

Delay-Aware Cross-Layer Design for Network Utility Maximization in Multi-hop Networks

Haozhi Xiong, Ruogu Li, Atilla Eryilmaz and Eylem Ekici

Abstract—We investigate the problem of designing *delay-aware* joint flow control, routing, and scheduling algorithms in general multi-hop networks for maximizing network utilization. Since the end-to-end delay performance has a complex dependence on the high-order statistics of cross-layer algorithms, earlier optimization-based design methodologies that optimize the long term network utilization are not immediately well-suited for delay-aware design. This motivates us in this work to develop a novel design framework and alternative methods that take advantage of several unexploited design choices in the routing and the scheduling strategy spaces. In particular, we reveal and exploit a crucial characteristic of back pressure-type controllers that enables us to develop a novel link rate allocation strategy that not only optimizes long-term network utilization, but also yields *loop free multi-path routes* between each source-destination pair. Moreover, we propose a regulated scheduling strategy, based on a token-based service discipline, for shaping the per-hop delay distribution to obtain highly *desirable end-to-end delay performance*. We establish that our joint flow control, routing, and scheduling algorithm achieves loop-free routes and optimal network utilization. Our extensive numerical studies support our theoretical results, and further show that our joint design leads to substantial end-to-end delay performance improvements in multi-hop networks compared to earlier solutions.

I. INTRODUCTION

Communication networks are expected to serve a variety of essential applications that demand high long-term throughput and low end-to-end delay. Over the last decade, we have witnessed the development of increasingly sophisticated optimization and control techniques targeting flow control, routing and scheduling components to address the cross-layer resource allocation problems for communication networks. Among various different developed policies, an important class of *throughput-optimal policies* has evolved from the seminal work [1] of Tassiulas and Ephremides, where they proposed the well-known back-pressure scheduling/routing policy. This policy utilizes properly maintained queue-length information to dynamically determine scheduling and routing decisions that optimize the long-term maximal throughput levels between all source-destination pairs. More recent works (e.g. [2], [3], [4], [5]; also see [6], [7], [8] and references therein) extended this framework by developing an optimization-based design methodology for the development of joint flow control, routing, and scheduling algorithms to maximize the long-term utilization of the network resources, measured through proper functions of throughput.

Haozhi Xiong, Ruogu Li, Atilla Eryilmaz and Eylem Ekici ({xiong, lir, eryilmaz, ekici}@ece.osu.edu) are with the Department of Electrical and Computer Engineering at The Ohio State University, Columbus, Ohio 43210 USA.

However, existing works have dominantly concentrated only on the long-term performance metrics of throughput or utilization, while ignoring the end-to-end delay metric that is crucial to many essential applications. This restriction allowed for the formulation of the design problem as a static optimization problem in terms of the mean behavior of the network algorithm operation. In particular, long-term design objectives can easily be described in terms of the mean flow rates and the mean link rates that the network algorithm provides to the applications. Unfortunately, this approach is no longer applicable when end-to-end delay is taken into account as end-to-end delay is a complex stochastic process that depends on the higher order statistics of the network algorithm operation. This calls for a more sophisticated delay-aware cross-layer design framework, and novel strategies that provide favorable end-to-end delay performance, while preserving long-term optimality characteristics.

With this motivation, in this paper, we are interested in the well-founded design of cross-layer network algorithms for general multi-hop networks that not only utilizes the network resources for long-term throughput optimality, but also exhibits desirable end-to-end delay characteristics. Our contributions in this direction can be summarized as follows:

- We propose a novel design paradigm that decouples the objectives of long-term utility maximization from the end-to-end delay-aware resource allocation. This framework is expected to enable the systematic development of future schemes, in addition to the one we develop in this paper.
- We reveal the solution given by the back-pressure policy has a unique structure which facilitate us to establish *loop-free* multi-path routes that *guarantee long-term network utility maximization*. This new routing-scheme not only inherits the adaptive and optimal nature of the long-term optimal algorithms, but also eliminates the unnecessary loops to reduce the end-to-end delay without sacrificing from throughput.
- We combine this loop-free route construction strategy with a token-based scheduling discipline that regulates the higher-order statistics of service processes to achieve drastic reductions in the end-to-end delay performance, while guaranteeing long-term optimality characteristics.

The above points also constitute the main differences of this work from the recent efforts in the design of algorithms with low end-to-end delay performance (e.g. [9], [10]). We also note that there has been recent interest in deriving fundamental bounds on the delay performance ([11], [12], [13]). In our future work, we are interested in utilizing and extending these results to study the gap between these fundamental bounds and the delay performance of our algorithms.

The remainder of the paper is organized as follows. Section II introduces our system model and objectives together with the description of several existing algorithms. In Section III, we propose our delay-aware design framework that describes the desired characteristics and interconnections of the routing, scheduling, and flow control components of the network algorithm. In Section IV, we build upon the proposed design framework to construct a delay-aware cross-layer algorithm that performs the desired tasks. The numerical result of the policy and our concluding remarks are provided in Section V and VI, respectively.

II. SYSTEM MODEL AND OBJECTIVE

In this work, we study wired networks to concentrate on the delay behavior of network algorithms without the additional complications of interference limited wireless communications. Discussion of possible extension of our proposed frameworks to interference limited networks is provided in the conclusions. We consider a fixed multi-hop network represented by graph $\mathcal{G} = (\mathcal{N}, \mathcal{L}, \mathbf{c})$, where \mathcal{N} is the set of nodes, \mathcal{L} is the set of *bidirectional* links (i, j) where $i, j \in \mathcal{N}$. We use $\mathbf{c} = (c_{ij})_{\{(i,j) \in \mathcal{L}\}}$ to denote the vector of *bidirectional link capacities* in packets per slot, i.e., both c_{ij} and c_{ji} refer to the bidirectional link capacity of the bidirectional link (i, j) . For clarity, we use $\langle i, j \rangle$ to denote the *directed* link from node i to node j . Time is slotted in our system, and external packets arrive at the beginning of each time slot.

The network resources are to be shared by a set of commodities. We distinguish different *commodities* by their destinations. We define \mathcal{D} to be the set of all destination nodes. In this paper, we are interested in designing joint flow-control, scheduling, and routing policies with desirable long-term throughput and short-term delay characteristics. Our dual goal will be discussed in further detail after we introduce some notations.

In each time slot, the service on the link $\langle i, j \rangle$ of commodity d is denoted by $R_{ij}^d[t]$, which is assumed to be a stationary ergodic stochastic process. It is determined by the scheduling and routing policy. We let $r_{ij}^d = \lim_{t \rightarrow \infty} \frac{1}{t} \sum_{\tau=0}^t \mathbb{E}(R_{ij}^d[\tau])$ to be the link rate of commodity d on link $\langle i, j \rangle$, and define $\mathbf{r} = (r_{ij}^d)_{i,j,d}$ to be the vector of all such link rates. Under flow control mechanism, the number of the exogenous packets that arrives at node s destined to node d at time slot t is denoted by $X_s^d[t]$, which is also assumed to be stationary and ergodic. Similarly, $x_s^d = \lim_{t \rightarrow \infty} \frac{1}{t} \sum_{\tau=0}^t \mathbb{E}(X_s^d[\tau])$ is the corresponding rate, and we let $\mathbf{x} = (x_s^d)_{s,d}$ to be exogenous arrival rate vector. A utility function $U_{sd}(x_s^d)$ is associated with each source-destination pair (s, d) . We make the following typical assumption on the utility function:

Assumption 1: The utility functions $\{U_{sd}(x_s^d)\}_{s,d}$ are strictly concave, twice differentiable and increasing functions.¹

A. Objective

Our goal is to develop a joint flow-control, scheduling, and routing algorithm that not only optimizes long-term network

utilization, but also provides desirable delay characteristics. We discuss these two goals next.

Utility Maximization: For the network $(\mathcal{N}, \mathcal{L}, \mathbf{c})$, the network utilization maximization optimization problem is defined as:

$$\max_{\{\mathbf{X}[t], \mathbf{R}[t]\}} \sum_{s,d} U_{sd}(x_s^d) \quad (1)$$

$$s.t. \quad X_s^d[t] \geq 0, \forall s, d \in \mathcal{N}, \forall t \geq 0, \quad (2)$$

$$R_{ij}^d[t] \geq 0, \forall (i, j) \in \mathcal{L}, \forall t \geq 0, \quad (3)$$

$$\sum_d R_{ij}^d[t] + \sum_d R_{ji}^d[t] \leq c_{ij}, \\ \forall (i, j) \in \mathcal{L}, d \in \mathcal{D}, \forall t \geq 0, \quad (4)$$

$$x_i^d + \sum_{\langle m,i \rangle \in \mathcal{L}} r_{mi}^d \leq \sum_{\langle i,j \rangle \in \mathcal{L}} r_{ij}^d, \\ \forall i \in \mathcal{N}, \forall d \in \mathcal{D}, i \neq d. \quad (5)$$

We define the *feasible* solution as:

Definition 1: (Feasible solution) A solution $(\mathbf{X}[t], \mathbf{R}[t])$ is *feasible* if it satisfies conditions (2) to (5). \diamond

We use $\{\mathbf{X}^*[t], \mathbf{R}^*[t]\}$ to denote the optimal solution to (1), and we define

$$\mathbf{x}^* = \lim_{t \rightarrow \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbf{X}^*[\tau], \quad (6)$$

$$\mathbf{r}^* = \lim_{t \rightarrow \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbf{R}^*[\tau]. \quad (7)$$

Note that under Assumption 1, \mathbf{x}^* is unique while \mathbf{r}^* generally belongs to a set of optimal link rates denoted by \mathcal{R}^* .

In this paper, we define the stability as follows:

Definition 2: (Network stability) We say the network is *stable* in the mean sense if for any queue (price) $P_{nj}^d, \forall n, j \in \mathcal{N}, \forall d \in \mathcal{N}$, we have $\limsup_{t \rightarrow \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}[P_{nj}^d[\tau]] < \infty$. \diamond

Delay Improvement: A step beyond just solving the above optimization, our second goal is to develop a new mechanism that reduces the end-to-end delay experienced by the traffic while maintains the utility maximizing nature. The end-to-end delay experienced by one packet is defined as the difference between the time instance of injection at the source and reception at the destination, which is a short-term metric instead of a long-term throughput. We also aim to find a new architecture that can decouple the different objectives.

B. Background

The back-pressure algorithm maintains a queue for each commodity d in each node i whose length at time t is denoted by $P_i^d[t]$. The algorithm is proposed for a general network model, and specifically, the back-pressure algorithm under our system model is given by

The Discrete Time Back-Pressure (DTBP) Policy ([14], [5])

- **Queue (Price) Evolution:**

Each queue evolves as

$$P_i^d[t+1] = \left(P_i^d[t] - \sum_{\langle i,j \rangle \in \mathcal{L}} R_{ij}^d[t] \right)^+ + X_i^d[t] + \sum_{\langle m,i \rangle \in \mathcal{L}} R_{mi}^d[t] \quad (8)$$

¹This is not a critical assumption, but will make our analysis clear.

- **Rate control:**

In time slot t , the flow controller determines the mean of the injection as

$$\mathbb{E}(X_s^d[t] | P_s^d[t]) = \min\{U_{sd}^{\prime-1}(P_s^d[t]/K), X_{max}\},$$

where X_{max} is some finite upper bound for the arrival rate.

- **Scheduling and Routing:**

Each link $l = (i, j)$ selects

$$d_l^*[t] = \operatorname{argmax}_d |P_i^d[t] - P_j^d[t]|, l \in \mathcal{L}, \quad (9)$$

and assign

$$R_{ij}^d[t] = \begin{cases} c_{ij}, & \text{if } d = d_l^*[t] \text{ and } P_i^d[t] - P_j^d[t] > 0; \\ 0, & \text{otherwise.} \end{cases} \quad (10)$$

In the event of multiple commodities satisfying Equation (9), we arbitrarily choose one such commodity with equal probability as a *tie-breaking rule*.

The constant K in the algorithm is a design parameter that determines how close the algorithm can converge to its optimal solution (see [14], [5]). Also we shall see in Section IV this solution is related to dual decomposition.

Remark 1: In our model, since there is no interference between links, the back-pressure algorithm reduces to making decisions on each link independently. Since each link can only transmit in one direction at a given time slot, it is necessary to use the weight defined in Equation (9) to determine which one of the directional links should transmit.

Remark 2: Note that the rate assignment in Equation (10) implies that we only allow transmission on those links with *strictly positive* maximum back-pressure. This does not affect the optimality of the DTBP algorithm.

Remark 3: The price evolution in Equation (8) implies that in the case of a node has less packets than scheduled service, dummy packets are transmitted. Thus the allocated service $R_{ij}^d[t]$ always *equals* to number of packets transmitted over link $\langle i, j \rangle$ at time t as suggested in Equation (8).

There are many follow up works of the back-pressure algorithm. In particular, in [10], an interesting *min-resource algorithm* is proposed based on the back-pressure algorithm. In this implementation, instead of using the queue-length difference $|P_i^d[t] - P_j^d[t]|$ as the weight of a link, $(|P_i^d[t] - P_j^d[t]| - M)$ is used, where M is some positive constant. This modification discourages the use of links unless the queue-length differences are sufficiently high, and hence reduces possible loops in the network.

III. DELAY-AWARE DESIGN FRAMEWORK

In this section, we expose the delay deficiencies of long-term utility maximizing designs such as DTBP and its variants, both conceptually and through numerical studies. In particular, we reveal that both the multi-path routing and the scheduling components of the earlier designs must be significantly changed or enhanced to obtain favorable end-to-end delay performance. Based on the observations, we propose a general delay-aware design framework whereby the utility maximization is combined with delay-aware routing and scheduling components to achieve our dual objective.

A. Deficiencies in multi-path routing

Long-term performance optimizing algorithms share the common characteristic of continuously searching for routes to maximize the end-to-end throughput of the flows. However, this potentially causes loops and unnecessarily long routes for a large subset of the packets, as we shall demonstrate in the following example.

Consider a 6×6 grid with unit capacity links serving two commodities as shown in Fig. 1. The source-destination node pair for Commodity 1 and Commodity 2 are marked with square and circle respectively. Under the operation of DTBP, the hop-count distribution of Commodity 1 packets converges to the plotted distribution. We can observe that the hop count exceeds the maximum possible loop-free path length of 35 for a non-negligible fraction of the packets. It turns out that such behavior is typical for similar long-term optimal policies under different setups. In this work, we are interested in preserving the long-term optimality of such solutions, but also eliminating loops and hence significantly improving the delay performance. We will explore the diversity in the set of optimal link rates \mathcal{R}^* to achieve this.

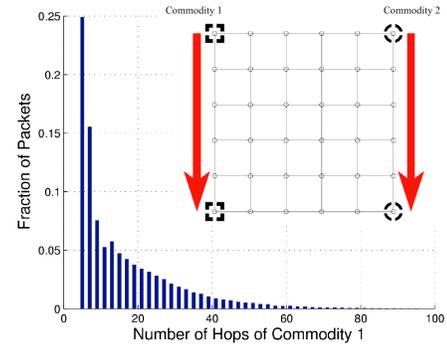


Fig. 1. The distribution of hop-count for flow 1 packets indicates the presence of unnecessary loops in their routes

B. Deficiencies in scheduling and flow control

Long-term utility maximizing policies such as DTBP converge to a set of mean flow and link rates that solves (1). Yet, the end-to-end delay of these policies is a complex function of higher order moments of the packet arrival and service processes that are determined by the flow controller and the scheduler.

As a simple example, it is well-known ([15]) that in a G/G/1 queueing system, the mean waiting time W is bounded by

$$W \leq \frac{\sigma_a^2 + \sigma_b^2}{2\bar{t}(1 - \rho)}, \quad (11)$$

where σ_a^2 is the variance of inter-arrival time, σ_b^2 is the variance of service time, \bar{t} is the mean of inter-arrival time, and ρ is the utility factor. This formulation suggests that reducing the variance of the arrival and service processes is crucial in reducing the delay.

In a large scale multi-hop network, it is not possible to formulate the end-to-end delay as a function of the arrival and service processes. However, motivated by the above observation, in this work, we are interested in regulating the flow

controller and scheduling components of the long-term optimal design to improve delay performance.

C. Proposed multi-layer design framework

The previous two subsections expose deficiencies in the routing, scheduling, and flow control components of existing long-term utility maximizing algorithms, and reveal new opportunities in the design of delay-aware cross layer algorithms. These motivate us in this section, to propose a novel *multi-layer design framework* for multi-purpose policy design.

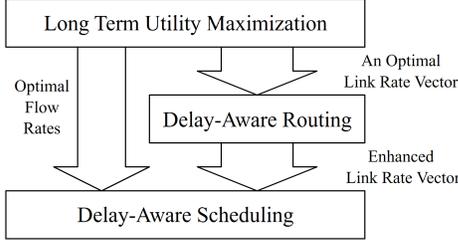


Fig. 2. Multi-Layer Design Framework

The qualitative structure of our proposed framework is shown in Fig. 2, where the goals of long-term optimality, loop-free routing, and regulated flow control and scheduling are decoupled into different layers of the conceptual framework.

In the first layer, we solve the problem of utility maximization. Note that under Assumption 1, the optimal solution for flow rates, \mathbf{x}^* is unique, while \mathbf{r}^* may have multiple solutions. The delay-aware routing layer exploits the freedom amongst these link-rate solutions and obtains a set of more desirable link rates that improves delay performance while preserving the utility maximization. Acquiring the utility maximization flow rates from the first layer and the delay-aware link-rates from the second layer, the delay-aware scheduler regulates the arrival and service process to achieve better delay performance. As such, this establishes a design framework for developing delay-aware cross-layer algorithms that possess the above characteristics and hence is expected to be useful for algorithm designs for diverse network scenarios. In particular, we shall next present specific algorithms with attractive delay characteristics.

IV. DELAY-AWARE ROUTING AND SCHEDULING

In this section, we present our delay-aware routing and delay-aware scheduling algorithms for our proposed framework. Our delay-aware routing algorithm preserves utility maximizing properties while avoiding loops. Our delay-aware scheduling algorithm implements a network stabilizing token-based scheme that significantly reduces queue lengths for any feasible rate assignment. We then combine these two schemes in our delay-aware, cross layer congestion control, routing and scheduling policy following our proposed architecture of Section III-C.

A. Delay-aware route construction

To overcome the disadvantages observed in the back-pressure algorithm, a natural idea is to restrict the direction

in which packets can be transmitted over a link for a certain commodity. To achieve this, we first introduce a fluid approximation associated with (1) for the following discussion.

$$\max_{\mathbf{x}, \mathbf{r}} \sum_{s,d} U_{sd}(x_s^d) \quad (12)$$

$$s.t. \quad x_s^d \geq 0, \forall s, d \in \mathcal{N}, \quad (13)$$

$$r_{ij}^d \geq 0, \forall (i, j) \in \mathcal{L}, \quad (14)$$

$$\sum_d r_{ij}^d + \sum_d r_{ji}^d \leq c_{ij}, \forall (i, j) \in \mathcal{L}, d \in \mathcal{D}, \quad (15)$$

$$x_i^d + \sum_{(m,i) \in \mathcal{L}} r_{mi}^d \leq \sum_{(i,j) \in \mathcal{L}} r_{ij}^d, \quad \forall i \in \mathcal{N}, \forall d \in \mathcal{D}, i \neq d. \quad (16)$$

Note that if an optimal solution of (1) is $\{\mathbf{X}^*[t], \mathbf{R}^*[t]\}$, it is shown in previous works (e.g. [14], [5]) that the associated $(\mathbf{x}^*, \mathbf{r}^*)$ defined in (6) and (7) is an optimal solution of (12).

An Alternate Optimal Solution: To ensure the optimality of the proposed algorithm, we explore the solution space for an alternate optimal solution with a particular structure. First, we define the following mapping of the link rates:

Definition 3: (Delay-aware link rate mapping) The delay-aware link rate mapping $\hat{R}_{ij}^d[t]$ on the directed link $\langle i, j \rangle$ for commodity d at time t is defined as

$$\hat{R}_{ij}^d[t] = \frac{1}{t} \left(\sum_{\tau=0}^{t-1} (R_{ij}^d[\tau] - R_{ji}^d[\tau]) \right)^+, \quad (17)$$

where $d \in \mathcal{N}$, $(i, j) \in \mathcal{L}$ and $(Z)^+ = \max(0, Z)$. Note that this measures the running average of the *net rate* of commodity d traffic traversing link $\langle i, j \rangle$. \diamond

Suppose we use $R_{ij}^d[t] = R_{ij}^{*d}[t]$ in (17). Let $\hat{r}_{ij}^{*d} = \lim_{t \rightarrow \infty} \hat{R}_{ij}^d[t]$ and $\hat{\mathbf{r}}^* = (\hat{r}_{ij}^{*d})_{i,j,d}$ then $\hat{\mathbf{r}}^*$ is related to \mathbf{r}^* as:

$$\hat{r}_{ij}^{*d} = (r_{ij}^{*d} - r_{ji}^{*d})^+, \quad (18)$$

$$\hat{r}_{ji}^{*d} = (r_{ji}^{*d} - r_{ij}^{*d})^+. \quad (19)$$

We shall show that the link rate mapping (18) and (19) preserves flow rate optimality.

Proposition 1: If $(\mathbf{x}^*, \mathbf{r}^*)$ is an optimal solution to (12), then $(\hat{\mathbf{x}}^*, \hat{\mathbf{r}}^*)$ with $\hat{\mathbf{x}}^* = \mathbf{x}^*$ and $\hat{\mathbf{r}}^*$ being as defined in Equations (18) and (19) is also an optimal solution.

Proof: The detailed proof is provided in Appendix A. \blacksquare

Remark: Note that under this link rate assignment, for each $(i, j) \in \mathcal{L}$, at least one of \hat{r}_{ij}^{*d} and \hat{r}_{ji}^{*d} equals to zero as if the corresponding bidirectional link becomes a unidirectional link. Clearly, those links with equal rates on both directions will have a net rate of 0. Also, note that this link rate mapping preserves the optimality of $(\hat{\mathbf{x}}^*, \hat{\mathbf{r}}^*)$ when applied to *any* optimal solution $(\mathbf{x}^*, \mathbf{r}^*)$ to problem (1) which is not necessarily given by the DTBP algorithm.

²More precisely, $(\mathbf{x}^*, \mathbf{r}^*)$ is within $O(1/K)$ of the optimal solution of (12). Nevertheless, we still call $(\mathbf{x}^*, \mathbf{r}^*)$ an optimal solution of (12) since the converge can be precise if a diminishing step-size (in our notation, if K grows linearly with time t) is used.

The Steady-State Behavior of the DTBP Algorithm:

Previous works (e.g. [14]) also show the stability of the DTBP algorithm. In other words, the Markov chain with the queue-lengths $\mathbf{P} = (P_i^d)_{i,d}$ being its states is positive recurrent and ergodic under the DTBP algorithm. We use $\pi(\mathbf{P})$ to denote the steady state distribution under the DTBP algorithm, i.e., the probability of the queue-lengths being \mathbf{P} is $\pi(\mathbf{P})$ after the convergence of the DTBP algorithm and $\sum_{\mathbf{P}} \pi(\mathbf{P}) = 1$. Also, for queue P_i^d , the following holds:

$$\lim_{t \rightarrow \infty} \frac{1}{t} \sum_{\tau=0}^t \mathbb{E}(P_i^d[\tau]) = \sum_{\mathbf{P}} P_i^d \pi(\mathbf{P}), \quad (20)$$

and we use

$$\bar{P}_i^{*d} = \sum_{\mathbf{P}} P_i^d \pi(\mathbf{P}) \quad (21)$$

to denote the optimal average queue-length of queue P_i^d .

Intuitively, the link rate assignment equation (10) of the DTBP algorithm implies that if there is a positive net flow rate from node i to node j for destination d , then the average queue-lengths should be *strictly decreasing*, i.e., $\bar{P}_i^{*d} > \bar{P}_j^{*d}$, since i can send a packet to j only when $P_i^d[t] > P_j^d[t]$. In certain simple topologies, this statement can be proven rigorously. For a tandem network (see Fig. 3 for an example) with unit link capacities and one end-to-end flow, we have the following results:

Lemma 1: In such a tandem network with no more than 3 hops, the average queue-length of the DTBP algorithm is strictly decreasing from the source to the destination for any arrival rate $a \in (0, 1)$ under a Bernoulli arrival process.

Proof: This lemma can be proven by constructing the queue-evolution Markov chain and solve for its steady distribution. The detailed proof is provided in [16]. ■

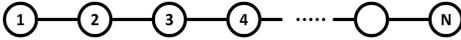


Fig. 3. A tandem network with $N - 1$ hops.

The method used in the proof of Lemma 1 can be extended to a k -hop tandem network with a Bernoulli arrival. However the analysis becomes cumbersome as k increases. Thus we take a different approach to extend the above result to a more general k -hop tandem network.

Lemma 2: In a tandem network under our model with unit link capacities and one end-to-end flow with a Bernoulli arrival with rate $a \in (\frac{1}{2}, 1)$, the average queue-length \bar{P}_i^{*d} is strictly decreasing from the source to the destination under DTBP.

Proof: The detailed proof is provided in [16]. ■

Due to the complex interactions in the dynamics of the queue-length in the neighboring nodes in a general network, a rigorous proof of a generalization of Lemma 2 remains an open research problem. However it can be observed from numerical studies (see [16] for more examples) of more general networks that, under the DTBP algorithm, the average queue-lengths are strictly decreasing over links with a positive net flow rate.

With such results and observations, we present the following assumption on the DTBP algorithm:

Assumption 2: Under our system model, the DTBP algorithm converges to a optimal solution of (1) $\{\mathbf{X}^*[t], \mathbf{R}^*[t]\}$

and the corresponding queue evolution $\mathbf{P}^*[t]$ as defined in (8). The optimal solutions satisfy the following property:

$$\text{If } r_{ij}^{*d} > r_{ji}^{*d} \text{ then } \bar{P}_i^{*d} > \bar{P}_j^{*d}, \forall (i, j) \in \mathcal{L}, d \in \mathcal{D},$$

where r_{ij}^{*d} and r_{ji}^{*d} are defined as in (7), and \bar{P}_i^{*d} and \bar{P}_j^{*d} are defined as in (21).

Remark: Note that the converse of Assumption 2 is not necessarily true, i.e., $\bar{P}_i^{*d} > \bar{P}_j^{*d}$ does not necessarily imply that $r_{ij}^{*d} > r_{ji}^{*d}$, which can be disproved by counter examples.

Loop-Free Route Construction: Consider the network $\mathcal{G} = (\mathcal{N}, \mathcal{L}, \mathbf{c})$. For each commodity d , we define a subgraph $\hat{\mathcal{G}}^d = (\hat{\mathcal{N}}^d, \hat{\mathcal{L}}^d, \hat{\mathbf{c}}^d)$ as:

$$\begin{aligned} \hat{\mathcal{N}}^d &= \mathcal{N}; \\ \hat{\mathcal{L}}^d &= \mathcal{L} \setminus \{(i, j) \in \mathcal{L} : \hat{r}_{ij}^{*d} = 0\}; \\ \hat{\mathbf{c}}_{ij}^d &= \begin{cases} 0 & \text{if } \langle i, j \rangle \notin \hat{\mathcal{L}}^d \text{ and } \langle j, i \rangle \notin \hat{\mathcal{L}}^d; \\ c_{ij} & \text{otherwise.} \end{cases} \end{aligned}$$

This $\hat{\mathcal{G}}^d$ is the restricted network topology that commodity d sees after applying the delay-aware link rate assignment.

Then we have the following proposition:

Proposition 2: Under Assumption 2, given a network $\mathcal{G} = (\mathcal{N}, \mathcal{L}, \mathbf{c})$ and its optimal solution $(\mathbf{x}^*, \mathbf{r}^*)$ given by the back-pressure algorithm to problem (12), the subgraph $\hat{\mathcal{G}}^d$ defined as above is loop-free $\forall d \in \mathcal{D}$.

Proof: Assume $\hat{\mathcal{G}}^d$ has a cycle that is composed of links $\{(n_{c_1}, n_{c_2}), (n_{c_2}, n_{c_3}), \dots, (n_{c_{k-1}}, n_{c_k}), (n_{c_k}, n_{c_1})\} \subseteq \hat{\mathcal{L}}^d$. By the definition of subgraph $\hat{\mathcal{G}}^d$, we can assume without loss of generality that $\hat{r}_{c_i c_{i+1}}^{*d} > 0$ for $0 \leq i \leq k-1$ and $\hat{r}_{c_k c_1}^{*d} > 0$. Then by Assumption 2, we have $\bar{P}_{c_1}^{*d} > \bar{P}_{c_2}^{*d} > \dots > \bar{P}_{c_k}^{*d} > \bar{P}_{c_1}^{*d}$, which is impossible. Therefore $\hat{\mathcal{G}}^d$ is loop-free. ■

Remark: As mentioned in the proposition itself, the loop-freeness holds only when the delay-aware link rate assignment is applied to the optimal solution given by DTBP.

Example: We use the topology shown in Fig. 4(a) as an example to illustrate how the delay-aware link rate assignment works with the optimal solution given by DTBP. Each link in the network is assumed to have unit capacity. Node 1 is the source and Node 3 is the destination.

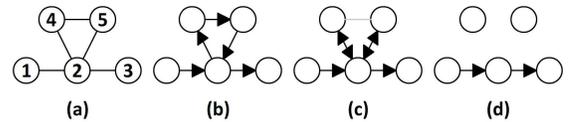


Fig. 4. An example to illustrate the DTBP solution with delay-aware link rate assignment.

Shown in Fig. 4(b) is a possible optimal solution of the problem, with $r_{12} = r_{23} = 1$ and $r_{24} = r_{45} = r_{52} = r$, where $r \in (0, 1)$. Note that there exists a loop ($2 \rightarrow 4 \rightarrow 5 \rightarrow 2$) in this optimal solution. However, analyzing the corresponding Markov chain shows that the DTBP algorithm can never converge to such an optimal solution. In fact, the DTBP converges to a solution shown in Fig. 4(c), where $r_{12} = r_{23} = 1$, link (2, 4) and (2, 5) have equal and non-zero transmission rate in either direction, and there is no transmission on link (4, 5). Note that this confirms Assumption 2 in this particular topology, and also confirms that the optimal solutions given

by DTBP is a subset of all optimal solutions with a special structure.

If we were to apply the delay-aware link rate assignment to the optimal solutions shown in Fig. 4(b), it can be verified that the loop (2 → 4 → 5 → 2) still exists in the resulting topology. On the other hand, Fig. 4(d) shows the resulting topology when the delay-aware link rate assignment being applied to the optimal solution given by DTBP algorithm, which is a loop-free route.

B. Delay-aware scheduler design

Using the delay-aware routing of Section IV-A, utility maximization can be achieved while avoiding loops, which leads to lower average end-to-end delays. Here, we seek to improve the delay performance further through delay-aware scheduling at each link. As argued in Section III-B, significant per hop delay improvement can be achieved by reshaping the service distribution through regulating the scheduler operation. In previous works [17], the notion of regulators were introduced to help stabilize the network with fixed single-path routing. Adopting a similar concept, we propose a delay-aware scheduler with regulated service that stabilizes $(\mathcal{N}, \mathcal{L}, c)$ when there exists a loop-free $\mathbf{R}[t]$ such that $\mathbf{X}[t]$ satisfies $\mathbf{x} \in \text{int}(\Lambda(\mathcal{N}, \mathcal{L}, c))$. We assume that the arrivals take place after the service in each time slot, i.e., the arrivals cannot be served in the arriving time slot. We further assume that from now on all links have the same link capacity denoted by c .

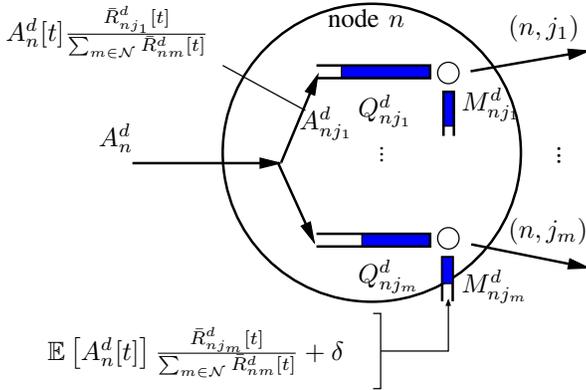


Fig. 5. Delay-Aware Scheduler Structure

Definition 4: (Delay-aware scheduler with regulated service) The delay-aware scheduler consists of *per-neighbor-per-commodity* queues, i.e., the scheduler at node n maintains queues Q_{nj}^d for each next-hop neighbor j and each commodity d . The servers apply regulated service discipline to shape the traffic. Let the service and departure of the server be $S(t)$ and $D(t)$, respectively.

Let $A_n^d[t]$ denote the commodity d packets entering node n in slot t . Mathematically,

$$A_n^d[t] = X_n^d[t] \mathbb{1}_{\{n \in \mathcal{N}_{source}^d\}} + \sum_{k \in \mathcal{N}} D_{kn}^d[t],$$

where \mathcal{N}_{source}^d denotes the source node set with respect to commodity d . Let $A_{nj}^d[t]$ denote the number of packets of commodity d arriving at queue Q_{nj}^d in slot t and $A_n^d[t] =$

$\sum_{j \in \mathcal{N}} A_{nj}^d[t]$. The queue dynamics for each queue is given as:

$$Q_{nj}^d[t+1] = \left(Q_{nj}^d[t] - D_{r,nj}^d[t] \right)^+ + A_{nj}^d[t], \quad \forall n \neq d, \quad (22)$$

The server of the delay-aware scheduler operates with a token-based service discipline which consists of three components:

- **Arrival Splitting:**

Each commodity d packet arriving at node n goes into the per-neighbor-per-commodity queue Q_{nj}^d with probability $\frac{\bar{R}_{nj}^d[t]}{\sum_{m \in \mathcal{N}} \bar{R}_{nm}^d[t]}$, resulting in

$$\mathbb{E} \left[A_{nj}^d[t] \mid A_n^d[t] \right] = A_n^d[t] \frac{\bar{R}_{nj}^d[t]}{\sum_{m \in \mathcal{N}} \bar{R}_{nm}^d[t]}, \quad (23)$$

where $\bar{R}_{nj}^d[t] = \frac{1}{t} \sum_{\tau=0}^{t-1} R_{nj}^d[\tau]$. This splitting preserves the same mean link rates as the original $\mathbf{R}[t]$. As an example, the arrival splitting can be done using a token-based scheme.

- **Token Generation:**

For each link $\langle i, j \rangle \in \mathcal{L}$, the system maintains *token counters* m_{nj}^d for each regulator queue Q_{nj}^d as shown in Fig. 5. As the input to the delay-aware scheduler is loop-free, the transmissions can only be unidirectional for certain commodity d over link (i, j) . Hence m_{nj}^d and m_{jn}^d cannot be non-zero simultaneously. Then we define $M_{nj}^d = m_{nj}^d + m_{jn}^d$. The token arrives at each counter with rate

$$S_{nj}^d = \mathbb{E} \left[A_n^d[t] \right] \frac{\bar{R}_{nj}^d[t]}{\sum_{m \in \mathcal{N}} \bar{R}_{nm}^d[t]} + \delta, \quad (24)$$

where $\delta > 0$. Note that the token arrival rates are enlarged by δ which is allowed by the excess capacity from the utility maximizing solution for $(\mathcal{N}, \mathcal{L}, c - \epsilon)$.

- **Service:**

In each time slot t , the token-based server chooses a winner commodity d_{nj}^* with a non-zero backlog for each link (n, j) as follows:

$$d_{nj}^*[t] = \begin{cases} \operatorname{argmax}_d (M_{nj}^d[t] - c), & \text{if } \max_d [M_{nj}^d[t] - c] > 0; \\ \emptyset, & \text{otherwise.} \end{cases} \quad (25)$$

The server serves the commodity with the largest token count that exceeds the link capacity c . Note that the link capacity c is shared by transmissions on link (n, j) in both directions. Then

$$D_{nj}^d[t] = \begin{cases} c, & \text{if } d = d_{nj}^*; \\ 0, & \text{if } d \neq d_{nj}^*. \end{cases} \quad (26)$$

In case of insufficient packets in queue, the server sends out dummy packets to ensure a departure of c packets. After service, the token counter M_{nj}^d for the commodity d_{nj}^* is decreased by c . \diamond

The above algorithm does not require the knowledge of $\mathbf{R}[t]$ as long as $\bar{\mathbf{R}}[t] := (\bar{R}_{nj}^d[t])_{n,j,d}$ is provided. Next we define the network capacity region for multihop networks.

Definition 5: (Network capacity region) The *network capacity region* is $\Lambda(\mathcal{N}, \mathcal{L}, \mathbf{c}) = \text{Co}\{\mathbf{x}\}$, where \mathbf{x} is the exogenous arrival rate vector such that $\mathbf{x} \succeq 0$ and such that there exists a link rate vector \mathbf{r} satisfying

$$\begin{aligned} \mathbf{r} &\succeq 0, \\ r_{nj}^d &= 0, \forall (n, j) \notin \mathcal{L}, \\ r_{nn}^d &= 0, \forall n \in \mathcal{N}, \\ r_{nj}^d &= 0, \forall n = d, \\ x_n^d &\leq \sum_j r_{nj}^d - \sum_n r_{mn}^d, \forall n \neq d, \\ \sum_d r_{nj}^d + \sum_d r_{jn}^d &\leq c_{nj}, \forall (n, j) \in \mathcal{L}. \quad \diamond \end{aligned}$$

Notice that the solution $(\mathbf{X}[t], \mathbf{R}[t])$ is *feasible* if (\mathbf{x}, \mathbf{r}) satisfies Definition 5 such that $\mathbf{x} \in \Lambda(\mathcal{N}, \mathcal{L}, \mathbf{c})$, where $\mathbf{x} = \lim_{t \rightarrow \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbf{X}[\tau]$, $\mathbf{r} = \lim_{t \rightarrow \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbf{R}[\tau]$.

Based on the defined network capacity region, we now introduce the following proposition:

Proposition 3: The delay-aware scheduler operating in the network $(\mathcal{N}, \mathcal{L}, \mathbf{c})$ stabilizes the queues $(Q_{nj}^d)_{n,j,d}$, if there exists any loop-free solution $(\mathbf{X}[t], \mathbf{R}[t])$ such that $\mathbf{x} \in \Lambda(\mathcal{N}, \mathcal{L}, \mathbf{c} - \epsilon)$, $\forall \epsilon > 0$.

Proof: The detailed proof is provided in [16]. Due to the limited space, here we briefly outline the proof. We first define the *token count process*, which describes the token counter behavior. We prove the lemma on the fact that the token count process keeps track of the assigned token generation rate and the deviation is always bounded. From the design of the delay-aware scheduler, the service rate of the next-hop neighbor is larger than its previous-hop neighbor by δ . And this δ is allowed by the excess capacity when the solution of the network $(\mathcal{N}, \mathcal{L}, \mathbf{c} - \epsilon)$ is used as input to control the network $(\mathcal{N}, \mathcal{L}, \mathbf{c})$. We make use of the service rate difference to prove that the T -step drift of the Lyapunov function is negative, where T is a finite value. Then applying the previous results in [14], we manage to prove the proposition. ■

Remark: To stabilize the network, the proposed delay-aware scheduler benefits from the special properties of the token count process. In fact, we have proved that the delay-aware scheduler structure can be extended to utilize a general service process that satisfies certain regulation constraints and still achieves network stability. The existence and characterization of other delay-aware service processes remains an open research problem.

C. Delay-aware cross-layer policy

The approaches we proposed in Section IV-A and IV-B can be used in the multi-layer architecture of Section III-C. To attain a utility maximization solution that reduces the expected end-to-end delay, we incorporate back-pressure solution to problem (1) together with delay-aware routing and delay-aware scheduling in the policy design. In the following policy, we call the queue length seen by DTBP the price, which is

a value reflecting the queue length price instead of being an actual queue.

Delay-Aware Cross-Layer Policy

- The first layer runs DTBP policy (see Section II-B) to generate $(\tilde{\mathbf{X}}[t], \tilde{\mathbf{R}}[t])$ for $(\mathcal{N}, \mathcal{L}, \mathbf{c} - \epsilon)$ in a virtual implementation using counters for prices.
- The second layer performs the delay-aware link rate mapping from the first layer and generates $\hat{\mathbf{R}}[t]$ as in (17) with $\mathbf{R}[t] = \tilde{\mathbf{R}}[t]$.
- The third layer uses delay-aware schedulers with regulated service in the network $(\mathcal{N}, \mathcal{L}, \mathbf{c})$. Solution $\mathbf{X}[t] = \tilde{\mathbf{X}}[t]$ is received from the first layer. Solution $\bar{\mathbf{R}}[t] = \hat{\mathbf{R}}[t]$ is received from the second layer. Then the delay-aware scheduling is performed as described in Section IV-B which determines the actual queueing dynamics.

This cross-layer policy is an online policy that starts to function while the DTBP is converging. All layers run different algorithms in parallel and dynamic solution updates can be performed among layers.

The cross-layer policy inherits optimality (Proposition 1) and loop-freeness (Proposition 2) characteristics of the routing component, and the stability (Proposition 3) of the delay-aware scheduling component, which lead to the following fundamental result:

Theorem 1: Delay-aware cross-layer policy results in loop-free and stable rate assignments that arbitrarily approaches maximum utility solutions for a network $(\mathcal{N}, \mathcal{L}, \mathbf{c})$.

V. NUMERICAL RESULTS

In this section, we present numerical results for our proposed delay-aware cross-layer policy. We compare the delay performance of our policy with the back-pressure policy, which has long term optimality guarantees, and the more recent min-resource algorithm (see Section II-B), which has superior routing characteristics over back-pressure policy. As a representative topology, simulations are carried out in a 6×6 grid network. Simulation duration is 30000 time slots. Moving average method is used to update $\hat{\mathbf{R}}[t]$ as follows:

$$\hat{R}_{nj}^d[t] = \left[\sum_{\tau=t_0-W}^{t_0} \frac{R_{nj}^d[\tau] - R_{jn}^d[\tau]}{W} \right]^+, \quad t_0 \leq t < t_0 + T,$$

where W is the window size and T is update period. W and T are both set to 5000. We assume that all links have unit capacity. The K parameter for the flow controller of the back-pressure policy is chosen to be 200. We implement min-resource algorithm as described in Section II-B with $M = 3$. For fair comparison, all queues are implemented as per-commodity queues.

The simulated scenario has parallel traffics with two commodities. For better demonstration, the routing information is embedded in delay distribution graph. Since the source-destination pairs are symmetric, results are given for one commodity only. Squares and circles indicate source and destination locations for commodity 1 and 2, respectively, both flowing from top to bottom. As is shown in Fig. 6(a), the

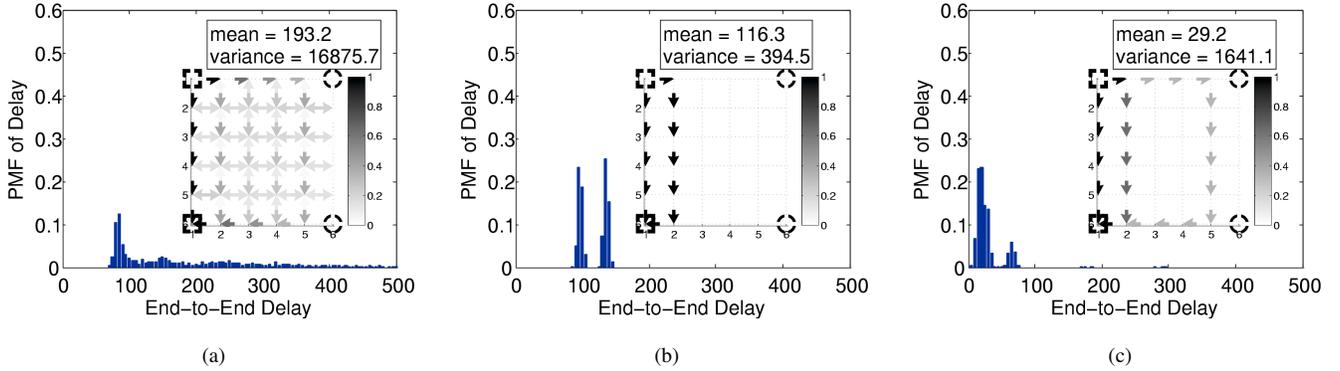


Fig. 6. (a) Back-Pressure Policy (b) Min-Resource Algorithm (c) Delay-Aware Cross-Layer Policy

back-pressure policy results in loops. The delay distribution is heavy-tailed which also indicates that packets may experience looping. Fig. 6(b) demonstrates that the min-resource algorithm utilizes minimal number of links, thus attaining shortest paths from source to destination for each commodity. Since loops are avoided, the delay distribution is improved with the delay peaks moved towards lower values. Fig. 6(c) shows the network topology for our policy where the selected paths are loop-free. From the delay distribution, one observes that the majority of packets follow the shortest path and the others are routed over longer paths. The mean delay performance is improved over other two solutions.

We further study the effect of the design parameter K used in the DTBP policy. It is known that the allocated flow rates approach the optimal rate x^* as $K \rightarrow \infty$. However, this also increases the equilibrium queue lengths in the DTBP and causes delay degradation ([18], [4], etc.)

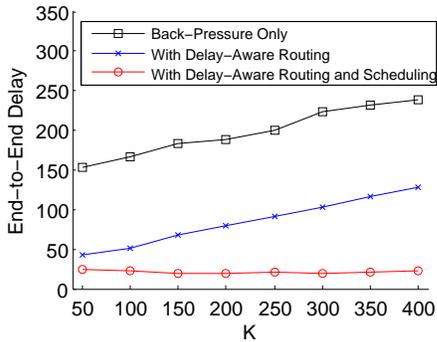


Fig. 7. End-to-End Delay under Different K Values

The delay-aware cross-layer policy offers a solution to this dilemma. In multi-layer architecture, the delay-aware cross-layer policy keeps real layer queue length separated from back-pressure layer queue length. Then the dependence on K is relaxed. Fig. 7 shows that in the simulated scenario, the sensitivity of the delay on K is significantly reduced with delay-aware cross-layer policy. As expected, the delay under back-pressure policy grows as K increases. In this figure, we also show that the end-to-end delay can be reduced by eliminating loops alone. However, further delay improvement can be achieved by utilizing the delay-aware scheduling with delay-aware routing.

VI. CONCLUSION

In this work, we exposed the delay deficiencies of many long-term optimal policies when operated in multi-hop networks, and identified several unexploited design choices within the routing and scheduling strategy spaces that yield drastic delay improvements. In particular, we exploit: the flexibility in link rate assignments to eliminate loops in routing; and the service shaping opportunities in scheduling to reduce queue-sizes, while preserving the long-term optimality characteristics.

Along with these algorithms, we also present a generic delay-aware design framework that can be used to develop other long-term optimal algorithms with favorable delay characteristics. This framework provides a unique and methodical approach to the design of modular solutions where end-to-end delay characteristics can be improved without sacrificing from long-term optimality. As such, our framework offers a promising direction for tackling the challenging problem of delay-aware algorithm design for wireless multi-hop networks, as well. To that end, in our future work, we will extend the insights and techniques developed in this work to the wireless domain, while accounting for the intrinsic challenges associated with the interference-limited nature of wireless communications.

REFERENCES

- [1] 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. 36, pp. 1936–1948, December 1992.
- [2] X. Lin and N. Shroff, "Joint rate control and scheduling in multihop wireless networks," in *Proceedings of IEEE Conference on Decision and Control*, Paradise Island, Bahamas, December 2004.
- [3] A. Stolyar, "Maximizing queueing network utility subject to stability: Greedy primal-dual algorithm," *Queueing Systems*, vol. 50, no. 4, pp. 401–457, 2005.
- [4] M. Neely, E. Modiano, and C. Li, "Fairness and optimal stochastic control for heterogeneous networks," in *Proceedings of IEEE Infocom*, Miami, FL, March 2005, pp. 1723–1734.
- [5] A. Eryilmaz and R. Srikant, "Joint congestion control, routing and mac for stability and fairness in wireless networks," *IEEE Journal on Selected Areas in Communications, special issue on Nonlinear Optimization of Communication Systems*, vol. 14, pp. 1514–1524, August 2006.
- [6] X. Lin, N. B. Shroff, and R. Srikant, "A tutorial on cross-layer optimization in wireless networks," *IEEE Journal on Selected Areas in Communications, special issue on Nonlinear Optimization of Communication Systems*, vol. 14, no. 8, Aug 2006.
- [7] L. Georgiadis, M. Neely, and L. Tassiulas, "Resource allocation and cross-layer control in wireless networks," vol. 1, no. 1, pp. 1–144, 2006.

- [8] S. Shakkottai and R. Srikant, "Network optimization and control," vol. 2, 2007.
- [9] L. Ying, S. Shakkottai, and A. Reddy, "On combining shortest-path and back-pressure routing over multihop wireless networks," in *Proceedings of IEEE Infocom*, Rio de Janeiro, Brazil, April 2009, pp. 1674–1682.
- [10] L. Bui, R. Srikant, and A. Stolyar, "Novel architectures and algorithms for delay reduction in back-pressure scheduling and routing," in *Proceedings of IEEE Infocom*, Rio de Janeiro, Brazil, April 2009, pp. 2936–2940.
- [11] G. Gupta and N. Shroff, "Delay analysis for multi-hop wireless networks," in *INFOCOM 2009. The 28th Conference on Computer Communications. IEEE*, April 2009, pp. 2356–2364.
- [12] K. Kar, X. Luo, and S. Sarkar, "Delay guarantees for throughput-optimal wireless link scheduling," in *INFOCOM 2009. The 28th Conference on Computer Communications. IEEE*, April 2009, pp. 2331–2339.
- [13] M. Neely, "Delay analysis for maximal scheduling in wireless networks with bursty traffic," in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, April 2008, pp. 6–10.
- [14] —, "Dynamic power allocation and routing for satellite and wireless networks with time varying channels," Ph.D. dissertation, Massachusetts Institute of Technology, LIDS, 2003.
- [15] L. Kleinrock, *Queueing Systems, Volume 2: Computer Applications*. New York: Wiley-Interscience, 1975.
- [16] H. Xiong, R. Li, A. Eryilmaz, and E. Ekici, "Delay-aware cross-layer design for network utility maximization in multi-hop networks," 2010, technical Report, available online at <http://www.ece.osu.edu/~li/xiolieryeki10report.pdf>.
- [17] X. Wu and R. Srikant, "Regulated maximal matching: A distributed scheduling algorithm for multi-hop wireless networks with node-exclusive spectrum sharing," in *Proceedings of IEEE Conference on Decision and Control*, 2005.
- [18] A. Eryilmaz and R. Srikant, "Fair resource allocation in wireless networks using queue-length based scheduling and congestion control," in *Proceedings of IEEE Infocom*, vol. 3, Miami, FL, March 2005, pp. 1794–1803.

APPENDIX

A. Proof of Proposition 1

Since $\hat{\mathbf{x}}^* = \mathbf{x}^*$, the optimal value of the objective function (12) is not affected. Thus, to show that $(\hat{\mathbf{x}}^*, \hat{\mathbf{r}}^*)$ is also an optimal solution to (12), it is sufficient to show that the pair $(\hat{\mathbf{x}}^*, \hat{\mathbf{r}}^*)$ satisfies the constraints (13)–(16).

- Verifying Equation (13) and (14): $\hat{x}_s^{*d} \geq 0$ because $x_s^{*d} \geq 0$ for all $s, d \in \mathcal{D}$; $\hat{r}_{ij}^{*d} \geq 0$ for all $(i, j) \in \mathcal{L}$ and $d \in \mathcal{D}$ by its definition in Equations (18) and (19).

- Verifying Equation (15): By the definition of \hat{r}_{ij}^{*d} and the fact that $r_{ij}^{*d} \geq 0, \forall (i, j) \in \mathcal{L}, \forall d \in \mathcal{D}$, we have $\hat{r}_{ij}^{*d} \leq r_{ij}^{*d}, \forall (i, j) \in \mathcal{L}, \forall d \in \mathcal{D}$. Then $\sum_d \hat{r}_{ij}^{*d} \leq \sum_d r_{ij}^{*d}$. So $\sum_d \hat{r}_{ij}^{*d} \leq c_{ij}$.

- Verifying Equation (16): Notice that for any real number Z , $(Z)^+ = \frac{1}{2}(Z + |Z|)$, Equations (18) and (19) can be rewritten as: $\hat{r}_{ij}^{*d} = ((r_{ij}^{*d} - r_{ji}^{*d}) + |r_{ij}^{*d} - r_{ji}^{*d}|)/2$ and $\hat{r}_{ji}^{*d} = ((r_{ji}^{*d} - r_{ij}^{*d}) + |r_{ji}^{*d} - r_{ij}^{*d}|)/2$. Subtracting these two equations, we have $\hat{r}_{ij}^{*d} - \hat{r}_{ji}^{*d} = r_{ij}^{*d} - r_{ji}^{*d}$. Thus, we have the following

$$\begin{aligned}
& \hat{x}_n^{*d} + \sum_{\langle m, i \rangle \in \mathcal{L}} \hat{r}_{mi}^{*d} - \sum_{\langle i, j \rangle \in \mathcal{L}} \hat{r}_{ij}^{*d} \\
&= \hat{x}_i^{*d} + \sum_j (\hat{r}_{ij}^{*d} - \hat{r}_{ji}^{*d}) \\
&= x_i^{*d} + \sum_j (r_{ij}^{*d} - r_{ji}^{*d}) \leq 0.
\end{aligned}$$

Therefore, $(\hat{\mathbf{x}}^*, \hat{\mathbf{r}}^*)$ is also an optimal solution to (12).