

RESEARCH

Open Access

Minimum-length scheduling with rate control in wireless networks: a shortest path approach

Anna Pantelidou^{1*} and Anthony Ephremides²

Abstract

In this paper, the minimum-length scheduling problem in wireless networks is studied, where each source of traffic has a finite amount of data to deliver to its corresponding destination. Our objective is to obtain a joint scheduling and rate control policy to minimize the total time required to deliver this finite amount of data from all sources. First, networks with time-invariant channels are considered. An optimal solution is provided by formulating the minimum-length scheduling problem as finding a shortest path on a single-source directed acyclic graph. However, finding the shortest paths is computationally hard since the number of vertices and edges of the graph increases exponentially in the number of network nodes, as well as in the initial traffic demand values. Toward this end, a simplified version of the problem is considered for which we explicitly characterize the optimal solution. Next, our results are generalized to time-varying channels. First, it is shown that in case of time-varying channels, the minimum-length scheduling problem can be formulated as a stochastic shortest path problem and then an optimal policy is provided that is based on stochastic control. Finally, our analytical results are illustrated with a set of numerical examples.

Keywords: Cross-layer design, Minimum-length scheduling, Rate control, Stochastic shortest paths

1. Introduction

The problem of minimum-length scheduling involves obtaining a sequence of activations of wireless nodes so that a finite amount of data residing at a subset of the nodes in the network reaches its intended destinations in minimum time. It is closely related to the problems of network throughput or stable throughput maximization, since minimizing the time to deliver a fixed amount of data can be seen as maximizing the effective rate at which data traverse the network. However, it also differs from them in that some network resources become available to heavily loaded nodes when the lightly loaded ones become relieved. Furthermore, it can be introduced as a useful alternative metric that characterizes the traffic-carrying capabilities of wireless networks with non-stationary and non-ergodic channel variations, where the commonly used performance criteria of stable throughput and network capacity are not well defined. Although in this paper we focus on networks with stationary and ergodic channel behavior, we

expect our analysis to yield valuable insights regarding the more general case of non-ergodic and non-stationary wireless channels.

The topic of obtaining schedules of minimum length has attracted the attention of the research community, as in [1-10] to sample a few. The first formulations of the problem of scheduling for efficient access to a shared channel, which we are aware of, appeared in [1,2]. A simple collision channel model had been considered but with the possibility of spatial reuse. That is, an “interference map” was assumed in terms of a graph that described all the independent sets of nodes in the graph, namely those sets of nodes that do not include “adjacent” nodes. The objective was to determine the shortest length of a frame of slots that would allow all nodes to transmit once in the frame without violating the “interference” constraint imposed by the interference rules on the graph. It was shown that the problem is NP-complete, and a distributed heuristic was developed that showed decent performance compared to the optimum that was computable in “small” instances of the problem.

* Correspondence: Anna.Pantelidou@renesasmobile.com

¹Renesas Mobile Corporation, Elektronikkatie 13, 90590 Oulu, Finland
Full list of author information is available at the end of the article

This problem was revisited in more generality through a continuous approximation of the structure of the frame schedule in [3], where the authors obtain a centralized, polynomial time algorithm for static networks that finds a schedule of minimum length satisfying a set of link-traffic requirements. However, in [3], modeling of the physical layer is overly simplified as it is assumed that any two links can be successfully activated simultaneously as long as they do not share any common vertices. This simplification relates the minimum-length scheduling problem to the problem of obtaining a maximal matching in a non-bipartite graph [11]. However, due to the broadcast nature of the wireless medium *all*, concurrent transmissions can potentially contribute to the total amount of interference at each receiver and, thus, cause a reception to fail.

A commonly accepted model for capturing the effects of interference is the Signal to Interference plus Noise Ratio (SINR) criterion under which the outcome of a transmission depends on the ratio of the signal power at a receiver to the noise and the total interference. If this ratio exceeds a certain threshold, then the transmission is assumed to be successful. Although still an approximation, this model is reasonable and captures the overall interference generated by the simultaneous transmissions in the network. A variation of this problem formulation that incorporated some physical layer attributes was studied in [4,5].

In [5], the authors consider the problem of obtaining a schedule of minimum length under the SINR interference model. They assume that the transmission rates are fixed and that each transmitting node selects optimally its transmission power. In [5], the minimum-length scheduling problem is formulated as a linear program [12] that can possibly have a prohibitively large number of variables and thus is hard to solve. In [6-9], the authors consider the minimum-length scheduling problem for different sets of optimization parameters. Specifically, they consider the cases where (1) both the transmission powers and rates are fixed, (2) the transmission powers can be optimized, but the transmission rates are fixed, and (3) the transmission powers are fixed and each transmitter is allowed to choose its rate from a predetermined, finite set of rates, common to all transmitters. In [6-9], the minimum-length scheduling problem is also formulated as a complex linear program, with a relatively small number of constraints and a large number of variables. To address the high complexity, the authors employ the technique of column generation [12], whose running time is faster, on average, than that of the original linear program.

Most of the prior work on the minimum-length scheduling problem assumes that the transmission rates are fixed. However, due to the broadcast nature of the

wireless medium, the parameters of the physical layer, such as the transmission powers and rates, are coupled with the scheduling decisions that can be made at the medium access control. In the recent years, there is ample evidence of how the variables associated with a particular layer depend on variables associated with other layers. Exploiting the links and dependencies between such variables usually yields superior solutions (see e.g., [13-17]). This is similar, if not equivalent, to maximizing a function of multiple variables by considering its dependence on all the variables rather than on a subset of them alone. Therefore, due to this coupling between the physical layer and the medium access control in wireless systems, it is clear that a joint optimization of link activation and rate control will yield a better performance, which is the focus of this paper.

In [10], the authors consider a cross-layer view of the minimum-length scheduling problem for static, single-hop networks through rate control and formulate the problem as a shortest path on a directed acyclic graph (DAG). In the first part of this paper, we consider static networks where the channel effect is due to pure path loss. We first assume a slotted-time model and formulate the minimum-length scheduling problem as a shortest path between a given source-destination pair on a DAG. We obtain an optimal joint scheduling and rate control solution that provides a shortest path. Although finding a shortest path on a DAG has a polynomial complexity in the number of vertices and edges, this number grows exponentially as the size of the network and initial data traffic load increase. For this reason, we make the following simplifications. We first map the discrete-time problem to a continuous-time equivalent, where slots are eliminated. We then restrict the possible scheduling and rate control decisions to either communication “one at a time”, in a Time Division Multiple Access (TDMA) fashion, or “all together” for all time. A similar approach for the problems of sum-rate maximization and proportional fairness was considered in [18]. We explicitly characterize the optimal solution of this reduced problem. Understanding the behavior of the optimal policy, even for the reduced problem, is significant since it provides valuable intuition about when scheduling and rate control actions for one or the other extreme are preferable. This intuition, for example, can improve the performance of the column generation technique in [6-9] by providing the algorithm with those scheduling and rate control actions that are expected to be employed by an optimal policy in the reduced problem and thus improve that heuristic.

However, the cited work (i.e., [1-10]) studies the minimum-length scheduling problem *only* for time-invariant wireless networks. Since the wireless channel in reality is time-varying, in the second part of this paper, we

consider the time-varying case. Our goal then becomes to find an optimal policy that minimizes the *expected time* required to deliver all the traffic to its respective destinations. We solve the minimum-length scheduling problem by formulating it as a *stochastic shortest path*, which is a special case of a Markov Decision Process (MDP) [19]. We obtain an optimal scheduling and rate control policy through stochastic control methods. Our approach is similar to the works in [20] and [21] where the minimum-length scheduling problem is formulated as an MDP under exact and statistical knowledge of the underlying channel conditions, respectively. For time-invariant channel processes, this model reduces to finding a shortest path on a DAG and methods described in the first part of this paper are applicable to compute the optimal solution.

Our work differs from [1-3] since we model the interference more accurately through the SINR interference model. We follow a different approach from [4-9] since we formulate the minimum-length scheduling problem as finding a shortest path on a single-source DAG, and we give an optimal graph-theoretic algorithm. Furthermore, we provide an explicit characterization of an optimal policy for a simplified, continuous-time model that is obtained by reducing the set of feasible scheduling and rate control decisions to either transmission in the “one at a time” fashion, as in TDMA, or in the “all together” mode. Our results are different from [4,5] since we consider joint scheduling and rate control decisions. Finally, and more importantly, we generalize existing work by considering the more realistic case of time-varying channels.

The rest of this paper is organized as follows. In Section II, we present the network model. In Section III, we consider static wireless networks. In particular, in Section III-A, we present a graph-theoretic formulation of the minimum-length scheduling problem as a shortest path problem on a single-source directed acyclic graph. In Section III-B, we first map the problem to a continuous-time model and we then restrict the set of feasible scheduling and rate control actions that can be employed. By doing so, we are able to explicitly characterize an optimal policy that finds a schedule of minimum length. Then, in Section IV, we consider time-varying wireless networks. In Section IV-A, we formulate the minimum-length scheduling problem as a stochastic shortest path, and in Section IV-B, we provide an optimal solution by employing the principles of stochastic control theory. Specifically, we use the value iteration method [19] to find a stochastic shortest path. In Section V, we complement our analytical results with some numerical experiments, and finally, in Section VI, we conclude the paper. In Appendix 1, a table of

variables is provided to make the notation comprehensively clearer to those who prefer to see the full notational picture. An optimal algorithm that computes a shortest path on a DAG is given in Appendix 2. The proofs of our results appear in Appendices 3-4.

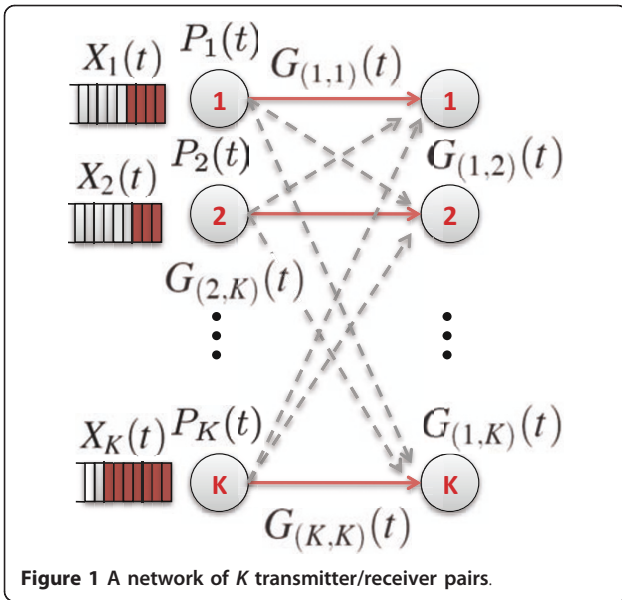
II. Model formulation

We consider a slotted-time, single-hop, wireless network consisting of K transmitter and receiver pairs. Without loss of generality, we assume that the slot duration is equal to 1 sec. Each transmitter has a finite amount of data units, e.g., a file of packets or bits to deliver to its corresponding receiver. The objective is to activate the transmitters so that the time to deliver all the traffic to the intended receivers is minimized. The single-hop network assumption, albeit simplifying, is interesting and highly non-trivial since it captures the fundamental problems that arise due to interference, when multiple nodes attempt simultaneous channel access. We denote by $\mathcal{X} = \{1, \dots, K\}$ the set of all transmitter and receiver pairs in the network. At every time slot, each transmitter $k \in \mathcal{X}$ can either transmit at its maximum transmission power P_k^{\max} or remain silent. We denote the transmission power level of the k th transmitter at time slot t by $P_k(t)$, where $P_k(t) \in \{0, P_k^{\max}\}$.

It is assumed that each transmitter k has a *fixed* amount of d_k bits to deliver to its corresponding destination. We denote by $\mathbf{d} = (d_1, \dots, d_K)$ the vector of initial data traffic at each transmitter. We also denote by $X_k(t)$ the queue size at transmitter k at time slot t and by $\mathbf{X}(t) = (X_1(t), \dots, X_K(t))$ the corresponding vector of queue sizes at all transmitters in the network. The queue size of each transmitter at time slot 0 is equal to its initial data traffic, i.e., $\mathbf{X}(0) = \mathbf{d}$. The state space of the process $\{\mathbf{X}(t)\}_{t=0}^{\infty}$ is denoted by \mathcal{X} .

We also consider a channel process $\{\mathbf{G}(t)\}_{t=0}^{\infty}$ that takes values from a finite set \mathcal{G} . For every time slot t , the channel state $\mathbf{G}(t) = (G_{(k,j)}(t), \forall k, j \in \mathcal{X})$ gives the channel quality between every transmitter k and receiver j in the network. This model captures the effects of channel variations due to e.g., node mobility, fading, or fixed path loss. It is assumed that the channel follows a block fading model with block length equal to the duration of a time slot. Hence, the channel conditions change *only* at the beginning of each time slot and remain constant throughout the slot duration. The network model under consideration is depicted in Figure 1.

We include the physical layer effects by adopting the Signal to Interference plus Noise Ratio (SINR) criterion. Specifically, a transmission from transmitter k to receiver k is successful if the ratio of the received signal power to the sum of the thermal noise and the total interference exceeds a certain threshold. The exact value



of the SINR threshold depends on various factors, such as the transmission rate, the target probability of bit error, the coding and modulation techniques employed at the transmission, etc. In this paper, we focus on the dependence of this threshold on the transmission rate and assume that the rest of the parameters affecting it are fixed. We denote by $\gamma_{t,k}(r_k(t))$ the SINR threshold value at receiver k that must be met or exceeded in order to receive successfully from transmitter k at rate $r_k(t)$ at time slot t . Consequently, we say that at slot t transmitter k transmits successfully to receiver k at rate $r_k(t)$ if

$$\text{SINR}_k(t) = \frac{P_k(t)G_{(k,k)}(t)}{N_k + \sum_{j=1, j \neq k}^K P_j(t)G_{(j,k)}(t)} \geq \gamma_{t,k}(r_k(t)), \quad (1)$$

where N_k is the thermal noise power at receiver k .

It is known that the maximum transmission rate is an increasing function of the SINR threshold (see e.g., [22]). This gives rise to the following *trade-off*: By increasing the transmission rate, the number of transmitters that can successfully satisfy the SINR criterion concurrently decreases. On the other hand, by decreasing the transmission rate, a higher number of transmitters can jointly satisfy the SINR criterion. Thus, it is not clear whether allowing more concurrent transmissions (less time sharing) at lower rates is preferable to allowing fewer concurrent transmissions (more time sharing) at higher rates. The answer is tightly dependent on the performance objective and on the network parameters, such as the transmission powers, channel conditions, etc.

A joint scheduling and rate control policy at any given time needs to decide (a) which transmitters to activate and (b) their respective transmission rates. This

information can be captured by the K -dimensional rate vector $\mathbf{r}(t) = (r_1(t), \dots, r_K(t))$, where $r_k(t)$ is the rate of transmitter k at slot t . If a transmitter is assigned a zero rate, then it is not activated by the policy. In other words, a transmission rate vector implicitly specifies the scheduling decisions. We define the set of all feasible rate vectors to contain those that are obtained by the following two-step procedure: We first identify all possible subsets of activated transmitters (by assigning to each transmitter k either power 0 or P_k^{\max}), and then, we assign them the maximum rates that allow all activated transmitters to jointly satisfy the SINR criterion. Thus, there exist $2^K - 1$ such K -dimensional transmission power vectors, each of which corresponds to an achievable rate vector. Clearly, the set of achievable rates depends on the current channel state $\mathbf{g} \in \mathcal{G}$. Hence, for every channel state \mathbf{g} , we denote by $\mathcal{R}(\mathbf{g})$ the finite, discrete set of feasible K -dimensional rate vectors. Then, the cardinality of $\mathcal{R}(\mathbf{g})$, i.e., $|\mathcal{R}(\mathbf{g})|$, is equal to $2^K - 1$ for every channel state $\mathbf{g} \in \mathcal{G}$.

In this paper, we are interested in obtaining optimal policies that take joint scheduling and rate control decisions under the objective of minimizing the (expected) time to deliver all data to the intended destinations. The policies we consider are aware of the network queue sizes at all times. Furthermore, unlike in [21], they are assumed to know the current channel conditions in order to make accurate scheduling decisions. For every slot t , the pair of the channel state $\mathbf{G}(t)$ and queue sizes $\mathbf{X}(t)$ constitutes the system state $\mathbf{S}(t)$. We denote by \mathcal{S} the state space of the system state process $\{\mathbf{S}(t)\}_{t=0}^{\infty}$, which is given by

$$\mathcal{S} = \{(\mathbf{x}, \mathbf{g}) : \mathbf{x} \in \mathcal{X}, \mathbf{g} \in \mathcal{G}\}. \quad (2)$$

We restrict our attention to *stationary* policies that take decisions merely based on the *current system state information*. Consider a state $\mathbf{s} = (\mathbf{x}, \mathbf{g}) \in \mathcal{S}$, and let the system state at time slot t satisfies $\mathbf{S}(t) = \mathbf{s}$. Let us also define the set $\mathcal{A}(\mathbf{s})$ to be a subset of the overall feasible scheduling and rate control decisions corresponding to state $\mathbf{s} = (\mathbf{x}, \mathbf{g})$, i.e., $\mathcal{A}(\mathbf{s}) \subseteq \mathcal{R}(\mathbf{g})$. Then, the policies we consider are given by the mapping

$$\mathbf{r}(t) = \boldsymbol{\pi}(\mathbf{s}), \quad \boldsymbol{\pi} : \mathcal{X} \times \mathcal{G} \rightarrow \mathcal{A}(\mathbf{s}) \subseteq \mathcal{R}(\mathbf{g}). \quad (3)$$

That is, we allow for the possibility of restricting the space of actions by limiting the range of allowable policies. If $\mathcal{A}(\mathbf{s})$ is a strict subset of the overall feasible scheduling and rate control decisions, then scheduling will be suboptimal in general at the benefit of decreased complexity. Furthermore, by selectively choosing the elements of the set $\mathcal{A}(\mathbf{s})$, it is possible to obtain performance close to optimal while achieving considerable reduction in computational complexity.

We assume that every admissible policy uses the channel state information rationally so that a scheduled transmission must always be successful. Naturally, as reflected by the cardinality of the set $\mathcal{R}(\mathbf{g})$, the policies we consider are *non-idling*, i.e., they always activate at least one transmitter that has a non-empty queue until all the queues in the network are empty. We call the class of stationary, non-idling policies given by the mapping (3) as *admissible*, and denote them by Π .

The queue size process evolves according to

$$\mathbf{X}(t+1) = [\mathbf{X}(t) - \mathbf{r}(t)]^+, \quad t \geq 0, \quad (4)$$

where the rate control and scheduling vectors $\mathbf{r}(t)$ are given according to the mapping in (3) and where $[z]^+ = \max\{z, 0\}$.

Clearly, the queue size at each transmitter k takes its maximum value at time slot 0, when it is equal to the initial demand d_k and, due to the absence of external arrivals, it keeps decreasing over time until it reaches zero. Under the above model, we proceed to formulate the minimum-length scheduling problem for static and time-varying networks.

III. Static networks

In this section, we restrict our attention to static networks, where the channel qualities $G_{(k,j)}(t)$ are equal for every time slot t , i.e., we ignore effects of fading or user mobility. Thus, the cardinality of the set \mathcal{G} is equal to one. To simplify notation, in this section, we denote the channel quality $G_{(k,j)}(t)$ as $G(k,j)$. We will drop this assumption in Section IV where we will consider time-varying channel processes. Furthermore, since there is a single channel state \mathbf{g} , to simplify notation, we will write \mathcal{X} to denote $\mathcal{R}(\mathbf{g})$ for $\mathbf{g} \in \mathcal{G}$ and \mathcal{A} to denote $\mathcal{A}(\mathbf{s})$ for $\mathbf{s} = (\mathbf{x}, \mathbf{g})$, $\mathbf{g} \in \mathcal{G}$. At every time slot t , the scheduling and rate control policy identifies a rate vector $\mathbf{r}(t) = (r_1(t), \dots, r_K(t)) \in \mathcal{A} \subseteq \mathcal{R}$ that specifies which transmitters are activated and their respective rates.

We can formulate the minimum-length scheduling problem as follows:

$$\text{minimize: } T \quad (5)$$

$$\text{subject to: } \mathbf{X}(T) = 0, \mathbf{X}(0) = \mathbf{d}, \quad (6)$$

$$T \in \mathbb{N}. \quad (7)$$

In the specific case of pure TDMA scheduling, combined with rate control, where only a single transmitter can be active at any given time, the solution of the above problem becomes trivial. Specifically, each transmitter must be active for as many time slots as needed to empty its queue. The required number of such time

slots for each transmitter k is equal to the ratio of its initial demand d_k divided by its corresponding rate when it accesses the channel individually, rounded upwards to the closest integer value. Then, the minimum total time that is needed until all the queues are empty is equal to the sum of the time slots required by each transmitter. The order in which the transmitters must be activated is immaterial; they can be chosen in a round-robin or random fashion.

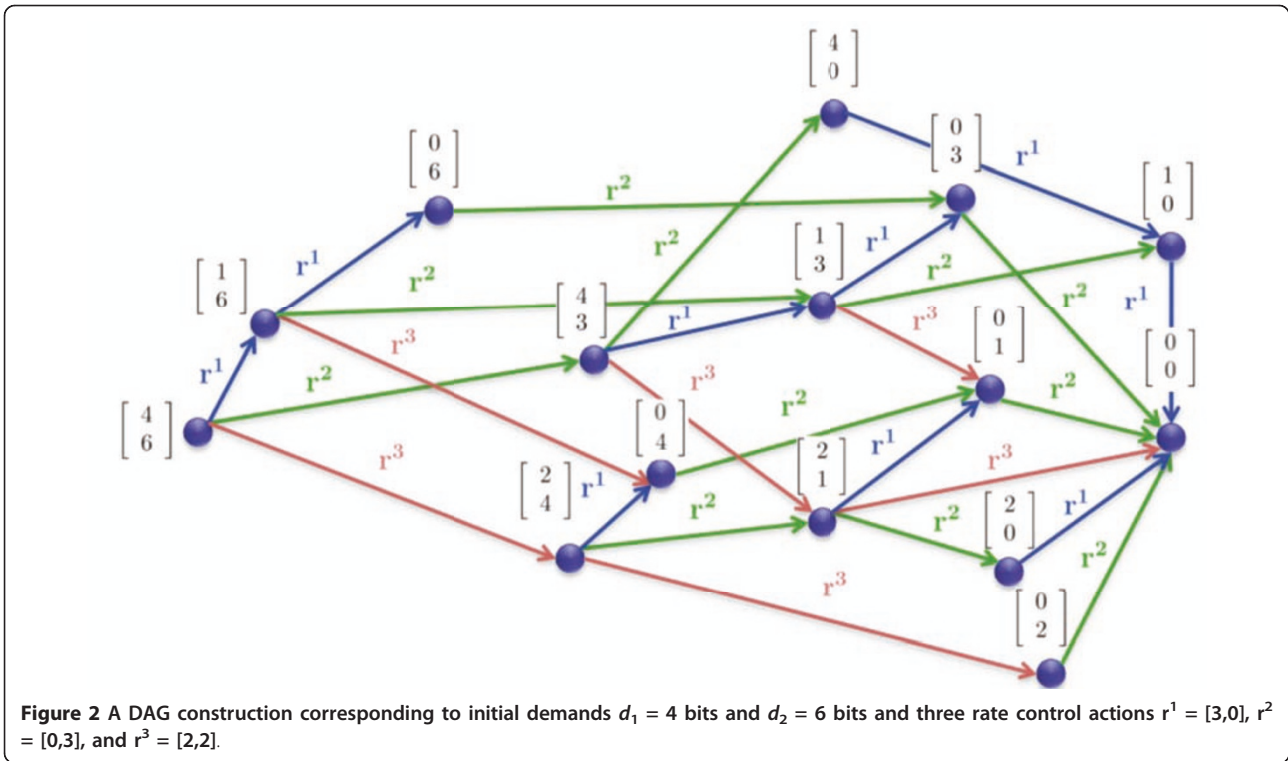
However, the solution of the optimization problem given by (5)-(7) is, in general, a non-trivial discrete optimization problem. In the following subsections, we provide an optimal graph-theoretic algorithm by mapping it to a shortest path problem on a DAG, and in the sequel, we give an explicit characterization of the optimal policy for a reduced version of this problem.

A. The equivalent DAG representation

We construct the weighted DAG $\tilde{G} = (V, E)$ as follows: We assume that every vertex $\mathbf{u} \in V$ of the DAG represents a queue size vector that can be obtained, starting from a vector of queues $\mathbf{X}(t)$, by employing some scheduling and rate control action chosen from the set \mathcal{A} . Furthermore, every directed edge $(\mathbf{u}, \mathbf{v}) \in E$ represents one such action in \mathcal{A} . We say that the edge (\mathbf{u}, \mathbf{v}) is *incident from* \mathbf{u} and *incident to* \mathbf{v} . Hence, from every vertex \mathbf{x}_i , we can have $|\mathcal{A}|$ edges that are incident from \mathbf{x}_i , each corresponding to a different rate vector \mathbf{r}^i , $i = 1, \dots, |\mathcal{A}|$. Each such edge is incident to a graph node $\mathbf{y}_i = [\mathbf{x}_i - \mathbf{r}^i]^+$. We disallow those edges that correspond to rate vectors, which activate transmitters with empty queues. Therefore, the actual number of edges that are incident from a vertex can be less than $|\mathcal{A}|$. The weight of each edge is equal to one. From now on, we will refer to action \mathbf{r}^i by means of the edge $(\mathbf{x}_i, \mathbf{y}_i)$. The unique source node \mathbf{x}_0 of the DAG corresponds to the vector of initial demands, $\mathbf{X}(0)$.

In Figure 2, we give an example of such a graph for a network of two transmitter and two receiver pairs. We assume that the initial demands are $d_1 = 4$ bits and $d_2 = 6$ bits and that we have three possible scheduling and rate control actions, namely (1) only transmitter 1 accesses the channel at a rate of 3 bits/sec, (2) only transmitter 2 accesses the channel at a rate of 3 bits/sec, and (3) both transmitters concurrently transmit at a rate of 2 bits/sec each. Figure 2 depicts the DAG that is obtained by these three actions. Note that from each vertex all the three rate control actions are allowed, as long as each action schedules transmitters with non-empty queues. For example, in Figure 2, the only viable rate control action for the queue size vector $[4,0]$ is to activate transmitter 1 individually.

As we observe from Figure 2 for any path of vertices $\langle \mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m \rangle$, the queue size vector of each vertex



in the path has to be component-wise larger or equal to the queue size of any other vertex that succeeds it in the path and the queue size vectors of any two vertices on the graph cannot be the same. As a result, the overall graph of the different queues is directed and acyclic. Finally, it is clear that every path starting at the source \mathbf{x}_0 ends at the $\mathbf{0}$ -vector. Moreover, the weight of any sub-path $\langle \mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m \rangle$ is equal to its length m , which is effectively the number of time slots required to go from vertex \mathbf{x}_0 to vertex \mathbf{x}_m along the specified path, since by construction of this DAG the weight of each edge is equal to one time slot. Thus, the initial problem given by (5)-(7) is transformed into a shortest path problem on a weighted single-source DAG. A shortest path on the DAG $G = (V, E)$ can be obtained through the DAG-SHORTEST-PATHS(\bar{G}, \mathbf{x}_0) algorithm. The exact algorithm, taken from [11], is given in Appendix 2.

For the example given in Figure 2, the shortest path algorithm selects the sequence of actions $\mathbf{r}^3, \mathbf{r}^3, \mathbf{r}^2$, and the minimum schedule length is equal to 3 sec (slots). Note that the sequences of actions $\mathbf{r}^2, \mathbf{r}^3, \mathbf{r}^3$ and $\mathbf{r}^3, \mathbf{r}^2, \mathbf{r}^3$ are also optimal as the order in which the actions are taken is immaterial in terms of minimizing the time needed to empty the queues, under the assumption of static channels. Also, it is worth mentioning that the length of the optimal schedule obtained through rate control is, naturally, no longer than that of TDMA (employing only actions \mathbf{r}^1 and \mathbf{r}^2), which, in this

example, has length 4. Furthermore, it is reasonable to expect that the difference in the schedule lengths under the two schemes can become significant when the number of transmitter/receiver pairs in the network increases or when the values of initial demands are large. This will also be illustrated through a set of numerical results in Section V.

The optimality of DAG-SHORTEST-PATHS(\bar{G}, \mathbf{x}_0) can easily be verified (see e.g., [11], Theorem 24.5). Also, it is easy to see that its overall running time is $\Theta(|V| + |E|)$. Hence, the number of operations needed to compute a shortest path on a single-source DAG is of polynomial complexity in the number of vertices and edges. However, in our DAG construction, this number grows exponentially (1) in the number of transmitters when $\mathcal{A} = \mathcal{K}$ since from every vertex there exist $2^K - 1$ potential edges that are incident from it and (2) as the initial demands increase. Therefore, the overall complexity of the algorithm becomes exponential, and despite its theoretical merit, it is rendered impractical.

B. Continuous-time model

To decrease the complexity that stems from the discrete nature of the minimum-length scheduling problem, we can map the problem given by (5)-(7) to a continuous-time one. Therefore, instead of seeking the minimum number of time slots required to deliver all data traffic to its respective destinations, we will be interested to

obtain the minimum *duration* or *period of time* that has to elapse until all network queues empty. In this way, the minimum-length scheduling problem becomes a linear program with a relatively small number of constraints and a large number of variables as in the formulations of [5], [6], and [7]. In order to solve this linear program, we follow a different approach than [5], [6], and [7]. In particular, we reduce the number of variables involved, i.e., the scheduling and rate control decisions that the policy employs, and then obtain an optimal solution for this reduced problem.

As is commonly done, to understand the essential features of a difficult problem, one needs to consider a simplified version that inherits and keeps the key features of the problem. Specifically, here, we restrict the set \mathcal{A} to contain only feasible rate vectors obtained by two simple schemes, namely scheduling a *single* transmitter at a time (in a TDMA fashion) or concurrently activating *all* the transmitters, as considered in [23,24]. By doing so, we decrease the cardinality of \mathcal{A} to $K + 1$. Clearly, such a reduction is expected to yield suboptimal results. However, this simplification focuses attention on the trade-off between transmitting more frequently (at a reduced rate) and transmitting less frequently (at a higher rate). We expect that this approach will help us gain valuable insights regarding the nature of optimal scheduling and rate control for the general problem.

We define Action k for $k \in \mathcal{X}$ to consist of individually activating transmitter k and Action 0 to be the corresponding action when all K transmitters are activated simultaneously. Let the rate of transmitter k under individual operation be r_k^k and the corresponding

rate under concurrent operation be r_k^0 . Furthermore, let us denote by τ_i for $i \in \{0, \dots, K\}$ the period of time that Action i is utilized. The above are illustrated in Figure 3.

Then, the continuous-time equivalent of the problem given in (5)-(7) under the reduced space of actions is:

$$\text{minimize : } \sum_{i=0}^K \tau_i \tag{8}$$

$$\text{subject to: } d_k \leq \tau_k r_k^k + \tau_0 r_k^0, \quad \forall k \in \mathcal{X} \tag{9}$$

$$\tau_i \geq 0, \quad i \in \{0, \dots, K\}. \tag{10}$$

The following theorem characterizes an optimal scheduling and rate control policy that solves (8)-(10).

Theorem 1: A minimum-length scheduling and rate control policy solving (8)-(10) takes actions according to the following:

1) If it is true that

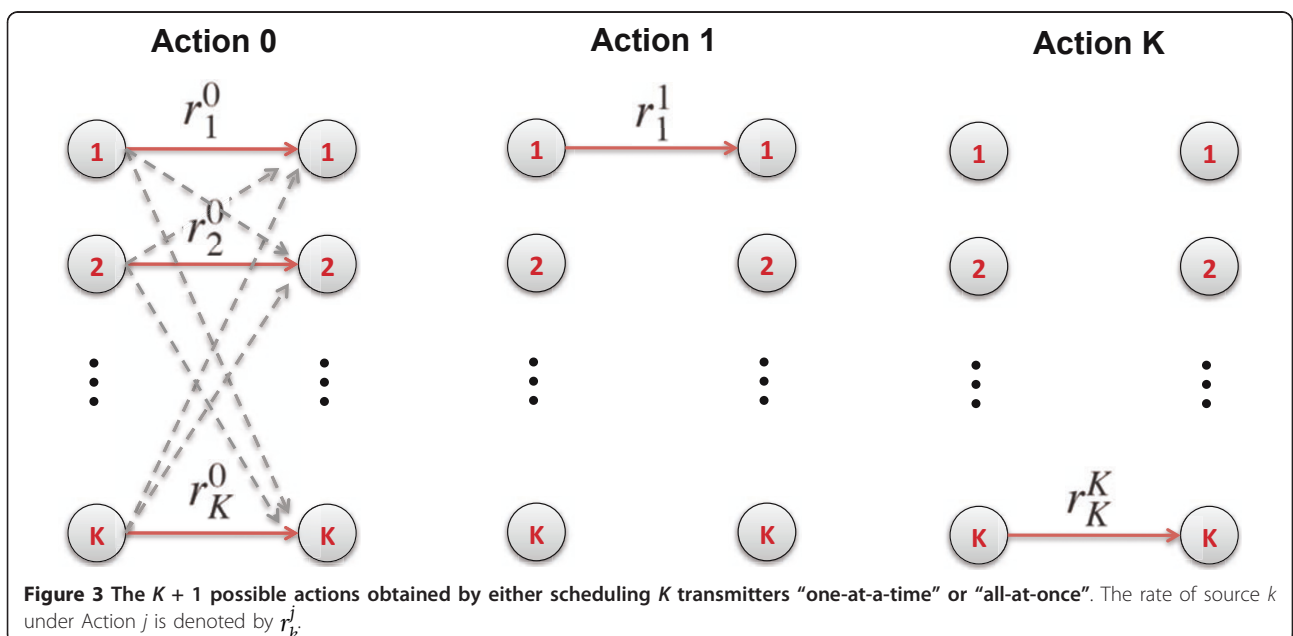
$$\sum_{k=1}^K \frac{r_k^0}{r_k^k} \leq 1, \tag{11}$$

then Action k is chosen ($k = 1, \dots, K$) for a duration of

$$\tau_k = \frac{d_k}{r_k^k}, \tag{12}$$

while Action 0 is never employed, i.e.,

$$\tau_0 = 0. \tag{13}$$



2) If it is true that

$$\sum_{k=1}^K \frac{r_k^0}{r_k^k} \geq 1, \quad (14)$$

then a subset of transmitters \mathcal{J} , where $\mathcal{J} \subseteq \mathcal{K}$, is chosen such that for every $k \in \mathcal{J}$, Action k is chosen for a duration of

$$\tau_k = \frac{d_k - \tau_0 r_k^0}{r_k^k}, \quad (15)$$

and Action 0 is selected for a period of

$$\tau_0 = \max_{i \in \mathcal{K} \setminus \mathcal{J}} \frac{d_i}{r_i^0}, \quad (16)$$

where for any two sets \mathcal{X}, \mathcal{Y} the operation \setminus is the set difference operation defined as

$$\mathcal{X} \setminus \mathcal{Y} = \{x : x \in \mathcal{X} \text{ and } x \notin \mathcal{Y}\}. \quad (17)$$

The proof appears in Appendix 3. To completely characterize the policy, we need to specify the set \mathcal{J} , which results from the following lemma.

Lemma 1: Consider an ordering of the transmitters in decreasing order of the values d_k/r_k^0 for every $k \in \mathcal{K}$. Let the corresponding indexing of the transmitters be $\{\ell_k\}_{k=1}^K$, that is $d_{\ell_1}/r_{\ell_1}^0 \geq \dots \geq d_{\ell_K}/r_{\ell_K}^0$. Then, the set \mathcal{J} contains those transmitters with the highest ratios d_k/r_k^0 and the cardinality $|\mathcal{J}|$ of the set \mathcal{J} is given by

$$|\mathcal{J}| = \arg \min_{k \in \{0, \dots, K\}} \left\{ \left(\max_{i \in \mathcal{K} \setminus \{0, \dots, k\}} \frac{d_{\ell_i}}{r_{\ell_i}^0} \right) + \sum_{\{j: j \geq 1, j \leq k\}} \frac{d_{\ell_j}}{r_{\ell_j}^j} - \left(\max_{i \in \mathcal{K} \setminus \{0, \dots, k\}} \frac{d_{\ell_i}}{r_{\ell_i}^0} \right) \sum_{\{j: j \geq 1, j \leq k\}} \frac{r_{\ell_j}^0}{r_{\ell_j}^j} \right\}. \quad (18)$$

The proof of the lemma appears in Appendix 4.

From the above, we conclude that the set \mathcal{J} contains the transmitters with the highest $|\mathcal{J}|$ values of d_k/r_k^0 , where $|\mathcal{J}|$ is given by Lemma 1. Hence, an optimal scheduling and rate control policy individually activates the transmitters that either have a very high initial demand or whose rates under concurrent operation are very low, e.g., due to excessive amounts of interference caused by other concurrent transmissions. Those transmitters must be further assisted toward emptying their queues by being granted individual access to the channel.

IV. Time-varying networks

In the previous section, we focused on time-invariant channels. However, in reality, the wireless channel is time-varying, due to the effects of fading, node mobility, etc. In this section, we extend our model by considering

time-varying channels. We make the following assumption on the wireless channel process $\{\mathbf{G}(t)\}_{t=0}^{\infty}$.

Assumption 1: The channel process $\{\mathbf{G}(t)\}_{t=0}^{\infty}$ varies according to a stationary Markov Chain with transition probability from some channel state $\mathbf{g} \in \mathcal{G}$ to another channel state $\mathbf{g}' \in \mathcal{G}$ given by

$$p_{\mathbf{G}}(\mathbf{g}, \mathbf{g}') = \text{Prob}[\mathbf{G}(t+1) = \mathbf{g}' | \mathbf{G}(t) = \mathbf{g}], \quad \forall \mathbf{g}, \mathbf{g}' \in \mathcal{G}. \quad (19)$$

Due to the time variability of the channel process, the length of the schedule T is a random variable and thus “minimum-length” is meant “in the expected sense”.

This can be formula ted as follows:

$$\text{minimize : } \mathbb{E}[T] \quad (20)$$

$$\text{subject to : } \mathbf{X}(T) = 0, \mathbf{X}(0) = \mathbf{d}, \quad (21)$$

$$T \in \mathbb{N}, \quad (22)$$

where the expectation is with respect to the stationary probability distribution of the channel process.

We proceed to present a solution to the problem of (20)-(22) through stochastic control methods by considering admissible policies in the class Π .

A. Stochastic shortest path formulation

Since the wireless channel process $\{\mathbf{G}(t)\}_{t=0}^{\infty}$ is Markov and since the process of the queue sizes evolves according to (4), for every admissible policy, it is easy to show that the system process $\{\mathbf{S}(t)\}_{t=0}^{\infty}$ is also a Markov Chain, with state space \mathcal{S} given by (2). The Markovianness of the system process is shown and verified in Appendix 5. We further define a subset $\mathcal{S}_{\text{term}}$ of the state space \mathcal{S} to be the set of *terminating states* that correspond to empty queues, i.e.,

$$\mathcal{S}_{\text{term}} = \{(\mathbf{x}, \mathbf{g}) : \mathbf{x} = \mathbf{0}, \mathbf{g} \in \mathcal{G}\}. \quad (23)$$

Evidently, from (4), this Markov Chain is absorbing and from every non-terminating state a terminating state is reached with probability one in finite time under all admissible policies. Furthermore, once the system reaches any state in $\mathcal{S}_{\text{term}}$, it remains there forever. Hence, the objective becomes to reach a terminating state in minimum expected time by choosing the next state. This will yield a schedule of minimum expected length. This is a *stochastic shortest path* problem, which is a special case of an MDP. If there is no randomness in the channel state, i.e., the entire wireless channel realization is known at priori at the very first time slot, our results of Section III follow from this model as a special case.

The set of feasible scheduling and rate control actions corresponding to each system state $\mathbf{s} = (\mathbf{x}, \mathbf{g}) \in \mathcal{S}$ is the

set $\mathcal{A}(\mathbf{s}) \subseteq \mathcal{R}(\mathbf{g})$. The system is driven by the time-varying channel process $\{\mathbf{G}(t)\}_{t=0}^{\infty}$. Taking an action leads to different states with different probabilities depending on the evolution of the channel process unless the system has already reached a terminating state.

Let $p_{\mathbf{r}}(\mathbf{s}, \mathbf{s}')$ be the transition probability from system state $\mathbf{s} = (\mathbf{x}, \mathbf{g})$ to state $\mathbf{s}' = (\mathbf{x}', \mathbf{g}')$ by taking action $\mathbf{r} = \boldsymbol{\pi}(\mathbf{x}, \mathbf{g}) \in \mathcal{A}(\mathbf{s})$. Then we have

$$p_{\mathbf{r}}(\mathbf{s}, \mathbf{s}') = \text{Prob}[\mathbf{X}(t+1) = \mathbf{x}', \mathbf{G}(t+1) = \mathbf{g}' | \mathbf{X}(t) = \mathbf{x}, \mathbf{G}(t) = \mathbf{g}, \boldsymbol{\pi}(\mathbf{x}, \mathbf{g}) = \mathbf{r}]. \quad (24)$$

From (4) and Assumption 1, it is easy to see that $p_{\mathbf{r}}(\mathbf{s}, \mathbf{s}')$ can be written as

$$p_{\mathbf{r}}(\mathbf{s}, \mathbf{s}') = \begin{cases} p_{\mathbf{G}}(\mathbf{g}, \mathbf{g}'), & \text{if } (\mathbf{x} - \mathbf{r})^+ = \mathbf{x}', \mathbf{s}, \mathbf{s}' \in \mathcal{S} \\ 0, & \text{otherwise.} \end{cases} \quad (25)$$

Note that from the Markovianess of the channel process and the admissibility of the policy $\boldsymbol{\pi}$, the transition probability $p_{\mathbf{r}}(\mathbf{s}, \mathbf{s}')$ is time invariant and does not depend on the previous system states.

We define the cost of taking action \mathbf{r} and going from state \mathbf{s} to state \mathbf{s}' as $\tilde{c}_{\mathbf{r}}(\mathbf{s}, \mathbf{s}')$. For every system state \mathbf{s} , action $\mathbf{r} \in \mathcal{A}(\mathbf{s})$ and system state \mathbf{s}' such that $p_{\mathbf{r}}(\mathbf{s}, \mathbf{s}') > 0$, we assume that $\tilde{c}_{\mathbf{r}}(\mathbf{s}, \mathbf{s}') = 1$. This represents the fact that in order to go from state \mathbf{s} to state \mathbf{s}' by taking this action one needs to spend one time slot. Let us further define the *cost per stage* $c_{\mathbf{r}}(\mathbf{s})$ to be the expected cost when, being at state $\mathbf{s} \in \mathcal{S} \setminus \mathcal{S}_{\text{term}}$, control action $\mathbf{r} \in \mathcal{A}(\mathbf{s})$ is chosen. It is clear that $c_{\mathbf{r}}(\mathbf{s}) = \sum_{\mathbf{s}' \in \mathcal{S}} p_{\mathbf{r}}(\mathbf{s}, \mathbf{s}') \tilde{c}_{\mathbf{r}}(\mathbf{s}, \mathbf{s}') = 1, \forall \mathbf{s} \in \mathcal{S} \setminus \mathcal{S}_{\text{term}}$. Once a terminal state $\mathbf{s} \in \mathcal{S}_{\text{term}}$ is reached, no more cost is incurred and the system remains there forever, i.e., $c_{\mathbf{r}}(\mathbf{s}) = 0, \forall \mathbf{r} \in \mathcal{A}(\mathbf{s}), \mathbf{s} \in \mathcal{S}_{\text{term}}$. Having said the above, the minimum-length scheduling problem can be formulated as a stochastic shortest path. Next, we provide an optimal policy that is obtained through Dynamic Programming [19].

B. An optimal policy

Let $\mathcal{T}^{\boldsymbol{\pi}}(\mathbf{s})$ be the expected time to empty the queues in the network starting from state \mathbf{s} under a policy $\boldsymbol{\pi} \in \Pi$. Clearly, $\mathcal{T}^{\boldsymbol{\pi}}(\mathbf{s}) = 0$ under any policy $\boldsymbol{\pi} \in \Pi$ for every terminating state $\mathbf{s} \in \mathcal{S}_{\text{term}}$. Then, the minimum expected schedule length $\mathcal{T}^*(\mathbf{s})$ is given by

$$\mathcal{T}^*(\mathbf{s}) = \min_{\boldsymbol{\pi} \in \Pi} \mathcal{T}^{\boldsymbol{\pi}}(\mathbf{s}), \quad \forall \mathbf{s} \in \mathcal{S} \setminus \mathcal{S}_{\text{term}}. \quad (26)$$

A policy $\boldsymbol{\pi}^*$ is optimal if it achieves the minimum $\mathcal{T}^*(\mathbf{s})$ for every non-terminating state $\mathbf{s} \in \mathcal{S} \setminus \mathcal{S}_{\text{term}}$, i.e.,

$$\mathcal{T}^{\boldsymbol{\pi}^*}(\mathbf{s}) = \mathcal{T}^*(\mathbf{s}), \quad \forall \mathbf{s} \in \mathcal{S} \setminus \mathcal{S}_{\text{term}}. \quad (27)$$

To solve the above shortest path problem, two commonly used methods are the *policy iteration* and the *value iteration* [19]. Due to the large state space of the problem, value iteration is easier to compute and, hence, will be used here. Consider the value iteration algorithm and the corresponding “expected” time $\mathcal{T}_k(\mathbf{s})$ to empty the queues starting from state \mathbf{s} at the k th iteration. Assume that $\mathcal{T}_0(\mathbf{s}) = \infty$ for all states $\mathbf{s} \in \mathcal{S}$. We borrow the following properties from [19].

Lemma 2: The value iteration method converges to the optimal cost function, i.e.,

$$\mathcal{T}^*(\mathbf{s}) = \lim_{k \rightarrow \infty} \mathcal{T}_k(\mathbf{s}), \quad \forall \mathbf{s} \in \mathcal{S} \setminus \mathcal{S}_{\text{term}}, \quad (28)$$

where

$$\mathcal{T}_{k+1}(\mathbf{s}) = 1 + \min_{\mathbf{r} \in \mathcal{A}(\mathbf{s})} \sum_{\mathbf{s}' \in \mathcal{S}} p_{\mathbf{r}}(\mathbf{s}, \mathbf{s}') \mathcal{T}_k(\mathbf{s}'), \quad \mathbf{s} \in \mathcal{S} \setminus \mathcal{S}_{\text{term}}. \quad (29)$$

Lemma 29, borrowed from [19], shows the optimality of the value iteration method to solve problems of stochastic control. Therefore, by employing the value iteration method, we can obtain a solution to the minimum-length scheduling problem in time-varying networks.

Lemma 3: The optimal solution to a stochastic shortest path problem must satisfy Bellman’s equation, i.e., for every non-terminating state $\mathbf{s} \in \mathcal{S} \setminus \mathcal{S}_{\text{term}}$, it must be true that

$$\mathcal{T}^*(\mathbf{s}) = 1 + \min_{\mathbf{r} \in \mathcal{A}(\mathbf{s})} \left\{ \sum_{\mathbf{s}' \in \mathcal{S}} p_{\mathbf{r}}(\mathbf{s}, \mathbf{s}') \mathcal{T}^*(\mathbf{s}') \right\}. \quad (30)$$

Hence, the optimal scheduling and rate control policy $\boldsymbol{\pi}^*$ for every state $\mathbf{s} \in \mathcal{S} \setminus \mathcal{S}_{\text{term}}$ is given by

$$\boldsymbol{\pi}^*(\mathbf{s}) = \arg \min_{\mathbf{r} \in \mathcal{A}(\mathbf{s})} \left\{ \sum_{\mathbf{s}' \in \mathcal{S}} p_{\mathbf{r}}(\mathbf{s}, \mathbf{s}') \mathcal{T}^*(\mathbf{s}') \right\}, \quad \forall \mathbf{s} \in \mathcal{S} \setminus \mathcal{S}_{\text{term}}. \quad (31)$$

Although the value iteration method optimally solves the aforementioned stochastic shortest path problem, in general it may require an infinite number of iterations until it converges. However, if the Markov Chain of the system evolution is acyclic, then it was shown in [19] that the value iteration method for each state converges in a finite number of iterations (at most as many as the number of non-terminating states of the Markov Chain). It is easy to see that the Markov Chain driving our system is acyclic. This is because starting from one of the states whose queue size satisfies $\mathbf{X}(0) = \mathbf{d}$, the queue sizes in the network are non-increasing with time as given in (4) under any admissible policy.

V. Numerical results

In this section, we consider slotted-time, static and time-varying single-hop wireless networks and illustrate our analytical results with a few numerical calculations. We did not find it essential to provide numerical results on the continuous-time reduced problem since for that case we explicitly characterize the corresponding policy. Thus, we only provide numerical results for the value iteration method in the discrete-time case for static and time-varying networks. We consider a network of two transmitter/receiver pairs as in Figure 1 with $K = 2$. The channel process $\{G(t)\}_{t=0}^{\infty}$ is Markov and switches between two states, namely a *good* state, \mathbf{G} , and a *bad* state, \mathbf{B} . When the channel is in the good state, both transmitters have channels of good quality to their receivers; otherwise, both channels are bad. The transition probabilities of this Markov Chain are shown in Figure 4.

Since we have 2 transmitter/receiver pairs, there exist 3 possible rate vectors corresponding to each channel state, denoted by $\mathbf{r}^k(\mathbf{g})$, $k = 1, 2$, when only the k th transmitter is activated and by $\mathbf{r}^3(\mathbf{g})$, when both transmitters are activated, under channel state $\mathbf{g} \in \{\mathbf{B}, \mathbf{G}\}$.

We first assume that the initial demands are $d_1 = 4$ bits and $d_2 = 6$ bits, which is the case discussed in Section III. We consider 3 scenarios associated with different achievable rates for each channel state.

- **Scenario 1:** When the k th transmitter is activated alone, its achievable rate is 3 bits/sec, and when both transmitters are activated simultaneously, the corresponding rate is 2 bits/sec for each. Under this scenario, the channel realization is immaterial and the minimum expected time to empty the queues is 3 sec (slots), i.e., equal to the result of the static network example of Section III.
- **Scenario 2:** Under the good channel state, \mathbf{G} , the achievable rates are equal to the case of Scenario 1, i.e., when the k th transmitter is activated alone, its achievable rate is 3 bits/sec and when both transmitters are activated simultaneously, the corresponding rate is 2 bits/sec for each. However, under the bad channel, \mathbf{B} , the achievable rates are strictly worse (2

bits/sec for individual transmission and 1 bit/sec for each transmitter under concurrent transmission). Naturally, we anticipate that the expected time required to empty the queues is more than 3 sec (slots).

- **Scenario 3:** We assume that under the bad channel state, \mathbf{B} , the achievable rates are equal to the ones in Scenario 1, but the good channel is better and thus allows higher rates (4 bits/sec when a transmitter is activated individually and 3 bits/sec when they are both activated simultaneously). Naturally, the expected time to empty the queues will decrease to a value less than 3 sec (slots).

The above 3 scenarios and the corresponding minimum expected schedule lengths are shown in Table 1, where we have assumed that the channel starts from a good channel state. Similar results were observed for higher initial demands ($d_1 = d_2 = 100$ bits), which are also given in Table 1. Note that the above values of initial demands and transmission rates can be scaled accordingly to give meaningful values for real systems. Note also that although time is measured in terms of time slots, the average minimum expected length takes fractional values since it is computed by taking the expectation with respect to the channel state probability distribution.

In Figure 5, we illustrate the performance comparison between the optimal policy and a pure TDMA scheme that activates only a single transmitter at a time. Specifically, we consider the same single-hop network of two

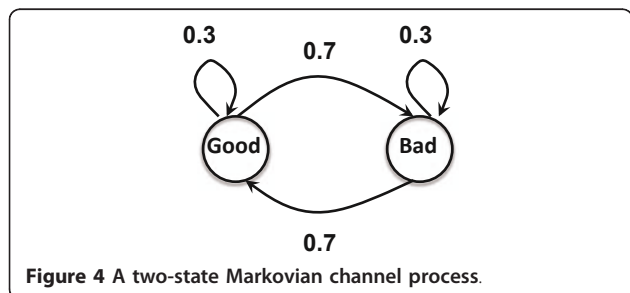


Figure 4 A two-state Markovian channel process.

Table 1 Expected time required to empty the queues for different values of initial demands, under Scenarios 1-3, assuming that the channel starts from a good state.

Demands	Good Channel			Bad Channel			$E[T]$
	$\mathbf{r}^1(\mathbf{G})$	$\mathbf{r}^2(\mathbf{G})$	$\mathbf{r}^3(\mathbf{G})$	$\mathbf{r}^1(\mathbf{B})$	$\mathbf{r}^2(\mathbf{B})$	$\mathbf{r}^3(\mathbf{B})$	
$\begin{bmatrix} 4 \\ 6 \end{bmatrix}$	$\begin{bmatrix} 3 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 3 \end{bmatrix}$	$\begin{bmatrix} 2 \\ 2 \end{bmatrix}$	$\begin{bmatrix} 3 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 3 \end{bmatrix}$	$\begin{bmatrix} 2 \\ 2 \end{bmatrix}$	3.00
	$\begin{bmatrix} 3 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 3 \end{bmatrix}$	$\begin{bmatrix} 2 \\ 2 \end{bmatrix}$	$\begin{bmatrix} 2 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 2 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$	3.62
$\begin{bmatrix} 4 \\ 4 \end{bmatrix}$	$\begin{bmatrix} 4 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 4 \end{bmatrix}$	$\begin{bmatrix} 3 \\ 3 \end{bmatrix}$	$\begin{bmatrix} 3 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 3 \end{bmatrix}$	$\begin{bmatrix} 2 \\ 2 \end{bmatrix}$	2.79
	$\begin{bmatrix} 3 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 3 \end{bmatrix}$	$\begin{bmatrix} 2 \\ 2 \end{bmatrix}$	$\begin{bmatrix} 2 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 2 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$	66.86
$\begin{bmatrix} 100 \\ 100 \end{bmatrix}$	$\begin{bmatrix} 4 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 4 \end{bmatrix}$	$\begin{bmatrix} 3 \\ 3 \end{bmatrix}$	$\begin{bmatrix} 3 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 3 \end{bmatrix}$	$\begin{bmatrix} 2 \\ 2 \end{bmatrix}$	40.25

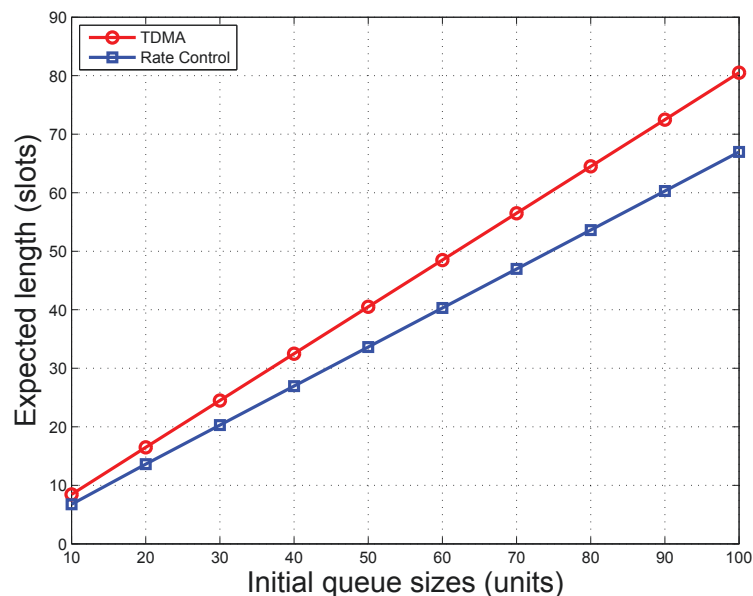


Figure 5 Performance comparison of the optimal policy with respect to TDMA scheduling.

transmitter/receiver pairs discussed above under Scenario 2. We plot the expected schedule length for the above two schemes as a function of the values of the initial data traffic, where slots are in units of seconds and queue sizes are in bits. For simplicity, the initial queue sizes at each transmitter are assumed to be equal. As expected, we observe that the difference between the expected time to empty the queues under the optimal policy and under the TDMA scheme increases as the initial queue sizes increase. This result illustrates the fact that permitting concurrent transmissions through rate control can provide considerable gains.

VI. Conclusions

In this paper, we focused on the problem of joint scheduling and rate control in single-hop wireless networks under the objective of minimizing the required time to deliver all data traffic to the intended destinations. First, we focused on networks with time-invariant links. We presented a graph-theoretic formulation for the minimum-length scheduling problem as a shortest path on a single-source directed acyclic graph. Motivated by the combinatorial nature of the minimum-length scheduling problem, and we then mapped it to a continuous-time formulation and restricted the set of feasible scheduling and rate control actions. By doing so, we were able to explicitly characterize an optimal policy for the reduced problem that finds a schedule of minimum length. Finally, we considered time-varying wireless networks. We formulated the minimum-length scheduling problem as a stochastic shortest path and presented an

optimal policy by employing the principles of stochastic control theory.

Appendix 1: Table 2 table of variables

Here, we give a list with the most commonly appearing variables in the paper to facilitate the reader. The variables are listed in the order they appear in Table 2 below.

Appendix 2: Finding a shortest path on a dag

Shortest path problems on single-source DAGs can be solved optimally in polynomial time [11]. Below, we provide an optimal algorithm, taken from [11], that finds a shortest path on a DAG.

In order to compute a shortest path, we first need to sort the DAG in *topological order* and then use a sequence of edge *relaxations* until we obtain a shortest path from the source \mathbf{x}_0 to the vertex corresponding to the $\mathbf{0}$ vector. Topological order is a linear ordering of all the vertices of the DAG so that for every edge $(\mathbf{x}_i, \mathbf{x}_j)$, the vertex \mathbf{x}_i appears before \mathbf{x}_j in the ordering. The process of edge relaxation verifies whether the current best-known path from the source \mathbf{x}_0 to a vertex \mathbf{y} can be improved by passing through a different vertex \mathbf{x} .

We proceed with a few definitions that will be useful in the rest of this appendix. We define the *distance* of a vertex \mathbf{x} to be the minimum distance in terms of edges that must be traversed from the source to reach \mathbf{x} . We also denote by $\delta[\mathbf{x}]$ an *upper bound* on the distance of vertex \mathbf{x} . For every edge (\mathbf{x}, \mathbf{y}) , we say that \mathbf{x} is the predecessor of \mathbf{y} and we write $\mathbf{x} = \text{Pred}[\mathbf{y}]$. We denote by

Table 2 Table of variables

Variable	Explanation
K	Number of transmitter and receiver pairs
\mathcal{X}	Set of transmitter and receiver pairs
P_k^{\max}	Maximum instantaneous transmission power of transmitter k
$P_k(t)$	Transmission power level of transmitter k at time slot t
d_k	Initial data traffic at transmitter k
\mathbf{d}	Vector of initial traffic at all transmitters
$X_k(t)$	Queue size of transmitter k at time slot t
$\mathbf{X}(t)$	Queue size vector of all transmitters in the network at slot t
$\{\mathbf{X}(t)\}_{t=0}^{\infty}$	Process of queue sizes of all transmitters
\mathcal{X}	State space of the process $\{\mathbf{X}(t)\}_{t=0}^{\infty}$
$G_{(k,j)}(t)$	Channel quality between transmitter k and receiver j at slot t
$\mathbf{G}(t)$	Channel state at time slot t
$\{\mathbf{G}(t)\}_{t=0}^{\infty}$	Channel process
\mathcal{G}	State space of the channel process
$\gamma_{t,k}(r_k(t))$	SINR threshold at receiver k at slot t to receive at rate $r_k(t)$
$r_k(t)$	Transmission rate of transmitter k at time slot t
N_k	Thermal noise power at receiver k
$\mathbf{r}(t)$	Vector of transmission rates of all transmitters at time slot t
$\mathcal{R}(\mathbf{g})$	Set of all achievable rates under channel state \mathbf{g}
$\{\mathbf{S}(t)\}_{t=0}^{\infty}$	System process
\mathcal{S}	State space of the system process
$\mathcal{A}(\mathbf{s})$	Subset of the total rate control actions corresponding to state $\mathbf{s} = (\mathbf{x}, \mathbf{g})$
π	Admissible policy
Π	Class of admissible policies
$G(k,j)$	Channel between transmitter k and receiver j in static networks
\mathcal{R}	Set of all achievable rates under static networks
\mathcal{A}	Subset of the total rate control actions in static networks
T	Timeslot number
$\bar{\mathcal{G}} = (V, E)$	Graph $\bar{\mathcal{G}}$ with V vertices and E edges
(\mathbf{u}, \mathbf{v})	Edge that is incident from vertex \mathbf{u} and incident to vertex \mathbf{v}
m	Path length on a DAG
Action k	Action activating only transmitter k
Action 0	Action activating all transmitters concurrently
r_k^k	Rate of transmitter k under Action k
r_k^0	Rate of transmitter k under Action 0
τ_i	Time duration that Action $i \in \{0, \dots, K\}$ is used
\mathcal{J}	Set of transmitters to be activated individually
$p_{\mathcal{G}}(\mathbf{g}, \mathbf{g}')$	Transition probability from channel state \mathbf{g} to \mathbf{g}'
$\mathcal{S}_{\text{term}}$	Set of terminating states
$p_r(\mathbf{s}, \mathbf{s}')$	Transition probability from system state \mathbf{s} to \mathbf{s}' under action \mathbf{r}
$\tilde{c}_r(\mathbf{s}, \mathbf{s}')$	Cost of taking action \mathbf{r} and going from state \mathbf{s} to state \mathbf{s}'
$c_r(\mathbf{s})$	Expected cost when being at state \mathbf{s} action \mathbf{r} is chosen
$\mathcal{T}^{\pi}(\mathbf{s})$	Expected time to empty the queues starting from \mathbf{s} under policy π
$\mathcal{T}^*(\mathbf{s})$	Minimum expected time to empty the queues starting from state \mathbf{s}
$\mathcal{T}_k^*(\mathbf{s})$	Expected time to empty the queues starting from \mathbf{s} at the k th iteration
π'	Optimal rate control policy

Adj[x] a list that contains all the vertices \mathbf{y} that are adjacent to \mathbf{x} , i.e., such that there exists an edge $(\mathbf{x}, \mathbf{y}) \in E$. The pseudo-code of the algorithm is provided below:

```
DAG-SHORTEST-PATHS( $\bar{G}, \mathbf{x}_0$ )
1 topologically sort the vertices of  $\bar{G}$ 
2 INITIALIZE-SINGLE-SOURCE( $\bar{G}, \mathbf{x}_0$ )
3 for each vertex  $\mathbf{x}$  taken in topologically sorted order
4   do for each vertex  $\mathbf{y} \in \text{Adj}[\mathbf{x}]$ 
5     do RELAX( $\mathbf{x}, \mathbf{y}$ )
```

The topological sorting of the first line of the algorithm can be completed in $\Theta(|V| + |E|)$ time, by running a Depth-First Search (DFS) [11]. The second line of the algorithm involves the initialization of various variables as shown next:

```
INITIALIZE-SINGLE-SOURCE( $\bar{G}, \mathbf{x}_0$ )
1 for each vertex  $\mathbf{x} \in V$ 
2   do  $\delta[\mathbf{x}] \leftarrow \infty$ 
3     Pred[ $\mathbf{x}$ ]  $\leftarrow NIL$ 
4    $\delta[\mathbf{x}_0] \leftarrow 0$ 
```

This process requires time order $\Theta(|V|)$. Finally, in lines 3-5 of the DAG-SHORTEST PATHS(\bar{G}, \mathbf{x}_0) algorithm, at each time step, the next vertex in the topological order is selected and a sequence of relaxations over all edges that are incident from this vertex is performed. The procedure RELAX(\mathbf{x}, \mathbf{y}), given next, verifies whether the current shortest path from \mathbf{x}_0 to \mathbf{y} can be improved by passing through \mathbf{x} .

```
RELAX( $\mathbf{x}, \mathbf{y}$ )
1 if  $\delta[\mathbf{y}] > \delta[\mathbf{x}] + 1$ 
2   then  $\delta[\mathbf{y}] \leftarrow \delta[\mathbf{x}] + 1$ 
3   Pred[ $\mathbf{y}$ ]  $\leftarrow \mathbf{x}$ 
```

Appendix 3: Proof of Theorem 1

We can write the Lagrangian of the problem described by (8)-(10) as:

$$L(\boldsymbol{\tau}, \boldsymbol{\mu}, \boldsymbol{\lambda}) = - \sum_{i=0}^K \tau_i + \sum_{k=1}^K \mu_k (\tau_k r_k^k + \tau_0 r_k^0 - d_k) + \sum_{i=0}^K \lambda_i \tau_i,$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\lambda}$ represent the Lagrange multipliers. The Karush-Kuhn-Tucker (KKT) conditions yield:

1) For every Action $k \in \mathcal{K}$ we have

$$\frac{\partial L(\boldsymbol{\tau}, \boldsymbol{\mu}, \boldsymbol{\lambda})}{\partial \tau_k} = -1 + \mu_k r_k^k + \lambda_k = 0. \quad (32)$$

2) For Action 0 we have

$$\frac{\partial L(\boldsymbol{\tau}, \boldsymbol{\mu}, \boldsymbol{\lambda})}{\partial \tau_0} = -1 + \sum_{k=1}^K \mu_k r_k^0 + \lambda_0 = 0. \quad (33)$$

3) For every Action $k \in \mathcal{K}$ it must be true that

$$\begin{aligned} \mu_k (\tau_k r_k^k + \tau_0 r_k^0 - d_k) = 0 &\Rightarrow \\ \mu_k \geq 0, \quad \tau_k r_k^k + \tau_0 r_k^0 &\geq d_k. \end{aligned} \quad (34)$$

4) For all actions $i \in \{0, \dots, K\}$ we have

$$\lambda_i \tau_i = 0 \Rightarrow \lambda_i \geq 0, \quad \tau_i \geq 0. \quad (35)$$

Consider the following cases:

Case 1: Assume that Action 0 is never employed, i.e., $\tau_0 = 0$. Since the traffic demands of every transmitter must be met, we have that $\tau_k > 0$ for every $k \in \mathcal{K}$. Hence, from (35), it follows that $\lambda_0 \geq 0$ and $\lambda_k = 0$ for every $k \in \mathcal{K}$. From (32), we obtain,

$$\mu_k = \frac{1}{r_k^k}. \quad (36)$$

Furthermore, since $\mu_k > 0$ and $\tau_0 = 0$, (34) yields

$$d_k = \tau_k r_k^k,$$

i.e., for every $k \in \mathcal{K}$ we obtain

$$\tau_k = \frac{d_k}{r_k^k}. \quad (37)$$

Finally, from (33) and (36) it follows that

$$\sum_{k=1}^K \frac{r_k^0}{r_k^k} \leq 1. \quad (38)$$

Case 2: Assume that Action 0 is employed for a non-negative amount of time and that also a subset \mathcal{J} of the transmitters is further selected to transmit individually. This implies that $\tau_0 > 0$, $\tau_j > 0$ for every $j \in \mathcal{J}$ and $\tau_i = 0$ for every $i \in \mathcal{K} \setminus \mathcal{J}$. Hence, (35) yields $\lambda_0 = 0$, $\lambda_j = 0$ for every $j \in \mathcal{J}$ and $\lambda_i \geq 0$ for every $i \in \mathcal{K} \setminus \mathcal{J}$. Also, for every $j \in \mathcal{J}$, (32) yields

$$\mu_j = \frac{1}{r_j^j}, \quad (39)$$

and for every $i \in \mathcal{K} \setminus \mathcal{J}$, it follows that

$$\mu_i \leq \frac{1}{r_i^i}. \quad (40)$$

Moreover, from (34) and (39), for every $j \in \mathcal{J}$, we obtain

$$\tau_j = \frac{d_j - \tau_0 r_j^0}{r_j^j}, \quad (41)$$

and from (34) and (40), for every $i \in \mathcal{X} \setminus \mathcal{J}$, we have

$$d_i \leq \tau_0 r_i^0, \quad (42)$$

or equivalently

$$\tau_0 \geq \max_{i \in \mathcal{X} \setminus \mathcal{J}} \frac{d_i}{r_i^0}. \quad (43)$$

Finally, from (33), (39), and (40), it follows that

$$\sum_{k=1}^K \frac{r_k^0}{r_k} \geq 1. \quad (44)$$

■

Appendix 4: Proof of Lemma 1

From (41), (42), and the fact that $\tau_j > 0$ for every $j \in \mathcal{J}$, for $i \in \mathcal{X} \setminus \mathcal{J}$, it follows that

$$0 < \tau_j \leq \frac{d_j r_i^0 - r_j^0 d_i}{r_j^0 r_i^0},$$

which yields that

$$\frac{d_i}{r_i^0} < \frac{d_j}{r_j^0}. \quad (45)$$

Consider now a new ordering $\{\ell_k\}_{k=1}^K$ of the transmitters in decreasing order of the values d_k/r_k^0 for every $k \in \mathcal{X}$, i.e., $d_{\ell_1}/r_{\ell_1}^0 \geq \dots \geq d_{\ell_K}/r_{\ell_K}^0$. Hence, from (45), it follows that there exists a threshold, i.e., a transmitter index value in the new ordering, below which all the transmitters must belong to the set \mathcal{J} and above which (that is, the remaining ones), they must belong to the set $\mathcal{X} \setminus \mathcal{J}$.

Since the objective is to minimize $\tau_0 + \sum_{j \in \mathcal{J}} \tau_j$, from (41) and (43), it follows that

$$\begin{aligned} |\mathcal{J}| &= \arg \min_{k \in \{0, \dots, K\}} \left\{ \tau_0 + \sum_{\{j \geq 1, j \leq k\}} \tau_{\ell_j} \right\} \\ &= \arg \min_{k \in \{0, \dots, K\}} \left\{ \tau_0 + \sum_{\{j \geq 1, j \leq k\}} \frac{d_{\ell_j} - \tau_0 r_{\ell_j}^0}{r_{\ell_j}^0} \right\} \\ &= \arg \min_{k \in \{0, \dots, K\}} \left\{ \left(\max_{i \in \mathcal{X} \setminus \{0, \dots, K\}} \frac{d_{\ell_i}}{r_{\ell_i}^0} \right) + \sum_{\{j \geq 1, j \leq k\}} \frac{d_{\ell_j}}{r_{\ell_j}^0} - \left(\max_{i \in \mathcal{X} \setminus \{0, \dots, K\}} \frac{d_{\ell_i}}{r_{\ell_i}^0} \right) \sum_{\{j \geq 1, j \leq k\}} \frac{r_{\ell_j}^0}{r_{\ell_j}^0} \right\}. \end{aligned}$$

■

Appendix 5: Markovian property of the system process

To show that the process $\{\mathbf{S}(t)\}_{t=0}^{\infty}$ is Markov, we need to prove that for any values that the system state can take $\mathbf{s}_0 = (\mathbf{x}_0, \mathbf{g}_0)$, ..., $\mathbf{s}_{t+1} = (\mathbf{x}_{t+1}, \mathbf{g}_{t+1})$ it is true that

$$\begin{aligned} \text{Prob}[\mathbf{S}(t+1) = \mathbf{s}_{t+1} | \mathbf{S}(t) = \mathbf{s}_t, \mathbf{S}(t-1) = \mathbf{s}_{t-1}, \dots, \mathbf{S}(0) = \mathbf{s}_0] \\ = \text{Prob}[\mathbf{S}(t+1) = \mathbf{s}_{t+1} | \mathbf{S}(t) = \mathbf{s}_t]. \end{aligned} \quad (46)$$

Then, we have the following:

$$\begin{aligned} \text{Prob}[\mathbf{X}(t+1) = \mathbf{x}_{t+1}, \mathbf{G}(t+1) = \mathbf{g}_{t+1} | \mathbf{X}(t) = \mathbf{x}_t, \mathbf{G}(t) = \mathbf{g}_t, \\ \mathbf{X}(t-1) = \mathbf{x}_{t-1}, \mathbf{G}(t-1) = \mathbf{g}_{t-1}, \dots, \mathbf{X}(0) = \mathbf{x}_0, \mathbf{G}(0) = \mathbf{g}_0] = \\ \text{Prob}[\mathbf{X}(t) = \mathbf{x}_t, \mathbf{G}(t) = \mathbf{g}_t | \mathbf{X}(t-1) = \mathbf{x}_{t-1}, \mathbf{G}(t-1) = \mathbf{g}_{t-1}, \dots, \mathbf{X}(0) = \mathbf{x}_0, \mathbf{G}(0) = \mathbf{g}_0] = \end{aligned} \quad (47)$$

$$\begin{aligned} \text{Prob}[\mathbf{x}_t = \mathbf{x}_t, \mathbf{g}_t = \mathbf{g}_t | \mathbf{x}_{t-1} = \mathbf{x}_{t-1}, \mathbf{g}_{t-1} = \mathbf{g}_{t-1}, \dots, \mathbf{x}_0 = \mathbf{x}_0, \mathbf{g}_0 = \mathbf{g}_0] = \\ \text{Prob}[\mathbf{x}_t = \mathbf{x}_t, \mathbf{g}_t = \mathbf{g}_t | \mathbf{x}_{t-1} = \mathbf{x}_{t-1}, \mathbf{g}_{t-1} = \mathbf{g}_{t-1}, \dots, \mathbf{x}_0 = \mathbf{x}_0, \mathbf{g}_0 = \mathbf{g}_0] = \\ \text{Prob}[\mathbf{X}(t+1) = \mathbf{x}_{t+1}, \mathbf{G}(t+1) = \mathbf{g}_{t+1} | \mathbf{X}(t) = \mathbf{x}_t, \mathbf{G}(t) = \mathbf{g}_t] = \end{aligned} \quad (48)$$

where (47) follows from (3) and (4) and where (48) follows from Assumption 1 on the Markovian property of the channel process. This completes the proof.

■

Acknowledgements

This work was supported by the Department of Defense under MURI grants W911NF-05-1-0246 and W911NF-08-1-0238 by the National Science Foundation under the grant CCF0728966 and by Renesas Mobile Corporation.

Author details

¹Renesas Mobile Corporation, Elekroniikkatie 13, 90590 Oulu, Finland
²Department of Electrical and Computer Engineering, Institute for Systems Research, University of Maryland, College Park, MD 20742, USA

Competing interests

The authors declare that they have no competing interests.

Received: 23 November 2010 Accepted: 30 September 2011

Published: 30 September 2011

References

1. Ephremides, TV Truong, Distributed algorithms for efficient and interference-free broadcasting in radio networks, in *Proceedings of IEEE Infocom* (1988)
2. Ephremides, TV Truong, Scheduling broadcasts in multihop radio networks. *IEEE Trans Commun.* **38**(4) (1990)
3. Hajek, G Sasaki, Link scheduling in polynomial time. *IEEE Trans Inf Theory* **34**(5) (1988)
4. SA Borbash, *Design Considerations in Wireless Sensor Networks* (PhD thesis, University of Maryland, 2004)
5. SA Borbash, A Ephremides, Wireless link scheduling with power control and SINR constraints. *IEEE Trans Inf Theory* **52**(11) (2006)
6. S Kompella, JE Wieselthier, A Ephremides, A cross-layer approach to optimal wireless link scheduling with SINR constraints, in *Proceedings of IEEE Military Communications Conference (MILCOM)* (2007)
7. S Kompella, JE Wieselthier, A Ephremides, Revisiting the optimal scheduling problem, in *Proceedings of Conference on Information Sciences and Systems (CISS)* (2008)
8. S Kompella, JE Wieselthier, A Ephremides, HD Sherali, A cross-layer approach to end-to-end routing and SINR-based scheduling in multi-hop wireless networks, in *Proceedings of 6th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt)* (2008)
9. S Kompella, JE Wieselthier, A Ephremides, HD Sherali, GD Nguyen, On optimal SINR-based scheduling in multihop wireless networks. *IEEE/ACM Transactions on Networking*, **18**(6) (2010)

10. A Pantelidou, A Ephremides, Minimum schedule lengths with rate control in single-hop wireless networks, in *Proceedings of IEEE Military Communications Conference (MILCOM)* (2008)
11. TH Cormen, CE Leiserson, RL Rivest, C. Stein, *Introduction to Algorithms* (The MIT Press, 2001)
12. D Bertsimas, JN Tsitsiklis, *Introduction to Linear Optimization* (Athena Scientific, 1997)
13. S Shakkottai, TS Rappaport, P Karlsson, Cross-layer design for wireless networks. *IEEE Commun Mag.* **41**(10) (2003)
14. V Srivastava, M Motani, Cross-layer design: a survey and the road ahead. *IEEE Commun Mag.* **43**(12) (2005)
15. M van Der Schaar, NS Shankar, Cross-layer wireless multimedia transmission: challenges, principles, and new paradigms. *IEEE Wirel Commun.* **12**(4) (2005)
16. X Lin, NB Shroff, R Srikant, A tutorial on cross-layer optimization in wireless networks. *IEEE J Sel Areas Commun.* **24**(8) (2006)
17. MS. Kuran, G Gür, T Tugcu, F Alagöz, Applications of the cross-layer paradigm for improving the performance of WiMAX. *IEEE Wirel Commun.* **17**(3) (2010)
18. A Pantelidou, A Ephremides, A cross-layer view of optimal scheduling. *IEEE Trans Inf Theory* **56**(11) (2010)
19. DP Bertsekas, in *Dynamic Programming and Optimal Control*, vol. 2, 3rd edn. (Athena Scientific, 2007)
20. A Pantelidou, A Ephremides, Minimum-length scheduling and rate control for time-varying wireless networks, in *Proceedings of IEEE Military Communications Conference (MILCOM)* (2009)
21. A Pantelidou, A Ephremides, Minimum-length scheduling for multicast traffic under channel uncertainty, in *Proceedings of IEEE Global Communications Conference (GLOBECOM)* (2009)
22. A Goldsmith, *Wireless Communications* (Cambridge University Press, 2005)
23. A Pantelidou, A Ephremides, Optimal rate control policies for proportional fairness in wireless networks, in *Proceedings of Conference on Information Sciences and Systems (CISS)* (2008)
24. A Pantelidou, A Ephremides, What is optimal scheduling in wireless networks?, in *Proceedings of 4th International Wireless Internet Conference (WICON)* (2008)

doi:10.1186/1687-1499-2011-115

Cite this article as: Pantelidou and Ephremides: Minimum-length scheduling with rate control in wireless networks: a shortest path approach. *EURASIP Journal on Wireless Communications and Networking* 2011 **2011**:115.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- ▶ Convenient online submission
- ▶ Rigorous peer review
- ▶ Immediate publication on acceptance
- ▶ Open access: articles freely available online
- ▶ High visibility within the field
- ▶ Retaining the copyright to your article

Submit your next manuscript at ▶ springeropen.com
