Optimal Scheduling in High Speed Downlink Packet Access Networks

HUSSEIN AL-ZUBAIDY, IOANNIS LAMBADARIS and JEROME TALIM
Carleton University

We present an analytic model and a methodology to determine the optimal packet scheduling policy in a High Speed Downlink Packet Access (HSDPA) system. The optimal policy is the one that maximizes cell throughput while maintaining a level of fairness between the users in the cell. A discrete stochastic dynamic programming model for the HSDPA downlink scheduler is presented. Value iteration is then used to solve for the optimal scheduling policy. We use a FSMC (Finite State Markov Channel) to model the HSDPA downlink channel. A near optimal heuristic scheduling policy is developed. Simulation is used to study the performance of the resulted heuristic policy and compare it to the computed optimal policy. The results show that the performance of the heuristic policy is very close to that of the optimal policy. The heuristic policy has much less computational complexity which makes it easy to deploy, with only slight reduction in performance compared to the optimal policy.

Categories and Subject Descriptors: C.2.1 [**Computer Systems Organization**]: Computer Communication Networks, Network Architecture and Design—*wireless communication*; G.1.6 [**Mathematics of Computing**]: Numerical Analysis, Optimization—*Stochastic programming*; *unconstrained optimization*; G.1.10 [**Mathematics of Computing**]: Numerical Analysis—*applications*; G.3 [**Mathematics of Computing**]: Probability and Statistics—*stochastic processes*; *Markov processes*; I.6.3 [**Computing Methodologies**]: Simulation and Modeling—*applications, model development*

General Terms: Algorithms, Design, Performance, Theory

Additional Key Words and Phrases: Markov decision process, dynamic programming, optimal scheduling, resource allocation, HSDPA systems, 3G wireless networks, cross-layer design

1. INTRODUCTION

In this paper, we investigate the problem of scheduling in emerging wireless networks, specifically the High Speed Downlink Packet Access (HSDPA). We focus on theoretical, performance and implementation sides of the optimal scheduling policy in this system. HSDPA is a 3G wireless network that provides high cell peak data rate (up to 14.4 Mbps for Revision 5) on the downlink by incorporating Adaptive Modulation and Coding (AMC), Hybrid ARQ and fast scheduling [Holma and Toskala 2004][Castro 2004]. The development of this system represents an im-

Author's address: Systems and Computer Engineering, Carleton University, 1125 Colonel By Drive, Ottawa, ON, Canada, K1S 5B6. Email: {hussein, ioannis, jtalim}@sce.carleton.ca
This work was supported by Mathematics of Information Technology and Complex Systems (MITACS) and Natural Sciences and Engineering Research Council of Canada (NSERC).

portant step in the effort to establish an ubiquitous wireless access paradigm that provides a host of services in a cost effective, high speed and reliable manner. This motivated our initial interest in this problem.

Our work in this paper also lies in the broader area of optimal control of queuing systems. There are three major techniques that were used extensively to study optimal control in such systems; Dynamic Programming (DP), Linear Programming (LP) and Stochastic Dominance. Dynamic Programming is the traditionally favored technique for optimization [Stidham 1985][Hajek 1984][Rosberg et al. 1982][Lin and Kumar 1984]. In this technique, a cost function is defined as a function of the system state and the sequence of decisions made by the working policy. Then using *value iteration* or *policy iteration*, one can identify the policy (or at least some of its properties) that minimizes the cost function. When the underlying random process resulted from the control problem is a Markov Decision Process (MDP), then a feasible solution to the formulated DP equation is possible in most cases. However, when the DP equation becomes somewhat complicated (i.e., has complex formulation), then a solution may be hard to find. Similar statement is true when the underling process is a Semi-Markov Decision Process. The DP equation can also be solved numerically. However, when the state space and/or action space grow large, then the computational complexity will rapidly increase and may become a prohibiting factor. For more details on Dynamic Programming, the reader may refer to [Puterman 1994][Ross 1983][Kumar and Varaiya 1986].

Linear Programming (including Integer LP) is another technique that can be used to solve optimization problems in queuing systems. LP concerns the problem of maximizing (or minimizing) a linear functional over a polyhedron [Schrijver 1986][Rosberg et al. 1982]. In this technique, a linear functional, representing the objective or cost, is maximized (or minimized) over a set of constraints (linear system of inequalities). Stochastic dominance is another important technique in stochastic modeling, analysis and optimization [Stoyan 1983]. A useful result from stochastic dominance is the *stochastic coupling* of random variables and processes. Stochastic coupling is used in literature to prove optimality in many scheduling and routing problems, for example, [Walrand 1984][Tassiulas and Ephremides 1993][Ganti et al. 2007][Al-Zubaidy et al. 2009].

Scheduling in HSDPA systems involves not only *transmission time intervals (TTI) allocation* but also *code allocation* [Castro 2004]. On the downlink, HSDPA uses Code and Time Division Multiplexing (CDM/TDM) and has 15 codes to be allocated per TTI. Most of the available work in scheduler design (e.g. [Wei and Izmailov 2004], [Bonald 2004] and [Jeon et al. 2004]) is based on the intuition and creativity of the designers. The designer usually selects an optimization criterion that represents some important performance measure (in his/her opinion), builds an algorithm based on that criterion and then tries to establish confidence in it using backward analysis or simulation. This method can be described as a *procedural approach.* This, most likely, will result in a suboptimal algorithm at the best, that performs well in some setups and poor in others. This happens especially in systems such as HSDPA, since it uses a very complex set of features such as Hybrid Automatic Repeat reQuest (H-ARQ) and Adaptive Modulation and Coding (AMC). These features introduced many new and interrelated tuning parameters which cannot be grasped by one selected optimality criterion. Another observation is the lack of work on schedulers that dynamically allocate codes as well as time slots to the users in an HSDPA system. Scheduling related processes in this system span across both layer 1 (physical) and layer 2 (media access control). Hence, a cross-layer design is desirable for such systems [Jiang et al. 2005].

In literature there are many efforts to model and optimize schedulers in wireless systems. In one well-known study [Tassiulas and Ephremides 1993], the authors investigated an optimal scheduling problem in wireless networks with one server that is shared by several users (queues) with random connectivity. They proved that the LCQ (Longest Connected Queue) policy minimizes the total number of packets in the system. In [Koole et al. 2001] a model similar to that of [Tassiulas and Ephremides 1993] was studied. In that work, the Best User (BU) policy was proven to maximize the expected discounted number of successful transmissions. Liu et al studied the optimality of opportunistic schedulers (e.g., Proportional Fair (PF) scheduler) under certain conditions [Liu et al. 2003]. They presented the characteristics and optimality conditions for such schedulers. However, Andrews showed that there are six implementation algorithms of PF scheduler, none of them are stable [Andrews 2004].

In this work we built and analyzed a realistic model of an optimal HSDPA scheduler using stochastic dynamic programming. This model introduces a simplifying abstraction of the real scheduler which estimates system behavior under different operating conditions and describes the role of various system components. This model can be readily solved numerically to obtain the optimal code allocation policy for a given *objective function*.

This approach can be considered as a *unified approach* since the same model can be used when solving for different objective function by simply changing the reward associated with the model to reflect a new optimality objective. Different objective functions may result in different optimal policies. The proposed approach produces an optimal scheduling policy in the sense that it maximizes cell throughput for a given fairness. It provides an elegant and presentable analytic foundation for scheduling problems and may be used as a *benchmarking tool* for other (suboptimal) schedulers or to test against heuristics.

The presented approach can be used to tackle optimal packet scheduling in the most recent releases of 3GPP (e.g., Long Term Evolution (LTE)). These systems use the same basic components at the link layer level (e.g., HARQ and AMC) that are used in HSDPA system. They also use other new technologies in the physical layer (that was not used in HSDPA systems) to increase downlink/uplink data rates up to 150Mbps/50Mbps, namely orthogonal frequency-division multiplexing (OFDM) and multiple-input multiple-output (MIMO) antenna system [Ekstrom et al. 2006][Peisa et al. 2007]. The packet scheduler in these systems is responsible for frequency allocation (frequency carriers that correspond to OFDM) as well as CDM codes during each TTI [Kela et al. 2008][Monghal et al. 2008]. The model we are presenting here may be extended for the LTE system. Overall, the contributions of this work can be summarized by the following:

(1) We provide analytical approach to model the downlink scheduler in 3G HSDPA system. This approach can be extended to other 3G/4G wireless systems.

(2) Using the theory of Dynamic Programming we present an optimization framework for the determination of the optimal policy for the HSDPA downlink packet scheduler. The applicability of this framework is demonstrated using a two user/queue case (in this case the pictorial visualization of the optimal policy structure is possible).

(3) A near optimal *heuristic* policy is proposed based on the structural properties of the optimal policy and its behavior with the changing system parameters.

(4) We conducted a simulation study to quantify the effect of different model parameters on the behavior of the optimal policy. We also studied the performance of the proposed heuristic policy and compared it to that of the optimal policy.

The rest of this paper is organized as follows; Section 2 defines the problem. The optimal policy is studied in section 3. In section 4, the optimal policy for 2-user case is computed. In section 5, we extend our model to include packet retransmissions. In sections 6 and 7, we present a near optimal heuristic policy for scheduling. Performance analysis and comparison are given in section 8. Conclusions are given in section 9. We also provide an online supplement as an *electronic appendix* that contains the appendices with material that could not be otherwise included due to space limitation. Directions on accessing this supplement is given at the end of this paper.

2. PROBLEM DEFINITION

2.1 HSDPA General Description

Third generation release R'5 [3GPP 2004], also called High-Speed Downlink Packet Access (HSDPA), is an IP-based network that can offer users a high speed asymmetric radio link with downlink peak bit rate up to 14.4 Mbps. The HSDPA uses a single time shared channel, called High Speed Downlink Shared CHannel (HS-DSCH), per cell/sector. This channel is divided into 2 ms Transmission Time Intervals (TTI). Each TTI may be used to transfer packets to one or more users at a rate that depends on their User Equipment (UE) capabilities and needs. The UE can use up to 15 codes simultaneously to achieve higher rate. More than one user can share the same slot by dividing the available 15 CDM codes between them. In such case, the scheduler need to choose not only the user/users to be served in the next time slot, but also the number of codes each user will receive.

The HSDPA system uses Adaptive Modulation and Coding (AMC) technique to adapt the transmission rate to the user's channel conditions. The selected modulation scheme and coding rate are chosen such that a fixed low error rate is achieved (usually 10%). The erroneous packets will be retransmitted during the next scheduled TTI using Hybrid ARQ [Lin and Costello 1983]. In this technique, a retransmitted packet will be soft combined at the receiver with the previous unsuccessful transmitted versions of itself (i.e., combining the signal energy of multiple retransmissions of the same packet) to increase it's SNR and it's detection probability. HSDPA supports two combining techniques: Chase Combining (CC) and Incremental Redundancy (IR). In CC, the base station (which is called NodeB in the 3GPP technical specification [3GPP 2004]) retransmits the exact same set of coded symbols of the original packet. Then the receiver combines the packet energy with the energy of previously received unsuccessful transmissions of this packet. With IR, the same principle is used except that different redundancy information can be sent (i.e., using different error coding rate) in every re-transmissions. This will result in incremental increase in the coding gain and hence, fewer retransmissions will be needed (compared to CC) and is particularly useful when the initial transmission uses high coding rates. However, it increases the complexity requirements for the UE [Frenger et al. 2001].

Our objective is *to investigate a methodology that determines the optimal scheduling regime, controlling the allocation of the time-code resources fairly between all the active users while maximizing the overall cell throughput.* The desired scheduling algorithm should have the following characteristics: channel awareness, fairness and high speed resource allocation.

2.2 HSDPA Downlink Scheduler Abstraction

The HSDPA downlink channel uses a mix of Time Division Multiplexing and Code Division Multiplexing:

—Time is slotted into fixed length 2 ms TTIs.

—During each TTI, there are 15 available codes that may be allocated to one or more users.

During one TTI, the channel capacity associated to one single user depends on the number of allocated codes and on the channel condition. This is mainly due to the fact that HSDPA uses AMC to adapt the transmission rate to the current channel conditions. A mobile user with good channel conditions will experience higher data rate than the other users.

The diagram in Figure 1 depicts a conceptual realization of the HSDPA downlink scheduler. Different users have separate buffers in the base station (Node-B

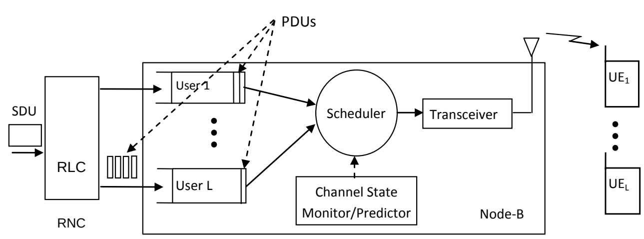


Fig. 1. HSDPA scheduler model (downlink)

according to 3GPP), and they are competing for the system resources. a channel state monitor/predictor is necessary to monitor current channel conditions of each user and predict his channel state during the next TTI. This information will then be used to adapt the transmission rate to the expected channel conditions. The arrived Service Data Units (SDU) are assumed to be segmented by the Radio Link Control (RLC) into u_i fixed size Protocol Data Units (PDU) (for user i) before delivering them to Node-B. The PDUs then will be classified and inserted into the proper buffers awaiting transmission to the intended user. RNC is the Radio Network Controller unit which implements the RLC protocol.

2.3 HSDPA Downlink Channel Model

The wireless channel for the HSDPA system is modeled as a Finite-State Markov Channel (FSMC) following [Wang and Moayeri 1995]. The FSMC was proved to be a good model of the wireless medium and has been shown to be in good agreement with realistic cases (c.f. [Zhang and Kassam 1999][Hassan et al. 2004]). FSMC modeling is done by partitioning the signal to noise ratio (SNR) into finite number of intervals, each representing a state in a Markov Chain. Assuming that the fading is slow enough then the channel states for consecutive time epochs are neighboring states. In this case, the model will be reduced into a discrete time birth and death process, as shown in Figure 2.

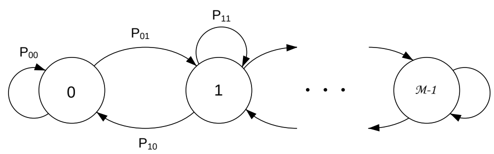


Fig. 2. FSMC model for HSDPA downlink channel.

Depending on the expected SNR state, different modulation and error-correcting coding rates can be dynamically selected from a set of Modulation and Coding Schemes (MCS) [Kolding et al. 2002]. The higher the order of the MCS selected the higher the transmission rate. The SNR is mapped directly into MCS and hence into data rates. In light of this, the states in our channel model will equivalently represent data rate levels rather than SNR.

3. OPTIMAL POLICY DETERMINATION

In this section, we propose an approach, based on Markov Decision Process (MDP), to find the optimal code allocation policy for the HSDPA downlink scheduler. We present a general model for this system and suggest a reward function.

To describe a system as a MDP model, the states, actions, rewards and transition probabilities have to be defined first. In our proposed model, time is slotted in constant intervals of size Δt. Let T denote the set of decision epochs of the system (it is assumed that the decision epochs are at the beginning of each time slot, PDUs that arrive during any time slot can only be considered for transmission in the subsequent time slots), and $T = \{1, 2, \ldots\}$. At time $t \in T$, we denote by $\boldsymbol{s}(t)$ and $\boldsymbol{a}(\boldsymbol{s})$, the system state and the action taken at that state (to be defined later) respectively. HSDPA downlink scheduler is modeled by the 5-tuple $(T, S, A, P_{\boldsymbol{ss'}}(\boldsymbol{a}), R(\boldsymbol{s}, \boldsymbol{a}))$, where S and A are the state and action spaces, $P_{\boldsymbol{ss'}}(\boldsymbol{a}) \triangleq Pr(\boldsymbol{s}(t+1) = \boldsymbol{s'} | \boldsymbol{s}(t) = \boldsymbol{s}, \boldsymbol{a}(\boldsymbol{s}) = \boldsymbol{a})$ is the state transition probability, and $R(\boldsymbol{s}, \boldsymbol{a})$ is the immediate reward function (to be introduced shortly) when at state $\boldsymbol{s}$ and taking action $\boldsymbol{a}$.

3.1 Basic Assumptions

There are L active users in the cell. A user $i \in I = \{1, 2, \ldots, L\}$ is allocated a buffer of finite size B. Initially, error free transmission will be assumed to eliminate the need for retransmission queues and to reduce the model complexity. Later (in section 5), the more general problem with retransmission will be examined. SDUs arrive at the RNC during the current TTI will be segmented by RLC into a fixed number of PDUs (u_i) and delivered to Node-B to be inserted into their respected buffer at the beginning of the next TTI. For each user $i \in I$ and slot $t \in T$, we define:

—$y_i(t)$ the number of scheduled PDUs,

—$x_i(t) \in \mathcal{X} = \{0, 1, 2, ..., B\}$ the queue size,

—$z_i(t) \in \{0, u_i\}$ the number of arriving PDUs per slot.

The SDUs destined to user i arrives at the RNC during one TTI according to the Bernoulli distribution with parameter q_i. Arrivals are assumed to be independent of the system state and of each other. The PDU size is chosen to be equal to the minimum Transport Format and Resource Combination (TFRC) for one code (i.e., one code is needed to transmit one PDU when the channel is in state 1). The scheduler can assign the available 15 codes as chunks of c codes at a time to active users in the system. The chunk size c must divide the total number of codes (15); therefore, $c \in \{1, 3, 5, 15\}$. For example, choosing $c = 5$ means that the policy can assign 0, 5, 10, or 15 of the available 15 codes to any user at any given TTI. We introduced the chunk size c in our analysis in order to reduce the action set size and the computational complexity.

3.2 FSMC State Space

The channel state of user i during time slot t is denoted by $\gamma_i(t)$; and its associated channel state space is the set $\mathcal{M} = \{0, 1, \dots, M-1\}$, where M is the total number of available channel states. $\mathcal{M}$ constitutes a subset of the available MCS set recommended by 3GPP. The elements of $\mathcal{M}$ were ordered in a way such that $\gamma_i(t) \in \mathcal{M}$ is directly proportional to the number of PDUs that can be transmitted by user i in one TTI. This assumption is in good agreement with the corresponding RFC [3GPP 2004]. Furthermore, we assume that user i channel can handle up to $\gamma_i(t)$ PDUs per code, i.e., a $\gamma_i(t) = 2$ means that at time t, user i can transmit two PDUs using one code and up to 30 PDUs when using all the 15 codes. The state transition probability $P_{\gamma_i \gamma_i'}$ is assumed known since it can be calculated (from SNR measurements) for any mobile environment with Rayleigh fading channel [Wang and Moayeri 1995].

3.3 State Space and Action Set

The system state $\boldsymbol{s}(t) \in S$ is a vector comprised of multiple state variables representing the queue sizes and the channel states for the L users. In other words,

$$\boldsymbol{s}(t) = (x_1(t), x_2(t), \dots, x_L(t), \gamma_1(t), \gamma_2(t), \dots, \gamma_L(t)) \tag{1}$$

and the state space of the system, $S = \{\mathcal{X} \times \mathcal{M}\}^L$, has finite size since the buffers have finite sizes and the channel state spaces are also finite.

The action space A is the set of all possible actions. The action $\boldsymbol{a}(\boldsymbol{s}) \in A$ is taken when the system is in state $\boldsymbol{s}$. The action taken at each slot corresponds to the number of codes allocated to each user. Let $D = \{0, 1, \dots, 15/c\}$ be the action space for a single user, where c is the code chunk size (the minimum number of codes that can be allocated at any given time). For example, if $c = 5$ then $D = \{0, 1, 2, 3\}$. Let $a_i(\boldsymbol{s}) \in D$ be the number of code chunks allocated to user i when in state $\boldsymbol{s}$.

Then the number of codes allocated to user i during time slot t is $a_i(t)c$ and the number of scheduled (for transmission) PDUs from queue i (corresponding to user i) is $y_i(t) = a_i(t)\gamma_i(t)c$. In this case, $\boldsymbol{a}(\boldsymbol{s})$ will be the collection of code allocation to all users, that is

$$\boldsymbol{a}(\boldsymbol{s}) = (a_1(\boldsymbol{s}), a_2(\boldsymbol{s}), \ldots, a_L(\boldsymbol{s})) \tag{2}$$

subject to

$$\sum_{i=1}^{L} a_i(\boldsymbol{s}) \cdot c \leq 15, \quad \text{and} \quad a_i(\boldsymbol{s}) \leq \left\lceil \frac{x_i(t)}{\gamma_i(t)c} \right\rceil$$

The first constraint means that the policy can not allocate more than the available 15 codes at each time slot. The second makes the policy conserving by allocating no more codes to user i than that required to empty its buffer. The right side of the second constraint represents the number of code chunks required to empty queue i.

3.4 Reward Function

In this subsection, we introduce the reward function used to determine the optimal allocation policy. As stated previously, the objective is to maximize the throughput while maintaining fairness between active users. Let the *fairness factor*, denoted by $\sigma \geq 0$, be a parameter that reflects the significance of fairness in the reward function and hence the optimal policy structure (i.e., the fairness of the resulted optimal policy increases as σ increases). Define $\bar{x}$ as the average instantaneous size of the L queues in the system at time t, i.e., $\bar{x} = \frac{1}{L}\sum_{i=1}^{L} x_i$, (we suppressed the time index to simplify notation). The reward function $R(\boldsymbol{s}, \boldsymbol{a})$ will have two components corresponding to two objectives (throughput and fairness) and it is given by

$$R(\boldsymbol{s}, \boldsymbol{a}) = \sum_{i=1}^{L} y_i - \sigma \sum_{i=1}^{L} (x_i - \bar{x})\, \mathbf{1}_{\{x_i = B\}} = \sum_{i=1}^{L} a_i \gamma_i c - \sigma \sum_{i=1}^{L} (B - \bar{x})\, \mathbf{1}_{\{x_i = B\}} \tag{3}$$

where $\mathbf{1}_{\{\cdot\}}$ is the indicator function. The positive term of the reward relates to the cell throughput. If the reward is composed of this part only, then the policy will always favor the users with good channel conditions. Therefore the users with less favorable channels will starve; their queues will grow larger and start loosing packets. We introduced the second term, which guarantees a level of fairness and reduces dropping probability. Lower σ will result in a policy that favors cell throughput over fairness, while higher σ has the opposite effect. Overall, $R(\boldsymbol{s}, \boldsymbol{a})$ will produce a policy that maximizes cell throughput for a given σ.

3.5 State Transition Probability

$P_{\boldsymbol{ss'}}(\boldsymbol{a})$ denotes the probability that choosing an action $\boldsymbol{a}$ at time t when in state $\boldsymbol{s}$ will lead to state $\boldsymbol{s'}$ at time $t+1$. Using equations (1) and (2), $P_{\boldsymbol{ss'}}(\boldsymbol{a})$ can be stated as follows

$$\begin{aligned} P_{\boldsymbol{ss'}}(\boldsymbol{a}) &\triangleq Pr(\boldsymbol{s}(t+1) = \boldsymbol{s'} | \boldsymbol{s}(t) = \boldsymbol{s}, \boldsymbol{a}(t) = \boldsymbol{a}) \\ &= Pr(x'_1, \ldots, x'_L, \gamma'_1, \ldots, \gamma'_L | x_1, \ldots, x_L, \gamma_1, \ldots, \gamma_L, a_1, \ldots, a_L) \end{aligned} \tag{4}$$

The evolution of the queue size (x_i) is given by

$$x'_i = \min\left([x_i - y_i]^+ + z'_i, B\right) = \min\left([x_i - a_i\gamma_i c]^+ + z'_i, B\right) \tag{5}$$

where, z_i' is the arrival to queue i at $t+1$, $[e]^+$ equals e if $e \geq 0$ and 0 otherwise. The channel state γ_i depends only on the previous channel state, that is $Pr(\gamma_i'|\boldsymbol{s}) = Pr(\gamma_i'|\gamma_i) = P_{\gamma_i\gamma_i'}$. Accordingly, we can write equation (4) as follows (Appendix A)

$$P_{\boldsymbol{ss'}}(\boldsymbol{a}) = \prod_{i=1}^{L} \left(P_{x_ix_i'}(\gamma_i, a_i)\, P_{\gamma_i\gamma_i'}\right) \tag{6}$$

where $P_{\gamma_i\gamma_i'}$ is the Markov transition probability of the FSMC. Define $W1$ and $W2$ as follows

$$W1 = [x_i - a_i\gamma_i c]^+ + u_i; \qquad W2 = [x_i - a_i\gamma_i c]^+$$

$P_{x_ix_i'}(\gamma_i, a_i)$ can be analytically derived using equation (5) and the total probability law (refer to Appendix B), and is given by the following expression

$$P_{x_ix_i'}(\gamma_i, a_i) = \begin{cases} 1 & \text{if } x_i' = x_i = B \ \&\ a_i\gamma_i = 0, \\ q_i & \text{if } x_i' = x_i = B \ \&\ 0 < a_i\gamma_i c \leq u_i, \\ q_i & \text{if } x_i' = B \ \&\ x_i < B \ \&\ W1 \geq B, \\ q_i & \text{if } x_i' < B \ \&\ x_i' = W1, \\ 1 - q_i & \text{if } x_i' < B \ \&\ x_i' = W2, \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

The first three cases in equation (7) correspond to the boundary state/condition (i.e., when the queue length reaches the buffer size), while the remaining cases correspond to interior states.

3.6 Dynamic Programming Formulation

In this work, we investigate an infinite-horizon MDP. We use the total expected discounted reward optimality criterion with discount factor λ, where $0 < \lambda < 1$, in order to find the policy π among all policies, that maximizes the *value function* $V^\pi(s)$. Let $V^*(\boldsymbol{s})$ be the maximal discounted value function (i.e., $V^*(\boldsymbol{s}) = \sup_\pi V^\pi(\boldsymbol{s})$), attained when applying the optimal policy π^*. Then the following optimality equation (also known as Bellman equation) is used to characterize the optimal policy [Bellman 1957][Sennott 1999]

$$V^*(\boldsymbol{s}) = \max_{\boldsymbol{a} \in A}[R(\boldsymbol{s}, \boldsymbol{a}) + \lambda \sum_{\boldsymbol{s'} \in S} P_{\boldsymbol{ss'}}(\boldsymbol{a}) V^*(\boldsymbol{s'})] \tag{8}$$

Value iteration [Puterman 1994] (also known as successive approximation) is used to solve this model numerically. The first step is to define $V_0(\boldsymbol{s})$ to be an arbitrary bounded function. Then $V_n(\boldsymbol{s})$, $n > 0$ is determined by the following recursion

$$V_n(\boldsymbol{s}) = \max_{\boldsymbol{a} \in A}[R(\boldsymbol{s}, \boldsymbol{a}) + \lambda \sum_{\boldsymbol{s'} \in S} P_{\boldsymbol{ss'}}(\boldsymbol{a}) V_{n-1}(\boldsymbol{s'})]$$

V_n converges to V^* as $n \to \infty$ [Ross 1983]. For a given $\epsilon > 0$, the algorithm can be stopped after n iterations, provided that

$$\|V_{n+1} - V_n\| < \epsilon(1-\lambda)/2\lambda \tag{9}$$

where $\|v\| = \sup_{s \in S} |v(s)|$. If (9) holds, then $\|V_{n+1} - V^*\| < \epsilon/2$, according to [Puterman 1994].

The results obtained here can be extended to the infinite horizon expected average reward using results from [Puterman 1994] and [Ross 1983]. This extension is out of the scope of this work and will not be presented here.

4. TWO USERS WITH 2-STATE FSMC

The approach presented earlier was used to model the case when there are two users (i.e., $L = 2$) sharing the same cell. The channel for user i is modeled as a two-state FSMC with transition probability matrix P_i

$$P_i = \begin{bmatrix} 1-\alpha_i & \alpha_i \\ \beta_i & 1-\beta_i \end{bmatrix} \tag{10}$$

The two user case will yield a policy that is easy to visualize, evaluate, and to deduct conclusions for the optimal policy. It also serves as a verification for the proposed approach, since it is possible to visualize and plot the optimal policy in this case as will be shown later. The obtained results can then be generalized to more complex cases involving more than 2 queues.

User i is said to be *connected* when $\gamma_i = 1$ with probability $P(\gamma_i = 1) = \alpha_i/(\alpha_i + \beta_i)$, and *not connected* ($\gamma_i = 0$) with probability $P(\gamma_i = 0) = \beta_i/(\alpha_i + \beta_i)$.

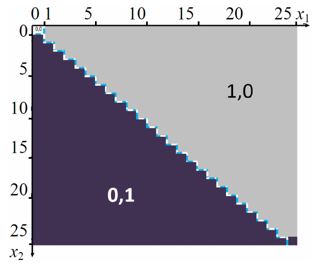
The remaining parameters were chosen as follows: $B = 25$, $\sigma = 0.5$, $\lambda = 0.95$, $\epsilon = 0.1$, and c =3, 5 or 15. The action space depends on the value of c. For example, if $c = 5$ then there are *four* possible actions for each user (i.e., $D = \{0, 1, 2, 3\}$). Since $\boldsymbol{a} = (a_1, a_2)$ corresponds to $a_1 c$ codes assigned to user 1 and $a_2 c$ codes assigned to user 2, we get $A = \{(0,0), (0,1), (0,2), (0,3), (1,0), (1,1), (1,2), (2,0), (2,1), (3,0)\}$. Similarly, when $c = 15$ then there are *two* possible actions per user ($D = \{0, 1\}$) and when $c = 3$ then there are *six* possible actions per user ($D = \{0, 1, 2, 3, 4, 5\}$).

4.1 Optimal Policy Structure

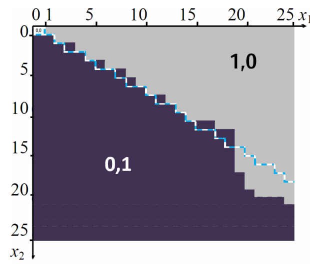
The model is solved using value iteration (section 3.6) to determine the optimal scheduling policy. The effect of the channel quality and arrival probability on the behavior of the optimal policy was studied. Figures 3-5 provide general structure of the optimal policy for c =15, 5, and 3 respectively. For the sake of brevity and due to space limitations Figures 4 and 5 contains only two cases. The remaining cases for these figures can be found in the electronic appendix (check the end of the paper).

4.1.1 *Optimal Policy for Two Symmetrical Users.* The optimal policy for two symmetrical users with the same channel characteristics ($\alpha_i = \beta_i = p$) for all $0 \leq p \leq 1$ and with $P(z_i = 5) = 0.5$ for all $i \in \{1, 2\}$ is shown in Figures 3-5(a). Results for the case when the two users are connected, i.e., $\gamma_i = 1$, is shown here, since the two users are competing for the system resources. The other three cases when one or both of them are not connected, i.e., $\gamma_i = 0$, resulted in a policy that assigns all the codes (required) to the connected user and nothing to the other.

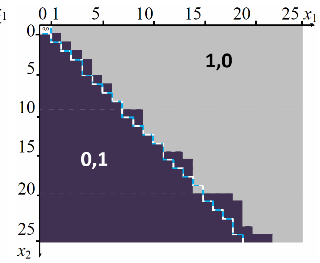
The optimal policy in this case can be described as follows: *divide the codes between the connected users in proportion to their queue length.* When $c = 15$, the action space will be reduced to $A = \{(0,0), (0,1), (1,0)\}$ and the policy will be



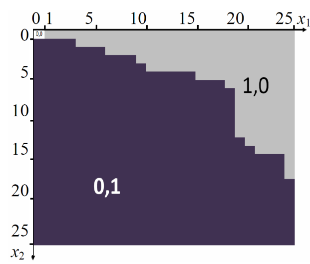
(a) Symmetrical case



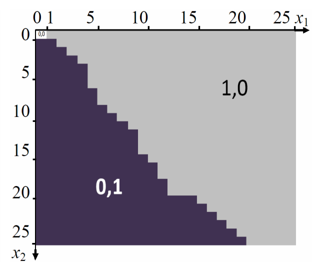
(b) $P(\gamma_1 = 1) = 0.8$ and $P(\gamma_2 = 1) = 0.5$



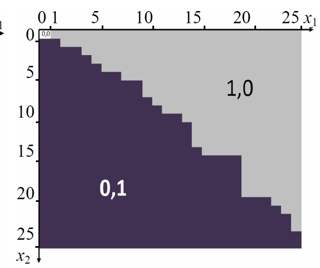
(c) $P(z_1 = 5) = 0.8$ and $P(z_2 = 5) = 0.5$



(d) $P(\gamma_1 = 1) = 0.8$ and $P(\gamma_2 = 1) = 0.3$

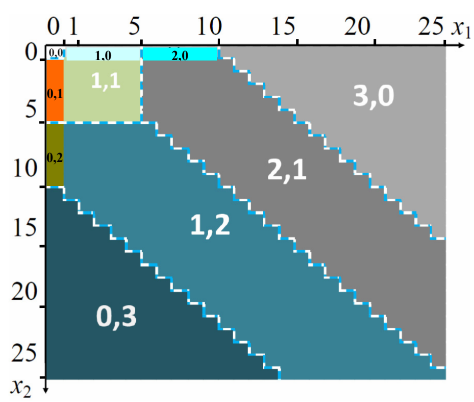


(e) $P(z_1 = 5) = 0.8$ and $P(z_2 = 5) = 0.3$

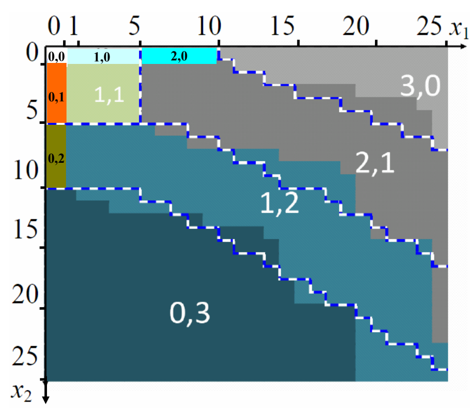


(f) $P(z_1 = 5) = 0.8$, $P(z_2 = 5) = 0.5$, $P(\gamma_1 = 1) = 0.8$ and $P(\gamma_2 = 1) = 0.5$

Fig. 3. Optimal and heuristic (dotted line) policies for two user case; $c = 15$ (i.e., 0 or 1 chunks of size 15 codes can be assigned to a user), arrival batch size $u = 5$.



(a) Symmetrical case

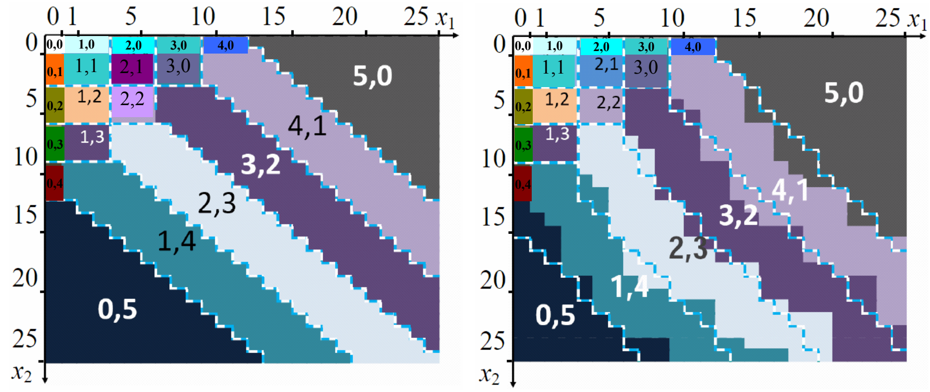


(b) $P(\gamma_1 = 1) = 0.8$ and $P(\gamma_2 = 1) = 0.5$

Fig. 4. Optimal and heuristic (dotted line) policies for two user case; $c = 5$ (i.e., 0,1,2 or 3 chunks of size 5 can be assigned to a user), $u = 5$.

equivalent to *serve the longest connected queue (LCQ)*, which makes intuitive sense and matches with the findings in [Tassiulas and Ephremides 1993] for a case similar to our $c = 15$ case.

4.1.2 *The Effect of Channel Quality on Policy Structure.* The effect of the channel quality on the optimal policy structure when $\gamma_1 = \gamma_2 = 1$ is shown in Figures 3-5(b). When $P(\gamma_1 = 1) > P(\gamma_2 = 1)$ the policy favors user 2 *which is less likely to*



(a) Symmetrical case (b) $P(z_1\!=\!5)\!=\!0.8$ and $P(z_2\!=\!5)\!=\!0.5$

Fig. 5. Optimal and heuristic (dotted line) policies for two user case; $c = 3$ (i.e., 0,1,2,3,4 or 5 chunks of size 3 can be assigned to a user), $u = 5$.

be connected compared to user 1. The bias in favor of user 2 is depicted in Figure 3(b) by a larger dark area, which corresponds to optimal action (0,1) and (0,3) respectively, compared to Figure 3(a). We noticed that this bias increases as the difference between $P(\gamma_1 = 1)$ and $P(\gamma_2 = 1)$ increases. The reason is that using an LCQ in this situation will result in unbalanced system (queue 2 will grow larger and will have more packet drops than queue 1). User 2 will start experiencing unfairness in terms of higher delay and more drops. Hence, more resources have to be assigned to the user with the worst channel to avoid such result. The resource sharing in this case will depend on the difference $\Delta P_\gamma = P(\gamma_1 = 1) - P(\gamma_2 = 1)$ as well as their relative queue length.

4.1.3 *The Effect of Arrival Probability on Policy Structure.* The arrival probability has an analogous effect on the optimal policy structure. The relative increase in one of the users arrival probability will result in more traffic inserted in that user's buffer and it will require more resources to keep the queue length stable and achieve fairness between the two users.

Figure 3(c) shows the optimal policy when $P(z_1 = 5) = 0.8$ and $P(z_2 = 5) = 0.5$ and both users have the same channel quality. The policy shifts in favor of the user with higher arrival probability (user 1 in this case). By comparing Figure 3(c) to Figure 3(e), we noticed that this shift is proportional to the difference $\Delta P_z = P(z_1 = u) - P(z_2 = u)$.

Figure 3(f) shows that when $\Delta P_z = \Delta P_\gamma = 0.3$ the optimal policy favors user 2, i.e., the user with less connectivity. This agrees with intuition since both users share the available server capacity (the number of codes). The exogenous arrivals will always be added to the corresponding buffer (provided the availability of space in that buffer), while departures depend not only on the channel connectivity, but also on the maximum server capacity (total number of departures is bounded by the server capacity during each time slot; 15 PDUs in this case).

5. HSDPA SYSTEM WITH RETRANSMISSION

We now expand our model to include the case of packet retransmission for unsuccessful packet transmissions. The resulting model will have two queues per user, a transmission queue and a retransmission queue. The state space for this system will be $S = \{\mathcal{X} \times \mathcal{W} \times \mathcal{M}\}^L$, where $\mathcal{W} = \{0, 1, \ldots, B_r\}$ is the state space for the retransmission queue and B_r is the retransmission buffer size. $\mathcal{X}$, $\mathcal{M}$ are the same as defined earlier. The computational complexity for finding the optimal policy for such system is substantially greater than the previous model and could become a prohibitive factor for a system with a large number of users. Instead, we present an alternative model for this system, having less computational complexity. In order to avoid the increased dimensionality due to the retransmission queue, we consider a Bernoulli random process $\{\mu_i(t)\}_{t=1}^{\infty}$ that indicates the status of a packet transmission. This will result in a retransmission model with i.i.d. geometric service time, an assumption well vetted in the literature, e.g.,[Tassiulas and Ephremides 1993] and [Bambos and Michailidis 2002]. When codes are allocated to user i at time slot t, then the transmission is successful ($\mu_i(t) = 1$) with a probability ξ (corresponding to the probability of successful transmission). When $\mu_i(t) = 0$, then the transmission is unsuccessful and the PDUs have to be retransmitted. This happens with probability $1 - \xi$. The HSDPA scheduler with retransmissions is modeled as shown in Figure 6. The evolution of the queue size (x_i) in this case is given by

$$\begin{aligned} x_i(t+1) &= \min\Big([x_i(t) - y_i(t)\mu_i(t)]^+ + z_i(t+1)\,, B\Big) \\ &= \min\Big([x_i(t) - \gamma_i(t)a_i(\boldsymbol{s})c\mu_i(t)]^+ + z_i(t+1), B\Big) \end{aligned} \tag{11}$$

where $a_i(\boldsymbol{s})c$ is the number of codes allocated to user i when in state $\boldsymbol{s}$ and $\boldsymbol{a}(\boldsymbol{s}) = (a_1(\boldsymbol{s}), a_2(\boldsymbol{s}), \ldots, a_L(\boldsymbol{s}))$. Equation (11) means that PDUs removed from the head of the queue only when a transmission is successful, and remain there otherwise.

5.1 Reward Function

The reward function $R(\boldsymbol{s}, \boldsymbol{a})$ is similar as before and is given by

$$R(\boldsymbol{s}, \boldsymbol{a}) = \sum_{i=1}^{L}(y_i\mu_i) - \sigma \sum_{i=1}^{L}\Big[(x_i - \bar{x})\, \mathbf{1}_{\{x_i = B\}}\Big] \tag{12}$$

where $y_i = a_i c \gamma_i$. Similar as before, the objective is to maximize throughput while providing a fair allocations of resources (depending on the value of σ) to all users in the system.

5.2 Transition Probability Function

The MDP state transition probability $P_{\boldsymbol{s}\boldsymbol{s}'}(\boldsymbol{a})$ can be formed as before, namely equations (4) and (6). In this case however, the queue state transition probability depends on μ_i. Using the total probability law, this probability can be rewritten as

$$\begin{aligned} P_{x_i x_i'}(\gamma_i, a_i) &\triangleq P(x_i' | x_i, \gamma_i, a_i) \\ &= P_{x_i x_i' | \mu_i = 1}(\gamma_i, a_i) P(\mu_i = 1) + P_{x_i x_i' | \mu_i = 0}(\gamma_i, a_i) P(\mu_i = 0) \\ &= P_{x_i x_i' | \mu_i = 1}(\gamma_i, a_i) \cdot \xi + P_{x_i x_i' | \mu_i = 0}(\gamma_i, a_i) \cdot (1 - \xi) \end{aligned} \tag{13}$$

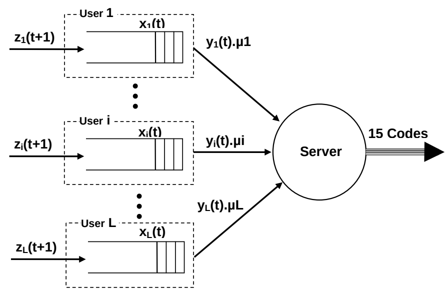


Fig. 6. model for HSDPA downlink scheduler with retransmission

5.3 Queue State Transition Probability

The marginal queue state transition probabilities needed to find $P_{x_i x_i'}(\gamma_i, a_i)$ are given below (refer to Appendix C for complete derivation). We remind the reader that B is the buffer size for each user buffer in the model.

$$P_{x_i x_i' | \mu_i = 1}(\gamma_i, a_i) = \begin{cases} 1 & \text{if } x_i' = x_i = B, \gamma_i a_i = 0 \\ q_i & \text{if } x_i' = x_i = B, 0 < y_i \leq u_i \\ q_i & \text{if } x_i' = B, x_i < B, [x_i - y_i]^+ + u_i \geq B \\ q_i & \text{if } x_i' < B, x_i' = [x_i - y_i]^+ + u_i \\ 1 - q_i & \text{if } x_i' < B, x_i' = [x_i - y_i]^+ \\ 0 & \text{Otherwise} \end{cases} \tag{14}$$

and,

$$P_{x_i x_i' | \mu_i = 0}(\gamma_i, a_i) = \begin{cases} 1 & \text{if } x_i' = x_i = B \\ q_i & \text{if } x_i' = B, x_i < B, x_i + u_i \geq B \\ q_i & \text{if } x_i' < B, x_i' = x_i + u_i \\ 1 - q_i & \text{if } x_i' < B, x_i' = x_i \\ 0 & \text{Otherwise} \end{cases} \tag{15}$$

The overall queue transition probability $P_{x_i x_i'}$ can be determined by substituting equations (14) and (15) in (13).

6. THE HEURISTIC POLICY

In this section, we will present a heuristic approach for code allocation in our HSDPA model, by focusing on the cases where c =15, 5 or 3. We will utilize the information (regarding the structure of the optimal policy) we obtained from the results of section 4.

6.1 Optimal Policy Structural Analysis and Weight Function Estimation

Analyzing the results obtained in sections 4.1.1-4.1.3 we observe that the optimal policy exhibits the following structural characteristics:

— When there is only one connected user then the optimal policy allocates all the 15 codes to that user or as many codes as required to empty its queue if its queue length is less than 15 PDUs.

— When both users are connected then both are competing for the available 15 codes. From Figures 3–5 we observe the following trends:

(1) The optimal action a_1 (respectively a_2) is increasing in x_1 (respectively x_2).
(2) There is one switch-over region for every action vector. These regions depicted by different colors (or shades) and labeled by their corresponding action vector in Figures 3–5).
(3) When $x_1 + x_2 \leq 15$, then the optimal policy allocates no more codes to a user than that required to empty its queue.
(4) In the symmetrical case and when $x_1 + x_2 > 15$, then the border(s) between neighboring regions is (are) a staircase function that can be approximated by a line with slope 1 (Figure 4(a)). Those border lines define parallel stripes that have a constant width.
(5) In the asymmetrical cases and when $x_1 + x_2 > 15$ (Figures 3(b)-(f)), the aforementioned regions while still equidistant, are no longer delimited by linear boarders. The approximate boarder slope as a function of x_1 and x_2 depends on the difference in arrival rates ΔP_z and the difference in the connectivities ΔP_γ. For example the approximate slope (the dotted line) in Figure 3 is: 1 in the symmetrical case (Figure (a)), less than 1 when $\Delta P_\gamma = 0.3$ (Figure (b)) and more than 1 when $\Delta P_z = 0.3$ (Figure (c)).

From observation (4) above, we can approximate each of the switch-over boundary lines between the policy switch-over regions for the symmetrical case by a straight line with slope 1. This approximation is a good fit in the symmetrical case as shown in Figure 3(a). To extend this approximation to the asymmetrical cases, we introduce the weight vector $\mathbf{w} = (w_1, w_2)$, where $w_i, i = 1, 2$ is a function of ΔP_z and ΔP_γ. Then the boundary lines between the different regions can be linearly approximated by the following equation:

$$w_1 x_1 = w_2 x_2 + C \tag{16}$$

where C is a constant. Observations of Figures 4 and 5 suggest that C is a multiple of $\pm 2c$ ($C = \pm 2ck, k = 0, 1, \ldots, 0.5(15/c - 1)$). Equation (16) defines a family of lines that have a slop equals to w_2/w_1. For the symmetrical case, $w_1 = w_2 = 1$ and equation (16) will be reduced to $x_1 = x_2 + C$.

The weight w_1 (respectively w_2) is increasing (respectively decreasing) in ΔP_z and decreasing (respectively increasing) in ΔP_γ (the reader may verify this from the figures, for example Figures 3(b) and 3(c)). We observed the behavior of the optimal policy under different arrivals and connectivity parameters, by solving the dynamic programming equation (equation 8) numerically for these cases. The results obtained were analogous to the ones in Figures 4-5. Following these observations, we approximated w_1 and w_2 as follows

$$w_1 \approx 1 + 1.5[-\Delta P_\gamma]^+ - 0.7[-\Delta P_z]^+, \text{ and } w_2 \approx 1 + 1.5[\Delta P_\gamma]^+ - 0.7[\Delta P_z]^+ \tag{17}$$

The coefficients for ΔP_γ and ΔP_z were chosen empirically based on a variety of computations (value iterations to determine the optimal policy) under different channel connectivity and arrival parameters.

The linear approximation may not be perfect for asymmetric cases (see for example Figures 3(b) and (c)). A non-linear approximation would result in a better fit in those cases (and may be a future research problem). However, (as we will show in section 7.3) the heuristic scheduling policy, resulted from our linear approximation, compares favorably (in throughput and fairness/queuing delay terms) with the optimal policy.

6.2 Detailed Characterization of The Heuristic Policy

(1) Case for $c = 15$. In this case, the optimal policy is a switch-over policy as depicted in Figure 3. We can identify three regions which correspond to the three possible actions: (0,0), (1,0) and (0,1). The heuristic policy is a weighted LCQ and it assigns codes to users according to the following rules:

—Rule1: when there is only one connected user then assign all the needed codes to that user,

—Rule2: when both users are not connected (i.e., $\gamma_1 = \gamma_2 = 0$) then no codes will be allocated to any user,

—Rule3: when the two users are connected allocate code chunks according to (18)

$$\boldsymbol{a}(t) = \begin{cases} (1,0) & \text{if } w_1 x_1 > w_2 x_2, \\ (0,1) & \text{if } w_1 x_1 \leq w_2 x_2 \end{cases} \tag{18}$$

(2) Case for $c = 5$. The optimal policy defines ten regions specifying the optimal code allocation. However, only four of these regions are of interest. They lie within the area where the demand exceeds the available resources as shown in Figure 4. Based on this observation, the heuristic policy partitions the state space into four major regions that corresponds to the actions (3,0), (2,1), (1,2), and (0,3). The policy rules 1 and 2 are the same as before. Rule3 is modified as follows:

—Rule3: when the two users are connected, *if* $x_1 + x_2 < 15$ *then* allocate codes to the two users in proportion to their queue length, *else* allocate the code chunks as follows

$$\boldsymbol{a}(t) = \begin{cases} (3,0) & \text{if } w_1 x_1 > w_2 x_2 + 10, \\ (2,1) & \text{if } w_2 x_2 < w_1 x_1 \leq w_2 x_2 + 10, \\ (1,2) & \text{if } w_2 x_2 - 10 \leq w_1 x_1 \leq w_2 x_2, \\ (0,3) & \text{if } w_1 x_1 < w_2 x_2 - 10, \end{cases} \tag{19}$$

(3) Case for $c = 3$. There are 21 different regions in the state space as shown in Figure 5. The heuristic rules used earlier can be extended to this case. Again only Rule3 need to be modified as shown below

—Rule3: when the two users are connected, *if* $x_1 + x_2 < 15$ *then* allocate codes to the two users in proportion to their queue length, *else* allocate the code chunks as follows

$$\boldsymbol{a}(t) = \begin{cases} (5,0) & \text{if } w_1 x_1 > w_2 x_2 + 12, \\ (4,1) & \text{if } w_2 x_2 + 6 < w_1 x_1 \leq w_2 x_2 + 12, \\ (3,2) & \text{if } w_2 x_2 < w_1 x_1 \leq w_2 x_2 + 6, \\ (2,3) & \text{if } w_2 x_2 - 6 < w_1 x_1 \leq w_2 x_2, \\ (1,4) & \text{if } w_2 x_2 - 12 < w_1 x_1 \leq w_2 x_2 - 6, \\ (0,5) & \text{if } w_1 x_1 \leq w_2 x_2 - 12, \end{cases} \tag{20}$$

Figures 3(a)-(c), 4(a)-(b) and 5(a)-(b) show the heuristic policy (the dotted line) superimposed on the optimal policy from section 4 for different loading and channel quality conditions. From these figures, it is fair to say that the heuristic policy reasonably approximated the slope of border lines between the different regions of the optimal policy.

6.3 Extended Heuristic Policy

The optimal policy for three users or more still has a switch-over structure. However, it is not possible to visualize on a two dimensional plane (e.g., for the case of three users, the possible action can be described by a three-dimensional vector, i.e., $\boldsymbol{a}(\boldsymbol{s}) = (a_1, a_2, a_3)$, therefore, it is not possible to pictorially visualize the structure of the policy in this case). Motivated by the heuristic policy construction for the two-user case, we can extend this heuristic policy (presented in the previous subsection) to any finite number of users. We can do this by ordering all users in the system according to their weighted queue lengths. The weight will be a function of the different users' arrival probabilities and channel states. We can then allocate the available code chunks to the connected queues according to their weighted queue length using rules analogous to those in the previous section. The added complexity of this extension (compared to the heuristic policy we presented in the previous section) is minimal and is mainly due to the users ordering phase of the algorithm. Due to space limitation, we will not present this extension here.

7. PERFORMANCE EVALUATION

In this section, we study the performance of the optimal policy and compare it with that of the devised heuristic policy. We also compared both policies with the Round Robin fair queuing, keeping the same assumptions as previously. The evaluation is based on the simulation of several cases with constant buffer size $B = 50$. All simulations were performed using C programming language. The simulation ran for 50,000 time slots and 10000 replications. The 95% confidence interval (CI) was computed for all simulation results. For the sake of clarity of presentation, only the CI's with maximum width among simulated performances for each offered load (corresponding to a point on the x-axis) are plotted at the top of each figure.

7.1 The Effect of Code Allocation Granularity

The total number of codes available in one TTI is 15 codes according to [3GPP 2004] and in our approach, the scheduler allocates chunks of these codes to active users. The chunk size c was introduced in section 3.1. It ranges from the finest (when $c = 1$) to the coarsest (when $c = 15$); in the former case, the policy can assign as little as 1 code to a user at a time, while in the latter case, all the 15 codes are assigned to one single user at a time. Figures 7-10 show the effect of the chunk size c on the system performance for various offered load $\rho = \sum_i P_{z_i} u_i / r_\pi$ with r_π as the system capacity under the policy π. The channel state probability for the two users is set to $P(\gamma_1 = 1) = 0.8$ and $P(\gamma_2 = 1) = 0.5$. Using (10), The channel model parameters (α_i and β_i) for each user were selected as follows

$$P_1 = \begin{bmatrix} 0.4 & 0.6 \\ 0.6 & 0.4 \end{bmatrix}, \qquad \text{and} \qquad P_2 = \begin{bmatrix} 0.85 & 0.15 \\ 0.6 & 0.4 \end{bmatrix}$$

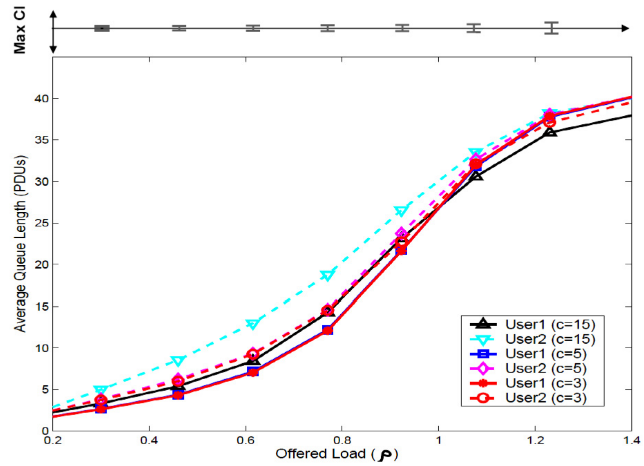


Fig. 7. The effect of policy granularity on queue length

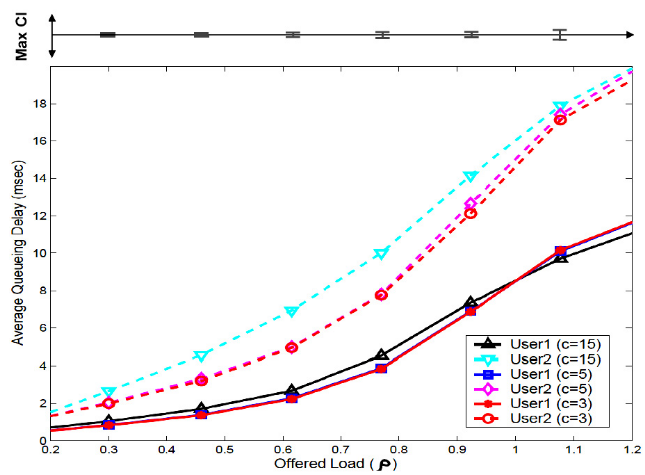


Fig. 8. The effect of policy granularity on the queuing delay experienced by the two users

The results show that in light and moderate load conditions, the average queue length is shorter when the assigned code chunks have finer granularity. However, when $\rho \to 1$ the difference becomes smaller (and within the confidence interval) when ρ becomes greater or equal to 1.

Another observation is that the performance gain when moving from $c = 5$ to $c = 3$ is less significant and may not justify the added implementation and computational cost. It is interesting to see that the optimal policy under all of the three values of c achieved approximately the same throughput (see Figure 9). The slight throughput loss when $c = 15$ in moderate to high load is within the confidence interval and is due to the increased drops under these particular conditions as it is shown in Figure 10. The intuition behind this behavior can be summarized as follows:

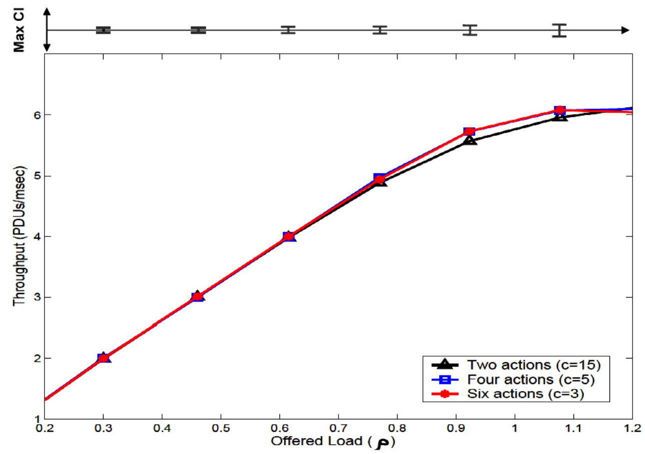


Fig. 9. The effect of policy granularity on scheduler throughput

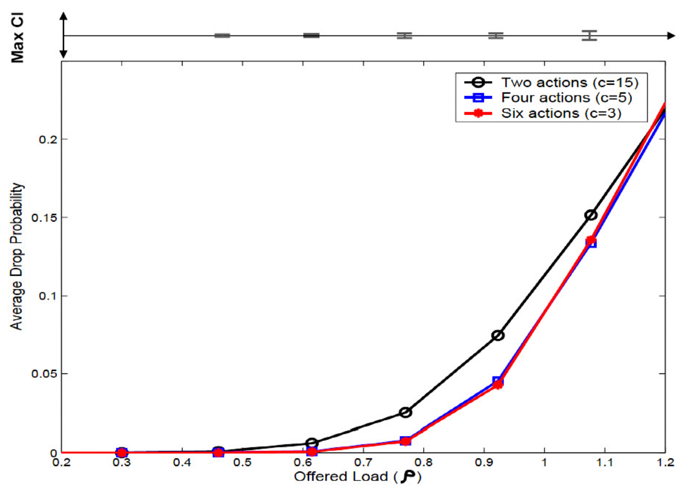


Fig. 10. The effect of policy granularity on scheduler dropping probability

(1) For light loads ($\rho < 0.7$), the chunk size (c) does not have a significant effect on the dropping probability in the two queues since the drops are unlikely in this case (Figure 10). Therefore, the optimal policy for any value of c achieves throughput that is not substantially different.

(2) As the load increases ($\rho = 0.7 \sim 1.1$) the drops become *more likely* to occur. A coarser code allocation granularity ($c = 15$) will serve one queue only at a given TTI thus resulting in more drops for the other, while a finer granularity ($c = 5$) would keep both queue sizes balanced (thus resulting in a lower total loss).

(3) For higher load ($\rho >\sim 1.1$), the losses escalate dramatically and chunk granularity has no significant effect on drops between the two queues in this case.

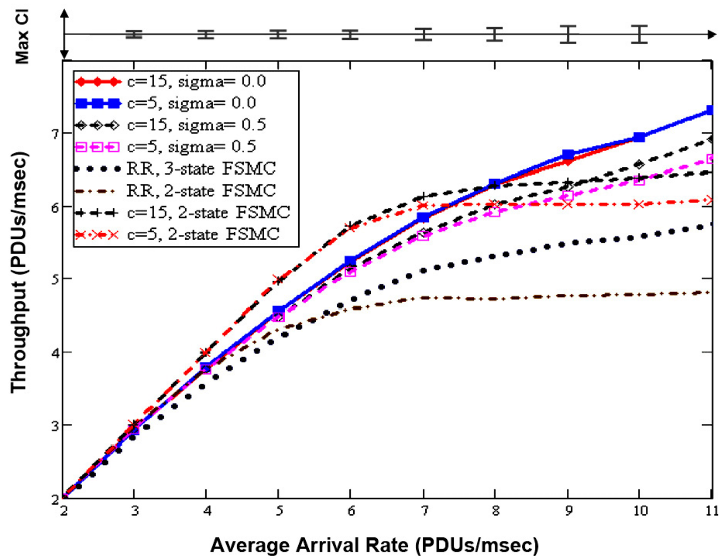


Fig. 11. System throughput vs. average arrival rate to the system. First four cases in legend corresponds to the optimal policy with 3-state FSMC model.

7.2 The Effect of Channel Model

The performance of the optimal policy, when using 3-state FSMC channel model, is evaluated using simulation and compared to the 2-state FSMC model. The channel models used are

—Two-state FSMC with $P(\gamma_1 = 1) = 0.8$, $P(\gamma_2 = 1) = 0.5$.

—Three-state FSMC with $P(\gamma_1 = 1) = 0.4$, $P(\gamma_1 = 2) = 0.4$, $P(\gamma_2 = 1) = 0.25$ and $P(\gamma_2 = 2) = 0.25$.

The two cases above are analogous in the sense that in both cases user 1 channel (respectively user 2 channel) is connected with probability $P(\gamma_1 \geq 1) = 0.8$ (respectively $P(\gamma_2 \geq 1) = 0.5$). To achieve this, the 3-state FSMC model parameters for each user were selected as follows

$$P_1 = \begin{bmatrix} 0.4 & 0.6 & 0.0 \\ 0.3 & 0.3 & 0.4 \\ 0.0 & 0.4 & 0.6 \end{bmatrix}, \qquad \text{and} \qquad P_2 = \begin{bmatrix} 0.8 & 0.2 & 0.0 \\ 0.4 & 0.3 & 0.3 \\ 0.0 & 0.3 & 0.7 \end{bmatrix}$$

Figures 11 and 12 show the system throughput and the drop probability versus average arrival rate ($\sum_i P_{z_i} u_i$ for all $i \in \{1, 2\}$). In the 2-state model, the system reaches its capacity at about 6.5 $PDUs/sec$ (Figures 11). For the case of a 3-state FSMC model, the saturation performance is better since more PDUs can be transmitted on average (compared to the 2-state FSMC model) when connected (the FSMC model is presented in section 3.2), for the given state transition probabilities.

Figure 13 depicts the average queue length behavior for both users as a function of the arrival rate; we can see that when the fairness factor $\sigma = 0.5$ the policy keeps almost equal queue occupancy for both users. On the other hand, when $\sigma = 0.0$ the difference between the average queue length is more than 10 PDUs between the two

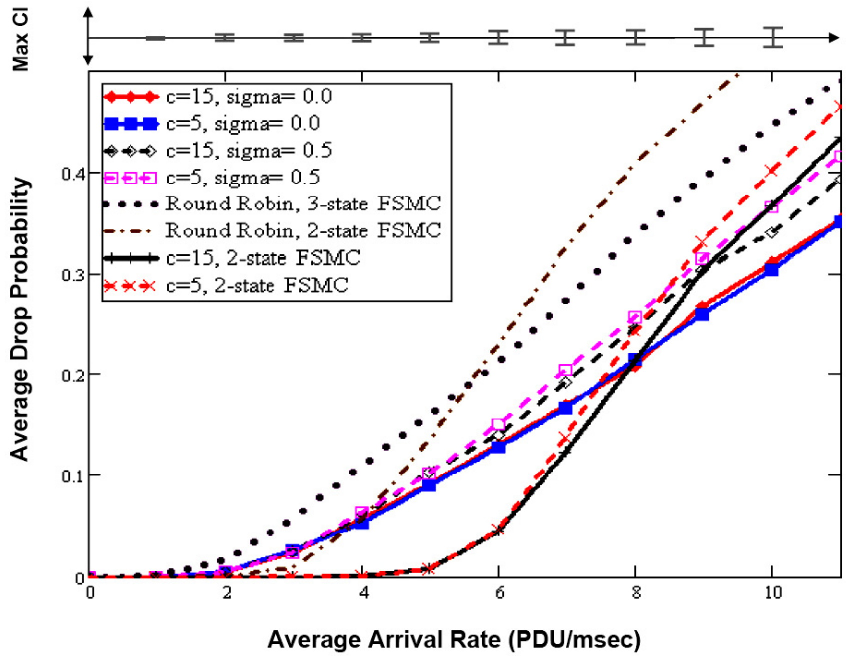


Fig. 12. Average drop probability (average dropped/average arrived PDUs). First four cases in legend corresponds to the optimal policy with 3-state FSMC model.

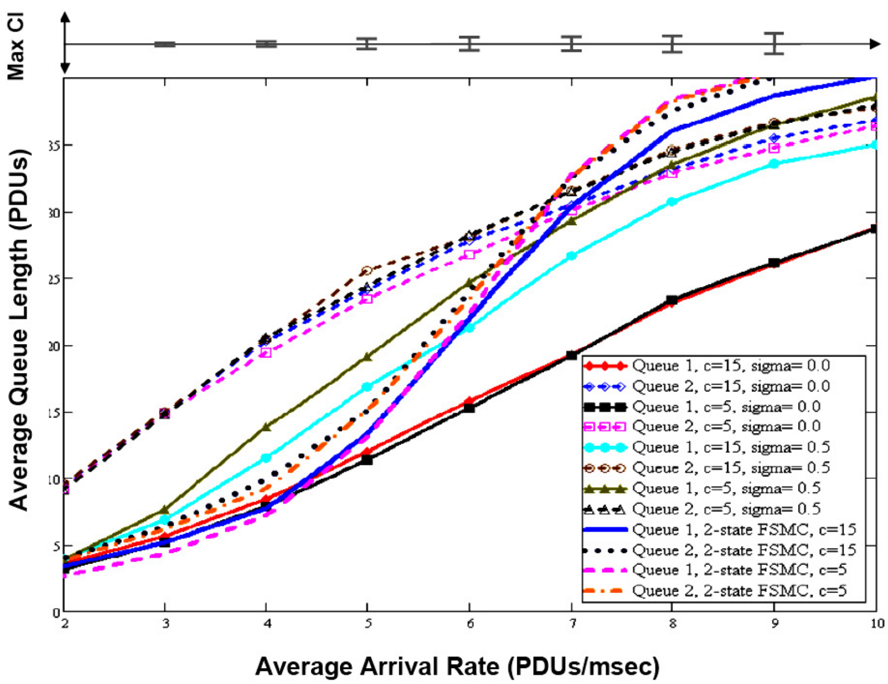


Fig. 13. Average queue length vs. load compared to the 2-state FSMC model. First eight cases in legend corresponds to the optimal policy with 3-state FSMC model.

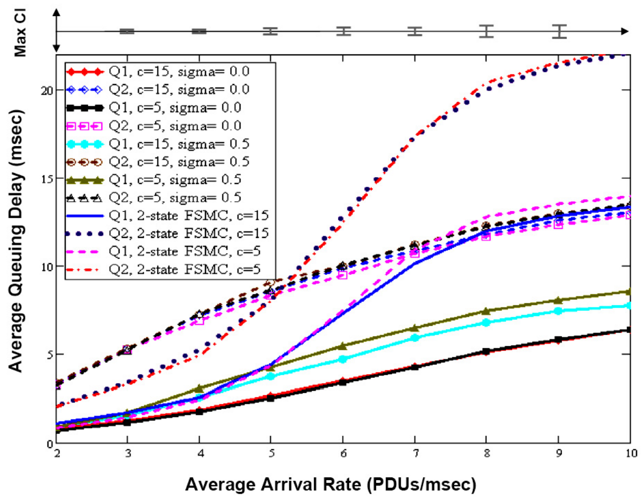


Fig. 14. Average queuing delay vs. load; 3-state FSMC model compared to 2-state FSMC. First eight cases in legend corresponds to the optimal policy with 3-state FSMC model.

users. This will result in unfairness and increased losses to user 2 traffic. In this case, the queuing delay increases and results in poor delay performance provided to user 2 as shown in Figure 14.

The fairness factor (σ), that did not have an effect in the 2-state case, has a significant effect when using the 3-state FSMC model as seen in Figure 13. When the load increases, the optimal policy with $\sigma = 0.0$ achieves higher system throughput compared to that with $\sigma = 0.5$. This agrees with intuition since the policy corresponding to $\sigma = 0.0$ aims at optimizing the throughput at the expense of fairness. Higher σ will result in more fairness at the cost to the overall system throughput as shown in Figures 13 and 14.

Remark: The 3-state FSMC model is obviously a more accurate representation of the HSDPA downlink transmission system than the 2-state model. In 3GPP R'5 standard [3GPP 2004] there are 31 transmission levels (TFRC) which corresponds to 32 channel states. However, in real life only a subset of that range (usually 6 TFRCs) is used (see [Holma and Toskala 2004] for an example).

7.3 Heuristic Policy Evaluation

The system throughput when using the heuristic policy we developed in section 6 compared to the optimal and Round Robin policies is shown in Table I for different loading conditions. The channel model parameters was chosen (2-state FSMC channel is assumed) such that $P(\gamma_1 = 1) = 0.8$ and $P(\gamma_2 = 1) = 0.5$. Table I shows that the throughput performance of the heuristic policy is very close to that of the optimal policy. It also shows that RR performance is close to that of the optimal policy in case of light loading (e.g., $\rho = 0.5$). However, it performs around 27% worse than the optimal policy in heavy load conditions when $\rho = 1.2$.

Queuing delay performance of the heuristic policy is shown in Table II in comparison with that of RR and the optimal policy. Queue lengths for the two users are shown in Tables III and IV for a different channel and arrival parameters. From

Table I. System throughput (PDUs/msec) for different policies and loading conditions.

ρ	*Optimal*	*Heuristic*	*RoundRobin*
0.5	3.25(±0.12)	3.25(±0.12)	3.25(±0.13)
0.8	5.0(±0.13)	4.95(±0.12)	4.45(±0.14)
1.2	6.73(±0.18)	6.71(±0.17)	4.9(±0.18)

Table II. Queuing delay performance for different policies, $P(\gamma_1=1)=0.8$, $P(\gamma_2=1)=0.5$, $q_1=0.8$, $q_2=0.5$ and $u=10$.

	Optimal	*Heuristic*	*RoundRobin*
*User*1	8.0(±0.42)	7.4(±0.38)	13.6(±0.46)
*User*2	10.8(±0.55)	12.2(±0.6)	19.9(±0.71)
Difference	2.8	4.8	6.3

Table III. Queue length, $\rho=0.75$, $P(\gamma_1=1)=0.8$, $P(\gamma_2=1)=0.5$, $q_1=0.5$, $q_2=0.5$ and $u=10$.

	Optimal	*Heuristic*	*RoundRobin*
*User*1	11.5(±0.54)	11.0(±0.45)	16.5(±0.6)
*User*2	14.5(±0.63)	16.0(±0.72)	34.5(±0.95)
Difference	3.0	5.0	18.0

these tables, the following conclusions can be drawn:

— The proposed heuristic policy compares favorably to the optimal one in terms of throughput and delay performance.

— The optimal policy provides better fairness in comparison to the heuristic and RR policies. This is apparent from the simulation results since the optimal policy achieved the smallest difference between the two users queuing delay. The heuristic policy provides a comparable performance to that of the optimal policy.

— The difference in queue lengths of the two users resulted from using the heuristic policy is reasonably close to that of the optimal policy under different channel and arrival parameters (Tables III and IV).

— The performance of the RR policy is highly dependent on the loading conditions. The results obtained proved that RR has poor performance in wireless channel.

8. CONCLUSION

In this work, we developed an MDP model for scheduling in 3G-HSDPA wireless systems. We used dynamic programming and value iteration to determine numerically the optimal scheduling policy. Value iteration is computationally demanding especially for large state space (e.g., larger number of users and/or wireless channel states). To counter this, we developed a heuristic approach to obtain a near-optimal policy. The suggested approach involves studying the structure and behavioral characteristics of the optimal policy using the MDP model. Based on that, we then construct a near-optimal heuristic scheduling policy that is shown to compare favorably with the optimal one.

Towards the derivation of the heuristic policy, we approximated the policy switching curves linearly (equations (18)-(20)). A non-linear approximation could be used, although we suspect that it may not provide substantial improvement. Extensions to more than two users can be realized following a similar approach.

Table IV. Queue length, $\rho = 1.1$, $P(\gamma_1 = 1) = 0.6$, $P(\gamma_2 = 1) = 0.6$, $q_1 = 0.8$, $q_2 = 0.5$ and $u = 10$.

	Optimal	*Heuristic*	*RoundRobin*
User1	33.0(±1.1)	33.5(±0.98)	45.5(±1.22)
User2	28.5(±0.8)	27.5(±0.74)	32.5(±0.92)
Difference	4.5	6.0	13.5

Furthermore, we studied the effect of the code allocation granularity on the optimal policy performance. Our results showed that a policy with finer granularity will perform better in light to moderate loading conditions, while a coarse policy is more desirable in heavy loading conditions. We also showed that the performance gain when using $c < 5$ is marginal and does not justify the added complexity. Our results also proved that Round Robin schedulers are not desirable in HSDPA system due to their poor performance and lack of fairness.

ELECTRONIC APPENDIX

The electronic appendix for this article can be accessed in the ACM Digital Library by visiting the following URL: `http://www.acm.org/pubs/citations/journals/tomacs/20YY-V-N/p1-zubaidy`.

REFERENCES

3GPP. 2004. High speed downlink packet access (hsdpa): Overall description (release 5). *3GPP Technical specification 5.7.0,* TS 25.308 (Dec.).

Al-Zubaidy, H., Lambadaris, I., and Viniotis, I. 2009. Optimal resource scheduling in wireless multi-service systems with random channel connectivity. In *(to appear in) Proceeding of IEEE Global Communications Conference (GLOBECOM 2009)*. IEEE, Honolulu, HI, USA.

Andrews, M. 2004. Instability of the proportional fair scheduling algorithm for hdr. *IEEE Transactions on Wireless Communications 3*, 1422–1426.

Bambos, N. and Michailidis, G. 2002. On parallel queuing with random server connectivity and routing constraints. *Probability in Engineering and Informational Sciences 16*, 185–203.

Bellman, R. 1957. *Dynamic Programming.* Princeton University Press, Princeton, USA.

Bonald, T. 2004. A score-based opportunistic scheduler for fading radio channels. In *Proceeding of the 5th European Wireless Conference*. IEEE, Barcelona, Spain.

Castro, J. P. 2004. *All IP in 3G CDMA Networks.* John Wiley & Sons Inc., NY, USA.

Ekstrom, H., Furuskar, A., Karlsson, J., Meyer, M., Parkvall, S., Torsner, J., and Wahlqvist, M. 2006. Technical solutions for the 3g long-term. evolution. *IEEE Communications Magazine*, 38–45.

Frenger, P., Parkvall, S., and Dahlman, E. 2001. Performance comparison of harq with chase combining and incremental redundancy for hsdpa. In *Proceedings of 54th IEEE Vehicular Technology Conference.* Vol. 3. Atlantic City, NJ, USA, 1829–1833.

Ganti, A., Modiano, E., and Tsitsiklis, J. N. 2007. Optimal transmission scheduling in symmetric communication models with intermittent connectivity. *IEEE Transactions on Information Theory 53*, 998–1008.

Hajek, B. 1984. Optimal control of two interacting service stations. *IEEE Trans. Automatic Control 29*, 491–499.

Hassan, M., Krunz, M., and Matta, I. 2004. Markov-based channel characterization for tractable performance analysis in wireless packet networks. *IEEE Transactions on Wireless Communications*.

Holma, H. and Toskala, A. 2004. *WCDMA for UMTS, Radio Access for Third Generation Mobile Communication, 3rd ed.* John Wiley & Sons Inc., NY, USA.

JEON, W. S., JEONG, D. G., AND KIM, B. 2004. Packet scheduler for mobile internet services using high speed downlink packet access. *IEEE Transactions on Wireless Communications 3,* 35 (Sept.).

JIANG, H., ZHUANG, W., AND SHEN, X. S. 2005. Cross-layer design for resource allocation in 3g wireless networks and beyond. *IEEE Communications Magazine*.

KELA, P., PUTTONEN, J., KOLEHMAINEN, N., RISTANIEMI, T., HENTTONEN, T., AND MOISIO, M. 2008. Dynamic packet scheduling performance in utra long term evolution downlink. In *Proceedings of the International Symposium on Wireless Pervasive Computing (ISWPC08)*. Santorini, Greece.

KOLDING, T. E., FREDERIKSEN, F., AND MOGENSEN, P. E. 2002. Performance aspects of wcdma systems with hsdpa. *IEEE 56th IEEE Vehicular Technology Conference (VTC) 1*, 477–481.

KOOLE, G., LIU, Z., AND RIGHTER, R. 2001. Optimal transmission policies for noisy channels. *Operations Research 49*, 892–899.

KUMAR, P. R. AND VARAIYA, P. P. 1986. *Stochastic Systems: Estimation, Identification and Adaptive Control*. Prentice Hall, Englewood Cliffs, NJ.

LIN, S. AND COSTELLO, D. 1983. *Error Control Coding: Fundamentals and Applications*. Prentice-Hall, Englewood Cliffs, NJ.

LIN, W. AND KUMAR, P. 1984. Optimal control of a queuing system with two heterogeneous servers. *IEEE Transactions on Automatic Control*.

LIU, X., CHONG, E., AND SHROFF, N. 2003. Optimal opportunistic scheduling in wireless networks. In *Proceedings of IEEE 58th Vehicular Technology Conference*. Vol. 3. Orlando, USA, 1417–1421.

MONGHAL, G., PEDERSEN, K., KOVACS, I., AND MOGENSEN, P. 2008. Qos oriented time and frequency domain packet schedulers for the utran long term evolution. In *Proceedings of 67th IEEE Vehicular Technology Conference*. Singapore, 2532–2536.

PEISA, J., WAGER, S., SAGFORS, M., TORSNER, J., GORANSSON, B., FULGHUM, T., COZZO, C., AND GRANT, S. 2007. High speed packet access evolution - concept and technologies. In *Proceedings of 65th IEEE Vehicular Technology Conference*. Dublin, Ireland.

PUTERMAN, M. 1994. *Markov Decision Process: Discrete Stochastic Dynamic Programming*. John wiley & Sons Inc., NY, USA.

ROSBERG, Z., VARAIYA, P., AND WALRAND, J. 1982. Optimal control of service in tandem queues. *IEEE Trans. Automatic Control 27*, 600–610.

ROSS, S. M. 1983. *Introduction to Stochastic Dynamic Programming*. Academic Press, USA.

SCHRIJVER, A. 1986. *Theory of Linear and Integer Programming*. John wiley & Sons Inc., Essex, GB.

SENNOTT, L. I. 1999. *Stochastic Dynamic Programming and The Control of Queuing Systems*. Wiley Series in Probability and Stochastics, New York, NY, USA.

STIDHAM, S. J. 1985. Optimal control of admission to a queuing system. *IEEE Transactions in Automatic Control 30*, 705–713.

STOYAN, D. 1983. *Comparison Methods for Queues and other Stochastic Models*. John Wiley and Sons, Chichester.

TASSIULAS, L. AND EPHREMIDES, A. 1993. Dynamic server allocation to parallel queues with randomly varying connectivity. *IEEE Transactions on Information Theory 39*, 466–478.

WALRAND, J. 1984. A note on optimal control of a queuing system with two heterogeneous servers. *Systems and Control Letters*.

WANG, H. S. AND MOAYERI, N. 1995. Finite-state markov channel–a useful model for radio communication channels. *IEEE Transactions on Vehicular Technology 44*, 163–171.

WEI, H. AND IZMAILOV, R. 2004. Channel-aware soft bandwidth guarantee scheduling for wireless packet access. In *IEEE Wireless Communications and Networking Conference (WCNC 04)*. Atlanta, USA, 1276–1281.

ZHANG, Q. AND KASSAM, S. 1999. Finite-state markov model for rayleigh fading channels. *IEEE Transactions on Communications*.