The Impact of Imperfect Scheduling on Cross-Layer Congestion Control in Wireless Networks

Xiaojun Lin, *Member, IEEE* and Ness B. Shroff *Senior Member, IEEE*

***Abstract*— In this paper, we study cross-layer design for congestion control in multihop wireless networks. In previous work, we have developed an optimal cross-layer congestion control scheme that jointly computes both the rate allocation and the stabilizing schedule that controls the resources at the underlying layers. However, the scheduling component in this optimal cross-layer congestion control scheme has to solve a complex global optimization problem at each time, and is hence too computationally expensive for online implementation. In this paper, we study how the performance of cross-layer congestion control will be impacted if the network can only use an imperfect (and potentially distributed) scheduling component that is easier to implement. We study both the case when the number of users in the system is fixed and the case with dynamic arrivals and departures of the users, and we establish performance bounds of cross-layer congestion control with imperfect scheduling. Compared with a layered approach that does not design congestion control and scheduling together, our cross-layer approach has provably better performance bounds, and substantially outperforms the layered approach. The insights drawn from our analyses also enable us to design a *fully distributed* cross-layer congestion control and scheduling algorithm for a restrictive interference model.**

***Index Terms*— Cross-layer design, congestion control, multihop wireless networks, stability, imperfect scheduling, mathematical programming/optimization, stochastic processes/queueing theory.**

I. INTRODUCTION

Cross-layer design is becoming increasingly important for improving the performance of multihop wireless networks (see, e.g., [1]–[14] and the reference therein). By simultaneously optimizing the control across multiple protocol layers, cross-layer design can substantially increase the network capacity, reduce interference and power consumption.

In this paper, we study issues involved in the cross-layer design of multihop wireless networks *that employ congestion control* [9]–[14]. Congestion control is a key functionality in modern communication networks. With proper congestion control, each user regulates the rate at which it injects data into the network, so that (a) the network capacity is fully utilized, (b) excessive congestion inside the network is avoided, and (c) some form of fairness (in terms of the amount of service that each user receives) is ensured. Although congestion control has been studied extensively for *wireline* networks (see [15], [16] for good references), these results cannot be applied directly to multihop wireless networks. In wireline networks, the *capacity region* (i.e., the set of feasible data rates) is of a simple form, i.e., the sum of the data rates at each link should be less than the link capacity, which is usually known and fixed. In multihop wireless networks, the capacity of each radio link depends on the signal and interference levels, and thus depends on the power and transmission schedule at other links. Hence, the capacity region is usually of a complex form that critically depends on the way in which resources at the underlying physical and MAC layers are scheduled. One possible way to address this difficulty is to choose a *rate region* within the capacity region, which has a simpler set of constraints similar to that of wireline networks, and compute the rate allocation within this simpler rate region [17]–[19]. This approach essentially attempts to make congestion control oblivious of the dynamics of the underlying layers. Hence, we will refer to this approach as the *layered approach* to congestion control. However, it requires prior knowledge of the capacity region in order to choose such a rate region. For many network settings, even such a rate region is difficult to find. Further, because the rate region reduces the set of feasible rates that congestion control can utilize, the layered approach results in a *conservative* rate allocation.

On the other hand, the *cross-layer approach* to congestion control can allocate data rates without requiring precise prior knowledge of the capacity region [9]–[14]. Here, by the "cross-layer" approach to congestion control, we mean that the network jointly optimizes both the data rates of the users and the resource allocation at the underlying layers, which include modulation, coding, power assignment and link schedules, etc. (For the rest of the paper, we will use the term *scheduling* to refer to the joint allocation of these resources at layers under congestion control.) In our previous work [9], we have presented an optimal cross-layer congestion control scheme and we have shown that our scheme can fully utilize the capacity of the network, maintain fairness, and improve the quality of service to the users.

However, the *scheduling* component in the optimal cross-layer congestion control scheme of [9] (and also in most of the other related results in the literature) requires solving at each iteration a global optimization problem that is usually quite difficult. In some cases, the optimization problem does not even have a polynomial-time solution. In this work, our objective is to develop a framework for cross-layer congestion

Manuscript received xxxxxxxxx xx, 200x.

This work has been partially supported by the NSF grant ANI-0099137 and the Indiana 21st Century Center for Wireless Communications and Networking. An earlier version of this paper has appeared in the Proceedings of IEEE INFOCOM 2005 [1].

The authors are with the Center for Wireless Systems and Applications (CWSA) and the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907, USA (email: {linx,shroff}@ecn.purdue.edu).

control that is suitable for online (and potentially distributed) implementation. The complexity of the scheduling component has become the main *obstacle* to developing such a solution.

To overcome this difficulty, in this paper we take a different approach. We accept the possibility that only suboptimal solutions to the scheduling problem may be computable, which we will refer to as *imperfect schedules*. We then study the impact of imperfect scheduling on the optimality of cross-layer congestion control. In this paper, we have studied this impact for a large class of imperfect scheduling policies, both for the case when the number of users in the system is fixed, and for the case when users dynamically arrive and leave the network. When the number of users in the system is fixed, we are able to obtain some desirable, but weak, results on the fairness and convergence properties of cross-layer congestion control with imperfect scheduling. Surprisingly, we are able to obtain far stronger results on the performance of the system when we consider dynamic arrivals and departures of the users. Our numerical results suggest that, in many network configurations, cross-layer congestion control with *imperfect* scheduling can perform comparably to that with *perfect* scheduling, while significantly reducing the computation overhead of the scheduling component. Further, we find that our cross-layer approach can substantially outperform the layered approach. Finally, the insights drawn from our analysis allow us to develop a *fully distributed* congestion control and scheduling scheme in a more restrictive network setting.

The rest of the paper is structured as follows. The system model is presented in Section II. We review results with perfect scheduling in Section III, and study the impact of imperfect scheduling in Sections IV and V. In Section VI, we present a fully distributed cross-layer congestion control algorithm. Simulation results are presented in Section VII, and the conclusion is given in Section VIII. Due to space constraints, some of the proofs are omitted and they are provided in our technical report that is available online [20].

II. The System Model

We consider a multihop wireless network with N nodes. Let $\mathcal{L}$ denote the set of node pairs (i, j) (i.e., links) such that direct transmission from node i to node j is allowed. The links are assumed to be directional. Due to the shared nature of the wireless media, the data rate r_{ij} of a link (i, j) depends not only on its own modulation/coding scheme and power assignment P_{ij}, but also on the interference due to the power assignments on other links. Let $\vec{P} = [P_{ij}, (i, j) \in \mathcal{L}]$ denote the vector of global power assignments and let $\vec{r} = [r_{ij}, (i, j) \in \mathcal{L}]$ denote the vector of data rates. We assume that $\vec{r} = u(\vec{P})$, i.e., the data rates are completely determined by the global power assignment[1]. The function $u(\cdot)$ is called the *rate-power function* of the system. Note that the global power assignment $\vec{P}$ and the rate-power function $u(\cdot)$ summarize the *cross-layer* control capability of the network at both the physical layer and the MAC layer. Precisely, the global power assignment determines the Signal-to-Interference-Ratio (SIR) at each link. Given the SIR, each link can choose appropriate modulation and coding schemes to achieve the data rate specified by $u(\vec{P})$. Finally, the network can schedule different sets of links to be active (and to use different power assignments) at different times to achieve maximum capacity [4], [5], [7]. There may be constraints on the feasible power assignment. For example, if each node has a total power constraint $P_{i,\max}$, then $\sum_{j:(i,j)\in\mathcal{L}} P_{ij} \le P_{i,\max}$. Let Π denote the set of feasible power assignments, and let $\mathcal{R} = \{u(\vec{P}), \vec{P} \in \Pi\}$. We assume that $\text{Co}(\mathcal{R})$, the convex hull of $\mathcal{R}$, is closed and bounded. We assume that time is divided into slots and the power assignment vector $\vec{P}(t)$ is fixed during each time slot t. We will refer to $\vec{r}(t) = u(\vec{P}(t))$ as the *schedule* at time slot t.

In the rest of the paper, it is usually more convenient to index the links numerically (e.g., links $1, 2, ..., L$) rather than as node-pairs (e.g., link (i, j)). The power assignment vector and the rate vector should then be written as $\vec{P} = [P_1, ..., P_L]$ and $\vec{r} = [r_1, ..., r_L]$, respectively.

There are S users and each user $s = 1, ..., S$ has one path through the network[2]. Let $H = [H_s^l]$ denote the routing matrix, i.e., $H_s^l = 1$, if the path of user s uses link l, and $H_s^l = 0$, otherwise. Let x_s be the rate with which user s injects data into the network. Each user is associated with a utility function $U_s(x_s)$, which reflects the level of "satisfaction" of user s when its data rate is x_s. As is typically assumed in the congestion control literature, we assume that each user s has a maximum data rate M_s and the utility function $U_s(\cdot)$ is strictly concave, non-decreasing and twice continuously differentiable on $(0, M_s]$.

III. Cross-Layer Congestion Control with Perfect Scheduling

In this section, we review the optimal cross-layer congestion control scheme that we presented in [9]. We first define the *capacity region* of the system. We say that a system is *stable* if the queue lengths at all links remain finite. We say that a user rate vector $\vec{x} = [x_1, ..., x_s]$ is *feasible* if there exists a scheduling policy that can *stabilize* the system under user rates $\vec{x}$. We define the *capacity region* to be the set of *feasible* rates $\vec{x}$. It has been shown in [4], [5], [7] that the *optimal capacity region* Λ is a convex set and is given by $\Lambda = \left\{ \vec{x} \,\middle|\, [\sum_{s=1}^{S} H_s^l x_s] \in \text{Co}(\mathcal{R}) \right\}$. The convex hull operator $\text{Co}(\cdot)$ is due to a standard time-averaging argument [4], [5], [7]. Λ is optimal in the sense that no vector $\vec{x}$ outside Λ is feasible for any scheduling policy.

In [9], we have formulated and solved the following optimal cross-layer congestion control problem.

The Cross-Layer Congestion Control Problem:

- Find the user rate vector $\vec{x}$ in Λ that maximizes the total

[1] Here we have assumed that the channel condition is time-invariant. The extension to the case with channel variations is treated in Section V-B.

[2] Extensions to the case with multipath routing are also possible (see [9]).

system utility, i.e.,

$$\max_{0\le x_s\le M_s} \quad \sum_{s=1}^{S} U_s(x_s) \tag{1}$$

$$\text{subject to} \quad \sum_{s=1}^{S} H_s^l x_s \le r_l \text{ for all } l \in \mathcal{L} \tag{2}$$

$$\text{and} \quad [r_l] \in \text{Co}(\mathcal{R}).$$

- Find the associated scheduling policy that *stabilizes* the system.

Remark: Note that this problem is indeed a cross-layer optimization problem. The first element of the problem determines the rates with which users inject data into the network (i.e., the congestion control problem at the transport layer). The second element determines when and at what rate each link in the network should transmit (i.e., the scheduling problem at the MAC/physical layer). Maximizing the total system utility as in (1) has been shown to be equivalent to some *fairness* objectives when the utility functions are appropriately chosen [21]. For example, utility functions of the form

$$U_s(x_s) = w_s \log x_s \tag{3}$$

correspond to *weighted proportional fairness*, where $w_s, s = 1, ..., S$ are the weights. A more general form of utility function is

$$U_s(x_s) = w_s \frac{x_s^{1-\beta}}{1-\beta}, \beta > 0. \tag{4}$$

Maximizing the total utility will correspond to *maximizing weighted throughput* as $\beta \to 0$, *weighted proportional fairness* as $\beta \to 1$, *minimizing weighted potential delay* as $\beta \to 2$, and *max-min fainess* as $\beta \to \infty$ [22].

The solution to the optimal cross-layer congestion control problem is of the following form [9]. We associate an implicit cost q^l for each link l. Let $\vec{q} = [q^1, ..., q^L]$.

The Optimal Cross-Layer Congestion Control Algorithm:

At each iteration t:

- The data rates of the users are determined by

$$x_s(t) = \operatorname*{argmax}_{0\le x_s\le M_s} \left[U_s(x_s) - \sum_{l=1}^{L} H_s^l q^l(t) x_s \right]. \tag{5}$$

- The schedule is determined by

$$\vec{r}(t) = \operatorname*{argmax}_{\vec{r}\in\mathcal{R}} \sum_{l=1}^{L} q^l(t) r_l = \operatorname*{argmax}_{\vec{r}=u(\vec{P}),\vec{P}\in\Pi} \sum_{l=1}^{L} q^l(t) r_l. \tag{6}$$

- The implicit costs are updated by

$$q^l(t+1) = \left[q^l(t) + \alpha_l \left(\sum_{s=1}^{S} H_s^l x_s(t) - r_l(t) \right) \right]^+ . \tag{7}$$

Remark: This solution has an attractive *decomposition* property: the congestion control component and the scheduling component are decomposed by the implicit costs $\vec{q}$. Given $\vec{q}$, each user can determine its own data rate independently according to (5). The scheduling component (6) also uses $\vec{q}$ to compute the schedule independently of the data rates of the users.

The implicit cost updates in (7) can be viewed as subgradient descent iterations for the dual of problem (1). To be precise, the implicit costs q^l are the Lagrange multipliers for the constraint (2). We can construct the Lagrangian as

$$L(\vec{x}, \vec{r}, \vec{q}) = \sum_{s=1}^{S} U_s(x_s) - \sum_{l=1}^{L} q^l \left[\sum_{s=1}^{S} H_s^l x_s - r_l \right].$$

The dual of problem (1) is then

$$\min_{\vec{q}\ge 0} D(\vec{q}), \tag{8}$$

where

$$\begin{aligned} D(\vec{q}) &= \max_{0\le x_s\le M_s, s=1,..,S, \vec{r}\in \text{Co}(\mathcal{R})} L(\vec{x}, \vec{r}, \vec{q}) \\ &= \sum_{s=1}^{S} B_s(\vec{q}) + V(\vec{q}), \\ B_s(\vec{q}) &= \max_{0\le x_s\le M_s} \left[U_s(x_s) - \sum_{l=1}^{L} H_s^l q^l x_s \right], \quad (9) \\ V(\vec{q}) &= \max_{\vec{r}\in\mathcal{R}} \sum_{l=1}^{L} q^l r_l. \quad (10) \end{aligned}$$

The following proposition from [9] shows that, when the stepsizes α_l are small, the user rates $\vec{x}(t)$ will converge within a small neighborhood[3] of the optimal rate allocation $\vec{x}^*$.

Proposition 1: a) There is no duality gap, i.e., the minimal value of (8) coincides with the optimal value of (1).

b) Let Φ be the set of $\vec{q}$ that minimizes $D(\vec{q})$. For any $\vec{q} \in \Phi$, let $\vec{x}$ solve (5), then $\vec{x}$ is the unique optimal solution $\vec{x}^*$ of (1).

c) Assume that $\alpha_l = h\alpha_l^0$. Let $||\vec{q}||_A = \sum_{l=1}^{L} \frac{(q^l)^2}{\alpha_l^0}$ and $d(\vec{q}, \Phi) = \min_{\vec{p}\in\Phi} \sqrt{||\vec{q} - \vec{p}||_A}$. For any $\epsilon > 0$, there exists some $h_0 > 0$ such that, for any $h \le h_0$ and any initial implicit costs $\vec{q}(0)$, there exists a time T_0 such that for all $t \ge T_0$,

$$d(\vec{q}(t), \Phi) < \epsilon \text{ and } ||\vec{x}(t) - \vec{x}^*|| < \epsilon.$$

The proofs of part (a) and part (b) are quite standard (see, for example, Theorem 3.2.8 in [23, p44]). Part (c) is a consequence of Theorem 2.3 in [24, p26]. The details of the proof are provided in [20]. We will also draw the connection with Proposition 5 later in Section IV.

The optimal cross-layer congestion control algorithm (5)-(7) not only computes the optimal rate allocation, but also generates the stabilizing scheduling policy by solving (6) at each time slot t. In fact, let Q^l denote the queue size at link l. Then Q^l evolves approximately as[4]:

$$Q^l(t+1) \approx \left[Q^l(t) + \left(\sum_{s=1}^{S} H_s^l x_s(t) - r_l(t) \right) \right]^+ . \tag{11}$$

[3] If instead the stepsizes are time-varying and they are chosen such that $\alpha_l(t) = h_t \alpha_l^0$, $h_t \to 0$ as $t \to \infty$ and $\sum_{t=1}^{+\infty} h_t = +\infty$, then $d(\vec{q}(t), \Phi) \to 0$ and $\vec{x}(t) \to \vec{x}^*$ as $t \to \infty$.

[4] Note, (11) is an approximation because not all links are active at the same time. Hence, data injected to the network by each user at time t may take several time slots to reach downstream links.

Comparing (11) with (7), we can see that $Q^l(t) \approx q^l(t)/\alpha_l$. From here we can infer that $Q^l(t)$ is bounded, as formalized in the following proposition.

Proposition 2: If the stepsizes α_l are sufficiently small, then using the schedules determined by solving (6) at each time slot, along with an appropriate packet scheduling policy at each link, we have,

$$\sup_t Q^l(t) < +\infty \text{ for all } l \in \mathcal{L}.$$

We give the proof and the details of the packet scheduling policy in [20]. From Propositions 1 and 2, we observe an interesting tradeoff between optimality and the queue-length. As we choose smaller stepsizes α_l, we can obtain user rate allocation $\vec{x}$ as close to $\vec{x}^*$ as we want. However, the actual queue length $Q^l \approx q^l/\alpha_l$ also increases. It is possible to address this tradeoff by using "virtual queue" algorithms, and significantly reduce the actual queue length with only a minimum amount of reduction on the system throughput. For the detail, please refer to [9].

IV. The Impact of Imperfect Scheduling on Cross-Layer Congestion Control: The Static Case

In this paper, we are interested in developing cross-layer congestion control solutions that are suitable for online implementation. The main difficulty in implementing the optimal solution of Section III is the complexity of the scheduling component. Depending on the rate-power function $u(\cdot)$, the scheduling problem (6) is usually a difficult global optimization problem. In some cases, this optimization problem does not even have a polynomial-time solution. Hence, solving (6) exactly at every time slot is too time-consuming.

As discussed in the Introduction, in this paper, we take a different approach from that of finding *optimal* rate allocations. We will study systems that can only compute *imperfect schedules*, which are suboptimal solutions to the scheduling problem (6). We will investigate how imperfect scheduling impacts the optimality of cross-layer congestion control. Our objective is to find some imperfect scheduling policies that are easy to implement and that, when properly designed with congestion control, result in good overall performance.

We will particularly be interested in the following class of imperfect scheduling policies:

Imperfect Scheduling Policy S_γ:

Fix $\gamma \in (0,1]$. At each time slot t, compute a schedule $\vec{r}(t) \in \mathcal{R}$ that satisfies:

$$\sum_{l=1}^{L} r_l(t) q^l(t) \geq \gamma \max_{\vec{r} \in \mathcal{R}} \sum_{l=1}^{L} r_l q^l(t). \tag{12}$$

The parameter γ in (12) can be viewed as a tuning parameter indicating the degree of precision of the imperfect schedule. The complexity of finding a schedule $\vec{r}(t)$ satisfying (12) usually decreases as γ is reduced. The following proposition shows that an imperfect scheduling policy S_γ at most reduces the capacity region by a factor of γ. The proof is a straightforward extension of similar results in the switching literature (see [25]) and is also provided in [20].

Proposition 3: Fix $\gamma \in (0,1]$. If the user rates $\vec{x}$ lie strictly inside $\gamma\Lambda$ (i.e., $\vec{x}$ lies in the interior of $\gamma\Lambda$), then any imperfect scheduling policy S_γ can stabilize the system.

With an imperfect scheduling policy S_γ, the dynamics of cross-layer congestion control are summarized by the following set of equations:

$$x_s(t) = \operatorname*{argmax}_{0 \leq x_s \leq M_s} \left[U_s(x_s) - \sum_{l=1}^{L} H_s^l q^l(t) x_s \right], \tag{13}$$

$$\sum_{l=1}^{L} r_l(t) q^l(t) \geq \gamma \max_{\vec{r} \in \mathcal{R}} \sum_{l=1}^{L} r_l q^l(t), \quad \vec{r}(t) \in \text{Co}(\mathcal{R}), \tag{14}$$

$$q^l(t+1) = \left[q^l(t) + \alpha_l \left(\sum_{s=1}^{S} H_s^l x_s(t) - r_l(t) \right) \right]^+. \tag{15}$$

Note that when $\gamma = 1$, the dynamics (13)-(15) reduce to the case with perfect scheduling (as in Section III). Let $\vec{x}^{*,0}$ denote the optimal solution to the original cross-layer congestion control problem (1). The solution to the following problem turns out to be a good reference point for studying the dynamics (13)-(15) when $\gamma < 1$:

The γ-Reduced Problem:

$$\max_{0 \leq x_s \leq M_s} \sum_{s=1}^{S} U_s(x_s) \quad \text{subject to} \quad \vec{x} \in \gamma\Lambda. \tag{16}$$

Let $\vec{x}^{*,\gamma}$ denote the solution to the γ-reduced problem. (Note that $\vec{x}^{*,\gamma}$ exists and is unique for any given $\gamma > 0$.) Motivated by Proposition 3, we would expect that the rate allocation computed by the dynamics (13)-(15) will be "no worse than" $\vec{x}^{*,\gamma}$. However, this assertion is not quite true. We find that the interaction between cross-layer congestion control and imperfect scheduling is much more complicated. As the data rates of the users are reacting to the same implicit costs as the scheduling component is, there is a possibility that the system gets stuck into local sub-optimal areas. We can construct examples where, for a subset of users, their data rates determined by the dynamics (13)-(15) can be much smaller than the corresponding rate allocation computed by the γ-reduced problem. In fact, it is even possible to construct examples where certain components of $\vec{x}^{*,0}$ (which is the optimal rate allocation that the dynamics (13)-(15) converge to when $\gamma = 1$) are smaller than the corresponding components of $\vec{x}^{*,\gamma}$. (We provide detailed examples in our technical report [20].) Nonetheless, we are able to show the following weak but desirable results on the fairness and convergence properties of cross-layer congestion control with imperfect scheduling.

Proposition 4: Assume that the utility function is logarithmic (i.e., of the form in (3)). If the dynamics (13)-(15) converge, i.e., $\vec{x}(t) \to \vec{x}^{*,I}$ and $\vec{q}(t) \to \vec{q}_I^*$ as $t \to \infty$, then

$$\vec{x}^{*,I} \in \Lambda \text{ and } \sum_{s=1}^{S} \frac{w_s x_s^{*,\gamma}}{x_s^{*,I}} \leq \sum_{s=1}^{S} w_s. \tag{17}$$

The proof is available in Appendix A. Proposition 4 can be generalized to other forms of utility functions (as in (4)). This result can be viewed as a *weak fairness property* of the likely rate allocation under imperfect scheduling. It shows that, if the

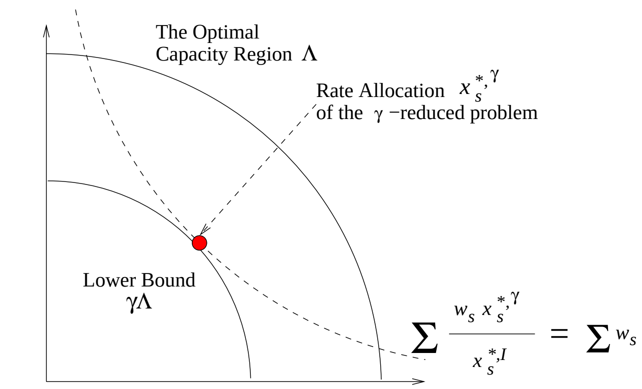


Fig. 1. The weak fairness property

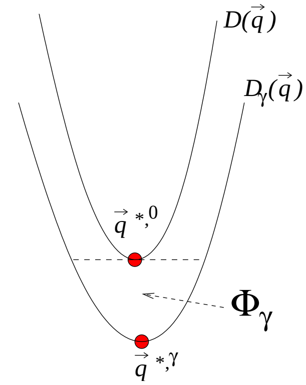


Fig. 2. The set Φ_γ

dynamics (13)-(15) converge, the rate allocation of the users will lie in a strip defined by (17) (see Fig. 1). Hence, the rate of each user is unlikely to be arbitrarily different from $\vec{x}^{\,*,\gamma}$ (thus the level of unfairness is bounded). In particular, if $w_s = 1$ for all s, then by (17), $x_s^{*,I}$ will be no smaller than $x_s^{*,\gamma}/S$.

We next study the question of convergence. Note that for any given $\gamma < 1$, there are many schedules that can satisfy the definition of S_γ policies. Hence, it is impossible for us to establish the convergence of the dynamics (13)-(15) to a particular point. Nonetheless, it is possible to find a "region of attraction." Using a duality approach analogous to that in Section III, we can define the dual of the γ-reduced problem as

$$D_\gamma(\vec{q}) \;=\; \sum_{s=1}^{S} B_s(\vec{q}) + \gamma V(\vec{q}),$$

where $B_s(\vec{q})$ and $V(\vec{q})$ are still defined as in (9) and (10), respectively. Note that both $D(\vec{q})$ and $D_\gamma(\vec{q})$ are convex functions and $D(\vec{q}) \geq D_\gamma(\vec{q})$. Let $\vec{q}^{\,*,0}$ denote a minimizer of $D(\vec{q})$ and $\vec{q}^{\,*,\gamma}$ denote a minimizer of $D_\gamma(\vec{q})$. (Note that $\vec{q}^{\,*,0}$ corresponds to $\gamma = 1$.) The following proposition shows that

$$\Phi_\gamma \triangleq \{\vec{q} : D_\gamma(\vec{q}) \leq D(\vec{q}^{\,*,0})\},$$

is a region of attraction for the dynamics (13)-(15).

Proposition 5: Assume that $\alpha_l = h\alpha_l^0$. Let $||\vec{q}||_A = \sum_{l=1}^{L} \frac{(q^l)^2}{\alpha_l^0}$ and $d(\vec{q}, \Phi) = \min_{\vec{p}\in\Phi} \sqrt{||\vec{q}-\vec{p}||_A}$. For any $\epsilon > 0$, there exists some $h_0 > 0$ such that, for any $h \leq h_0$ and any initial implicit costs $\vec{q}(0)$, there exists a time T_0 such that for all $t \geq T_0$,

$$d(\vec{q}(t), \Phi) < \max_{\vec{p}\in\Phi_\gamma} d(\vec{p}, \Phi) + \epsilon. \quad (18)$$

Remark: If $\gamma = 1$, we have $\Phi_\gamma = \Phi$ and the right hand side of (18) is equal to ϵ. We thus obtain part (c) of Proposition 1.

The proof of Proposition 5 is provided in Appendix B. Proposition 5 shows that, if the stepsizes α_l are sufficiently small, the dynamics (13)-(15) will eventually enter a neighborhood of the set Φ_γ. Note that both $\vec{q}^{\,*,0}$ and $\vec{q}^{\,*,\gamma}$ belong to the set Φ_γ (see Fig. 2). Hence, in a weak sense, the dynamics of the system are moving in the right direction. However, in general the set Φ_γ is quite large and does not provide much further insights on the eventual rate allocation. In fact, we can construct examples (see [20] for the details) where the dynamics (13)-(15) may converge to any point that satisfies (17), or may form loops and never converge at all.

To conclude this section, we have studied the impact of imperfect scheduling on the dynamics of cross-layer congestion control *when the number of users in the system is fixed*. We are able to show certain desirable, but weak, results on the fairness and convergence properties of the system. In the next section, we will turn to the case when users dynamically arrive and depart the network, and surprisingly, we will be able to show far stronger results on the performance of the system there.

V. Stability Region of Cross-Layer Congestion Control

In this section, we turn to the case when the number of users in the system is itself a stochastic process. The motivation for studying the dynamic case stems from the fact that, under the static setting in Section IV, we have been unable to obtain satisfactory results on the performance of cross-layer congestion control with imperfect scheduling. In fact, it appears that no such results are possible since we can construct examples where convergence does not hold or where the rate allocation of some users can be quite unfair. Thus, we are left with little knowledge about how well (or badly) a system will behave even with a moderate level of imprecision in the scheduling policy used. However, despite the existence of these negative examples, we feel that they may not be as pessimistic as they appear. In real networks, the user population keeps changing. It is unclear yet whether these unfavorable configurations are the ones that persist and dominate, or whether they will self-correct as the users arrive and leave the system.

These questions lead us to study the dynamic setting. Under the dynamic setting, we are no longer interested in the rate-allocation at each snapshot in time (since we know that there are always time instants when the rate-allocations are unsatisfactory). Instead, we are interested in performance measures that are more *global*, i.e., that summarize the system behavior over an interval of time. In this paper, we will use the *stability region* to quantify how imperfect scheduling impacts cross-layer congestion control under the dynamic setting. Here, by *stability*, we mean that the number of users in the system and the queue lengths at all links in the network remain finite. The *stability region* of the system is the set of offered loads under which the system is stable. We choose to study the stability region because past results have indicated that both fairness and convergence of congestion control are closely connected with stability. Previous works for wireline networks have shown that, by allocating data rates to the users *fairly*, i.e., by choosing data rates that maximize the total system utility in the forms of (3) or (4), the *largest possible* stability region can be achieved [21], [26]–[28]. Conversely, examples have been given in [21] that, if the rate allocation at each snapshot is unfair, the stability region may be much less than the optimal. These results are important as they tell us that fairness is not just a *static* and *aesthetic* property, but it actually has a strong *global performance* implication, i.e., in achieving the *largest possible* stability region. In this section, we will establish similar but stronger results for our cross-layer congestion control scheme with imperfect scheduling.

To be precise, instead of using the notation s for user s, we now use s to denote a class of users with the same utility function and the same path. We assume that users of class s arrive according to a Poisson process with rate λ_s and that each user brings with it a file for transfer whose size is exponentially distributed with mean $1/\mu_s$. The load brought by users of class s is then $\rho_s = \lambda_s/\mu_s$. Let $\vec{\rho} = [\rho_1, ..., \rho_S]$. Let $n_s(t)$ denote the number of users of class s that are in the system at time t, and let $\vec{n}(t) = [n_1(t), ..., n_S(t)]$. We assume that the rate allocations for users of the same class are identical. Let $x_s(t)$ denote the rate of each user of class s at time t. In the rate assignment model that follows, the evolution of $\vec{n}(t)$ will be governed by a Markov process. Its transition rates are given by:

$$\begin{aligned} n_s(t) &\to n_s(t)+1, && \text{with rate } \lambda_s, \\ n_s(t) &\to n_s(t)-1, && \text{with rate } \mu_s x_s(t) n_s(t) \\ & && \quad \text{if } n_s(t) > 0. \end{aligned}$$

As in [29], we say that the above system is *stable* if

$$\limsup_{t\to\infty} \frac{1}{t}\int_0^t \mathbf{1}_{\{\sum_{s=1}^{S} n_s(t) + \sum_{l=1}^{L} q^l(t) > M\}} dt \to 0,$$

as $M \to \infty$. This means that the fraction of time that the amount of "unfinished work" in the system exceeds a certain level M can be made arbitrarily small as $M \to \infty$. The *stability region* Θ of the system under a given congestion control and scheduling policy is the set of offered loads $\vec{\rho}$ such that the system is stable.

We next describe the rate assignment and implicit cost update policy. We assume that time is divided into slots of length T, and the schedules and implicit costs are only updated at the end of each time slot. However, users may arrive and depart in the middle of a time slot. Let $\vec{q}(kT)$ denote the implicit cost at time slot k. The data rates of the users are determined by the current implicit costs as in (5). For simplicity, we assume that the utility function is logarithmic (the result can be readily generalized to utility functions of other forms in (4)). Further, let M_s denote the maximum data rate for users of class s. The rate of each user of class s is then given by

$$x_s(t) = x_s(kT) = \min\left\{\frac{w_s}{\sum_{l=1}^{L} H_s^l q^l(kT)}, M_s\right\} \qquad (19)$$

for $kT \le t < (k+1)T$. The schedule $\vec{r}(kT)$ at time slot k is computed according to an imperfect scheduling policy S_γ based on the current implicit cost $\vec{q}(kT)$. Finally, at the end of each time slot, the implicit costs are updated as

$$\begin{aligned} q^l((k+1)T) = \Big[&q^l(kT) \\ &+ \alpha_l\left(\sum_{s=1}^{S} H_s^l \int_{kT}^{(k+1)T} n_s(t) x_s(kT) dt - r_l(kT)T\right)\Big]^+. \end{aligned}$$

The following proposition shows that, using the above cross-layer congestion control algorithm with imperfect scheduling policy S_γ, the stability region of the system is no smaller than $\gamma\Lambda$.

Proposition 6: If

$$\max_{l\in\mathcal{L}} \alpha_l \le \frac{1}{T\bar{S}\bar{L}} \min_s \frac{w_s}{4\rho_s M_s}, \qquad (20)$$

where $\bar{S} = \max_{l\in\mathcal{L}} \sum_{s=1}^{S} H_s^l$ is the maximum number of classes using any link, and $\bar{L} = \max_s \sum_{l=1}^{L} H_s^l$ is the maximum number of links used by any class, then for any offered load $\vec{\rho}$ that resides strictly inside $\gamma\Lambda$, the system described by the Markov process $[\vec{n}(kT), \vec{q}(kT)]$ is stable.

Several remarks are in order: Firstly, Proposition 6 shows that, when imperfect schedules are used, the stability region of the system employing cross-layer congestion control is no worse than the capacity region shown in Proposition 3 (and used by the γ-reduced problem). This result is interesting (and somewhat surprising) given the fact that, when the number of users in the system is fixed, the dynamics of cross-layer congestion control with imperfect scheduling can form loops or get stuck into local sub-optimal regions. *Nonetheless, Proposition 6 shows that, as far as the overall stability region of the system is concerned, these potential local sub-optimums are inconsequential when the arrivals and departures of the users are taken into account.*

Secondly, we do not need the rates of any users to converge. Previous results on the stability region of congestion control typically adopt a *time-scale separation assumption* [21], [26]–[28], which assumes that the rate allocation $\vec{x}(t)$ *perfectly solves (1) at each time instant* t. Such an approach is of little value for the model in this paper because the dynamics (13)-(15) with imperfect scheduling do not even converge in the first place! Further, the time-scale separation assumption is rarely realistic in practice: as the number of users in the system is constantly changing, the rate allocation may never have the time to converge. In Proposition 6, we establish the stability region of the system without requiring such a time-scale separation assumption. This result is of independent value. For the special case when $\gamma = 1$, it can be viewed as a stronger version of previous results in the literature (including those for wireline networks, e.g., Theorem 1 in [21]).

Finally, a simple stepsize rule is provided in (20). Note that when the number of users in the system is fixed, we typically require the stepsizes to be driven to zero for convergence to occur (see Proposition 1). However, in (20) the stepsizes can be chosen bounded away from zero. In fact, as the set $\gamma\Lambda$ is bounded, the stepsizes can be chosen independently of the offered load. The simplicity in the stepsize rule is another benefit we obtain by studying the dynamic arrivals and departures of the users.

A. The Main Idea of the Proof of Proposition 6

We now sketch the main idea of the proof for Proposition 6 so that the reader can gain some insight on the dynamics of the system. Define the following Lyapunov function,

$$\mathcal{V}(\vec{n},\vec{q}) = V_n(\vec{n}) + V_q(\vec{q}),$$

where $V_n(\vec{n}) = \sum_{s=1}^{S} \frac{w_s n_s^2}{2\lambda_s}$, and $V_q(\vec{q}) = \sum_{l=1}^{L} \frac{(q^l)^2}{2\alpha_l}$. We shall show below that $\mathcal{V}(\vec{n},\vec{q})$ has a negative drift. As a crude first-order approximation, assume that users arrive and depart only at the end of each time slot. Thus, $n_s(t) = n_s(kT)$ during the k-th time slot. We can show that (see [20] for the details),

$$\begin{aligned} &\mathbf{E}[V_n(\vec{n}((k+1)T)) - V_n(\vec{n}(kT))|\vec{n}(kT),\vec{q}(kT)] \\ \leq\ & T\sum_{s=1}^{S}\left[\frac{w_s}{x_s(kT)}\right][\rho_s - n_s(kT)x_s(kT)] + E_1(k), \end{aligned}$$

where $E_1(k)$ is an error term that is roughly on the order of $|\rho_s - n_s(kT)x_s(kT)|$. Since the rate allocation is determined by (19), we have (ignoring the maximum data rate M_s),

$$\begin{aligned} &\mathbf{E}[V_n(\vec{n}((k+1)T)) - V_n(\vec{n}(kT))|\vec{n}(kT),\vec{q}(kT)] \\ \leq\ & T\sum_{s=1}^{S}\left[\sum_{l=1}^{L} H_s^l q^l(kT)\right][\rho_s - n_s(kT)x_s(kT)] \\ &+E_1(k). \end{aligned} \tag{21}$$

We can also show that

$$\begin{aligned} &\mathbf{E}[V_q(\vec{q}((k+1)T) - V_q(\vec{q}(kT))|\vec{n}(kT),\vec{q}(kT)] \\ \leq\ & T\sum_{l=1}^{L} q^l(kT)\left[\sum_{s=1}^{S} H_s^l n_s(kT)x_s(kT) - r_l(kT)\right] \\ &+E_2(k), \end{aligned} \tag{22}$$

where $E_2(k)$ is an error term that is roughly on the order of $\left[\sum_{s=1}^{S} H_s^l n_s(kT)x_s(kT) - r_l(kT)\right]^2$. Hence, by adding (21) and (22), and by changing the order of the summation, we have

$$\begin{aligned} &\mathbf{E}[\mathcal{V}(\vec{n}((k+1)T),\vec{q}((k+1)T)) \\ &\quad -\mathcal{V}(\vec{n}(kT),\vec{q}(kT))|\vec{n}(kT),\vec{q}(kT)] \\ \leq\ & T\sum_{l=1}^{L} q^l(kT)\left[\sum_{s=1}^{S} H_s^l \rho_s - r_l(kT)\right] \\ &+E_1(k) + E_2(k). \end{aligned} \tag{23}$$

By assumption, $\vec{\rho}$ lies strictly inside $\gamma\Lambda$. Hence, there exists some $\epsilon > 0$ such that

$$[(1+\epsilon)\sum_{s=1}^{S} H_s^l \rho_s] \in \gamma \text{Co}(\mathcal{R}).$$

By the definition of the imperfect scheduling policy S_γ,

$$\sum_{l=1}^{L} q^l(kT)r_l(kT) \geq (1+\epsilon)\sum_{l=1}^{L} q^l(kT)\sum_{s=1}^{S} H_s^l \rho_s.$$

Substituting into (23), we have,

$$\begin{aligned} &\mathbf{E}[\mathcal{V}(\vec{n}((k+1)T),\vec{q}((k+1)T)) \\ &\quad -\mathcal{V}(\vec{n}(kT),\vec{q}(kT))|\vec{n}(kT),\vec{q}(kT)] \\ \leq\ & -T\epsilon\sum_{l=1}^{L} q^l(kT)\sum_{s=1}^{S} H_s^l \rho_s + E_1(k) + E_2(k). \end{aligned} \tag{24}$$

This shows that $\mathcal{V}(\cdot,\cdot)$ would drift towards zero when $||\vec{q}(kT)||$ is large and when the error terms $E_1(k)$ and $E_2(k)$ are bounded. We would then apply Theorem 2 of [29] to establish the stability of the system. The detailed proof is given in [20].

Remark: We may relax the definition of the imperfect scheduling policy S_γ from (12) to:

$$\sum_{l=1}^{L} r_l(t)q^l(t) \geq \gamma \max_{\vec{r}\in\mathcal{R}} \sum_{l=1}^{L} r_l q^l(t) - M_\gamma \tag{25}$$

for some positive constant M_γ. It is easy to see that the above argument (and consequently Proposition 6) will hold even with this new definition of the S_γ policy.

B. The Case with Channel Variations

Proposition 6 can be generalized to the case with channel variations (e.g., due to fading and/or mobility of the nodes). Assume that time is again divided into slots of length T. Further, we assume that the channel condition is fixed within each time-slot, and is chosen independently and identically among a set $\mathcal{K}$ of states. Let $\kappa(t) \in \mathcal{K}$ denote the channel condition at time-slot t, and let $\pi_\kappa, \kappa \in \mathcal{K}$ denote the probability that the channel condition at a particular time-slot is κ. The rate-power function now also depends on the channel condition, i.e., $\vec{r} = u(\vec{P},\kappa)$. Let $\mathcal{R}_\kappa = \{u(\vec{P},\kappa), \vec{P} \in \Pi\}$. We assume again that $\text{Co}(\mathcal{R}_\kappa)$ is closed and bounded for each $\kappa \in \mathcal{K}$.

Our cross-layer congestion control algorithm can be virtually unchanged with this new channel model. Both the congestion control component ((5) or (13)) and the implicit cost update ((7) or (15)) can remain the same. The only component that needs to be changed is the scheduling component ((6) or (14)). The scheduling policy S_γ should now compute an (imperfect) schedule based on the current channel condition κ:

$$\sum_{l=1}^{L} r_l(t)q^l(t) \geq \gamma \max_{\vec{r}\in\mathcal{R}_\kappa} \sum_{l=1}^{L} r_l q^l(t). \tag{26}$$

(The optimal schedule corresponds to $\gamma = 1$.) Note that no prior knowledge of the stationary distribution π of the channel condition is needed in our cross-layer congestion control algorithm.

We now illustrate how the sketch of the proof in Section V-A can be adapted to the case with channel variations. It has been shown in [4], [5], [7] that, with the above channel model, the capacity region Λ is given by:

$$\Lambda = \left\{ \vec{x} \left| [\sum_{s=1}^{S} H_s^l x_s] \in \sum_{\kappa \in \mathcal{K}} \pi_\kappa \mathrm{Co}(\mathcal{R}_\kappa) \right. \right\}. \tag{27}$$

Following the argument in Section V-A, we can show that,

$$\begin{aligned} &\mathbf{E}[\mathcal{V}(\vec{n}((k+1)T), \vec{q}((k+1)T)) \\ &\quad -\mathcal{V}(\vec{n}(kT), \vec{q}(kT)) | \vec{n}(kT), \vec{q}(kT)] \\ \leq\ & T \sum_{l=1}^{L} q^l(kT) \left[\sum_{s=1}^{S} H_s^l \rho_s - \mathbf{E}_\pi [r_l(kT)] \right] \\ &\quad + E_1(k) + E_2(k), \end{aligned} \tag{28}$$

where the expectation $\mathbf{E}_\pi[\cdot]$ is taken with respect to the stationary distribution π of the channel condition κ. (This inequality corresponds to Inequality (23) in Section V-A.) Now, if $\vec{\rho}$ lies strictly inside $\gamma\Lambda$, where Λ is given by (27), then there exists some $\epsilon > 0$ such that

$$[(1+\epsilon) \sum_{s=1}^{S} H_s^l \rho_s] \in \gamma \sum_{\kappa \in \mathcal{K}} \pi_\kappa \mathrm{Co}(\mathcal{R}_\kappa).$$

Using (26), we then have,

$$\sum_{l=1}^{L} q^l(kT) \mathbf{E}_\pi [r_l(kT)] \geq (1+\epsilon) \sum_{l=1}^{L} q^l(kT) \sum_{s=1}^{S} H_s^l \rho_s.$$

Substituting into (28), we have (24) again. We have thus shown that the stability region of the system employing cross-layer congestion control and imperfect scheduling policy S_γ is no worse than $\gamma\Lambda$ even when there are channel variations.

We now give two examples showing how efficient cross-layer congestion control schemes can be constructed by applying Proposition 6 to different network settings.

C. The Node Exclusive Interference Model

Proposition 6 is most useful when an imperfect schedule that satisfies (12) can be easily computed for some reasonable value of γ. This is the case under the following node exclusive interference model.

The Node Exclusive Interference Model:

- The data rate of each link is fixed at c_l.
- Each node can only send to or receive from one other node at any time.

This interference model has been used in earlier studies of congestion control in multihop wireless networks [17], [18]. Under this model, the *perfect* schedule (according to (6)) at each time slot corresponds to the **Maximum Weighted Matching (MWM)**, where the weight of each link is $q^l c_l$. (A *matching* is a subset of the links such that no two links share the same node. The *weight* of a matching is the total weight over all links belonging to the matching. A *maximum-weighted-matching (MWM)* is the matching with the maximum weight.) An $O(N^3)$-complexity algorithm for MWM can be found in [30], where N is the number of nodes. On the other hand, the following much simpler **Greedy Maximal Matching (GMM)** algorithm can be used to compute an imperfect schedule with $\gamma = 1/2$. Start from an empty schedule. From all possible links $l \in \mathcal{L}$, pick the link with the largest $q^l c_l$. Add this link to the schedule. Remove all links that are incident with either the sending node or the receiving node of link l. Pick the link with the largest $q^l c_l$ from the *remaining* links, and add to the schedule. Continue until there are no links left. The GMM algorithm has only $O(L \log L)$-complexity (where L is the number of links), and is much easier to implement than MWM. Using the technique in Theorem 10 of [25], we can show that the weight of the schedule computed by the GMM algorithm is at least $1/2$ of the weight of the *maximum-weighted-matching*. According to Proposition 6, the stability region will be at least $\Lambda/2$ using our cross-layer congestion control scheme with the GMM scheduling policy.

For the node-exclusive interference model, a *layered approach* to congestion control is also possible, which considers *separately* the dynamics of congestion control and scheduling [17], [18]. It has been shown that the optimal capacity region Λ in the node-exclusive interference model is bounded by $\frac{2}{3}\Psi_0 \subseteq \Lambda \subseteq \Psi_0$, where

$$\Psi_0 = \left\{ \vec{x} \left| \sum_{l:b(l)=i \text{ or } e(l)=i} \frac{1}{c_l} \sum_{s=1}^{S} H_s^l x_s \leq 1 \text{ for all } i \right. \right\}, \tag{29}$$

and $b(l)$ and $e(l)$ are the sending node and the receiving node, respectively, of link l. The layered approach then chooses the lower bound $\frac{2}{3}\Psi_0$ as the *rate region* for computing the rate allocation [17], [18]. On the other hand, when an imperfect GMM scheduling policy is used, the capacity region can be reduced by half in the worst case (according to Proposition 3). Hence, the layered approach then needs to use $\Psi_0/3 (\subseteq \Lambda/2)$ as the *rate region*. Note that for the layered approach with GMM scheduling, $\Psi_0/3$ is an *upper bound* for its stability region, which is smaller than the *lower bound* of the stability region of the corresponding cross-layer approach (which is $\Lambda/2$ according to Proposition 6). Hence, due to its *conservative* nature, the layered approach always suffers from *worst case* inefficiencies. In Section VII, we will use simulations to show that our cross-layer congestion control scheme can in practice substantially outperform the layered approach.

D. General Interference Models

Under general interference models, it may still be time-consuming to compute a schedule that satisfies (12) for a given value of γ. We now use Proposition 6 to develop a scheduling policy that can cut down the *frequency* of such computation, and hence effectively reduce the computation overhead. This idea is motivated by the observation that implicit costs, being updated by (15), cannot change abruptly. Hence, there is a high chance that a schedule computed earlier can be *reused* in subsequent time-slots. To see this, assume that we know a schedule $\vec{r}^{\,0}$ that satisfies (12) for an inefficiency factor $\gamma_0 > \gamma$ when the implicit cost vector is $\vec{q}^{\,0}$, i.e.,

$$\sum_{l=1}^{L} r_l^0 q_0^l \geq \gamma_0 \max_{\vec{r} \in \mathcal{R}} \sum_{l=1}^{L} r_l q_0^l. \tag{30}$$

Let the implicit cost vector at the current time slot be $\vec{q}$, and let $\vec{r}^*$ denote the corresponding (but unknown) perfect schedule. We can normalize $\vec{q}^0$ and $\vec{q}$ to be of unit length since the corresponding schedules will remain the same. We have,

$$\sum_{l=1}^{L} q^l r_l^* = \sum_{l=1}^{L}(q^l - q_0^l) r_l^* + \sum_{l=1}^{L} q_0^l r_l^* \leq \sum_{l=1}^{L}[q^l - q_0^l]^+ r_l^{\max} + \frac{\sum_{l=1}^{L} q_0^l r_l^0}{\gamma_0},$$

where $r_l^{\max}$ is the maximum rate of link l. Hence, if

$$\sum_{l=1}^{L} q^l r_l^0 \geq \gamma \left\{ \sum_{l=1}^{L}[q^l - q_0^l]^+ r_l^{\max} + \frac{\sum_{l=1}^{L} q_0^l r_l^0}{\gamma_0} \right\},$$

we can still use $\vec{r}^0$ as the imperfect schedule for $\vec{q}$. This approach is even more powerful when the network can remember multiple schedules from the past. Let $\vec{r}^k = [r_1^k, ..., r_L^k]$ and $\vec{q}^k = [q_1^1, ..., q_k^L]$, $k = 1, ..., K$. Assume that the schedules $\vec{r}^1, \vec{r}^2, ..., \vec{r}^K$ correspond to $\vec{q}^1, \vec{q}^2, ..., \vec{q}^K$, respectively, and each pair satisfies (30). Then, as long as

$$\max_{k=1,..,K} \sum_{l=1}^{L} q^l r_l^k \geq \min_{k=1,...,K} \gamma \left\{ \sum_{l=1}^{L}[q^l - q_k^l]^+ r_l^{\max} + \frac{\sum_{l=1}^{L} q_k^l r_l^k}{\gamma_0} \right\}, \quad (31)$$

we do not need to compute a new schedule. Instead, we can use the schedule that maximizes the left hand side of (31). By Proposition 6, the stability region of the system using the above scheduling policy is no smaller than $\gamma\Lambda$. In Section VII, we will use simulations to show that such a simple policy can perform very well in practice.

VI. A Fully Distributed Cross-Layer Congestion Control and Scheduling Algorithm

Proposition 6 opens a new avenue for studying cross-layer design for congestion control in multihop wireless networks. Instead of restricting our attention to the rate allocation at each snapshot of the system (as we did in Section IV where the results tend to be weaker), we can now study the entire time horizon by focusing on the stability region of such a cross-layer-designed system. Motivated by Proposition 6, we now present a *fully distributed* cross-layer congestion control and scheduling algorithm for the node-exclusive interference model in Section V-C. (In contrast, the GMM algorithm in Section V-C still requires centralized implementation.) This new algorithm can be shown to achieve a stability region no smaller than $\Lambda/2$.

The new algorithm uses **Maximal Matching (MM)** to compute the schedule at each time [31]. A *maximal matching* is a matching such that no more links can be added without violating the node-exclusive interference constraint. To be precise, let q_{ij} denote the implicit cost at link (i,j). (For convenience, in this section we will index a link by a node pair (i,j).) For each node i, let $I_i = \{(i,k) \in \mathcal{L}\} \cup \{(k,i) \in \mathcal{L}\}$ denote the set of links incident to node i. A maximal matching $\mathcal{M}$ is a subset of $\mathcal{L}$ such that $q_{ij} \geq 1$ for all $(i,j) \in \mathcal{M}$, and, for each $(i,j) \in \mathcal{L}$, one of the following holds:

$$q_{ij} < 1, \text{ or} \quad (32)$$
$$\text{some link in } I_i \cup I_j \text{ is included in } \mathcal{M}.$$

Note that a maximal matching can be computed in a distributed fashion as follows. When a link (i,j) is added to the matching, we say that both node i and node j are *matched*. For each node i, if it has already been matched, no further action is required. Otherwise, node i scans its neighboring nodes. If there exists a neighboring node j such that node j has not been matched, node i sends a matching request to node j. It is possible that a matching request conflicts with other matching requests. In this case, the nodes involved in the conflict can use some randomization and local coordination to pick any non-conflicting subset of the matching requests. For those nodes whose matching requests are declined, they can repeat the above procedure until every node in the network is either matched or has no neighbors that are not matched.

Let

$$Q_i = \sum_{j:(i,j)\in\mathcal{L}} q_{ij} + \sum_{j:(j,i)\in\mathcal{L}} q_{ji} \quad (33)$$

denote the total cost of the links that are incident to node i. Our new cross-layer congestion control and scheduling algorithm then proceeds as follows.

The Fully Distributed Cross-Layer Congestion Control Algorithm:

At each time slot $[kT, (k+1)T)$:

- A maximal matching $\mathcal{M}(kT)$ is computed based on the implicit costs $\vec{q}(kT)$.
- The data rate of each user of class s is determined by

$$x_s(t) = x_s(kT) = \min\left\{ \frac{w_s}{\sum_{(i,j)\in\mathcal{L}} H_s^{ij} \frac{Q_i(kT)+Q_j(kT)}{c_{ij}}}, M_s \right\} \quad (34)$$

  where c_{ij} is the capacity of link (i,j), and H_s^{ij} is defined as H_s^l, i.e., $H_s^{ij} = 1$, if users of class s use link (i,j); and $H_s^{ij} = 0$, otherwise.
- The implicit costs are updated by:

$$q_{ij}((k+1)T) = \Bigg[q_{ij}(kT) + \alpha\left(\sum_{s=1}^{S} H_s^{ij} \int_{kT}^{(k+1)T} \frac{n_s(t)x_s(kT)}{c_{ij}}dt - T\mathbf{1}_{\{(i,j)\in\mathcal{M}(kT)\}}\right)\Bigg]^+. \quad (35)$$

This new cross-layer congestion control and scheduling algorithm is similar to the algorithms of Sections IV and V in many aspects:

- A user reacts to congestion by reducing its data rate when the implicit costs along its path increase.
- The implicit cost at each link (i,j) is updated based on the difference between the offered load and the schedule of the link.

We can also show the following result on the stability region of the system, which is comparable to Proposition 6.

Proposition 7: If the stepsize α is sufficiently small, then for any offered load $\vec{\rho}$ that resides strictly inside $\Lambda/2$, the system with the above fully distributed cross-layer congestion control algorithm is stable.

Before we sketch the proof for Proposition 7, we note that there is a critical difference between the results in this section and those in Section V. When the maximal matching is computed, we do not care about the precise value of the implicit costs (see (32), where the maximal matching only depends on whether the implicit costs q_{ij} are larger than a chosen threshold). Hence, the maximal matching typically does not satisfy the requirement of the imperfect scheduling policy S_γ in (12), and thus Proposition 6 of Section V can not be directly applied.

Despite this difficulty, we next show that Proposition 7 can be derived from Proposition 6 through an appropriate mapping. Note that under the node-exclusive interference model, the set Ψ_0 is an upper bound on the capacity region Λ (see (29)). Imagine a fictitious system where the capacity constraints are exactly as specified by Ψ_0. Assuming logarithmic utility functions, and assigning a Lagrange multiplier Q_i to each constraint in Ψ_0, we can derive the "optimal" congestion controller for this fictitious system as,

$$x_s(t) = \min\left\{ \frac{w_s}{\sum_{(i,j)\in\mathcal{L}} H_s^{ij} \frac{Q_i(kT)+Q_j(kT)}{c_{ij}}}, M_s \right\} \quad (36)$$

$$Q_i(t+1) = \left[Q_i(t) + \alpha \left(\sum_{l\in I_i} \frac{1}{c_l} \sum_{s=1}^{S} H_s^l x_s - \tilde{r}_i(t) \right) \right]^+, \quad (37)$$

where $\tilde{r}_i(t) = 1$ for all t. These two equations correspond to the congestion control component (5) and the implicit cost update (7), respectively, in our optimal cross-layer congestion control algorithm in Section III. The choice $\tilde{r}_i(t) = 1$ corresponds to the perfect scheduling policy (i.e., (6)) for the fictitious system, and it achieves a weighted-rate-sum of

$$\sum_{i=1}^{N} Q_i(t)\tilde{r}_i(t) = \sum_{i=1}^{N} Q_i(t). \quad (38)$$

In order to apply Proposition 6 to the above fictitious system, we now show that maximal matching is an $S_{\frac{1}{2}}$ policy with respect to the weighted-rate-sum in (38). Using the mapping (33), we can map the fully distributed algorithm in (34-35) to the congestion controller (of the fictitious system) in (36-37) by setting

$$\tilde{r}_i(t) = \sum_{j:(i,j)\in\mathcal{L}} \mathbf{1}_{\{(i,j)\in\mathcal{M}(t)\}} + \sum_{j:(j,i)\in\mathcal{L}} \mathbf{1}_{\{(j,i)\in\mathcal{M}(t)\}}.$$

Note that now $\tilde{r}_i(t) \le 1$, and it corresponds to an imperfect scheduling policy. Further,

$$\begin{aligned}
&\sum_{i=1}^{N} Q_i(t)\tilde{r}_i(t) \\
&= \sum_{i=1}^{N} \left(\sum_{j:(i,j)\in\mathcal{L}} q_{ij}(t) + \sum_{j:(j,i)\in\mathcal{L}} q_{ji}(t) \right) \tilde{r}_i(t) \\
&= \sum_{(i,j)\in\mathcal{L}} q_{ij}(t) \left(\sum_{l\in I_i\cap I_j} \mathbf{1}_{\{l\in\mathcal{M}(t)\}} \right).
\end{aligned}$$

By the definition of the maximal matching, either $q_{ij}(t) < 1$ or $\sum_{l\in I_i\cap I_j} \mathbf{1}_{\{l\in\mathcal{M}(t)\}} \ge 1$. Thus,

$$\sum_{i=1}^{N} Q_i(t)\tilde{r}_i(t) \ge \sum_{(i,j)\in\mathcal{L}} (q_{ij}(t) - 1) \ge \frac{1}{2}\sum_{i=1}^{N} Q_i(t) - L.$$

Hence, any maximal matching satisfies the relaxed definition (25) of policy $S_{\frac{1}{2}}$ with respect to the weighted-rate-sum $\sum_{i=1}^{N} Q_i \tilde{r}_i$. By Proposition 6, the stability region of the system using the fully distributed cross-layer congestion control algorithm is at least $\Psi_0/2$. Since $\Lambda \subset \Psi_0$, we obtain Proposition 7. The details of the proof are available in [20].

VII. Numerical Results

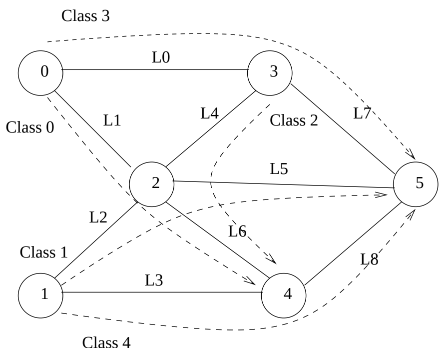
We now use simulations to illustrate the results in this paper. We use the network in Fig. 3. There are 5 classes of users, whose paths are shown in Fig. 3. Their utility functions are all given by $U_s(x_s) = \log x_s$. We first use the following interference model. The path loss $G(i,j)$ from a node i to a node j is given by $G(i,j) = d_{ij}^{-4}$ where d_{ij} is the distance from node i to node j (the positions of the nodes are also given in Fig. 3). We assume that the data rate r_{ij} at link $(i,j) \in \mathcal{L}$ is proportional to the SIR, i.e.,

$$r_{ij} = W \frac{G(i,j)P_{ij}}{N_0 + \sum_{(k,h)\in\mathcal{L},(k,h)\neq(i,j)} G(k,j)P_{k,h}},$$

where N_0 is the background noise and W is the bandwidth of the system. This assumption is suitable for CDMA systems with a moderate processing gain [7]. Each node i has a power constraint $P_{i,\max}$, i.e., the power allocation must satisfy $\sum_{j:(i,j)\in\mathcal{L}} P_{ij} \le P_{i,\max}$ for all i.

We first simulate the case when there is one user for each class. The top figure in Fig. 4 shows the evolution of the data rates for all five users when the network computes the perfect schedule according to (6) at every time slot. We have chosen $W = 10$, $N_0 = 1.0$, $P_{i,\max} = 1.0$ for all nodes i and $\alpha_l = 0.1$ for all links l. Note that the scheduling subproblem (6) for this interference model is a complex non-convex global optimization problem. In [9], we have given an $O(2^N)$ algorithm for solving the perfect schedule, where N is the number of nodes. Executing such an algorithm at every time-slot is extremely time-consuming.

We then simulate the imperfect scheduling policy outlined in Section V-D for general interference models. Such an imperfect scheduling policy attempts to *reuse* schedules that



Node	Position
0	(0.0, 0.0)
1	(0.0, 2.0)
2	(1.0, 1.0)
3	(2.2, 0.0)
4	(2.5, 2.0)
5	(3.5, 1.0)

Fig. 3. The Network Topology

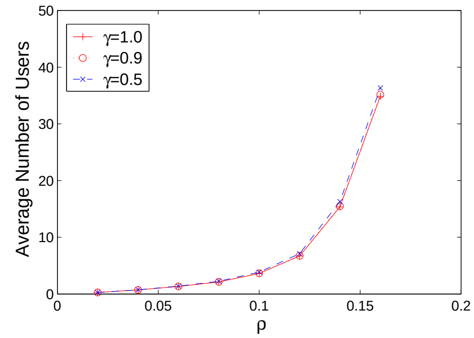


Fig. 5. The average number of users in the system versus load.

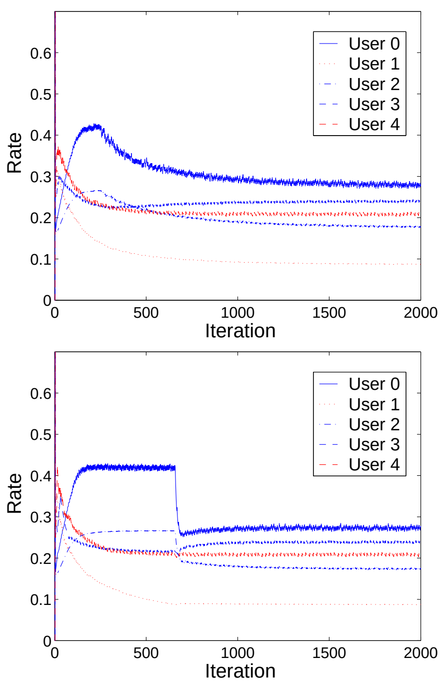


Fig. 4. The evolution of the data rates for all users with perfect scheduling (top) and with imperfect scheduling (bottom, $\gamma = 0.5$).

have already been computed in the past. In our simulation, we have chosen $\gamma_0 = 1.0$ in (30), i.e., each of these past schedules are perfect schedules. The computational complexity could have been further reduced if we had chosen $\gamma_0 < 1$. However, we leave this for future work. Instead, in this paper we focus on how the imperfect scheduling policy can *reduce the number of times that new perfect schedules have to be computed*. The system that we simulate can store at most 10 past schedules. If there are already 10 past schedules and a new perfect schedule is computed, the new schedule will replace the old one that has the smallest weighted-sum $\sum_{l=1}^{L} q^l r_l$. In the bottom figure of Fig. 4, we show the evolution of the data rates when $\gamma = 0.5$. Note that the rate allocation eventually converges to values close to that with perfect scheduling. (The abrupt transitions in the bottom figure indicate times when new schedules are computed.) We also record the number of times that perfect schedules are computed. When $\gamma = 0.5$, perfect schedules are computed in only 7 iterations among the entire 2000 iterations of the simulation, and most of these perfect schedules are computed at the initial stage of the simulation. We have simulated other values of γ and find similar results. In fact, by just reducing γ from 1.0 to 0.9, the number of times that perfect schedules have to be computed is reduced to 34 (over 2000 iterations of simulation). These results indicate that our cross-layer congestion control scheme with the imperfect scheduling policy in Section V-D can substantially reduce the computation overhead and still maintain good performance.

We then simulate the case when there are dynamic arrivals and departures of the users as in Section V. Users of each class arrive to the network according to a Poisson process with rate λ. Each user brings with it a file to transfer whose size is exponentially distributed with mean $1/\mu = 100$ unit. We vary the arrival rate λ (and hence the load $\rho = \lambda/\mu$) and record in Fig. 5 the average number of users in the system at any time for different choices of γ. Given γ, the average number of users in the system will increase to infinity as the offered load ρ approaches a certain limit. This limit can then be viewed as the capacity of the system. From Fig. 5, we observe that the capacity of the system is not significantly affected when γ is reduced from 1.0 to 0.5. On the other hand, the number of time-slots that new perfect schedules have to be computed is reduced to less than 1% of the total number of time-slots when $\gamma = 0.9$, and to less than 0.05% when $\gamma = 0.5$. These results confirm again the effectiveness of our cross-layer congestion control scheme with the imperfect scheduling policy in Section V-D, in reducing the computation overhead and achieving good overall performance.

We next turn to the node-exclusive interference model in Section V-C, where we can draw a comparison with the layered approach to congestion control [17], [18]. We still use the network topology in Fig. 3. The capacity of each link is now fixed at 10 units. Due to space constraints, we only report the result for the case when there are dynamic arrivals and departures of the users. Fig. 6 demonstrates the average number of users in the system versus load with different congestion control and scheduling schemes. We label each curve with the congestion control scheme (we use "Joint" to denote the cross-layer congestion control scheme and use "Layered" to denote the layered approach in [18]), followed by the scheduling policy. (Note that the curve for the cross-layer

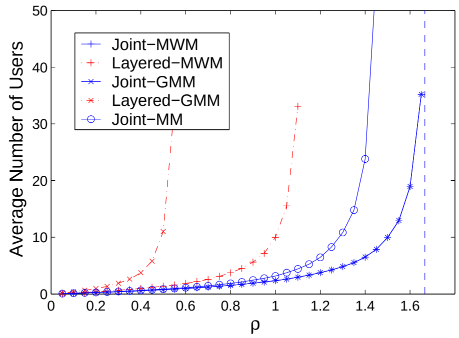


Fig. 6. The average number of users in the system versus load: the node-exclusive interference model

congestion control scheme with GMM scheduling, labeled as "Joint-GMM," in fact overlaps with the curve for the optimal cross-layer congestion control scheme with perfect MWM scheduling, which is the right most curve labeled as "Joint-MWM.") From Fig. 6, we observe that, regardless of the scheduling policy used (either MWM, GMM, or MM), the layered approach always performs *much poorer* than the corresponding cross-layer approach. The performance gap widens even more when an imperfect scheduling policy (such as GMM) is used. In particular, the fully distributed cross-layer congestion control and scheduling algorithm in Section VI (with *imperfect* maximal matching scheduling, labeled "Joint-MM"), actually performs even better than the layered approach with the *perfect* (and more complex) MWM scheduling (labeled "Layered-MWM"). These results demonstrate that the conservative nature of the layered approach indeed hurts the overall performance of the system, and an appropriately designed cross-layer congestion control scheme can perform very well in practice even with imperfect scheduling.

VIII. Conclusion

In this paper, we study how the performance of cross-layer congestion control will be impacted if the network can only use an imperfect (and potentially distributed) scheduling component. When the number of users in the system is fixed, we are able to show some desirable, but weak, results on the fairness and convergence properties of the system. We then turn to the case with dynamic arrivals and departures of the users, and establish stronger results bounding the stability region of the system. Compared with a layered approach that does not design congestion control and scheduling together, the cross-layer approach has provably better performance bounds, and usually substantially outperforms the layered approach. Hence, the cross-layer approach is much more robust to imperfect scheduling than the layered approach. The insights drawn from our analyses also enable us to design a *fully distributed* and high-performance cross-layer congestion control and scheduling algorithm for the node-exclusive interference model.

These results constitute an important step towards designing fully distributed cross-layer congestion control schemes for multihop wireless networks. Several directions for future work are possible. For example, Proposition 6 may be combined with a clustering scheme to design distributed cross-layer congestion-control solutions for large networks. We can also use similar techniques as in [9] to combine cross-layer congestion control with multipath routing. Our result on the fully distributed cross-layer congestion control and scheduling algorithm in Section VI only applies to the node-exclusive interference model. It would be interesting to study whether similar type of simple distributed algorithms can be used, and performance bounds established, for other more general interference models. It would also be important to study the impact of feedback delays, to address the effect of node mobility, and to extend our results to hybrid wireless-wireline networks.

Appendix

A. Proof of Proposition 4

Fix a positive number ϵ such that $\epsilon < \min_{l:q_I^{l,*}\neq 0} q_I^{l,*}$. By assumption, there exists a time slot T_0 such that for all $t \geq T_0$

$$|x_s(t) - x_s^{*,I}| \leq \epsilon, \text{ and } |q^l(t) - q_I^{l,*}| \leq \epsilon. \tag{39}$$

We will first show that there exists a large enough T such that for all l,

$$q_I^{l,*} \frac{1}{T} \sum_{t=T_0}^{T_0+T} r_l(t) \leq q_I^{l,*} \sum_{s=1}^{S} H_s^l x_s^{*,I} + O(\epsilon), \tag{40}$$

where we have used $O(\epsilon)$ to denote the class of functions $f(\epsilon)$ such that $f(\epsilon) < c\epsilon$ for some positive constant c. Note that Inequality (40) trivially holds if $q_I^{l,*} = 0$. If $q_I^{l,*} > 0$, then (39) implies that $q^l(t) > 0$ for all $t \geq T_0$. Hence,

$$\alpha_l \left(\sum_{s=1}^{S} H_s^l x_s(t) - r_l(t) \right) = q^l(t+1) - q^l(t), \text{ for all } t \geq T_0.$$

Summing over $t = T_0, T_0+1, ..., T_0+T$ and dividing by $\alpha_l T$, we obtain,

$$\left| \frac{1}{T} \sum_{t=T_0}^{T_0+T} \sum_{s=1}^{S} H_s^l x_s(t) - \frac{1}{T} \sum_{t=T_0}^{T_0+T} r_l(t) \right| < \frac{q_I^{l,*} + \epsilon}{\alpha_l T}.$$

We can thus pick T large enough such that (using (39) again),

$$\frac{1}{T} \sum_{t=T_0}^{T_0+T} r_l(t) \leq \sum_{s=1}^{S} H_s^l x_s^{*,I} + O(\epsilon).$$

Multiplying both sides by $q_I^{l,*}$, we obtain (40).

Next, since the utility function is logarithmic, we have,

$$x_s^{*,I} = \min \left\{ \frac{w_s}{\sum_{l=1}^{L} H_s^l q_I^{l,*}}, M_s \right\}, \text{ for all } s.$$

Let $J = \{s : x_s^{*,I} = M_s\}$. We have,

$$\sum_{l=1}^{L} q_I^{l,*} \sum_{s=1}^{S} H_s^l x_s^{*,\gamma} = \sum_{s=1}^{S} x_s^{*,\gamma} \sum_{l=1}^{L} H_s^l q_I^{l,*}$$

$$= \sum_{s \notin J} \frac{w_s x_s^{*,\gamma}}{x_s^{*,I}} + \sum_{s \in J} x_s^{*,\gamma} \sum_{l=1}^{L} H_s^l q_I^{l,*}.$$

Hence,

$$\begin{aligned}\sum_{s=1}^{S}\frac{w_s x_s^{*,\gamma}}{x_s^{*,I}} &= \sum_{s\notin J}\frac{w_s x_s^{*,\gamma}}{x_s^{*,I}}+\sum_{s\in J}\frac{w_s x_s^{*,\gamma}}{M_s}\\ &= \sum_{l=1}^{L} q_I^{l,*}\sum_{s=1}^{S}H_s^l x_s^{*,\gamma}-\sum_{s\in J}x_s^{*,\gamma}\sum_{l=1}^{L}H_s^l q_I^{l,*}\\ &\quad +\sum_{s\in J}\frac{w_s x_s^{*,\gamma}}{M_s}. \end{aligned} \tag{41}$$

Since $\vec{x}^{\,*,\gamma}\in\gamma\Lambda$ by definition, each $r_l(t)$ must satisfy

$$\begin{aligned}\sum_{l=1}^{L} q^l(t) r_l(t) &\geq \gamma\sum_{l=1}^{L} q^l(t)\frac{\sum_{s=1}^{S}H_s^l x_s^{*,\gamma}}{\gamma}\\ &= \sum_{l=1}^{L} q^l(t)\sum_{s=1}^{S}H_s^l x_s^{*,\gamma}\ \text{for all } t.\end{aligned}$$

Using (39) again, we obtain,

$$\sum_{l=1}^{L} q_I^{l,*} r_l(t)\geq \sum_{l=1}^{L} q_I^{l,*}\sum_{s=1}^{S}H_s^l x_s^{*,\gamma}-O(\epsilon)\ \text{for all } t\geq T_0.$$

Hence, combining with (40), we have,

$$\begin{aligned}&\sum_{l=1}^{L} q_I^{l,*}\sum_{s=1}^{S}H_s^l x_s^{*,\gamma}\\ \leq\ &\sum_{l=1}^{L} q_I^{l,*}\sum_{s=1}^{S}H_s^l x_s^{*,I}+O(\epsilon)\\ =\ &\sum_{s=1}^{S} x_s^{*,I}\Big(\sum_{l=1}^{L}H_s^l q_I^{l,*}\Big)+O(\epsilon)\\ \leq\ &\sum_{s\notin J} w_s+\sum_{s\in J}M_s\Big(\sum_{l=1}^{L}H_s^l q_I^{l,*}\Big)+O(\epsilon).\end{aligned}$$

Substituting into (41), we have,

$$\begin{aligned}\sum_{s=1}^{S}\frac{w_s x_s^{*,\gamma}}{x_s^{*,I}} &\leq \sum_{s\notin J}w_s+\sum_{s\in J}(M_s-x_s^{*,\gamma})\Big(\sum_{l=1}^{L}H_s^l q_I^{l,*}\Big)\\ &\quad+\sum_{s\in J}\frac{w_s x_s^{*,\gamma}}{M_s}+O(\epsilon)\\ &\leq \sum_{s=1}^{S}w_s+O(\epsilon),\end{aligned}$$

where in the last step we have used the fact that $x_s^{*,\gamma}\leq M_s$ and $\sum_{l=1}^{L}H_s^l q_I^{l,*}\leq w_s/M_s$ when $s\in J$, Finally, letting $\epsilon\to 0$, the result then follows.

B. Proof of Proposition 5

Let A denote the $L\times L$ diagonal matrix whose l-th diagonal element is α_l^0. Let H denote the $L\times S$ matrix whose (l,s)-element is H_s^l. Then $||\vec{q}||_A=\vec{q}^{\,\text{tr}}A^{-1}\vec{q}$, where $[\cdot]^{\,\text{tr}}$ denotes the transpose. For any $\vec{q}^{\,*,0}\in\Phi$, by (15), we have

$$\begin{aligned}&||\vec{q}(t+1)-\vec{q}^{\,*,0}||_A\\ \leq\ &||\vec{q}(t)-\vec{q}^{\,*,0}||_A+2h[H\vec{x}(t)-\vec{r}(t)]^{\,\text{tr}}[\vec{q}(t)-\vec{q}^{\,*,0}]\\ &+h^2[H\vec{x}(t)-\vec{r}(t)]^{\,\text{tr}}A[H\vec{x}(t)-\vec{r}(t)],\end{aligned}\tag{42}$$

Note that

$$\begin{aligned}D_\gamma(\vec{q}(t)) &= \sum_{s=1}^{S}U_s(x_s(t))-[H^{\,\text{tr}}\vec{q}(t)]^{\,\text{tr}}\vec{x}(t)\\ &\quad+\gamma\max_{\vec{r}\in\text{Co}(\mathcal{R})}\vec{r}^{\,\text{tr}}\vec{q}(t)\\ &\leq \sum_{s=1}^{S}U_s(x_s(t))-[H^{\,\text{tr}}\vec{q}(t)]^{\,\text{tr}}\vec{x}(t)+\vec{r}^{\,\text{tr}}(t)\vec{q}(t),\end{aligned}$$

and

$$\begin{aligned}D(\vec{q}^{\,*,0}) &= \max_{0\leq x_s\leq M_s}\left\{\sum_{s=1}^{S}U_s(x_s)-(H^{\,\text{tr}}\vec{q}^{\,*,0})^{\,\text{tr}}\vec{x}\right\}\\ &\quad+\max_{\vec{r}\in\text{Co}(\mathcal{R})}\vec{r}^{\,\text{tr}}\vec{q}^{\,*,0}\\ &\geq \sum_{s=1}^{S}U_s(x_s(t))-(H^{\,\text{tr}}\vec{q}^{\,*,0})^{\,\text{tr}}\vec{x}(t)+\vec{r}^{\,\text{tr}}(t)\vec{q}^{\,*,0}.\end{aligned}$$

Hence,

$$D(\vec{q}^{\,*,0})-D_\gamma(\vec{q}(t))\geq[H\vec{x}(t)-\vec{r}(t)]^{\,\text{tr}}[\vec{q}(t)-\vec{q}^{\,*,0}].$$

Substituting into (42), we have

$$\begin{aligned}&||\vec{q}(t+1)-\vec{q}^{\,*,0}||_A\\ \leq\ &||\vec{q}(t)-\vec{q}^{\,*,0}||_A+2h[D(\vec{q}^{\,*,0})-D_\gamma(\vec{q}(t))]\\ &+h^2[H\vec{x}(t)-\vec{r}(t)]^{\,\text{tr}}A[H\vec{x}(t)-\vec{r}(t)].\end{aligned}$$

Fix $\eta>0$. Let

$$\Phi_\gamma(\eta)=\{\vec{q}|D_\gamma(\vec{q})\leq D(\vec{q}^{\,*,0})+\eta\}.\tag{43}$$

Since both $\vec{x}(t)$ and $\vec{r}(t)$ are bounded, there exists $M<\infty$ such that

$$\max_{0\leq x_s\leq M_s,\vec{r}\in\text{Co}(\mathcal{R})}(H\vec{x}-\vec{r})^{\,\text{tr}}A(H\vec{x}-\vec{r})\leq M.$$

If we pick $h\leq\eta/M$, then as long as $\vec{q}(t)\notin\Phi_\gamma(\eta)$, we have

$$||\vec{q}(t+1)-\vec{q}^{\,*,0}||_A\leq||\vec{q}(t)-\vec{q}^{\,*,0}||_A-h\eta.$$

Hence, eventually, $\vec{q}(t)$ will enter the set $\Phi_\gamma(\eta)$. On the other hand, if we pick $h\leq\eta/\sqrt{M}$, then once $\vec{q}(t)\in\Phi_\gamma(\eta)$, we have

$$\begin{aligned}&\sqrt{||\vec{q}(t+1)-\vec{q}^{\,*,0}||_A}\\ \leq\ &\sqrt{||\vec{q}(t)-\vec{q}^{\,*,0}||_A}+\sqrt{||\vec{q}(t+1)-\vec{q}(t)||_A}\\ \leq\ &\sqrt{||\vec{q}(t)-\vec{q}^{\,*,0}||_A}+\eta.\end{aligned}\tag{44}$$

Since the inequality (44) holds for any $\vec{q}^{\,*,0}\in\Phi\subset\Phi_\gamma(\eta)$, it implies that

$$d(\vec{q}(t+1),\Phi)\leq d(\vec{q}(t),\Phi)+\eta.$$

Hence, if $h\leq\min\{\eta/M,\eta/\sqrt{M}\}$, then there exists time T_0 such that

$$d(\vec{q}(t),\Phi)\leq\max_{\vec{p}\in\Phi_\gamma(\eta)}d(\vec{p},\Phi)+\eta,\ \text{for all } t\geq T_0.$$

It is easy to show that the right hand side converges to $\max_{\vec{p}\in\Phi_\gamma}d(\vec{p},\Phi)$ as $\eta\to 0$, the result then follows.

ACKNOWLEDGMENT

We thank the anonymous reviewers for their insightful comments and suggestions.

REFERENCES

[1] X. Lin and N. B. Shroff, "The Impact of Imperfect Scheduling on Cross-Layer Rate Control in Multihop Wireless Networks," in *Proceedings of IEEE INFOCOM*, Miami, FL, March 2005.

[2] L. Xiao, M. Johansson, and S. Boyd, "Simultaneous Routing and Resource Allocation via Dual Decomposition," in *Proceedings of 4th Asian Control Conference*, Singapore, September 2002, pp. 29–34.

[3] M. Johansson, L. Xiao, and S. Boyd, "Simultaneous Routing and Power Allocation in CDMA Wireless Data Networks," in *Proceedings of IEEE International Conference on Communications*, Anchorage, Alaska, May 2003, pp. 51–55.

[4] M. J. Neely, E. Modiano, and C. E. Rohrs, "Dynamic Power Allocation and Routing for Time Varying Wireless Networks," in *Proceedings of IEEE INFOCOM*, San Francisco, April 2003.

[5] S. Toumpis and A. J. Goldsmith, "Capacity Regions for Wireless Ad Hoc Networks," *IEEE Transactions on Wireless Communications*, vol. 2, no. 4, pp. 736–748, July 2003.

[6] A. Eryilmaz, R. Srikant, and J. Perkins, "Stable Scheduling Policies for Fading Wireless Channels," *IEEE/ACM Transactions on Networking*, vol. 13, no. 2, pp. 411–424, April 2005.

[7] R. L. Cruz and A. V. Santhanam, "Optimal Routing, Link Scheduling and Power Control in Multi-hop Wireless Networks," in *Proceedings of IEEE INFOCOM*, San Francisco, April 2003.

[8] L. Tassiulas and A. Ephremides, "Stability Properties of Constrained Queueing Systems and Scheduling Policies for Maximum Throughput in Multihop Radio Networks," *IEEE Transactions on Automatic Control*, vol. 37, no. 12, pp. 1936–1948, December 1992.

[9] X. Lin and N. B. Shroff, "Joint Rate Control and Scheduling in Multihop Wireless Networks," in *Proceedings of the IEEE Conference on Decision and Control*, Paradise Island, Bahamas, December 2004.

[10] M. Johansson and L. Xiao, "Scheduling, Routing and Power Allocation for Fairness in Wireless Networks," in *IEEE VTC - Spring*, Milan, Italy, May 2004.

[11] M. Chiang, "To Layer or Not to Layer: Balancing Transport and Physical Layers in Wireless Multihop Networks," in *Proceedings of IEEE INFOCOM*, Hong Kong, March 2004.

[12] M. J. Neely, E. Modiano, and C. Li, "Fairness and Optimal Stochastic Control for Heterogeneous Networks," in *Proceedings of IEEE INFOCOM*, Miami, FL, March 2005.

[13] L. Chen, S. H. Low, and J. C. Doyle, "Joint Congestion Control and Media Access Control Design for Wireless Ad Hoc Networks," in *Proceedings of IEEE INFOCOM*, Miami, FL, March 2005.

[14] I. Paschalidis, W. Lai, and D. Starobinski, "Asymptotically Optimal Transmission Policies for Low-Power Wireless Sensor Networks," in *Proceedings of IEEE INFOCOM*, Miami, FL, March 2005.

[15] S. H. Low and R. Srikant, "A Mathematical Framework for Designing a Low-Loss Low-Delay Internet," *Network and Spatial Economics*, vol. 4, no. 1, pp. 75–102, March 2004.

[16] R. Srikant, *The Mathematics of Internet Congestion Control*. Birkhauser, 2004.

[17] S. Sarkar and L. Tassiulas, "End-to-end Bandwidth Guarantees Through Fair Local Spectrum Share in Wireless Ad-hoc Networks," in *Proceedings of the IEEE Conference on Decision and Control*, Maui, Hawaii, December 2003.

[18] Y. Yi and S. Shakkottai, "Hop-by-hop Congestion Control over a Wireless Multi-hop Network," in *Proceedings of IEEE INFOCOM*, Hong Kong, March 2004.

[19] Y. Xue, B. Li, and K. Nahrstedt, "Price-based Resource Allocation in Wireless Ad Hoc Networks," in *Proceedings of the Eleventh International Workshop on Quality of Service (IWQoS 2003), also Lecture Notes in Computer Science, ACM Springer-Verlag*, vol. 2707, Monterey, CA, June 2003, pp. 79–96.

[20] X. Lin and N. B. Shroff, "The Impact of Imperfect Scheduling on Cross-Layer Rate Control in Multihop Wireless Networks," *Technical Report, Purdue University, http://min.ecn.purdue.edu/∼linx/papers.html*, 2004.

[21] T. Bonald and L. Massoulie, "Impact of Fairness on Internet Performance," in *Proceedings of ACM Sigmetrics*, Cambridge, MA, June 2001, pp. 82–91.

[22] J. Mo and J. Walrand, "Fair End-to-End Window-Based Congestion Control," *IEEE/ACM Transactions on Networking*, vol. 8, no. 5, pp. 556–567, October 2000.

[23] J. M. Borwein and A. S. Lewis, *Convex Analysis and Nonlinear Optimization: Theory and Examples*. New York: Springer, 2000.

[24] N. Z. Shor, *Minimization Methods for Non-Differentiable Functions*. Berlin: Springer-Verlag, 1985.

[25] E. Leonardi, M. Mellia, F. Neri, and M. A. Marsan, "On the Stability of Input-Queued Switches with Speed-Up," *IEEE/ACM Transactions on Networking*, vol. 9, no. 1, pp. 104–118, February 2001.

[26] G. De Veciana, T. J. Lee, and T. Konstantopoulos, "Stability and Performance Analysis of Networks Supporting Elastic Services," *IEEE/ACM Transactions on Networking*, vol. 9, no. 1, pp. 2–14, February 2001.

[27] G. Fayolle, A. L. Fortelle, J. M. Lasgouttes, L. Massoulie, and J. Roberts, "Best Effort Networks: Modeling and Performance Analysis via Large Network Asymptotics," in *Proceedings of IEEE INFOCOM*, Anchorage, Alaska, April 2001.

[28] H. Q. Ye, "Stability of Data Networks Under an Optimization-Based Bandwidth Allocation," *IEEE Transactions on Automatic Control*, vol. 48, no. 7, pp. 1238–1242, July 2003.

[29] M. J. Neely, E. Modiano, and C. E. Rohrs, "Power Allocation and Routing in Multibeam Satellites with Time-Varying Channels," *IEEE/ACM Transactions on Networking*, vol. 11, no. 1, pp. 138–152, February 2003.

[30] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*. Englewood Cliffs, New Jersey: Prentice-Hall, 1982.

[31] J. G. Dai and B. Prabhakar, "The Throughput of Data Switches with and without Speedup," in *Proceedings of IEEE INFOCOM*, Tel Aviv, Israel, March 2000, pp. 556–564.

Xiaojun Lin (S'02 / M'05) received his B.S. from Zhongshan University, Guangzhou, China, in 1994, and his M.S. and Ph.D. degrees from Purdue University, West Lafayette, Indiana, in 2000 and 2005, respectively. He is currently an Assistant Professor of Electrical and Computer Engineering at Purdue University.

Dr. Lin's research interests are resource allocation, optimization, network pricing, routing, congestion control, network as a large system, cross-layer design in wireless networks, mobile ad hoc and sensor networks. He received the 2005 best paper of the year award from *Journal of Communications and Networks*. His paper was also one of two runner-up papers for the best-paper award at IEEE INFOCOM 2005.

Ness B. Shroff (S'91 / M'93 / SM'01) received his Ph.D. degree from Columbia University, NY in 1994. He has been at Purdue University since 1994, and is currently a Professor of Electrical and Computer Engineering at Purdue.

Dr. Shroff's research interests span the areas of wireless and wireline communication networks. He is especially interested in fundamental problems in the design, performance, control, security, and pricing of these networks.

Dr. Shroff is an editor for the IEEE/ACM Trans. on Networking and the Computer Networks Journal, and past editor of IEEE Communications Letters. He was the Technical Program co-chair for IEEE INFOCOM'03 (San Francisco, CA), the panel co-chair for ACM Mobicom'02 (Atlanta, GA), program co-chair for the symposium on high-speed networks, Globecom 2001 (San Francisco, CA), and conference chair for the 14th Annual IEEE Computer Communications Workshop (Estes Park, CO). He was the co-organizer of the NSF Workshop on "Fundamental Research in Networking" in April 2003. He has received numerous awards including the NSF CAREER award in 1996, the *Computer Network Journal's* best paper award of 2003, and the 2005 best paper of the year award from *Journal of Communications and Networks*.