Optimal and Approximate Mobility-Assisted Opportunistic Scheduling in Cellular Networks

Syed Hussain Ali, Student Member, IEEE, Vikram Krishnamurthy, Fellow, IEEE, and Victor C.M. Leung, Fellow, IEEE

Abstract—This paper considers the problem of scheduling multiple users in the downlink of a time-slotted cellular data network. For such a network, opportunistic scheduling algorithms improve system performance by exploiting time variations of the radio channel. We present novel optimal and approximate opportunistic scheduling algorithms that combine channel fluctuation and user mobility information in their decision rules. The algorithms modify the opportunistic scheduling framework of Liu et al. [5] with dynamic constraints for fairness. These fairness constraints adapt according to the user mobility. The adaptation of constraints in the proposed algorithms implicitly results in giving priority to the users that are in the most favorable locations. The optimal algorithm is an offline algorithm that precomputes constraint values according to a known mobility model. The approximate algorithm is an online algorithm that relies on the future prediction of the user mobility locations in time. We show that the use of mobility information in opportunistic scheduling increases channel capacity. We also provide analytical bounds on the performance of the approximate algorithm using the fundamental inequality of Dyer et al. [17] for linear programs. Simulation results on High Data Rate (HDR) illustrate the usefulness of the proposed schemes for elastic traffic and macrocell structures.

Index Terms—Wireless communication, mobile communication systems, algorithm design and analysis, linear programming, land mobile radio cellular systems, access control.

1 INTRODUCTION

We consider a downlink system model in a wireless cellular network where multiple data flows time-share a single channel. The model is inspired by third generation cellular systems such as High Data Rate (HDR) [1] and High-Speed Downlink Packet Access (HSDPA) [2]. The central scheduler at the base station (BS) operates in discrete time slots and transmits to one user at a time. The channel condition, being a random time-varying and user-dependent process, affects the service rate of a respective user. Better channel conditions result in a higher rate and vice versa. The channel conditions of all flows are conveyed to the scheduler as feasible rates before the start of the time slot. For this model, we consider infinitely backlogged flows, where the number of flows is constant and each flow always contains data for transmission. Such a model is suitable for elastic traffic that can tolerate variable delays.

In the wireless context, multiple flows correspond to download queues for corresponding users, channel condition means signal to interference plus noise ratio (SINR), and service rate equals the data rate. User equipment measures SINR in response to a BS pilot signal, where SINR accounts for path loss and fading effects [3]. BS receives channel information in the form of a feasible rate, that is, the rate at which the corresponding user can be served if scheduled in the time slot. Path loss variations occur at a slow timescale due to changes in distance between transmitter and receiver, whereas fading variations occur at a fast timescale due to multipath propagation.

For the above system model, scheduling plays an important part in the system performance because it decides which user is served in each time slot. If the scheduler selects a user with a high SINR value, then that user receives a high data rate. Conversely, a user selected with a low SINR value receives a low data rate. A scheduling scheme that gives preference to the users with a high data rate is called opportunistic scheduling [4]. Opportunistic scheduling takes advantage of the fast time fluctuations in the radio channel due to fading. It improves system performance by favoring users at a time when they have higher data rates. A trivial solution would be to always serve the best user, that is, maximum rate (MR) scheduling. However, this solution leads to unfairness because a few users with excellent wireless links can starve other users. Therefore, research in this area concentrates on proposing schemes that trade overall performance with quality-of-service (QoS) requirements like fairness (see, for example, [4], [5], [6], and [7]).

In this paper, our objective is to improve the long-term system data rate for elastic data traffic. To accomplish this objective, we exploit not only fading-related fast variations, but also path-loss-related slow variations in the feasible data rate. We present novel opportunistic scheduling schemes in which decision rules combine the user’s mobility and channel information. In a macrocellular environment with mobile user population, the average SINR of a mobile user will improve (or worsen) if the user moves toward (or away from) the BS. If the scheduler can track and anticipate user mobility, it can adapt scheduling
priorities with the help of dynamic constraints. It can give priority to users that are in the most favorable locations, for example, closer to the BS. Furthermore, adapting priorities should still satisfy the user’s long-term fairness constraints and prevent long-term starvation. The proposed mobility-assisted opportunistic scheduling (MaOs) algorithm achieves these objectives for infinitely backlogged elastic data traffic. First, the algorithm adapts user priorities by dynamically adjusting fairness constraints according to the user mobility information. In this way, it takes advantage of the slow time variations in the radio channel. Next, it uses these priorities to opportunistically schedule the users. Thus, it also exploits fast time fading variations.

There are two versions of the proposed algorithm: the optimal MaOs, which uses full state-space knowledge and precomputes user priorities for all possible mobility states, and the approximate MaOs, which relies on the future prediction of user mobility to compute user priorities in real time.

1.1 Related Work
1.1.1 Infinite Backlog Model
Liu et al. have proposed opportunistic scheduling algorithms that maximize the sum of the expected user data rates and satisfy long-term fairness constraints under an infinite backlog assumption [4], [5]. Their algorithms (abbreviated here as LCS algorithms) optimally exploit the fast time fading variations in the feasible rate. They consider several fairness constraints, but, for want of space, we discuss only the temporal fairness measure and present our proposed mobility-assisted solution for the same measure. We emphasize that the results and analysis presented in this paper are applicable for other measures considered in [5].

Let there be \( N \) users in a cell and let \( \mathbf{R} = (R_1, \ldots, R_N) \) be the feasible data rate vector at a generic scheduling time slot. Recall that the feasible rate \( R_i \) for user \( i \) is defined as the rate at which user \( i \) can be served if scheduled in the time slot. Further, assume that \( E\left(R_i 1_{(Q=1)}\right) \) is the average scheduled data rate of user \( i \) achieved by the scheduling policy \( Q \), where \( 1_{(\cdot)} \) is the indicator function. The overall system data rate is \( E\left(R_Q(\mathbf{R})\right) = \sum_{i=1}^{N} E\left(R_i 1_{(Q(\mathbf{R})=i)}\right) \).

The temporal fairness measure considers time as a resource and shares it among multiple users according to the given fairness constraints. Therefore, if \( P(Q(\mathbf{R}) = i) \) is the probability of scheduling user \( i \) by policy \( Q \) and \( r_i \) is the long-term minimum temporal requirement for user \( i \), where \( r_i \geq 0, \epsilon := \sum_{i=1}^{N} r_i \leq 1 \), then the LCS algorithm solves the following stochastic optimization problem:

\[
\max_Q E\left(R_Q(\mathbf{R})\right), \quad \text{s.t.} \quad P(Q(\mathbf{R}) = i) \geq r_i, \quad i = 1, 2, \ldots, N.
\]

(1)

According to the LCS algorithm, the optimal policy is \( Q^*(\mathbf{R}) = \arg\max_i (R_i + \nu_i^*) \),

(2)

where \( \nu_i^* \) are the true Lagrange multipliers that satisfy the complementary slackness conditions of the constraints in problem (1). The LCS algorithm has a polynomial complexity of \( O(N) \).

Let us consider an example to clarify the physical meanings of the temporal fairness measure. Assume that there are 10 users sharing a single channel with fairness constraints given as \( r_i = 1/12 \). These constraints mean that each user expects an allocation of at least 8.33 percent of the time slots for itself. This is a long-term fairness measure and considers only a fraction of the total time. Thus, the position of the allocated time slots or their order is not important.

1.1.2 Impact of Mobility
The positive impact of mobility on the performance of wireless systems has been an active research area after Grossglauser and Tse [8] showed that mobility increases the capacity of ad hoc mobile networks at the cost of increased delay.

For cellular networks, Donald et al. [9] and Borst et al. [10] have analyzed system performance for single-cell and multicell scenarios, respectively. According to these studies, intracell and intercell mobility increases performance. Certainly, in this context, our work enforces the results of the above studies that mobility increases system performance. However, our work is significantly different from [9] and [10]. We propose a new scheduling algorithm that optimally exploits user mobility and prevents long-term starvation.

1.1.3 Other Models and Objective Functions
Since our emphasis is on long-term fairness, we maximize the sum of expected data rates under an infinite backlog assumption. This assumption shields the scheduling process from delay distributions and arrival process dynamics and is appropriate for elastic traffic applications like file download.

Alternatively, for delay-constrained applications, different objective functions or models are used. For example, proportional fair scheduling (PF) [11], [12] maximizes the product of the expected data rates that is mathematically equivalent to \( \max \sum_{i=1}^{N} \log \left( E\left(R_i 1_{(Q(\mathbf{R})=i)}\right) \right) \) [13]. This objective function favors weak users over strong users. PF also exploits the time-varying channel conditions and provides proportional fairness that is different than the temporal fairness considered in this paper [4]. PF reduces delay variance among users, but in comparison to the class of algorithms that maximize the sum of the data rates, PF achieves, overall, lower data rates [4], [7]. Recall that LCS and MaOs fall into the category of algorithms that maximize the sum of the data rates. Like LCS, PF has a polynomial complexity of \( O(N) \).

Furthermore, in contrast to the backlog model, some studies consider another model in which arrival processes feed queues. For this model, popular scheduling algorithms based on the MaxWeight scheduling rule [14] aim to establish short-term fairness. Therefore, they combine the
queue length [15] or the head-of-line packet delay [16] with the channel information in their decision rules.

1.2 Contributions

The following are the main contributions of this paper:

1. We formulate a semi-Markov mobility model that is a coarse-grained mobility model suitable for user mobility among hot-spot locations within a cell. We then propose an MaOS algorithm that combines mobility and channel information in the scheduling rule. MaOS consists of two stages and, like LCS, considers time as a resource. In the first stage, the algorithm takes advantage of the slow time path loss variations and optimally distributes time fractions, which can be considered as priorities among all users. In the second stage, the algorithm exploits the fast time fading fluctuations by opportunistically scheduling users according to the resulting time fractions from the first stage. We prove that the optimal MaOS algorithm increases the channel capacity.

2. Because the optimal MaOS algorithm is computationally expensive, a computationally less intensive approximate MaOS algorithm is proposed that relies on time windowing and the sample path of the user mobility.

3. We provide analytical lower bounds on the performance of the approximate MaOS. For this purpose, we generalize the fundamental inequality of Dyer et al. [17], [18] (identified here as DFM) and analyze the performance of the approximate MaOS algorithm. The original inequality works for minimization linear programs (LPs) with certain conditions that make it unsuitable for maximization problems. We extend the inequality for maximization problems and analyze the approximate MaOS with respect to the sample path and the number of users. This analysis helps us to find lower bounds on the system performance of the approximate MaOS algorithm.

1.3 Outline

Section 2 briefly describes mobility and channel models. Section 3 presents the optimal MaOS algorithm followed by the approximate MaOS algorithm in Section 4. Section 5 analyzes the optimum algorithm in terms of the channel capacity gain and proposes bounds on the approximate algorithm. Implementation details of both algorithms are covered in Section 6. Numerical results of the optimal MaOS algorithm and a comparison with the LCS algorithm are given in Section 7. Section 8 compares the approximate MaOS with the LCS algorithm and Section 9 concludes this paper. Frequently used notations are summarized in Table 1.

2 Mobility and Channel Models

This section presents the mobility and channel models used in this paper. For the mobility model, we propose a novel finite-state semi-Markov model that is suitable for mobility between hot-spot locations in a cell or wireless coverage area. This model is a coarse-grained model because it divides the cell surface into nonoverlapping regions. The channel model is a popular finite-state Markov chain that models the Rayleigh fading process.

2.1 Finite-State Semi-Markov Mobility Model

This paper considers a discrete-state mobility model that divides cell geography into a set of nonoverlapping topological spaces \( S = \{1, \ldots, M\} \) called mobility states. Fig. 1 shows two examples of the mobility model where states are regular concentric rings (Fig. 1a) [19] or irregular cell areas depending on the accessibility and geography of the cell (Fig. 1b). These states are characterized by the

| \( N \) | number of users |
| \( R_i \) | feasible rate (FR) for user \( i \) and \( \overline{R} \) is FR vector |
| \( E(R_{Q(i)}) \) | average system data rate |
| \( E(R_i,1) \) | average scheduled data rate of user \( i \) |
| \( \gamma_i \) | probability of prediction error |
| \( v_i \) | Lagrange multiplier |
| \( S, \overline{m} \) | \( S = S_1 \times \ldots \times S_N \), \( \overline{m} \in S \) |
| \( Z() \) | SINR |
| \( \Gamma(m) \) | path loss in mobility state \( m \in S \) |
| \( I(m) \) | interference in mobility state \( m \in S \) |
| \( \mathcal{H} \) | ordered SINR levels |
| \( P_2(.) \) | distribution of SINR |
| \( \tilde{R}() \) | expected feasible rate for a user |
| \( \tilde{r}() \) | normalized time fractions (7) |
| \( \mathbf{H} \) | aggregate channel conditions \( \mathbf{H} = \mathcal{H}_1 \times \mathcal{H}_2 \) |
| \( E_{m}[\cdot], E_{\overline{m}}[\cdot] \) | expectation with respect to \( \overline{m} \) and \( \overline{H} \) respectively |
| \( \phi(\cdot), \omega(\cdot) \) | optimal solution of (14) and (17) respectively |
| \( E_\cap[\cdot] \) | time-average of expected feasible rate |
| \( c_{X}, x \) | coefficients and decision variables of (21) |
| \( B_{X} \) | feasible bases of (21) |
| \( Q \) | scheduling policy |
| \( 1_{Q(i)=i} \) | indicator function = 1 if \( i \) selected |
| \( \alpha \) | probability of scheduling user \( i \) |
| \( \theta, \beta, \delta \) | constants, see (3), (6), (23), and (26) respectively |
| \( r_i, \epsilon \) | time fraction allocation, \( \epsilon := \sum_{i=1}^{N} r_i \) |
| \( S, m \) | set of mobility states, \( m \in S \) |
| \( \Pi(\overline{m}), \pi(m) \) | stationary distribution of vector \( \overline{m} \) and state \( m \) |
| \( \rho(m) \) | mean SINR in mobility state \( m \) |
| \( P_t, P_l \) | transmit and interference power respectively |
| \( Y \) | cell radius |
| \( \zeta \) | fading process |
| \( p(.) \) | steady state distribution of channel |
| \( r^*(\cdot) \) | optimal solution of (6) |
| \( \tau, \Delta \) | mobility time window and \( \Delta = 1, \ldots, \tau \) |
| \( \overline{h} \) | aggregate channel condition vector \( \overline{h} \in \mathbf{H} \) |
| \( C_x \) | ergodic capacity where \( x = \{TD, LCS, MaOS\} \) |
| \( u^* \) | optimum value of linear program (21) |
| \( \bar{x}_j \) | any feasible solution of (21) |
| \( K,J \) | number of constraints and variables in (21) |
average SINR, $\rho(m)$, where $m \in S$. User mobility among these states follows a semi-Markov process with (known) generally distributed dwell times $P_d(m)$ and mean $\bar{d}(m)$. The stationary distribution of the underlying process is found as follows [20]:

$$\pi(m) = \frac{\pi'(m)\bar{d}(m)}{\sum_{m' \in S} \pi'(m')\bar{d}(m')} ,$$

where $\pi'(m)$ is the stationary distribution of the embedded Markov chain.

The above model is a coarse-grained model with discrete-state space. However, user mobility occurs in a continuous space where users follow fine-grained mobility patterns like random walk or random waypoint [21]. The proposed model is adaptable to any fine-grained mobility pattern and cell structure. For example, the speed and structure of $S$ affect the dwell-time distribution. The proposed model learns parameters $\pi'(\cdot)$, $P_d(\cdot)$, and $\bar{d}(\cdot)$ from simulation or actual data.

### 2.2 Finite-State Markov Chain Channel Model

The SINR is a function of path loss, interference, and fading processes [3].

#### 2.2.1 Path Loss and Interference

Path loss is a function of the distance from the BS, and interference is a function of the distance from the neighboring cells. Because our mobility model allows user mobility over large areas, the average SINR will change over time, resulting in a nonstationary channel model. To cope with the nonstationarity and to have a simpler model that allows analysis, we employ a piecewise stationary channel model.

For the piecewise stationary channel model, we assume that the path loss and interference remains constant within a mobility state. These values are computed at the center of the mobility state and used in the estimation of the channel model for that state. As long as a user remains in a mobility state, the user experiences channel conditions according to the stationary probability distributions for that state. Transitioning to another mobility state results in another channel model with a different set of probability distributions. The piecewise stationary model captures the essence of nonstationarity due to user mobility and allows a simpler analysis compared to a fully nonstationary channel model.

Let mobility state $m$ be at distance $y_m$ from the BS (see Fig. 1a). According to a simple propagation model, also considered in [19], the path loss at distance $y_m$ from the BS is given as

$$\Gamma(m) = \begin{cases} \frac{1}{y_m^\alpha} & \text{if } y_m \leq y_0 \\ 0 & \text{otherwise} \end{cases}$$

where $y_0$ is the maximum distance from the BS up to where full transmitted power $P_t$ is received and $\alpha$ is the path loss exponent.

For the network structure, we consider a seven-cell structure (Fig. 1c). The central cell forms the service area, and the other six cells act as sources of interference. The interference from the neighboring cells is approximated as [19]

$$I(m) = P_t \times \left( \Gamma(2Y - y_m) + 2\Gamma(\sqrt{(Y - y_m)^2 + 3Y^2}) \\ + 2\Gamma(\sqrt{Y^2 + y_m^2} + 3Y^2) + \Gamma(2Y + y_m) \right) ,$$

where $P_t$ is the transmit power of the interfering cell and $Y$ is the cell radius.

#### 2.2.2 Fading Process and SINR

For small-scale fading, a flat fading channel is assumed because, for systems like HDR and HSDPA, the feedback delay is relatively short compared to the fading frequency [1].

Let user $i$ be in mobility state $m$ at a generic time slot. According to the fading channel model, the received SINR behaves as follows [19]:

$$Z_i(m) = \zeta \frac{P_t \times \Gamma(m)}{\eta + I(m)} ,$$

where $\zeta$ is the time-dependent fading process, $P_t$ is the transmitted power, $\Gamma(m)$ and $I(m)$ are the path loss and interference experienced by the user, and $\eta$ is the background noise. Furthermore, the fading processes $\zeta$ are independent and identically distributed (i.i.d.) copies of an exponentially distributed stationary process with unit mean. Thus, the SINR process $Z$ also behaves as an exponentially distributed random process with mean $\rho(m) = P_t \times \Gamma(m)/(\eta + I(m))$ and, for $z \geq 0$, its distribution is given as follows [22]:

$$p_Z(z, m) = \frac{1}{\rho(m)} e^{-\frac{z}{\rho(m)}} .$$

Let $H = \{Z_0, Z_1, \ldots, Z_{H-1}\}$ represent the set of discrete SINR levels in ascending order. The fading channel takes level $h$ if the received SINR falls in the interval $[Z_h, Z_{h+1})$. The corresponding steady-state probability of level $h$ in mobility state $m$ is found as follows [22]:

$$p(h, m) = \int_{Z_h}^{Z_{h+1}} p_Z(z, m)dz = e^{-\frac{Z_h}{\rho(m)}} - e^{-\frac{Z_{h+1}}{\rho(m)}} .$$

The transition probabilities are computed according to the procedure given in [22] and [23]. The physical layer

1. The i.i.d. and unit mean assumptions are not necessary for the validity of our results; these assumptions are considered only to simplify discussion.
coding and modulation schemes map fading levels to corresponding data rates.

2.3 Hierarchical Time Scale

The overall system model employs a hierarchical timescale with two levels. The higher level, referred to as mobility slot, operates on coarse-grained user mobility and has a wider time slot. This time slot allows the proposed algorithm to take advantage of the slow time path loss variations in the feasible data rate. At the mobility slot level, the fast time fading variations in the feasible rate can be represented by an expected value. The lower level, referred to as scheduling slot, handles scheduling decisions and has a narrower time slot. The scheduling slot allows the algorithm to benefit from the fast time fading fluctuations. At the scheduling slot, the slow time path loss variations can be modeled as a constant path loss experienced at the center of the corresponding mobility state. The presence of two timescales helps to accommodate user mobility that evolves relatively slowly compared to the scheduling decisions.

Remark on Time Index. Channel fluctuations, feasible rates, and scheduling decisions depend on the scheduling timescale, whereas user mobility and expected feasible rates follow the mobility timescale. For simplicity of discussion, generally, we do not index variables with time. However, a time index is used for the mobility slot when considering the sample path of the user mobility. For other places, the use of the time index is clear according to the context.

3 PROPOSED OPTIMAL MAOS ALGORITHM

The proposed optimal MaOS algorithm consists of two stages. The first stage exploits the slow time path loss variations due to user mobility and the second stage takes advantage of the fast time fading fluctuations due to the multipath propagation.

3.1 Stage 1: Exploiting Slow Variations

Recall that, at the timescale of user mobility, the fading experienced by a user in a mobility state can be abstracted by the corresponding expected feasible rate. In order to improve the system data rate, it is imperative to give preference to users when they are in the most favorable locations, that is, having high expected feasible rates. Since the proposed algorithm considers time as a resource, if time fractions are distributed such that they maximize their product with the expected feasible rates of the corresponding users for all aggregate mobility states, then the algorithm can achieve its stated objectives of improving the system data rate. Therefore, this stage of the MaOS algorithm determines optimum time fractions that maximize the expected feasible rates of the users under the constraints of providing fairness and preventing starvation.

For $N$ users and $M$ mobility states per user, the set of possible aggregate states is an $N$-dimensional space given by $S = S_1 \times \ldots \times S_N$, where $\times$ denotes the Cartesian product and $S_i$ is the state space of user $i$ defined in Section 2.1. Thus, the aggregate mobility model contains $M^N$ states (that is, $|S| = M^N$), and a state is identified by a vector $\mathbf{m} = (m_1, \ldots, m_N)$ at a generic mobility slot. Assuming independence among users, the stationary distribution of state $\mathbf{m} \in S$ equates to the product of the individual stationary distributions, $\Pi(\mathbf{m}) = \prod_{i=1}^{N} \Pi(m_i)$. Further, assume that the scheduler knows the above mobility model, expected feasible rates $\bar{R}(\mathbf{m}, i)$ and $i$ and can track the user state accurately.

The proposed MaOS algorithm modifies the LCS algorithm by using dynamic fairness constraints. The algorithm first computes the optimal time-fraction allocations for every mobility state $\mathbf{m}$ that maximizes its product with the expected feasible rate summed over all states and for all users. For the notation used in (1), where $r_i$ is the long-term minimum temporal requirement for user $i$ and $\epsilon = \sum_{i=1}^{N} r_i \leq 1$, the proposed MaOS algorithm solves the following LP:

$$\begin{align*}
\max & \quad \sum_{\mathbf{m} \in S} \sum_{i=1}^{N} \bar{R}(\mathbf{m}, i)r(\mathbf{m}, i), \\
\text{s.t.} & \quad \sum_{i=1}^{N} r(\mathbf{m}, i) \leq \Pi(\mathbf{m}), \quad \forall \mathbf{m} \in S, \\
& \quad \sum_{\mathbf{m} \in S} r(\mathbf{m}, i) = r_i, \quad i = 1, \ldots, N, \\
& \quad r(\mathbf{m}, i) \geq \frac{\theta r_i}{|S|}, \quad \forall \mathbf{m} \in S, \forall i, \text{ and } 0 \leq \theta \leq \theta_{\text{max}},
\end{align*}$$

where $r(\mathbf{m}, i)$ are optimization variables and $\theta$ is a parameter.

The above LP maximizes the weighted sum of the expected feasible rate and complies with the stationary distribution of user mobility (6b), satisfies long-term temporal fairness constraints (6c), and prevents long-term starvation (6d). The time fraction $r_i$ of (1) behaves as a resource and the above LP optimally distributes it among all mobility states accessible to the user $i$.

The rational for $\theta > 0$ in (6d) is to prevent the long-term starvation of users. When $\theta = 0$, the resulting solution of (6) is a greedy solution where users are denied access in mobility states with weak channels. This denial results in the starvation of such users. Therefore, $\theta > 0$ guarantees that a minimum fraction of $r_i$, distributed over all mobility states, is assigned to every user even in mobility states with bad channels. In this way, the MaOS algorithm avoids long-term starvation. For the feasibility of LP, $\theta$ is upper bounded by $\theta_{\text{max}} = \min_{\mathbf{m}}(\Pi(\mathbf{m}))|S|/\epsilon$.

Let $r^*(\mathbf{m}, i)$ maximize the objective function (6a). This value is a fraction of $r_i$ because of the construction of LP (6). For example, when $\epsilon = 1$, then $r_i$ behaves like the probability of allocation of user $i$. The resulting $r^*(\mathbf{m}, i)$ is the joint probability of two events: user $i$’s allocation and the occurrence of mobility state $\mathbf{m}$. Thus, user $i$’s allocation probability given mobility state $\mathbf{m}$ can be found by dividing $r^*(\mathbf{m}, i)$ by the corresponding probability of the state. For a more general case when $\epsilon < 1$ and considering the feasibility of the constraint (6b), the resulting allocation probability, identified here as the normalized time fraction $\hat{r}(\cdot)$, is found as follows:

$$\hat{r}(\mathbf{m}, i) = \epsilon \frac{r^*(\mathbf{m}, i)}{\sum_{i=1}^{N} r^*(\mathbf{m}, i)}, \quad \forall \mathbf{m} \in S.$$
The next stage of the algorithm uses the normalized time fractions as the dynamic fairness constraints and opportunistically schedules users.

### 3.2 Stage 2: Exploiting Fast Variations

In this stage, the algorithm gains from the fast time fading fluctuations in the feasible data rate. For this purpose, it uses the LCS methodology for the normalized time fractions found in the first stage. It maximizes the expectation of the scheduled (system data) rate for each mobility state and satisfies the respective time-fraction allocation. For this objective, the algorithm solves the following stochastic optimization problem after modifying problem (1):

\[
\max_{Q} \quad E\left(R_{Q}(\tilde{m})\right), \quad \text{s.t.} \quad P\{Q(\tilde{m}) = i\} \geq \tilde{\alpha}(\tilde{m}, i),
\]
\[
i = 1, 2, \ldots, N.
\]

The resulting problem has dynamic constraint values that are functions of \(\tilde{m}\). The optimal scheduling policy is found by adapting the solution (2) for state \(\tilde{m}\):

\[
Q^*(\tilde{R}) = \arg \max_i (R_i + \nu_i^*(\tilde{m})),
\]

where \(\nu_i^*(\tilde{m})\) is the true Lagrange multiplier that satisfies the corresponding constraint in (8). Intuitively speaking, MaOS is a piecewise LCS algorithm, that is, one for every \(\tilde{m}\) state.

### 4 Proposed Approximate MaOS Algorithm

The optimal MaOS algorithm has exponential complexity in the number of users because it precomputes time fractions for all aggregate mobility states. The approximate algorithm computes time fractions in real time and relies on the future prediction of the mobility states of an individual user. Unlike the optimal MaOS, the approximate algorithm does not rely on the underlying mobility model. Rather, it depends on the sample path of the user mobility.

Like optimal MaOS, the approximate algorithm also consists of two stages. The first stage gains from the slow time path loss variations. Assume that, at (mobility) time \(t\), the scheduler can accurately predict the future state of all users for the next \(\tau\) mobility slots. Using this prediction during the first stage, the algorithm finds time-fraction values so that users close to the BS can be given preferences. During the second stage, the algorithm opportunistically schedules users according to the time fractions found in the first stage. The formulation of the second stage is given in the following paragraphs.

The first stage of the approximate MaOS finds \(\hat{\tau}(t + \Delta, i), \forall i\) and \(\Delta = 1, \ldots, \tau\), as a solution to the following LP:

\[
\max \quad \sum_{\Delta=1}^{\tau} \sum_{i=1}^{N} \tilde{R}(t + \Delta, i)\hat{\tau}(t + \Delta, i),
\]
\[
\text{s.t.} \quad \sum_{i=1}^{N} \hat{\tau}(t + \Delta, i) = \epsilon, \quad \forall \Delta,
\]
\[
\frac{1}{\tau} \sum_{\Delta=1}^{\tau} \hat{\tau}(t + \Delta, i) = r_i, \quad i = 1, \ldots, N,
\]

where \(\tilde{R}(t + \Delta, i)\) is the average feasible data rate of user \(i\) at \(t + \Delta\) (mobility) time slot in the future in accordance with the predicted mobility state. The constraint (10b) ensures that, for every \(\Delta\), the time fraction resource \(\epsilon\) is distributed among users, and constraint (10c) makes sure that, for every \(\tau\), the assigned fraction satisfies the long-term fairness requirement. The above LP (10) is solved after every \(\tau\) mobility slots.

Let \(\hat{\tau}(t + \Delta, i)\) maximize the objective function of LP (10). In the second stage, these values are used to modify and solve problem (1). The resulting stochastic optimization problem maximizes the expectation of the scheduled (system data) rate and satisfies the new fairness constraints as follows:

\[
\max_{Q} \quad E\left(R_{Q}(\tilde{R}(t + \Delta))\right), \quad \text{s.t.} \quad P\{Q(\tilde{R}(t + \Delta)) = i\} \geq \tilde{\alpha}(t + \Delta, i),
\]
\[
i = 1, 2, \ldots, N, \quad \Delta = 1, \ldots, \tau.
\]

**Remark.** The value of constraint constant \(\hat{\tau}(t + \Delta, i)\) changes only when a user makes a transition to a new mobility state. After that transition, it remains constant for several mobility time slots until the next transition. If need be, (10) can be modified by lower bounding \(r(t + \Delta, .)\) like (6d) in order to avoid possible long-term starvation.

The solution of problem (11) is a modified LCS policy that selects a user at every (scheduling) time slot according to the following rule:

\[
Q^*(\tilde{R}) = \arg \max_i (R_i + \nu_i^*(t + \Delta)),
\]

where \(\nu_i^*(t + \Delta)\) is the Lagrange multiplier for the constraint active at (mobility) time \(t + \Delta\). The resulting approximate MaOS algorithm has a complexity of \(O(N\tau)\).

### 5 Analysis of MaOS

This section analyzes the optimal and approximate MaOS algorithms. Section 5.1 proves that the optimal MaOS algorithm increases channel capacity. Section 5.2 provides bounds on the first stage (10) of the approximate MaOS algorithm.

#### 5.1 Capacity Gain of the Optimal MaOS

In this section, we use numerical methods to compute the ergodic capacity for the LCS and optimal MaOS algorithms in fading channel environments. The ergodic capacity is defined as [24] “the maximum long-term achievable rate averaged over all states of the time-varying channel.” Because the channel distribution is a function of the mobility state (Section 2.2), the ergodic capacity definition is modified as “the long-term achievable rate that is averaged over all mobility and fading states.” Furthermore, it is assumed that the LCS and MaOS algorithms are working in the restrictive constraint regime, that is, \(\epsilon \approx 1.0\). In this regime, constraints are almost always binding; therefore, LCS cannot overschedule any user beyond its fairness requirement. The allocation of users is spread all
over the scheduling time horizon. Using this fact, we generate the time division (TD) capacity equations of [24] for the ergodic capacity followed by similar equations for the LCS and MaOS algorithms.

We only consider the capacity region in two dimensions (that is, two users sharing a common channel) because the general characteristics of the spectrum sharing methods remain the same for a large number of users as in the two-user case [25]. Thus, for two users and $H$ channel conditions per user, the set of possible aggregate channel conditions is a 2D space given by $\mathcal{H}(\mathbf{m}) = \mathcal{H}_1 \times \mathcal{H}_2$, where $\times$ denotes the Cartesian product and $\mathbf{m}$ in parentheses signifies that $\mathbf{m}$ is a function of the aggregate mobility state. An aggregate channel condition is identified by a vector $\mathbf{h} = (h_1, h_2)$ and the corresponding feasible user data rate by $R(\mathbf{h}_i)$. Assuming independent channels among users, the steady-state distribution of state $\mathbf{h} \in \mathcal{H}(\mathbf{m})$, $P(\mathbf{h})$, is computed as a product of the individual steady-state distributions. Thus, using the modified definition of ergodic capacity, we can write the TD capacity of [24] as follows:

$$\mathcal{C}_{TD} = \bigcup_{\{r_1, \theta \leq r_1 \leq 1\}} \left(C_1 = E_m \left[ E_h \left[ r_1 R(\mathbf{h}_1, 1) \right] \right], C_2 = E_m \left[ E_h \left[ (1 - r_1) R(\mathbf{h}_1, 2) \right] \right]\right),$$

where $r_1$ is the fraction of time slots to be assigned to user 1 and the remaining fraction $1 - r_1$ to user 2; $E_h$ and $E_m$ are the expectations with respect to $\mathbf{h}$ and $\mathbf{m}$, respectively.

Equation (13) is the nonopportunistic TD capacity equation because it assigns an $r_i$ time fraction to user $i$ for all $\mathbf{h}$. However, the LCS algorithm is opportunistic; therefore, the time-fraction allocation also depends on the corresponding feasible data rate with an objective of maximizing the overall system rate and satisfying the fairness constraints. Thus, we need to discover the fraction of times the algorithm selects a user for every state $\mathbf{h} \in \mathcal{H}(\mathbf{m})$. We identify this fraction as $\phi(\mathbf{h}_i, i), i = 1, 2$. The corresponding LP that achieves the LCS objectives for every $\mathbf{m}$ is

\begin{align}
\max & \sum_{\mathbf{h} \in \mathcal{H}(\mathbf{m})} \sum_{i=1}^{2} R(\mathbf{h}_i, i) \phi(\mathbf{h}_i, i), \\
\text{s.t.} & \sum_{i=1}^{2} \phi(\mathbf{h}_i, i) \leq P(\mathbf{h}), \quad \forall \mathbf{h} \in \mathcal{H}(\mathbf{m}), \\
& \sum_{\mathbf{h}} \phi(\mathbf{h}_i, i) = r_1, \quad i = 1, 2, \quad r_2 = 1 - r_1, \\
& \phi(\mathbf{h}_i, i) \geq 0, \quad \forall \mathbf{h} \in \mathcal{H}(\mathbf{m}), \quad i = 1, 2.
\end{align}

Let $\phi(\mathbf{h}_i, i)$ maximize the objective function of (14). This value is a fraction of $r_i$; therefore, normalizing for aggregate channel condition $\mathbf{h}$ results in

$$\phi(\mathbf{h}_i, i) = \frac{\phi(\mathbf{h}_i, i)}{\sum_{i=1}^{2} \phi(\mathbf{h}_i, i)}, \quad \forall \mathbf{h} \in \mathcal{H}(\mathbf{m}).$$

The solution of (14) and the subsequent computation of (15) results in $\phi(\mathbf{h}_i, i)$, which are members of a set defined as $\mathcal{F}_{LCS} \triangleq \{ \phi(\mathbf{h}_i, i) : 0 \leq r_1 \leq 1 \text{ and Sol. of (14), (15)} \}$, where $\mathcal{F}_{LCS}$ is the closure of $\phi(\mathbf{h}_i, i)$ for all values of $r_1$ and $\mathbf{m}$. The resulting capacity region for LCS is expressed as follows:

$$\mathcal{C}_{LCS} = \bigcup_{\{\phi(\mathbf{h}_i, i) \in \mathcal{F}_{LCS}\}} \left( C_1 = E_m \left[ E_h [\phi(\mathbf{h}_1, 1) R(\mathbf{h}_1, 1)] \right], C_2 = E_m \left[ E_h [\phi(\mathbf{h}_2, 2) R(\mathbf{h}_2, 2)] \right]\right).$$

Next, we extend (16) for the optimal MaOS. Like LCS, we need to compute time-fraction allocations for a user for every $\mathbf{h} \in \mathcal{H}(\mathbf{m})$. However, this computation is different than the problem in (14) because the fraction values now depend on the average feasible data rates in aggregate mobility states. Using the notation $\omega(\cdot)$ in place of $\phi(\cdot)$, the modified LP for every $\mathbf{m}$ is

\begin{align}
\max & \sum_{\mathbf{h} \in \mathcal{H}(\mathbf{m})} \sum_{i=1}^{2} R(\mathbf{h}_i, i) \omega(\mathbf{h}_i, i), \\
\text{s.t.} & \sum_{i=1}^{2} \omega(\mathbf{h}_i, i) \leq P(\mathbf{h}), \quad \forall \mathbf{h} \in \mathcal{H}(\mathbf{m}), \\
& \sum_{\mathbf{h}} \omega(\mathbf{h}_i, i) = r(\mathbf{m}, i), \\
& \omega(\mathbf{h}_i, i) \geq 0, \quad \forall \mathbf{h} \in \mathcal{H}(\mathbf{m}), \quad i = 1, 2.
\end{align}

Comparing (17c) with (14c) shows that the values of $\omega(\cdot)$ depend on the solution of (6).

Let $\omega(\mathbf{h}_i, i)$ maximize the objective function of (17), which is a fraction of $r(\mathbf{m}, i)$. The following normalization computes the time-fraction allocation of user $i$ in channel condition $\mathbf{h}$:

$$\omega(\mathbf{h}_i, i) = \frac{\omega(\mathbf{h}_i, i)}{\sum_{i=1}^{2} \omega(\mathbf{h}_i, i)}, \quad \forall \mathbf{h} \in \mathcal{H}(\mathbf{m}).$$

The time-fraction allocations $\omega(\mathbf{h}_i, i)$ are members of a set $\mathcal{F}_{MaOS} \triangleq \{ \omega(\mathbf{h}_i, i) : 0 \leq r_1 \leq 1 \text{ and Sol. of (17), (18)} \}$. Thus, the resulting capacity region for MaOS is expressed as

$$\mathcal{C}_{MaOS} = \bigcup_{\{\omega(\cdot) \in \mathcal{F}_{MaOS}\}} \left( C_1 = E_m \left[ E_h [\omega(\mathbf{h}_1, 1) R(\mathbf{h}_1, 1)] \right], C_2 = E_m \left[ E_h [\omega(\mathbf{h}_2, 2) R(\mathbf{h}_2, 2)] \right]\right).$$

The following theorem, proof of which is given in Appendix A, formalizes the relationship of the channel capacities of the optimum MaOS and LCS algorithms.

**Theorem 1.** The channel capacity of the MaOS algorithm (19) is greater than or equal to the channel capacity of the LCS algorithm (16), that is, $\mathcal{C}_{MaOS} \geq \mathcal{C}_{LCS}$.

### 5.2 Bounds on the Approximate MaOS

Recall that the first stage of the approximate MaOS algorithm distributes time fractions among $N$ users according to the sample paths of their mobility. For this purpose, the algorithm solves LP (10) after every $\tau$ mobility time intervals, where $\tau$ is the length of the predicted time window. The coefficients of decision variables $\hat{R}(t + \Delta, \cdot)$

\[ \begin{align}
\hat{R}(t + \Delta, \cdot) &= \max \sum_{i=1}^{2} R(\mathbf{h}_i, i) \phi(\mathbf{h}_i, i), \\
\text{s.t.} & \sum_{i=1}^{2} \phi(\mathbf{h}_i, i) \leq P(\mathbf{h}), \quad \forall \mathbf{h} \in \mathcal{H}(\mathbf{m}), \\
& \sum_{\mathbf{h}} \phi(\mathbf{h}_i, i) = r_1, \quad i = 1, 2, \quad r_2 = 1 - r_1, \\
& \phi(\mathbf{h}_i, i) \geq 0, \quad \forall \mathbf{h} \in \mathcal{H}(\mathbf{m}), \quad i = 1, 2.
\end{align} \]
are expected feasible rates for users at (mobility) time $t + \Delta$. These coefficients are random processes that are functions of user mobility and cell characteristics. Thus, the resulting LP (10) is a stochastic LP with random rewards.

In contrast with MaOS, an opportunistic scheduling algorithm like LCS does not consider user mobility. Therefore, it does not take advantage of the slow time variations in the radio channel. Although it considers fast time channel variations in its scheduling decision, it also tries to satisfy long-term fairness constraints. Because user mobility takes place on a slower timescale than scheduling decisions, the LCS algorithm may end up satisfying portions of the fairness requirements on per mobility time slots. This behavior is common in the restrictive constraint regime, that is, when $\epsilon \approx 1$. Thus, the LCS behavior can be approximately modeled as a resource distribution system that assigns an $r_i$ fraction of time to user $i$ at every mobility slot. If we like to write an equivalent objective function similar to (10) for the LCS model, we will consider coefficients to be constant and equal to the time average of the expected feasible rates. By abuse of notation, we use $E_r[\hat{R}(\tau, i)]$ to represent the time average of the expected feasible rate of user $i$ for $\tau$ mobility slots. The resulting objective function values are

$$ u = \tau \sum_{i=1}^{N} E_r[\hat{R}(\tau, i)] r_i, \tag{20} $$

where $u$ gives the lower bound on the objective function of (10).

For the approximate MaOS, it is interesting to find the effects of the length of the sample path $\tau$ and the number of users $N$ on the quality of the solution of (10). For this purpose, we extend DFM, which gives upper bounds on the expected optimal cost of a minimization LP with random coefficients [17], [18]. Our extension allows the determination of lower bounds on the expected optimal value of a maximization problem. In particular, the extension provides a relationship between the LCS model’s objective function (20) and the optimal value of (10).

In this section, we briefly summarize the DFM result and then present our extension. For notational convenience, the general result relies on the notation of the standard form LP as follows:

\begin{align}
\text{min or max} \quad & u = \sum_{j=1}^{J} c_j x_j, \tag{21a} \\
\text{s.t.} \quad & \sum_{j=1}^{J} a_{kj} x_j = b_k, \quad k = 1, 2, \ldots, K, \tag{21b} \\
& x_j \geq 0, \quad j = 1, 2, \ldots, J, \tag{21c}
\end{align}

where $x_j$s are optimization variables. The coefficients $c_j$ are independent nonnegative random variables, but constraint coefficients $a_{kj}$ and the right-hand side $b_k$ are fixed constants. There are $J$ decision variables and $K$ equality constraints.

According to the DFM result [17], [18], if it is possible to find $\beta$, $0 < \beta \leq 1$, such that

$$ E(c_j \mid c_j \geq l) \geq E(c_j) + \beta l, $$

then, for the minimization LP (21), the following inequality gives the upper bound on the expected optimal objective function $E(u^*)$:

$$ \beta E(u^*) \leq \sum_{j=1}^{J} \hat{x}_j E(c_j), \tag{22} $$

where $\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_J$ is any fixed feasible solution.

In its present form, the inequality in (22) is not suitable for maximization problems because of two reasons. First, it requires positive coefficients. Second, the dual of maximization problem has deterministic coefficients. Therefore, the dual problem does not fall within the DFM scope, which requires random coefficients. The following theorem extends the DFM inequality for the maximization problem (21). The proof of this theorem is given in Appendix B.

**Theorem 2.** Suppose $c_j$, $1 \leq j \leq J$, are independent nonnegative random variables in the interval $[0, 1]$. Suppose that, for all $l > 0$ and $P(c_j \leq l) > 0$, we have

$$ E(c_j \mid c_j \leq l) \leq \beta l. \tag{23} $$

Let $\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_J$ be any fixed feasible solution and $u^*$ be the value of the optimal solution to the maximization LP (21); then, we have

$$ \sum_{j \in B_k} E(c_j) \hat{x}_j + \beta \sum_{j \notin B_k} E(c_j) \hat{x}_j \leq \beta E(u^*), \tag{24} $$

where $B_k$ are feasible bases.

In order to translate the above result to our problem (10), we assume that $\hat{R}(\tau, \cdot)$ are normalized average data rates in the interval $[0, 1]$, where, for convenience, we use $\hat{R}(\tau, \cdot)$ in place of $\hat{R}(t + \Delta, \cdot)$. Furthermore, we observe that $J = N \tau$, the size of $B_k = K$, and $K = N + \tau$. Thus, the first summation on the left side of (24) contains $J - K$ terms. Increasing $N$ and $\tau$ results in a linear increase in $K$ and a quadratic increase in $J$. Furthermore, $\beta < 1$; therefore, the relative contribution of the second sum on the left side of (24) decreases when $N$ and $\tau$ increase. Thus, we can ignore this term at high values of $N$ and $\tau$. The remaining first sum on the left side can be approximated to the lower bound of (20). Moreover, if we assume that $\hat{R}(\tau, i)$ are i.i.d. and $\epsilon = \sum_{i=1}^{N} r_i = 1$, then (24) reduces to

$$ E_r[\hat{R}(\tau, \cdot)] \leq \beta E(u^*)/\tau. \tag{25} $$

As an example, we consider a special case when $\hat{R}(\tau, \cdot)$ are uniform i.i.d. random variables in the interval $[0, 1]$. Then, $E_r[\hat{R}(\tau, \cdot)] = 1/2$ and $\beta = 1/2$. It means that, in order to satisfy (25), the optimal value per mobility slot of (10), which turns out to be $E(u^*)/\tau$, should approach 1.

The numerical results validate the above argument. Fig. 2 plots the expected optimal value per mobility slot of (10) when the coefficients of the decision variables are uniform i.i.d. random variables. By increasing $\tau$ and $N$, the expected optimal value per mobility slot approaches the ideal value of 1. This means that, if large number of mobile users ($N$) are sharing a channel and a sufficiently large window size ($\tau$) is employed, then the approximate MaOS can achieve a performance close to the maximum possible data rate.
In the real case, $R(\tau,.)$ would not be uniform i.i.d. random variables. However, for any general distribution, it is possible to find a $\beta$ that satisfies (23). Thus, (25) can still be used to provide a lower bound on the optimum solution.

### 6 Implementation Details

This section provides implementation details of the MaOS algorithms. It first identifies some of the candidate mobility estimation and prediction techniques from the literature. Next, it describes a stochastic approximation technique for the estimation of Lagrange multipliers.

#### 6.1 Mobility Estimation and Prediction

The optimal MaOS algorithm only needs to estimate the current mobility state of every user. It is possible to correctly estimate the user mobility state because the mobility model considers a large area within a cell as a state. A user anywhere in that area is considered to be in that state. Monitoring the respective radio signal strength or alternative technologies like the angle of arrival (AoA) and time delay of arrival (TDoA) may be used for location estimation [26]. Furthermore, the presence of two timescales helps in filtering out errors in the estimation process.

The approximate MaOS algorithm also needs to predict the mobility state in the future. We propose the following alternative techniques:

1. Straight line movement witnessed on highways mostly follows deterministic mobility with constant speed and velocity [27]. Therefore, the future trajectory can be accurately predicted.
2. For stochastic user mobility, the approximate MaOS algorithm can employ a hierarchical mobility model. The higher layer of this model will have a discrete-state space like the one presented in this paper (Section 2.1). The lower layer of the model will have a continuous-state space modeled by some linear dynamical system. This continuous-state-space model will help to monitor not only the state, but also the fine-grain location information in the form of coordinates, velocity, and acceleration [28], [29]. The fine-grain mobility information will help in predicting the future discrete states to be visited by a user. For example, [28] showed a fairly accurate prediction of the next state several seconds ahead of time. With the help of fingerprinting of the road network, the future prediction of mobility states can be further improved.

#### 6.2 Estimation of Lagrange Multipliers

Like LCS, the optimal and approximate MaOS algorithms also use a stochastic approximation technique for the estimation of Lagrange multipliers. The optimal MaOS sets the initial value of $v_{(.)} = 0 \forall i$ and $\mathbf{\bar{m}}$ and maintains a database for every mobility state $\mathbf{\bar{m}}$, where it holds the most recent estimated values of $v(.)$ and supporting parameters used in its computation. Whenever the system enters into mobility state $\mathbf{\bar{m}}$, the algorithm retrieves the stored values from the database. As long as the system remains in state $\mathbf{\bar{m}}$, it updates the values of $v(.)$ for every scheduling slot $k + 1$ using the following computation [4]:

$$v^{k+1}(\mathbf{\bar{m}}) = \max \left(v^k(\mathbf{\bar{m}}) + \delta \left(1_{(\mathbf{\bar{m}}(\mathbf{\bar{R}}(\mathbf{\bar{m}}) = \mathbf{\bar{m}}))} - \hat{v}(\mathbf{\bar{m}}, i)\right), 0\right).$$

(26)

We use a small constant $\delta$ as the step size because the corresponding LCS algorithm, being indifferent to the user mobility information, views the piecewise stationary channel as a nonstationary process. For nonstationary processes, the stochastic approximation technique recommends a constant step size [4]. The presence of a database to store the values of $v(.)$ and supporting parameters for future use and the difference in timescales between mobility transitions and scheduling decisions guarantees the saturation of (26).

The approximate MaOS cannot store the values of $v(.)$ into a database because it lacks the complete knowledge of the state space. Therefore, whenever a new state is visited, the algorithm resets $v(.)$ and the supporting parameters. Next, it uses an expression similar to (26) for subsequent scheduling and mobility slots until the next mobility transition. The timescale difference between mobility transitions and scheduling may allow the approximate MaOS to achieve saturation.

### 7 Numerical Results of the Optimal MaOS

In this section, we report our simulation results for optimal MaOS algorithms for the HDR data set. First, we report implementation details for the simulation setup, followed by the performance and the channel capacity results for the MaOS and LCS algorithms. We also compare these algorithms with the MR scheduling that always selects the best user.

#### 7.1 Basic HDR Simulation Setup

A hexagonal cell that has a maximum coverage distance of 5 km from the central BS to a mobile user forms the service area. The service area is divided into two concentric rings ($m = \{1, 2\}$). The innermost ring covers an area of up to 3 km from the BS. The BS’s maximum output power is
15 W, and 80 percent of this power is used for the shared data channel [30]. The system provides HDR data service with feasible data rates of

\[
\{2.457,6, 1.843,2, 1.228,8, 921.6, 614.4, 307.2, 204.8, 153.6, 102.6, 76.8, 38.4\}
\]

kilobits per second (Kbps) and corresponding SINRs of [1]

\[
\{9.5, 7.2, 3.0, 1.3, -1.0, -4.0, -5.7, -6.5, -8.5, -9.5, -12.5\}\text{ dB}.
\]

The average SINR \(\rho(n)\) at the center of the rings is found from the path loss model in (3) for \(\alpha = 4, y_0 = 1\text{ km},\) and \(\eta = 1\text{ W}.\) The steady-state channel distribution is determined using (5) for a speed of 60 km/h. The resulting expected feasible rate values are 1,438 Kbps in the inner ring and 116 Kbps in the outer ring. The scheduling decisions are made at every 1.67 ms.

For the two concentric rings, the coarse-grained mobility model is learned through the simulation of a user following a fine-grained mobility model. For our simulation, we considered random-walk mobility [21] with a constant speed of 60 km/h to model fine-grained user movement. Travel intervals are randomly distributed between 8 to 12 minutes. After completing one travel interval, the user selects a new direction randomly from 0 to \(2\pi.\) When the user moves out of the service area, it wraps around and reenters the service area. Mobility slots are of one second duration. We emphasize that the choice of identical mobility and channel distributions is only for the ease of simulation. The proposed algorithm can handle nonidentical cases equally well.

The following results are for eight users in the system with infinite backlog; Half of the users require (minimum) fairness requirements of \(\epsilon/6\) and the other half expect at least \(\epsilon/12\) fraction. In (26), we set \(\delta = 0.05.\)

### 7.2 Performance Comparison

Fig. 3a plots the system data rate (normalized with respect to the maximum achievable data rate of 2,457.6 Kbps) for the LCS and MaOS algorithms for \(\epsilon = \{0.99, 0.95, 0.90\}.\) For completeness, this figure also reports the performance of the MR algorithm, which is a greedy algorithm without any regard to fairness. Because, here, we are concentrating on performance, we set \(\theta = 0\) in (6).

The results show that MaOS performs better than LCS because it exploits not only the fast fading fluctuations but also the slow time path loss variations in the radio channel. First, through the solution of (6), MaOS finds optimum user priorities, in the form of time fractions that act as fairness constraints, to take advantage of the slow time path loss fluctuations. These priorities are such that users with strong channels, that is, those in the inner ring at any given time, are preferred over weak users, that is, those in the outer ring at the same time. Next, it applies these priorities opportunistically through the solution of (8) to gain from the fast time fading variations. This overscheduling of strong users allows MaOS to avoid them when they transition to weak mobility states. Similarly, weak users at any given time do not remain weak indefinitely. They move closer to the BS, and that is when MaOS compensates them for their loss. The net result is that all users get substantially data rates higher than or equal to LCS as shown in Fig. 3b. This figure compares the average data rates achieved by all eight users under the MR, LCS, and MaOS algorithms. Because the first four users have higher fairness requirements, their data rates are higher than the remaining four users.

On the other hand, LCS suffers performance losses in order to support constant fairness constraints. These constant fairness constraints force the LCS algorithm to schedule users when they are in the unfavorable location. Although the channel information is considered in scheduling, this information only allows it to benefit from the fast time fading while slow time path loss variations remain untapped. Thus, LCS ends up satisfying fairness constraints for users when they have weak channels.

The performance improvements seen in the case of MaOS have been achieved without sacrificing the long-term fairness requirements, as shown in Fig. 3c. This figure compares the temporal fairness requirements for all users to their actual allocations performed by the scheduling
algorithms. The MR, being a greedy algorithm, failed to support the required fairness measure. It gave equal access to all users, not because of any conscious decision on the part of the algorithm, but because of identical mobility and channel distributions used in our simulation. The other algorithms considered in the comparison, namely, MaOS and LCS, are able to satisfy the minimum fairness requirements of $r_{1...4} = \epsilon/6$ and $r_{5...8} = \epsilon/12$.

Returning to Fig. 3a, we observe that increasing the value of $\epsilon$ significantly increases the performance gap between MaOS and LCS. Although the MaOS performance does not change significantly, the LCS performance decreases with the increase of $\epsilon$. We observed that, for $\epsilon$ values of 0.90, 0.95, and 0.99, MaOS performs better than the LCS algorithm by a margin of 11.4, 16.8, and 23.2 percent, respectively. The performance improvement of MaOS in comparison with LCS is expected to be significant in the restrictive constraint regime when $\epsilon$ is close to 1, for example, $\epsilon = 0.99$. According to [5], in the restrictive constraint regime, the LCS algorithm has less opportunity to improve the system’s performance. This loss of opportunity is due to the fact that almost all constraints are binding for all time durations; therefore, the scheduler cannot overschedule any user. When a user has a weak channel for long duration of time, the binding constraint forces the scheduling of that user to satisfy its fairness requirement. On the other hand, MaOS handles such a user by scheduling it more than its long-term share when it is in favorable locations. This extra scheduling saves MaOS from scheduling that user when it is in unfavorable locations resulting in the performance improvement.

When $\epsilon$ decreases, fewer constraints are binding and LCS can overschedule strong users more than their required share. Thus, when these users are in unfavorable locations, the LCS algorithm can avoid them for a longer duration of time. This is the reason for the reduction in the performance gap between the MaOS and LCS algorithms when $\epsilon$ decreases.

### 7.3 Delay Comparison

In the previous section, MaOS performed significantly better than the LCS algorithm, but the improvement is at the cost of adverse delay statistics. The results presented there are for $\theta = 0$. For these results, the first batch of users, that is, users belonging to the set $\{1, 2, 3, 4\}$, experienced a head-of-line (HOL) packet delay of 8.59 s and a variance of 235.680 $s^2$. Here, the HOL delay is defined as the time gap between two consecutive allocations. The LCS algorithm faced an almost equal expected delay of 8.69 s, but its variance was only 34.728 $s^2$. The expected HOL delay is indifferent to both algorithms because of the temporal fairness measure that requires a certain minimum fraction of time for users.

The delay variance is significantly high for the MaOS algorithm. This high variance is because of the way MaOS distributes user priorities. When $\theta = 0$, MaOS completely avoids users in the unfavorable locations as long as there are other users in the favorable locations. Therefore, users in unfavorable locations can be starved of service. To overcome long-term starvation, the proposed algorithm provides a mechanism of ensuring minimum resource allocation to users even when they are in the most unfavorable locations so that their connection remains active and higher layer applications can perform reasonably. The mechanism is to use a nonzero value of $\theta$ in (6). When $\theta > 0$, MaOS lower bounds user preferences (that is, fairness) by the constraint (6d). The lower bound considerably decreases the MaOS delay variance, as shown in Table 2 for different values of $\theta$.

For example, even when $\theta$ is very small, like 0.1, the delay variance decreases to 2.366 $s^2$. This value is even better than the corresponding LCS value. By increasing $\theta$, we can reduce delay variance for MaOS. This pattern is observed for the second set of users $\{5, 6, 7, 8\}$ as well. For the data set considered in Table 2, $\theta_{max}$ is 0.4.

The decrease in MaOS delay variance is due to the way it computes (26) and uses Lagrange multipliers in (9). The algorithm associates Lagrange multipliers to the aggregate mobility state $\bar{m}$. Therefore, it is able to comply with the minimum fairness bounds for the $\theta > 0$ case for every mobility state resulting in lower variance values. On the other hand, the LCS algorithm does not associate corresponding Lagrange multipliers to the mobility states. It computes and uses them for the whole time horizon. Though it is able to provide asymptotic fairness, it does not guarantee strict fairness on a much reduced timescale like mobility state. If LCS is also forced to comply with the fairness for every mobility state $\bar{m}$, then we identify the resulting algorithm as LCS mobility state fair (MSF). For this compliance, LCS MSF associates Lagrange multipliers with $\bar{m}$ but retains the static constraints of LCS. With this modification, LCS MSF provides the best delay variance among the competition (see Table 2). For completeness, we also report the corresponding values for the MR algorithm. The MR algorithm does not differentiate users according to their fairness requirements. Therefore, all users experienced almost comparable values.

The decrease in delay variance achieved by MaOS for $\theta > 0$ and LCS MSF are not without loss. The loss is in the form of reduction in the average system data rate. Fig. 4 shows the average data rates for MR, LCS, LCS MSF, and MaOS. Recall that $\theta$ affects only the MaOS algorithm.

### Table 2

Comparison of Delay Statistics

<table>
<thead>
<tr>
<th>Users</th>
<th>Expected Delay (s)</th>
<th>Delay Variance ($s^2$)</th>
<th>Delay Variance ($s^2$) of MaOS as a function of $\theta$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LCS</td>
<td>MaOS</td>
<td>LCS MSF</td>
</tr>
<tr>
<td>1 → 4</td>
<td>8.69</td>
<td>8.59</td>
<td>8.43</td>
</tr>
<tr>
<td>5 → 8</td>
<td>17.10</td>
<td>17.44</td>
<td>18.03</td>
</tr>
</tbody>
</table>
7.4 Effects of Parameters

In another experiment, we change the system parameters to observe their impact on the performance of the proposed scheme. In this new experiment, every cell has a radius of 2,000 m and \( y_0 = 500 \) m. The mobility model consists of three mobility states of concentric rings. The ring radii in increasing order are 800, 1,400, and 2,000 m. For the fine-grain mobility model, users drive in straight lines with constant speeds. There were six users in the system with fairness requirements of \( r_{1-3} = \epsilon/9 \) and \( r_{1-6} = \epsilon/4.5 \). The other model parameters are identical to those reported in Section 7.1.

The average data rate of each algorithm in normalized units is given as follows: MR (0.82), LCS (0.56), MaOS (0.78 for \( \theta = 0 \)), and LCS MSF (0.44). MaOS is able to significantly improve performance in comparison to the LCS algorithm. In general, we have observed that, for macrocellular structures where free-space path loss variations significantly affect the channel quality and users experience those variations, MaOS can provide increased performance.

7.5 Capacity Gain

Using the mobility and channel parameters in Section 7.1, we plot the capacity region for two users using (13), (16), and (19) in Fig. 5. This figure also contains simulated capacity regions generated by the LCS and optimal MaOS algorithms for \( \epsilon = 0.99 \) and \( \theta = 0 \). The simulated and numerical results validate that MaOS provides a larger capacity region than the LCS algorithm.

8 Numerical Results of the Approximate MaOS

This section provides simulation results for the approximate MaOS algorithm. After describing the simulation setup, it reports performance comparisons and carries out a robustness analysis of the algorithm.

8.1 Extended HDR Simulation Setup

For the performance analysis of the optimal MaOS algorithm, a two or three-state mobility model was considered because of the exponential complexity in the number of states. However, for the approximate MaOS, we extend the model to a multiple-state model. The extended model has several mobility states in accordance with the average SINR values of the HDR data set.

We extend the topology shown in Fig. 1a with multiple concentric rings. In the extended model, each ring corresponds to a SINR level in the HDR data set; the ring order, from the innermost to the outermost, follows the descending order of the SINR levels. As there are 11 SINR levels in the HDR data set (Section 7.1), if the cell radius \( Y \) permits, then there can be 11 possible rings on the cell surface. A user at a distance \( y \) from the BS, where \( y \leq Y \), will fall in ring \( m \) if

\[
\begin{aligned}
\rho(1) &\leq \frac{P_{\text{TH}^m(1)}}{P_{\text{TH}^m(0)}} \quad m = 1, \\
\rho(m) &> \frac{P_{\text{TH}^m(0)}}{P_{\text{TH}^m(0)}} < \rho(m-1) \quad m = 2, \ldots, 11.
\end{aligned}
\]

For the 11 SINR levels in the HDR system, we compute the corresponding outer radii using (3), (4), and (27). The outer radii in increasing order are

\[
\{1,070, 1,222, 1,556, 1,715, 1,957, 2,325, 2,562, 2,682, 3,005, 3,181, 3,764 \} \text{ m}.
\]

These values are found for the following system parameters: \( y_0 = 1 \) km, \( \alpha = 4 \), \( Y = 4 \) km, \( \eta = 1 \) W, \( P_1 = 15 \) W, and \( P = 0.8P_1 \) [30]. The steady-state probability of channel conditions is found according to (5).

There are 12 users in the cell and they follow a random-walk mobility model [21]. The mobility model parameters are identical to those given in Section 7.1 except for user speed. For one experiment, users move with a speed of 60 km/h and, for the other experiment, they follow a speed of 30 km/h. The fairness requirements for each user are \( \epsilon/12 \), where \( \epsilon = 0.99 \).

8.2 Performance Comparison and Effects of the Future Sample Path

Fig. 6 plots the system data rate for the MR, MaOS, and LCS algorithms as a function of \( \tau \) in (10). Except MaOS, all the other algorithms are independent of \( \tau \). From this figure, it is clear that an increase in the sample path size (\( \tau \)) improves
the data rate of the approximate MaOS algorithm. This is in line with the theoretical result of Section 5.2.

However, for small values of $\tau$, the approximate MaOS can perform worse than LCS. For small $\tau$, users do not cover long distances; therefore, distance change does not significantly affect the data rate. Thus, the chances of performance improvement by MaOS are limited in this case. On top of that, because of the short time duration, the algorithm makes several quick changes in the constraints of (11). These changes make it difficult for the approximate MaOS to find true Lagrange multipliers. Therefore, in order to satisfy fairness, the algorithm ends up reducing overall performance. However, if the algorithm knows Lagrange multiplier values beforehand, then it can recover some of this loss. For example, Fig. 6 also plots the MaOS algorithm performance with known (estimated) values of Lagrange multipliers. These values were computed in an earlier simulation for the exact user mobility and channel statistics. With known multiplier values, the approximate MaOS achieved a comparable performance to LCS when $\tau \approx 120$ s, which is better than the original value of $\tau \approx 180$ s. With the increase of $\tau$, MaOS performs better because a larger time window allows it to take advantage of the path loss variations in the feasible rates. The LCS and MR algorithms are independent of the sample path size.

The increase in speed from 30 km/h to 60 km/h increases the performance of all algorithms because frequent changes in radio channel provide more opportunity to improve the data rate.

### 8.3 Robustness of the Approximate MaOS Algorithm

Future prediction of mobility states is crucial for the success of the MaOS algorithm. In this section, we simulate the effects of prediction errors on the performance of the algorithm. We study the approximate MaOS for the simulation setup in Section 8.1 under two scenarios.

The first scenario assumes that errors are independent and only restricted to the adjacent rings. In this scenario, the predictor makes an error with some probability $\gamma$. When it makes an error, it erroneously predicts one of the adjacent rings with equal probability. For situations where there is only one adjacent ring, that is, the correct ring is either the innermost or the outermost ring, the predictor selects the only adjacent ring.

The second scenario assumes that the errors are independent, but once an error is made with probability $\gamma$, the predictor can select any ring with equal probability. This is an extreme scenario as most of the prediction errors in real situations occur close to the actual location.

The MaOS performance is shown in Fig. 7 for both scenarios as a function of the probability of error $\gamma$ for the future prediction of 800 s. When $\gamma = 0$, the predictor is ideal with no errors. From this figure, we see that, for the first case, MaOS experiences graceful performance loss as $\gamma$ increases. Even when all the rings are erroneously replaced by their neighbors, that is, $\gamma = 1$, the MaOS algorithm achieved a performance of 0.51, which is higher than the LCS algorithm. The adjacent cell errors do not cause a serious dent in the performance of the algorithm because the expected data rates of adjacent rings do not differ significantly. For the second scenario, the performance loss is considerably higher because of the extreme types of errors. Even in this case, performance loss is graceful and it remains higher than the corresponding LCS for $\gamma$ as high as 0.5, that is, when half of the rings are incorrectly predicted and replaced by another ring with equal probability.

The prediction errors do not result in fairness violations as the temporal fairness measure considers fraction of time assignment, which remains independent of errors.

### 9 Conclusion

This paper proposes algorithms that combine channel and mobility information in the scheduling of downlink data packets in cellular networks. The proposed MaOS algorithms improve on the LCS algorithm in [5] by dynamically adapting long-term fairness constraints according to the mobility information of a user. The MaOS algorithms compute constraint values such that the scheduler gives preference to users that are in the most favorable locations with high expected feasible rates. This extra scheduling
helps the scheduler to avoid such users when they enter mobility states with bad channels. The optimal MaOS algorithm precomputes constraint values for all states according to a discrete-state-space model. The approximate MaOS algorithm computes constraint values according to the future prediction of mobility states. With the help of the extended DFM inequality, we prove that lengthening the future prediction and increasing the number of users in the network can achieve the maximum possible performance. The extended inequality is also capable of handling more general cases and provides lower bounds on the performance of the approximate MaOS. Simulation results show that the proposed algorithms perform better than LCS and satisfy the fairness constraints. Moreover, it is observed that the improvement is significant in the restrictive constraint regime. The improvement is observed without increasing the long-term expected delay, though delay variance increases. The proposed algorithm provides a mechanism to control the value of delay variance. The use of mobility information in opportunistic scheduling also increases channel capacity. The performance improvements are at the expense of complex algorithms that require additional information in the form of mobility state estimation and prediction.

**APPENDIX A**

**Proof of Theorem 1**

Proof. LP (14) and (17) differ from each other with respect to constraints (14c) and (17c). Thus, it is sufficient to show that (17c) results in a larger set than (14c).

Let \( \phi^*(\bar{m}, i) \) maximize the objective function of (14). Then, multiplying both sides of (14c) by \( \bar{R}(\bar{m}, i) \) and summing \( \forall i \) and \( \forall \bar{m} \in S \), we get

\[
q^* := \sum_{m \in S} \sum_{i=1}^{2} \bar{R}(\bar{m}, i) \sum_{i=1}^{2} \phi^*(\bar{m}, i) = \sum_{m \in S} \sum_{i=1}^{2} \bar{R}(\bar{m}, i) r_i.
\]

Similarly, let \( \omega^*(\bar{m}, i) \) maximize the objective function of (17). Repeating the above procedure on (17c), we get

\[
o^* := \sum_{m \in S} \sum_{i=1}^{2} \bar{R}(\bar{m}, i) \sum_{i=1}^{2} \omega^*(\bar{m}, i) = \sum_{m \in S} \sum_{i=1}^{2} \bar{R}(\bar{m}, i) \bar{r}(\bar{m}, i).
\]

From (7) and given \( r^*(\cdot) \), there is a linear mapping between \( o^* \) and the maximum objective function value of (6a). Thus, \( o^* \geq q^* \). \( \square \)

**APPENDIX B**

**Proof of Theorem 2**

Proof. Without loss of generality, assume that the constraint-coefficient matrix \( A = (a_{ij}) \) is of full rank \( K \), and (21) is nondegenerate. Further, assume that \( P \) is the polyhedral feasible region to the LP (21) and it has \( W \) feasible bases with \( B_1, B_2, \ldots, B_W \) being the corresponding index set. Let \( A_{B_j} \), \( 1 \leq j \leq W \), be a feasible base consisting of columns of \( A \) \( a_j = (a_{1j}, a_{2j}, \ldots, a_{Kj})' \), \( j \in B_e \), where \((\cdot)' \) represents the vector transpose operation.

By the optimality criterion of the maximization problem, \( B_e \) is optimal if and only if [18]

\[
c_j \leq c_{B_e} A_{B_e}^{-1} a_j, \text{ for all } j \notin B_e,
\]

where \( c_{B_e} \) is the row vector consisting of \( c_j \), \( j \in B_e \).

Let \( V_e \) represent an event when optimality condition (28) is satisfied. \( \cup_{j=1}^{W} V_e \) has a probability equal to one because problem (21) has at least one basic optimal solution. Applying optimality condition (28) and the conditional expectation hypothesis (23), we get an identity on expected values for \( c_j \) for \( j \notin B_e \):

\[
E(c_j \mid V_e, c_{B_e}) = E(c_j \mid c_j \leq c_{B_e} A_{B_e}^{-1} a_j) \leq \beta c_{B_e} A_{B_e}^{-1} a_j.
\]

Multiplying (29a) with feasible solution \( \hat{x}_j \) and summing over all \( j \), we get

\[
E \left( \sum_{j=1}^{J} c_j \hat{x}_j \mid V_e \right) \leq \sum_{j=1}^{J} c_j \hat{x}_j + \beta \sum_{j=1}^{J} \sum_{j \notin B_e} c_{B_e} A_{B_e}^{-1} a_j \hat{x}_j
\]

For optimal \( B_e \), the optimal reward \( u^* \) satisfies the following equation:

\[
u^* = c_{B_e} A_{B_e}^{-1} b.
\]

where \( b \) is the column vector consisting of the right side of (21b) constraints. Furthermore, the constraints of (21b) are satisfied, that is

\[
\sum_{j=1}^{J} a_j \hat{x}_j = b.
\]

Thus, applying (32) and (31) to (30b), we get

\[
E \left( \sum_{j=1}^{J} c_j \hat{x}_j \mid V_e \right) \leq \sum_{j=1}^{J} c_j \hat{x}_j + \beta \sum_{j=1}^{J} a_j \hat{x}_j + \beta E(u^* \mid V_e, c_{B_e}.)
\]

The \( a_j \) and \( A_{B_e} \) definitions result in

\[
c_j = c_{B_e} A_{B_e}^{-1} a_j, \quad j \in B_e.
\]

The above identity means that the coefficients of \( \hat{x}_j \) are \((1 - \beta)c_j \) in (33b). Thus, we get

\[
E \left( \sum_{j=1}^{J} c_j \hat{x}_j \mid V_e \right) + (\beta - 1) \sum_{j=1}^{J} c_j \hat{x}_j \leq \beta E(u^* \mid V_e, c_{B_e}.)
\]
Taking the expectation on values of $c_i$, conditioned on $B_i$, we get
\[
\sum_{e} P(V_e) \left( \sum_{j=1}^{J} c_j x_j | V_e \right) + \beta \sum_{e} \sum_{j \in B_i} E(c_j) x_j \\
\leq \sum_{j=1}^{J} E(c_j) x_j + \beta \sum_{j \in B_i} E(c_j) x_j \leq \beta E(u^*)
\]

\[\square\]

**ACKNOWLEDGMENTS**

This work was supported by a University of British Columbia Graduate Fellowship and by Canadian Natural Sciences and Engineering Research Council Grant STGP269872-03.

**REFERENCES**


**Syed Hussain Ali (S’05)** received the bache-
lor’s degree from Naderishaw Edulji Dinshaw (NED) University of Engineering and Technolog-
y, Karachi, Pakistan, in 1993 and the master’s degree from King Fahd University of Petroleum and Min-
ers (KFUPM), Dharhan, Saudi Arabia, in 1998, both in computer engineering. From 1993 to 1996, he worked on several commun-
ication and networking projects in Pakistan. From 1999 to 2003, he served as a lecturer and
etwork engineer at KFUPM, where he received several service awards for his performance. He is currently working toward the PhD degree in
the Department of Electrical and Computer Engineering, University of
British Columbia (UBC), Vancouver, Canada. He is a recipient of a UBC
graduate fellowship and a Josephine T. Berthier fellowship. His current re-
search interests include cross-layer design and resource allocation
strategies for wireless networks. He is a student member of the IEEE.
Vikram Krishnamurthy (S’90–M’91–SM’99–F’05) was born in 1966. He received the bachelor’s degree from the University of Auckland, New Zealand, in 1988 and the PhD degree from the Australian National University, Canberra, in 1992. Since 2002, he has been a professor and Canada research chair in the Department of Electrical Engineering, University of British Columbia, Vancouver, Canada. Prior to 2002, he was a chaired professor at the Department of Electrical and Electronic Engineering, University of Melbourne, Australia, where he also served as the deputy head of the department. His current research interests include stochastic modeling of biological ion channels, stochastic optimization and scheduling, and statistical signal processing. He has served as an associate editor for several journals including the IEEE Transactions on Signal Processing, IEEE Transactions on Aerospace and Electronic Systems, IEEE Transactions on Circuits and Systems B, IEEE Transactions on Nanobioscience, and Systems and Control Letters. He is a coeditor, along with S.H. Chung and O. Andersen, of the book Biological Membrane Ion Channels—Dynamics Structure and Applications (Springer-Verlag, 2006). He is a fellow of the IEEE.

Victor C.M. Leung (S’75–M’89–SM’97–F’03) received the BASc (Hons.) and PhD degrees in electrical engineering from the University of British Columbia (UBC), Vancouver, Canada, in 1977 and 1981, respectively. He was the recipient of many academic awards, including the Association of Professional Economists of British Columbia (APEBC) Gold Medal as the head of the 1977 graduating class in the Faculty of Applied Science, UBC, and the Natural Sciences and Engineering Research Council (NSERC) Postgraduate Scholarship. From 1981 to 1987, he was a senior member of the technical staff and a satellite systems specialist at MPR Teltech Ltd. in Burnaby, B.C., Canada. He was also a part-time visiting assistant professor in the School of Engineering Science at Simon Fraser University from 1986 to 1987. In 1988, he was a lecturer in the Department of Electronics, Chinese University of Hong Kong. He returned to UBC as a faculty member in 1989, where he is now a professor and holder of the TELUS mobility research chair in advanced telecommunications engineering in the Department of Electrical and Computer Engineering. He also holds an honorary guest professorship at Jilin University, China. His research interests are in mobile systems and wireless networks. He has contributed seven book chapters and coauthored more than 300 papers in international journals and conferences in these areas. He is a fellow of the IEEE and the IEEE Computer Society and a voting member of the ACM. He is an editor of the IEEE Transactions on Wireless Communications, an associate editor of the IEEE Transactions on Vehicular Technology, and an editor of the International Journal of Sensor Networks. He has also served on several journal issues. In addition to serving as a technical program committee (TPC) member in numerous international conferences, he has also served as the general cochair of the Eighth ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM 2005), the chair of the Symposium on Next Generation Mobile Networks (NGMN) and the local chair of the International Wireless Communication and Mobile Computing Conference (IWCMC) 2006, and the TPC associate chair for networking of the IEEE Wireless Communications and Networking Conference (WCNC) 2005. He is serving as the general chair of the Fourth International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness (QShine 2007) and the NGMN Symposium chair of IWCMC 2007.

For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.