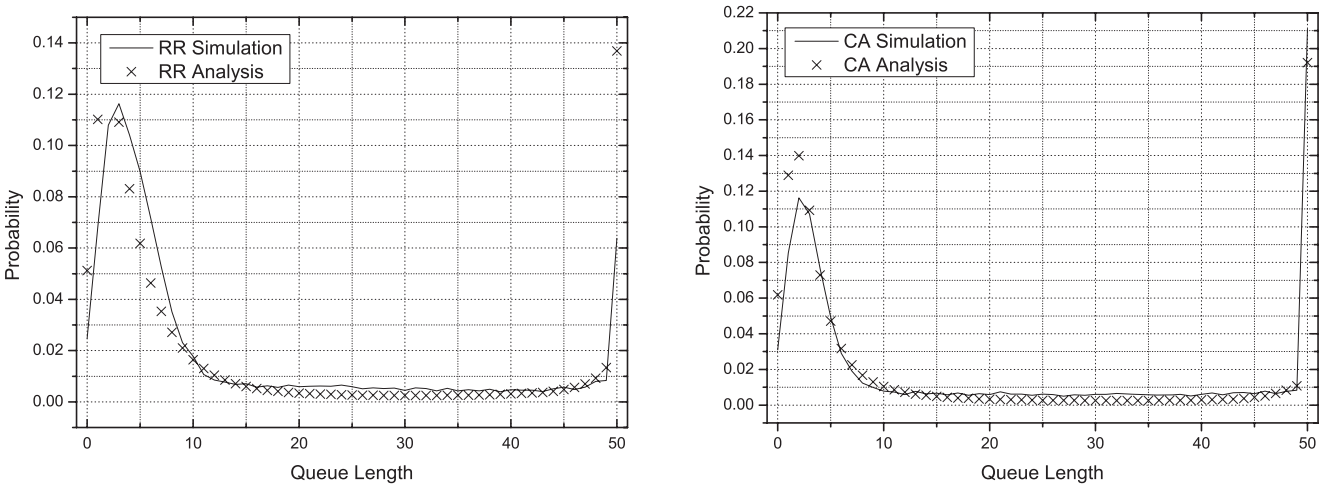
In order to simplify the simulations by further reducing the state space of the Markov model, we redefine the data unit in the queueing system, or equivalently the token unit in the Petri net model, so that one data unit represents 50 bits. This approximation is acceptable since the amount of data that arrive to or depart from the system in one time slot usually includes a large number of bits. After redefinition, both the transmission rate $R_{n,l}$ and the buffer size K should be divided by 50, and the arrival rate and the performance metrics derived in the follows are in terms of data units. Note that the maximum queue length becomes 50 data units.

TABLE I
SNR Threshold and Rates

SNR Threshold $\gamma_{n,l}$ (dB) $\geq$	Rates $R_{n,l}$ (Kbs)
-12.5	38.4
-9.5	76.8
-8.5	102.6
-6.5	153.6
-5.7	204.8
-4.0	307.2
-1.0	614.4
1.3	921.6
3.0	1228.8
7.2	1843.2
9.5	2457.6

Fig. 4 examines the accuracy of our analysis method described in Section III. The stationary queue length distributions from both analytical and simulation results are compared, which are denoted by "Analysis" and "Simulation" in the figure, respectively. In the simulation, the number of users is set to 2, and the mean data arrival rate per user is fixed at 1,500 data units per second. Each simulation runs for 20,000 time slots. Note that in the simulation, the analytical results converge after three iterations. In each iteration, the steady-state probabilities of the two embedded Markov chains for subsystems 1 and 2 are obtained, respectively. The state number of the Markov chain for each subsystem 1 or 2 equals 612, while the state number of the embedded Markov chain for directly modelling the whole system equals 374544. This further demonstrates the computational complexity reduction by the proposed approach.

In Fig. 4, two subfigures (a) and (b) correspond to the round robin (RR) and the channel-aware (CA) algorithms, respectively. From the figures, it can be observed that, for both scheduling algorithms, the analytical results match well with those from the simulation, i.e., the proposed analytical method is accurate. The same observation is also true for the queue-aware (QA) and the channel/queue-aware (CQA) algorithms, and the simulation and analytical results for these two algorithms are omitted. Since the buffer size K equals 50 data units and the arriving data is dropped if the buffer is full, the distribution of the queue length is truncated at 50. Since the queue length distribution at 50 represents the sum probability of the queue length equal to and larger than 50 if no buffer limitation exists, one peak at 50 is observed. In what follows, we will apply analytical results only to compare the performance of the four scheduling algorithms. The performance metrics include average queue length, mean throughput, average delay and dropping probability.



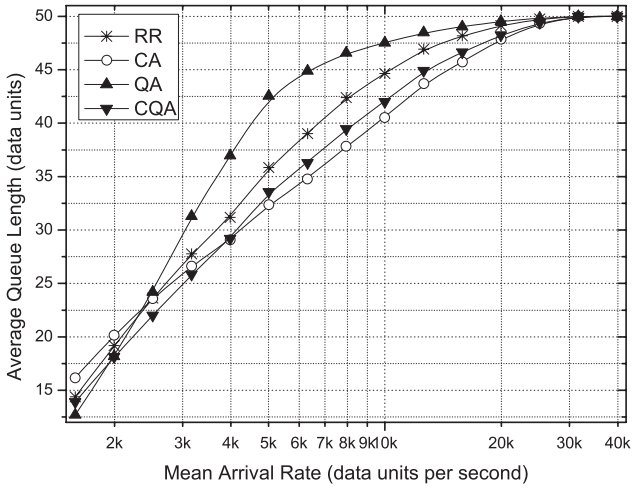
(a) round robin algorithm

(b) channel-aware algorithm

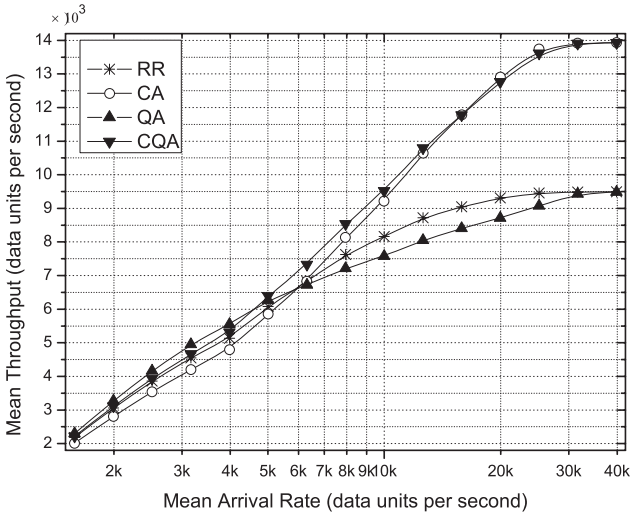
Fig. 4. Analytical and simulation results of queue length distribution for different schedulers.

Fig. 5 shows the performance comparison among the algorithms under different arrival rates (or traffic loads). The results are based on the fixed user number of 2 and the variant average data arrival rate per user from 1,000 to 40,000 data units per second. From the figure, it can be observed that 1) when the arrival rate is below a certain threshold (approximately 3,000-5,000 data units per second with respect to different performance metrics), the CA algorithm has the worst performance, while the QA algorithm achieves the best one; 2) when the arrival rate is higher than the threshold, the performance of the QA algorithm becomes the worst, while those of CA and CQA algorithms get very close and become the best; 3) the performance of the RR algorithm converges to that of the QA algorithm with the increase of the arrival rate. All these observations can be explained as follows. The CA algorithm tends to select a user with a better channel condition, while the QA algorithm favors a user with a larger queue length. When the traffic load is light, the instantaneous queue length is relatively short so that the actual transmission rate of a selected user n at any time slot t is mainly determined by its instantaneous queue length instead of its current channel condition. Therefore, in this case, the QA algorithm, which favors users with larger queue lengths, can maximize the transmission rate, while the CA algorithm performs worst due to the ignorance of queue length information. When the traffic load is heavy, on the other hand, the queue length is relatively large and the actual transmission rate of a selected user n at any time slot t is determined by its instantaneous channel condition. Therefore, the QA and the CA change their positions in system performance. Since the CQA algorithm considers both channel and queue states in user selection, it can balance the service rate and the queue length in both situations and thus achieve relatively good performance under all traffic conditions. When the traffic load is extremely heavy, i.e., the queues of the users are saturated, the queue length information is not important any more in user selection, which results in the performance convergence of the CA/CQA and RR/QA algorithms. Also, note that in Fig. 5(c), the average delay of QA algorithm is concave (increases before the arrival rate reaches 5,000 data units per second and decreases afterwards) while the average delay of others are monotonically decreasing with traffic load. This is because when the traffic load is light, the average queue length of the QA algorithm increases much faster than its mean throughput as shown in subfigures (a) and (b).

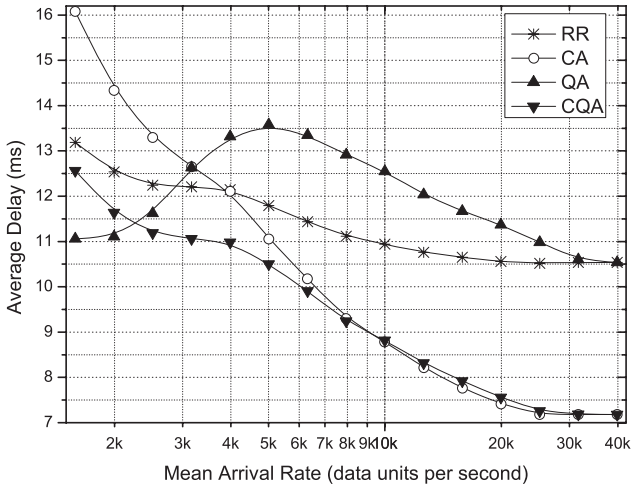
Fig. 6 shows the performance of the scheduling algorithms under different numbers of users. The average data arrival rates per user are set at the low (1,500 data units per second) and high (5,000 data units per second) levels. The number of users varies from 2 to 10. In infinite backlog traffic model [10], it is well-known that the OS algorithms achieve larger throughput than the RR algorithm, or the scheduling gain, defined as the ratio between the throughput of an OS algorithm and the RR algorithm, is larger than 1. Furthermore, such scheduling gain increases with the number of users due to the improved multiuser diversity effect. However, Figs. 6(a) and 6(b) reveal that for dynamic data arrivals, the above observations only hold for high average arrival rate (5,000 data units per second). When the average arrival rate is low (1,500 data units per second), the scheduling gain of the CA algorithm over the RR algorithm becomes smaller than 1, and decreases with the increase of the number of users. A similar observation can be obtained for the average delay and the dropping probability, as well. When the average arrival rate is high, the performance of CA algorithm deteriorates more slowly in terms of the average delay and the dropping probability than the RR algorithm as the number of users increases. But, when the average arrival rate is low, an opposite results can be observed. The analytical results for these two performance metrics are omitted here due to space limitation. The faster performance degradation of the CA algorithm under light traffic load results from the facts that 1) the multiuser diversity gain has little effects on the transmission rate, which is dominated by the queue length; and 2) such negative effects are enlarged in terms of delay and dropping probability when more users are waiting for transmission.



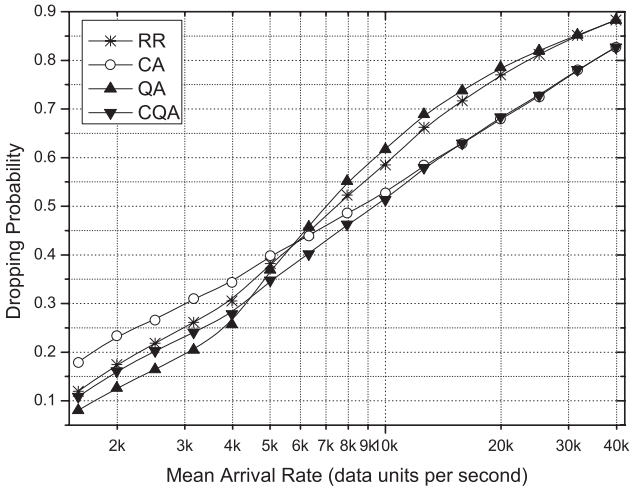
(a) average queue length (data units)



(b) mean throughput (data units per second)

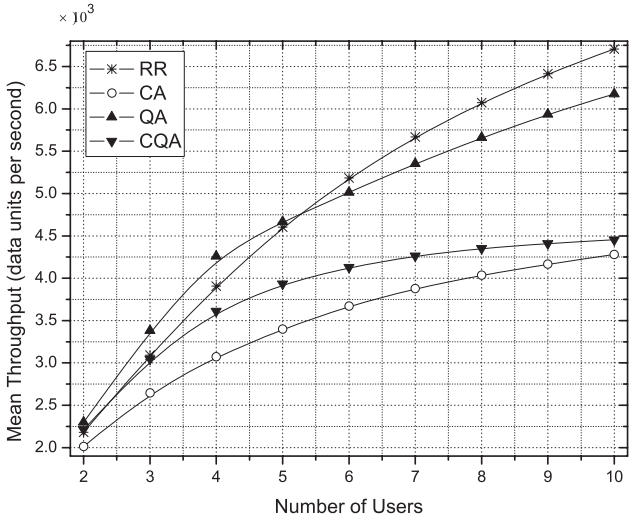


(c) average delay (ms)

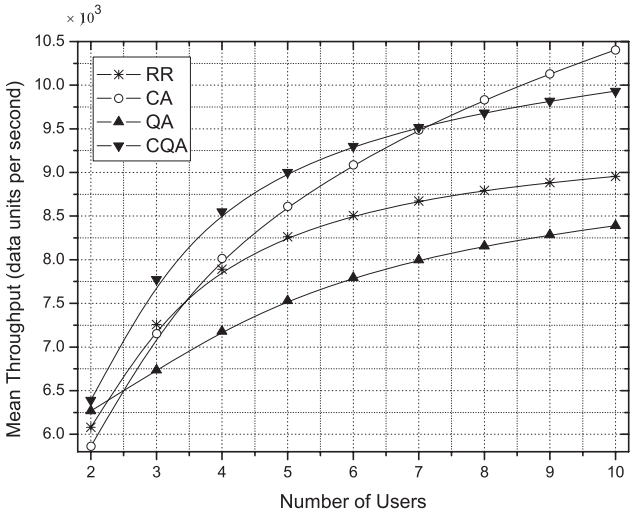


(d) dropping probability

Fig. 5. Performance comparison under different arrival rates.



(a) mean throughput - arrival rate 1.5Kps



(b) mean throughput - arrival rate 5Kps

Fig. 6. Performance comparison under different number of users.

V. Conclusions

In this paper, a general framework for analyzing performance of the wireless schedulers in multiuser systems has been discussed. The system behavior is formulated by an M/MMDP/1/K queueing model and the approximation for multiple performance metrics is derived with low computational complexity by applying the decomposition and the iteration techniques from SPN. Based on this framework, four classes of typical wireless schedulers, which are referred to as round robin, channel-aware, queue-aware and channel/queue-aware, are analyzed and compared in terms of average queue length, mean throughput, average delay and dropping probability. The analysis shows that

- while in heavy traffic regime the channel-aware scheduler indeed outperforms the round robin scheduler, this is not true when the traffic load is light;
- the performance of the channel/queue-aware scheduler is better than that of the channel-aware scheduler in the light traffic regime, and converges to the latter with the increase of the traffic load; and
- the ratio of the throughput between the channel-aware scheduler and the round robin scheduler, which is commonly referred to as the scheduling gain, decreases with the increase of the number of users when the traffic load per user is light.

Our future work will focus on extending the proposed analytical framework to multi-carrier and multi-antenna systems, where the system behavior can be formulated as an M/MMDP/M/K queueing model. Since the state space of the multi-server model is even larger than the single server model studied in this paper, it can be expected that applying the decomposition and iteration techniques may be more important in obtaining practical solutions.

Acknowledgement

The authors would like to thank Prof. Quanlin Li for his helpful discussion on the queueing model.

Appendix

A. Brief introduction of stochastic Petri nets (SPNs) [23], [24]

A Petri net is a directed bipartite graph with two types of nodes called *places* and *transitions* which are represented by circles and rectangles (or bars), respectively. Arcs connecting places to transitions are referred to as *input arcs*; while the connections from transitions to places are called *output arcs*. A non-negative integer (the default value is one) may be associated with an arc, which is referred to as *multiplicity* or *weight*. Places correspond to state variables of the system, while transitions correspond to actions that induce changes of states. A place may contain tokens that are represented by dots in the Petri net. The state of the Petri net is defined by its marking, which is represented by a vector $\mathbf{M} = (l_1, l_2, \ldots, l_k)$, where $l_k = M(p_k)$ is the number of tokens in place p_k. Here, $M(\cdot)$ is a mapping function from a place to the number of tokens assigned to it. A transition is enabled if the number of tokens in each of its input places is larger than the weight of its corresponding input arc. An enabled transition can fire, and as many tokens as the corresponding input arc's weight are moved from the input place to the output place.

Stochastic Petri nets (SPNs) are one kind of Petri nets in which an exponentially-distributed time delay is associated with each transition. Generalized Stochastic Petri nets (GSPNs) extends the modelling power of SPN, and divides the transitions into two classes: the exponentially-distributed timed transitions (represented by blank rectangles), which are used to model the random delays associated with the execution of activities, and immediate transitions (represented by bars), which are devoted to the representation of logical actions that do not consume time. Deterministic and stochastic Petri nets (DSPNs) further extends GSPN in that it allows timed transitions to have an exponentially-distributed time delay or an deterministic timed delay (represented by filled rectangles). The firing rate of the timed transitions may be marking-dependent.

SPN or GSPN can be mapped to continuous-time Markov chains (CTMC). If the resulting CTMC is irreducible, we can compute its steady-state probability vector. However, since the continuous-time stochastic process underlying DSPN is non-Markovian, the discrete-time embedded Markov chain has to be constructed in order to compute the steady-state solutions of DSPN.

B. Determination of $p_{l,m}^n$ in Rayleigh fading channel

For Rayleigh fading channel, $p_{l,m}^n$ can be determined as follows [18]. Assume the state transitions of the FSMC happen only between adjacent states, i.e.

$$p_{l,m}^n = 0, \quad |l-m| \geq 2. \tag{35}$$

Let $\gamma_{n,l}$, $(l = 1, \ldots, L-1)$, denotes the SNR threshold value between the l-th and $(l+1)$-th states of the FSMC model for user n. The adjacent-state transition probability can be calculated as

$$p_{l,l+1}^n = \frac{\chi(\gamma_{n,l+1})\Delta T}{\pi_{n,l}}, \quad l = 1, \ldots, L-1, \tag{36}$$

$$p_{l,l-1}^n = \frac{\chi(\gamma_{n,l})\Delta T}{\pi_{n,l}}, \quad l = 2, \ldots, L. \tag{37}$$

Here, $\chi(\gamma_n)$ denotes the level cross rate at an instantaneous SNR γ_n and is given by

$$\chi(\gamma_n) = \sqrt{\frac{2\pi\gamma_n}{\overline{\gamma}}} f_d^n \exp(-\frac{\gamma_n}{\overline{\gamma}_n}) \tag{38}$$

where f_d^n denotes the mobility-induced Doppler spread, $\overline{\gamma}_n = \mathbb{E}[\gamma_n]$ is the average received SNR, and $\pi_{n,l}(l \in \mathcal{L})$ denotes the stationary probability that the FSMC is in state l given by

$$\pi_{n,l} = \exp(\gamma_{n,l}/\overline{\gamma}_n) - \exp(\gamma_{n,l+1}/\overline{\gamma}_n). \tag{39}$$

Finally, $p_{l,l}^n$ can be derived from the normalizing condition $\sum_{m=1}^{L} p_{l,m}^n = 1$ as

$$p_{l,l}^n = \begin{cases} 1 - p_{l,l+1}^n - p_{l,l-1}^n, & (l = 2, \ldots, L-1) \\ 1 - p_{l,l+1}^n, & (l = 1) \\ 1 - p_{l,l-1}^n, & (l = L) \end{cases}. \tag{40}$$

C. Convergence of the fixed point iteration

According to Theorem 2 in [20], in order to prove the convergence of the fixed point iteration for the decomposed DSPN model as described in (34), it is sufficient to show that the following lemma is true.

Lemma 1: The embedded Markov chain $(\mathcal{H}_{n,t}, \mathcal{Q}_{n,t})$ for the n-th subsystem is irreducible, if $K \leq R_{n,L}\Delta T$.

Proof: It has been proved in [15] that the Markov chain for the single user system is irreducible. Similarly, we prove Lemma 1 by showing that for each transition from state (l,k) to (m,h), there exists a multi-transition path $(l,k) \rightarrow (l^*,k) \rightarrow (l^*,h) \rightarrow (m,h)$ with non-zero probability, where $R_{n,l^*}\Delta T \geq k$. Since $K \leq R_{n,L}\Delta T$, there always exists such l^* that satisfies this condition.

Since the FSMC model is irreducible, we have that p^n_{l,l^*}, $p^n_{l^*,l^*}$ and $p^n_{l^*,m}$ are all positive. Now we shall verify the following inequalities:

1) $\nu^{n,l}_{k,k}\widetilde{g}_n(\mathbf{M}) + \nu^{n,0}_{k,k}(1-\widetilde{g}_n(\mathbf{M})) > 0$;
2) $\nu^{n,l^*}_{k,h}\widetilde{g}_n(\mathbf{M}) + \nu^{n,0}_{k,h}(1-\widetilde{g}_n(\mathbf{M})) > 0$;
3) $\nu^{n,l^*}_{h,h}\widetilde{g}_n(\mathbf{M}) + \nu^{n,0}_{h,h}(1-\widetilde{g}_n(\mathbf{M})) > 0$.

For inequality 1), since $A_{n,t} = k - \max[0, k - R_{n,l}\Delta T] \geq 0$ ($R_{n,0} = 0$ included), we have $\nu^{n,l}_{k,k} > 0$ and $\nu^{n,0}_{k,k} > 0$. Therefore, inequality 1) is true with $\widetilde{g}_n(\mathbf{M}) \in [0,1]$. The proof of inequality 3) is similar.

For inequality 2), since $A_{n,t} = h - \max[0, k - R_{n,l^*}\Delta T] \geq 0$, we have $\nu^{n,l^*}_{k,h} > 0$, where $R_{n,l^*}\Delta T \geq k$. Now consider both the cases when the value of k is zero or not. If $k = 0$, we have $\widetilde{g}_n(\mathbf{M}) = 0$ and $\nu^{n,0}_{k,h} > 0$; otherwise, if $k > 0$, we have $\widetilde{g}_n(\mathbf{M}) > 0$ and $\nu^{n,0}_{k,h} \geq 0$. Thus, the inequality 2) is true under both cases.

According to (29), we have $p^n_{(l,k),(l^*,k)} > 0$, $p^n_{(l^*,k),(l^*,h)} > 0$ and $p^n_{(l^*,h),(m,h)} > 0$, where $R_{n,l^*}\Delta T \geq k$, which prove the existence of the multi-transition path with non-zero probability for each transition from state (l,k) to (m,h).

REFERENCES

[1] 3GPP2 C.S0024 Version 4.0, "CDMA 2000 High Rate Packet Data Air Interface Specification," Oct. 2002.

[2] 3GPP TS 25.848, "Physical layer aspects of utra high speed downlink packet access," v4.0.0, Release 4, 2001.

[3] M. Dianati, X. Shen, and K. Naik, "Cooperative fair scheduling for the downlink of CDMA celluar networks," *IEEE Trans. Veh. Technol.*, vol. 56, no. 4, pp. 1749-1760, 2007.

[4] M. Andrews, "Instability of the proportional fair scheduling algorithm for HDR," *IEEE Trans. Wireless Commun.*, vol. 3, no. 5, pp. 1422-1426, 2004.

[5] M. Andrews *et al.*, "Scheduling in a queueing system with asynchronously varying service rate," *Probability in the Engineering and Informational Sciences*, vol. 18, pp. 191-217, 2004.

[6] D. Wu and R. Negi, "Utilizing multiuser diversity for efficient support of quality of service over a fading channel," *IEEE Trans. Veh. Technol.*, vol. 54, no. 3, pp. 1198-1206, May 2005.

[7] F. Ishizaki and G. U. Hwang, "Queueing delay analysis for packet schedulers with/without multiuser diversity over a fading channel," *IEEE Trans. Veh. Technol.*, vol. 56, no. 5, pp. 3220-3227, Sept. 2007.

[8] S. Shakkottai, "Effective capacity and QoS for wireless scheduling," *IEEE Trans. Automatic Control*, Feb. 2008, to appear.

[9] D. Piazza and L. B. Milstein, "Analysis of multiuser diversity in time-varying channels," *IEEE Trans. Wireless Commun.*, vol. 6, no. 12, pp. 4412-4419, 2007.

[10] S. C. Borst, "User-level performance of channel-aware scheduling algorithms in wireless data networks," *IEEE/ACM Trans. Networking*, vol. 13, no. 3, pp. 636-647, 2005.

[11] R. Prakash and V. V. Veeravalli, "Centralized wireless data networks with user arrivals and depatures," *IEEE Trans. Inform. Theory*, vol. 53, no. 2, pp. 693-713, 2007.

[12] E. N. Gillbert, "Capacity of a burst-noise channel," *Bell Syst. Tech. J.*, vol. 39, pp. 1253-1265, Sept. 1960.

[13] E. O. Elliott, "Estimates of error rates for codes on burst-noise channels," *Bell Syst. Tech. J.*, vol. 42, pp. 1977-1997, Sept. 1963.

[14] H. S. Wang and N. Moayeri, "Finite-state Markov channel-A useful model for radio communication channels," *IEEE Trans. Veh. Technol.*, vol. 44, pp. 163-171, Feb. 1995.

[15] Q. Liu, S. Zhou, and G. B. Giannakis, "Queueing with adaptive modulation and coding over wireless links: cross-layer analysis and design," *IEEE Trans. Wireless Commun.*, vol. 50, no. 3, pp. 1142-1153, May 2005.

[16] G. Bolch and C. Bruzsa, "Modeling and simulation of Markov modulated multiprocessor systems with Petri nets," in *Proc. 7th European Simualtion Symprosium*, University of Erlangen, Germany, 1995.

[17] T. Yang and D. H. K. Tsang, "A novel approach to estimating the cell loss probability in an ATM multiplexer loaded with homogeneous on-off sources," *IEEE Trans. Commun.*, vol. 43, no. 1, pp. 117-126, Jan. 1995.

[18] Q. Zhang and S. A. Kassam, "Finite-state Markov model for Rayleigh fading channels," *IEEE Trans. Commun.*, vol. 47, no. 11, pp. 1688-1692, Nov. 1999.

[19] G. Ciardo and K. S. Trivedi, "A decomposition approach for stochastic reward net models," *Performance Evaluation*, vol. 18, pp. 37-59, 1993.

[20] V. Mainkar and K. S. Trivedi, "Fixed point iteration using stochastic reward nets," in *Proc. sixth International Workshop on Petri Nets and Performance Models*, pp. 21-30, 1995.

[21] Z. Shan, C. Lin, D. C. Marinescu, and Yang Y, "QoS-aware load balancing in web-server clusters: performance modeling and approximate analysis," *Computer Networks J.*, vol. 40, no. 2, pp. 235-256, Sept. 2002.

[22] P. Bender, *et al.*, "CDMA/HDR: a bandwidth-efficient high-speed wireless data service for nomadic users," *IEEE Commun. Mag.*, vol. 38, no. 7, pp. 70-77, 2000.

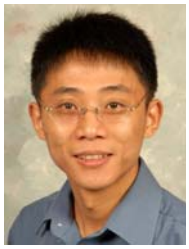
[23] J. L. Peterson, Petri Net Theory and the Modeling of Systems. Englewood Cliffs, NJ: Prentice-Hall, 1981.

[24] M. A. Marson and G. Chiola, "On Petri nets with deterministic and exponentially distributed firing times," *Advances in Petri Nets 1987*, *LCNS* vol. 266, pp. 132-145, 1987.

Lei Lei received a B.S. degree in 2001 and a PhD degree in 2006, respectively, from Beijing University of Posts & Telecommunications, China, both in telecommunications engineering. From 2006 to 2008, she was a postdoctoral fellow at Computer Science Department, Tsinghua University, Beijing, China. Her current research interests include performance evaluation, quality-of-service and radio resource management in wireless communication networks.

Chuang Lin (IEEE SM'04) is a professor of the Department of Computer Science and Technology, Tsinghua University, Beijing, China. He received the Ph.D. degree in Computer Science from the Tsinghua University in 1994. His current research interests include computer networks, performance evaluation, network security analysis, and Petri net theory and its applications. He has published more than 300 papers in research journals and IEEE conference proceedings in these areas and has published three books.

Professor Lin is a member of ACM Council, a senior member of the IEEE and the Chinese Delegate in TC6 of IFIP. He serves as the Technical Program Vice Chair, the 10th IEEE Workshop on Future Trends of Distributed Computing Systems (FTDCS 2004); the General Chair, ACM SIGCOMM Asia workshop 2005; the Associate Editor, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY; the Area Editor, JOURNAL OF COMPUTER NETWORKS; and the Area Editor, JOURNAL OF PARALLEL AND DISTRIBUTED COMPUTING.



Jun Cai (M'04) received the B.Sc. (1996) and the M.Sc. (1999) degrees from Xi'an Jiaotong University (China) and Ph.D. degree (2004) from University of Waterloo, Ontario (Canada), all in electrical engineering. From June 2004 to April 2006, he was with McMaster University as NSERC Postdoctoral Fellow. Since July 2006, he has been with the Department of Electrical and Computer Engineering, University of Manitoba, Canada, where he is an Assistant Professor. His current research interests include multimedia communication systems, mobility and resource management in 3G beyond wireless communication networks, cognitive radios, and ad hoc and mesh networks. He was a registration chair of the First International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness (QShine) 2005, a technical program committee co-chair of the International Wireless Communications and Mobile Computing Conference (IWCMC) 2008 General Symposium, and a technical program committee member for serval IEEE and other international conferences in wireless communications and networks. He is currently the NSERC Associate Industrial Research Chair.

Xuemin (Sherman) Shen (IEEE M'97-SM'02-F'09) received the B.Sc. (1982) degree from Dalian Maritime University (China) and the M.Sc. (1987) and Ph.D. degrees (1990) from Rutgers University, New Jersey (USA), all in electrical engineering. He is a University Research Chair Professor, Department of Electrical and Computer Engineering, University of Waterloo, Canada. His research focuses on mobility and resource management in interconnected wireless/wired networks, UWB wireless communications systems, wireless security, and vehicular ad hoc networks and sensor networks. He is a co-author of three books, and has published more than 400 papers and book chapters in wireless communications and networks, control and filtering. Dr. Shen serves as the Technical Program Committee Chair for IEEE Globecom'07, General Co-Chair for Chinacom'07 and QShine'06, the Founding Chair for IEEE Communications Society Technical Committee on P2P Communications and Networking. He also serves as a Founding Area Editor for IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS; Editor-in-Chief for PEER-TO-PEER NETWORKING AND APPLICATION; Associate Editor for IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY; KICS/IEEE JOURNAL OF COMMUNICATIONS AND NETWORKS, COMPUTER NETWORKS; ACM/WIRELESS NETWORKS; and WIRELESS COMMUNICATIONS AND MOBILE COMPUTING (Wiley), etc. He has also served as Guest Editor for IEEE JSAC, IEEE WIRELESS COMMUNICATIONS, and IEEE COMMUNICATIONS MAGAZINE. Dr. Shen received the Excellent Graduate Supervision Award in 2006, and the Outstanding Performance Award in 2004 and 2008 from the University of Waterloo, the Premier's Research Excellence Award (PREA) in 2003 from the Province of Ontario, Canada, and the Distinguished Performance Award in 2002 from the Faculty of Engineering, University of Waterloo. Dr. Shen is a registered Professional Engineer of Ontario, Canada.