Scheduling for End-to-End Deadline-Constrained Traffic with Reliability Requirements in Multi-Hop Networks

Ruogu Li and Atilla Eryilmaz

***Abstract*—We attack the challenging problem of designing a scheduling policy for end-to-end deadline-constrained traffic with reliability requirements in a multi-hop network. It is well-known that the end-to-end delay performance for a multi-hop flow has a complex dependence on the high-order statistics of the arrival process and the algorithm itself. Thus, neither the earlier optimization based approaches that aim to meet the long-term throughput demands, nor the solutions that focus on a similar problem for single-hop flows directly apply. Moreover, a dynamic programming-based approach becomes intractable for such multi-time scale Quality-of-Service(QoS)-constrained traffic in a multi-hop environment. This motivates us in this work to develop a useful architecture that enables us to exploit the degree of freedom in choosing appropriate service discipline. Based on the new architecture, we propose three different approaches, each leading to an original algorithm. We study the performance of these algorithms in different scenarios to show both optimality characteristics and to demonstrate the favorable service discipline characteristics they possess. We provide extensive numerical results to compare the performance of all of these solutions to throughput-optimal back-pressure-type schedulers and to longest waiting time based schedulers that have provably optimal asymptotic performance characteristics. Our results reveal that the dynamic choice of service discipline of our proposed solutions yields substantial performance improvements compared to both of these types of traditional solutions under non-asymptotic conditions.**

***Index Terms*—End-to-End Delay, Quality of Service, Service Discipline Choice, Scheduling, Earliest Deadline First**

I. Introduction

With the growth of wide area communication networks, we have witnessed the increasing deployment of a variety of real-time applications over the last few years, especially streaming multi-media applications. These real-time flows often have Quality-of-Service (QoS) requirements such as end-to-end deadline constraints and successful packet delivery requirements while traversing a multi-hop network. However, these QoS requirements, such as end-to-end delay performance, have a complex dependence on the higher-order statistics of the arrival process. Thus, the canonical optimization based approaches that aim to meet the long-term throughput requirements (e.g., [1], [2], [3], [4], [5]) do not apply. Subsequently, valuable efforts have been exerted for the design of algorithms with low end-to-end delay performance (e.g., [6], [7]) and the derivation of more fundamental bounds on the delay performance (e.g., [8]). However these works do not consider the scenario in which a strict per-packet delay bound exists.

Recently, a number of works (e.g., [9], [10], [11], [12]) have modeled and studied the scenario that the flow has a fixed per-packet delay constraint and a delivery ratio requirement under this delay constraint. Also, [13] presents some interesting results on developing an algorithm achieving constant delay bounds. However, all these works are based on a single-hop flow model, where the delay performance is more tractable.

One of many innovations of these works is a modeling in which a fixed number of consecutive time slots are grouped into frames. Packets are assumed to arrive in the beginning of the frame and expire at the end of the frame. In this work, we extend the model in two critical aspects: we extend the network and traffic model to multi-hop networks with multi-hop flows; and we allow a packet to spend multiple frames in the network before its expiry. These extensions significantly complicate the problem, and the approaches in the aforementioned works do not directly apply.

Rather than pursuing these traditional means, in this work, we propose a novel way to model this problem, that enables us to open up a new degree of freedom to exploit, namely the service discipline, in a tractable framework. Here we present a simple example to illustrate the delivery ratio gain we can get from carefully choosing the service discipline:

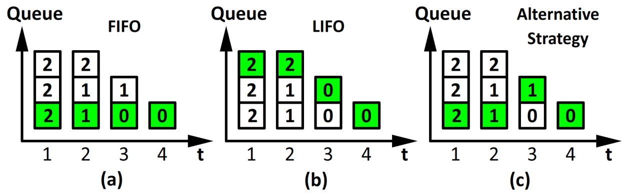


Fig. 1. A example showing the gain we can get by exploiting the service discipline

Example: Consider a one-hop flow over a unit-capacity link with a periodic arrival process of $3, 1, 0, 0$ packets in four consecutive time-slots. Each packet has an end-to-end deadline of 2 time-slots. Fig. 1 shows different service disciplines that can be used to serve this flow: First-In-First-Out (FIFO), Last-In-First-Out (LIFO) and an alternative strategy that serves packets more independently. The squares with numbers represent the packets and their remaining time before expiry. The shaded packets are the ones being served in that time slot. Both the FIFO discipline shown in Fig. 1(a) and the LIFO discipline shown in Fig. 1(b) serve 2 out of 4 packets successfully, while the service discipline shown in Fig. 1(c) serves 3 out of 4 packets, achieving a successful delivery ratio of 75%.

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

An earlier version of this work appeared in the Proceedings of Infocom, April, 2011.

This work was supported in part by Qatar National Research Fund (QNRF) under the National Research Priorities Program (NPRP) grant number NPRP 09-1168-2-455, DTRA Grant HDTRA 1-08-1-0016, and NSF Awards: CAREER-CNS-0953515 and CCF-0916664.

This example shows that to satisfy the QoS constraint that depends on the statistics of the arrival and the service process, the employed service discipline needs to be more flexible than a "fixed" strategy such as FIFO and LIFO. In this work, we aim to develop dynamic algorithms that automatically adapt the service discipline to the QoS requirements of the existing flows. Due to the complexity of the evolutions in a multi-hop network, the problem becomes intractable for the application of dynamic programming or opportunistic techniques used in previous works (e.g., [14], [15], [16], [17]). Thus new approaches must be used in solving this problem.

Our main contributions in addressing these challenges can be summarized as follows:

- We develop a model for deadline-constrained multi-hop traffic with delivery ratio requirements and propose a novel queueing architecture which enables us to exploit the new degree of freedom of choosing service discipline. This new model enables us to develop algorithms that utilize appropriate service disciplines to satisfy the delay constraint for different arrival process.
- Based on our model, we approach the problem in three different ways, namely an extension to a known optimal policy in a single-hop setup, the Lyapunov drift minimization approach and a pricing scheme based on a novel optimization formulation out of which three different reasonably practical algorithms are developed.
- We prove that two of the above approaches lead to algorithms which are optimal in the sense that they can support any arrival process whose delivery ratio requirements can be satisfied under single-hop, multiple flow scenario. We show their optimality in two extreme cases, namely when the deadline constraint is a single frame and when it is relaxed while the drop requirement is kept. We also demonstrate that the last approach indeed yields the adaptive service discipline strategy that has motivated this work.
- We perform extensive numerical studies to compare the performance of all of these solutions under non-asymptotic conditions. It is shown in the simulations that the proposed algorithms can significantly outperform existing throughput-optimal strategies as well as asymptotically delay optimal by dynamically selecting the proper service discipline.

The rest of our work is organized as follows. Section II introduces our system model and objective together with the description of existing algorithms that we use for comparison. We start to attack this problem in a simpler single-hop setup in Section III and then extend to the multi-hop setup in Section IV, where we present the three different approaches we used and the corresponding algorithms. The numerical results of our algorithms and comments on the results are presented in Section V. We make our concluding remarks as well as comments on possible future works in Section VI.

II. System Model

Network and Traffic Model: We consider a wired network $\mathcal{G} = (\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{L}$ denotes the set of links. We assume that the nodes operate in synchronized time slots with the capacity of each link l is normalized to one packet per slot. A set of delay-sensitive flows $\mathcal{F}$ traverses the network. Each flow f has a fixed source and a fixed destination, with a single fixed route of h^f hops connecting them. We denote the route for flow f by $\mathcal{R}^f$, which is a $|\mathcal{L}|$-dimensional vector such that $\mathcal{R}^f[l] = 1$ if link $l \in \mathcal{L}$ is on the route of flow f and 0 otherwise. We also use $l \in \mathcal{R}^f$ to denote that link l is on the route $\mathcal{R}^f$.

We call T consecutive time-slots a *frame* that captures the timescale at which new batches of packets arrive to the sources of the flows. We assume the exogenous arrivals only occur at the beginning of each frame. Also, all packets that are sent through a link $(i, j) \in \mathcal{L}$ during the k^{th} frame from node i are assumed to arrive at node j at the beginning of the $(k+1)^{st}$ frame. Flow f packets that arrive at the k^{th} frame are denoted by $A^f[k]$. We make the following assumption on the arrival process:

Assumption 1: The arrival process $A^f[k]$ is assumed to be stationary and ergodic with mean a^f for each flow f, and has a finite support for all $f \in \mathcal{F}$ and $k \in \mathbb{N}$, i.e., there exists $A < \infty$ such that $P(A^f[k] \leq A) = 1$ for all flows f in all frames k.

For each flow f, we number the links from 1 through h^f from the destination to the source along its route, i.e., the last hop to the destination is numbered by 1 and the first hop going out of the source is numbered by h^f. We use $f[i]$ to denote the link that is numbered i for flow f. Also, we use $N[l, f]$ to denote the downstream (next-hop) link of link l along the path of flow f, i.e., if link $l = f[i]$, then $N[l, f] = f[i-1]$. If the flow label f is apparent from the context, we briefly write $N[l, f]$ as $N[l]$. As an example, in Fig. 2, the link (b, c) can be denoted as link $f_1[2]$ and $f_2[3]$, while $N[(b, c), f_1]$ is link (c, d) and $N[(b, c), f_2]$ is link (c, f).

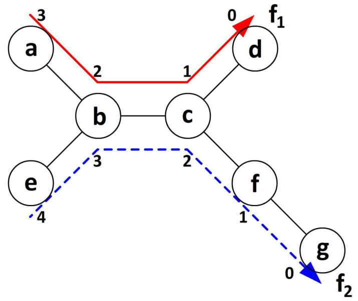


Fig. 2. A simple example: $h_{f_1} = 3$, $h_{f_2} = 4$, $(b, c) = f_1[2] = f_2[3]$

QoS Requirements: The QoS Requirements of each flow have a fixed end-to-end deadline for each packet and a corresponding delivery ratio requirement. We note that such requirements capture the demands of multi-media transmissions and real-time communication applications. Each packet for flow f has a fixed end-to-end delay constraint of $m^f T$ time-slots, or equivalently, m^f frames, after which the packet expires and becomes useless to the receiver. Without loss of generality, we assume $m^f \geq h^f$ for all f, since a packet for flow f needs at least h^f frames to reach its destination.

We assume flow f has a total end-to-end loss probability upper bound of $p^f \in (0, 1)$, i.e., its minimal delivery ratio requirement is $1-p^f$. The loss probability upper bound may also be divided among the links on its route accordingly, i.e., each link $f[i]$, $i=1,\dots,h^f$ has its loss probability upper bound of $p^f_{f[i]}$ which, for brevity, we write as $p^f_{[i]}$. For the convenience of our discussion, the probabilities $p^f_{[i]}$ are defined with respect to the arrival rate of flow f, i.e., we have $\sum_{i=1}^{h^f} p^f_{[i]} = p^f$ for all f. We also let $q^f_{[i]} \triangleq \sum_{k=i}^{h^f} p^f_{[k]}$, for all $f \in \mathcal{F}$ and $i=1,\dots,h^f$ be the total loss probability upper bound from the source of flow f up to and including link $f[i]$. We also write $q^f_{[i]}$ as q^f_l if $l=f[i]$.

For feasibility, we impose the following necessary condition on the arrival rate of the flows:

Assumption 2: Given the delivery ratio requirements $\{p^f\}_f$, the arrival rate of the flows satisfies

$$\sum_{f\in\mathcal{F}} a^f(1-q^f_l)\mathcal{R}^f[l] < T,\ \forall l \in \mathcal{L},$$

for some loss probability vector $\{p^f_{[i]}\}_{f,i}$ such that $\sum_{i=1}^{h^f} p^f_{[i]} = p^f$.

Assumption 2 suggests that there exists a (unknown) loss probability vector $\{p^f_{[i]}\}_{f,i}$ such that the corresponding minimal delivery rate requirements on each link l is less than its capacity.

Queueing Architecture: Each link l maintains one queue Q^f_l for each flow f that traverses it. The number of packets in Q^f_l at the beginning of a frame k is denoted by $Q^f_l[k]$. We also define $Q^f_{l,r}[k]$ to be the number of packets in $Q^f_l[k]$ that have r frames left before their expiry (we also refer to those packets as packets having r frames-to-go) at the beginning of frame k. Apparently, we have

$$\sum_{r=1}^{m^f} Q^f_{l,r}[k] = Q^f_l[k],\ \forall f\in\mathcal{F},\ \forall k.$$

In frame k, the service on link l for flow f is denoted by a vector $S^f_l[k] = \{S^f_{l,r}[k]\}_r$, where $S^f_{l,r}[k]$ is the number of packets being served that have r frames-to-go at the beginning of frame k. We have the following capacity constraint on the service:

$$\sum_f \sum_{r=1}^{m^f} S^f_{l,r}[k]\mathcal{R}^f[l] \le T,\ \forall l\in\mathcal{L}. \tag{1}$$

Also, the service should not exceed the number of available packets to prevent the under-utilization of the allocated resources. Thus, we need

$$S^f_{l,r}[k] \le Q^f_{l,r}[k],\ \forall l,r,k.$$

In each frame k, the policy may make service decisions for current frame, or also for future frames. In particular, we will present a policy making decision for both current and future frames based on current information in Section III-A.

At the end of the k^{th} frame, we denote $D^f_{l,r}[k]$ to be the number of unserved packets of flow f that had r frames-to-go at the beginning of frame k on link l. It, then, follows from above that $D^f_{l,r}[k] = Q^f_{l,r}[k] - S^f_{l,r}[k]$. By definition, those unserved packets that have r frames-to-go will have $(r-1)$ frames-to-go in the $(k+1)^{st}$ frame. Also, those served packets, upon the arrival to the next node, will have $(r-1)$ frames-to-go before their expiry. So we have the following queue evolution equation for each flow f on link $f[i]$:

$$Q^f_{[i],r}[k+1] = \begin{cases} A^f[k+1], & \text{if } i=h^f \text{ and } r=m^f \\ D^f_{[i],(r+1)}[k], & \text{if } i=h^f \text{ and } r<m^f \\ D^f_{[i],(r+1)}[k] + S^f_{[i+1],(r+1)}[k], & \text{otherwise} \end{cases}$$

which can be concisely written as

$$Q^f_{[i],r}[k+1] = D^f_{[i],(r+1)}[k] + S^f_{[i+1],(r+1)}[k], \tag{2}$$

where

$$\begin{aligned} D^f_{[i],(m^f+1)}[k] &= 0,\ \forall i=1,2,\dots,h^f,\ \forall k, \\ S^f_{[h^f+1],(r+1)}[k] &= 0,\ \forall r<m^f,\ \forall k, \\ S^f_{[h^f+1],(m^f+1)}[k] &= A^f[k],\ \forall k, \end{aligned}$$

and we also assume the system starts from a zero state, i.e. $D^f_{[l,r]}[0]=0$, $Q^f_{[l,r]}[0]=0$ for all links.

Note that the above queue evolution is different from a standard one in that at the end of each frame k, all packets $Q^f_{[i],r}[k]$ are either served or dropped. Thus the queue-length $Q^f_{[i],r}[k+1]$ does not depend on $Q^f_{[i],r}[k]$.

For flow f, the packets with 1 frame-to-go that are left unserved at link $f[i]$ or served at the upstream link $f[i+1]$ are placed into a *deficit queue* $Y^f_{[i]}$ at link $f[i]$, the length of which reflects the amount of missed service for flow f on link $f[i]$. The deficit queue $Y^f_{[i]}$ has a service of $a^f P^f_{[i]}[k]$ at frame k, where $P^f_{[i]}[k]$ is the loss probability upper bound to link $f[i]$ in frame k. It can be either predetermined and fixed for each frame k, or dynamically or probabilistically chosen as part of the control variable. $P^f_{[i]}[k]$ needs to satisfy the following constraints:

$$\begin{aligned} \lim_{K\to\infty}\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left[P^f_{[i]}[k]\right] &= p^f_{[i]},\ \forall f\in\mathcal{F}, \\ \sum_{i=1}^{h^f} P^f_{[i]}[k] &\le p^f,\ \forall f\in\mathcal{F},\ \forall k. \end{aligned}$$

These conditions ensure that the average total service rate of the deficit counters for flow f is $a^f p^f$. Thus, by stabilizing these deficit queues, we can satisfy the loss probability upper bounds. As described above, the deficit queue $Y^f_{[i]}$ evolves as

$$\begin{aligned} &Y^f_{[i]}[k+1] \\ &\quad= \left(Y^f_{[i]}[k] - a^f P^f_{[i]}[k]\right)^+ + D^f_{[i],1}[k] + S^f_{[i+1],1}[k], \end{aligned} \tag{3}$$

where $(x)^+ = x$ if $x>0$, and 0 otherwise, and $Y^f_l[0]=0$ for all links and flows.

The queueing architecture is illustrated in Fig. 3 for a flow with a deadline of 3 frames traversing a 2-hop network. It shows the "layered" queues $Q^f_{l,r}[k]$ in the link l for flow f. With this intuition of the queueing architecture in mind, we will also refer to $D^f_{l,r}[k]$ as the packets that are being dropped to a "lower layer queue".

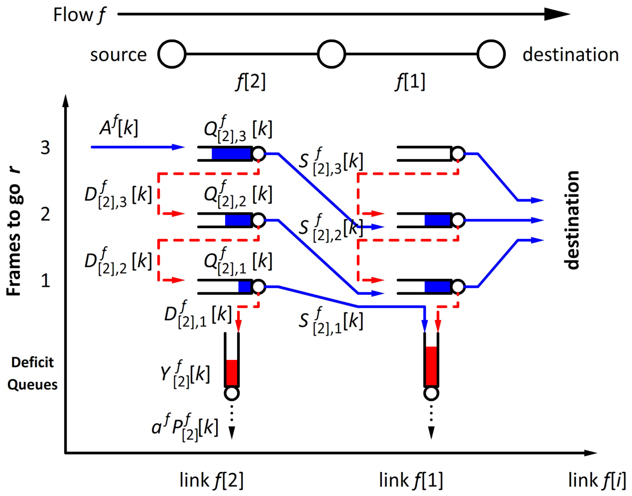


Fig. 3. An example showing the queueing architecture for the stochastic implementation in a 2-hop network with one flow whose deadline is 3 frames.

Remark: Although we refer to the evolution of $Q^f_{l,r}$ as queue evolution, $Q^f_{l,r}$'s are conceptual. In the actual implementation, the "queues" $Q^f_{l,r}$ for the packets that have different frames-to-go do not have to be separated. This can be achieved by using a single queue for each flow and tagging each packet with its corresponding remaining time before expiry. Also, the deficit queues can be implemented as counters rather than actual queues that store the expired packets. We also note that since the unserved packets at the end of each frame are flushed, the size of $Q^f_{l,r}$ cannot grow unboundedly. Thus the $Q^f_{l,r}$'s are guaranteed to be stable.

A. Objective

Note that in our system setup, the end-to-end delay requirements are satisfied by construction since each packet is dropped to one of the deficit queues if it is not delivered before its deadline. Thus, it is guaranteed that the packets received by the destination are delivered within their deadlines. Our remaining goal is to develop a scheduling policy that utilizes the freedom of choosing an appropriate service discipline that meets the delivery ratio requirements.

Let

$$s^f_{l,r} \triangleq \lim_{K\to\infty} \frac{1}{K} \sum_{k=1}^{K} S^f_{l,r}[k], \qquad (4)$$

$$d^f_{l,r} \triangleq \lim_{K\to\infty} \frac{1}{K} \sum_{k=1}^{K} D^f_{l,r}[k], \qquad (5)$$

respectively denote the service rate and the drop rate of the packets that have r frames-to-go for flow f at link l. The delivery ratio requirements can be written as

$$s^f_{[i+1],1} + d^f_{[i],1} \le p^f_{[i]} a^f, \forall f \in \mathcal{F}, i \in \{1, 2, \ldots, h^f\}. \qquad (6)$$

B. Background

In this subsection, we introduce three well-known scheduling policies, namely the Earliest Deadline First (EDF) policy (e.g., [18], [19]), the back-pressure (BP) policy (e.g. [1], [20], [21], [4]) and the Largest Weighted Delay First (LWDF) policy (see [22]). The first is a motivation for our first approach of the problem, while we will use the latter two as comparison benchmarks in our numerical studies.

The EDF Policy is a well known optimal policy in serving deadline-constrained traffic by a single server . It operates by always serving the packet that is closest to its expiry. The EDF Policy is optimal in the sense that if the deadline requirements of *all* incoming packets can be met by some policy, then the EDF Policy can satisfy all the deadline requirements. However, its performance in our setup, where a fraction of packet expiration is allowed, is unknown.

The BP policy is a scheduling/routing policy in multi-hop networks, which maintains a queue for each flow f in link l whose length at time t is denoted by $X^f_l[k]$. The scheduling policy chooses the flow $f^*_l[k]$ on a link l that satisfies

$$f^*_l[k] = \operatorname*{argmax}_{f\in\mathcal{F}} \left(X^f_l[k] - X^f_{N[l]}[k] \right)$$

and serves it with full capacity.

BP policy is known to be throughput-optimal, i.e., it stabilizes the network for any arrival rate vector that is supportable by any other policy. While the general BP algorithm can dynamically establish routes for the flows to increase the achieved throughput levels, such implementations exhibit poor end-to-end delay performance (e.g. see [6], [23]). Instead, we focus on the fixed route case, where the BP policy can be expected to have close-to-lower-bound average delay performance ([8]). Therefore, BP performance under fixed routes will serve as a benchmark for the performance of our proposed algorithms.

The LWDF policy keeps track of the waiting time of all the packets in the network. Let $W^f_l[k]$ denote the largest waiting time of the flow f packets at link l at the beginning of the k^{th} frame. Then, LWDF chooses to serve the packets with the largest waiting time of the flow $f_l[k]$ that maximizes the weighted delay $W^f_l[k]/\alpha^f$ over all flows, where $\{\alpha^f\}_f$ is a set of positive weighing coefficients associated the flows. It is shown in [22] that the LWDF policy is optimal in minimizing the asymptotic "tail" of the delay distribution, i.e., it solves the following minimization problem

$$\min_{f\in\mathcal{F}} \left[\alpha^f \lim_{m^f\to\infty} \frac{-1}{m^f} \log P(W^f > m^f) \right]$$

over all scheduling policies, where W^f denotes the steady-state end-to-end delay of flow f packets. However, the optimality of the LWDF policy is in the asymptotic sense where m^f goes to infinity, and its performance under a fixed finite m^f as in our model is unknown.

Notice that all the above policies are ignorant to the QoS requirements of the flows, and there exists no known strategy, to the best of our knowledge, that incorporates them into their operation. In fact, our work indicates that different queueing architectures and solution methodologies should be employed to account for these requirements.

III. Single-Hop Network Scenario

We study the problem by first considering a single-hop network scenario where all flows have the same deadline m. This setup is less complicated compared to our more general model, and the solution it yields can provide us some insights for solving the problem in the multiple-hop network setup. We note that by defining $m = \min_f m^f$, the results presented in this section can be extended to the case where the flows have different deadline requirements.

Throughout this section, we will omit the subscript indicating the links in our notations for brevity.

A. A Step beyond the EDF Policy

Motivated by the optimality of the EDF Policy and its ignorance to possible heterogeneous delivery ratio requirement, our first approach is to try to develop an extension for the EDF Policy by looking at its operation in the following novel perspective.

The queue evolution established by Equation (2) implies that the arrivals $A^f[k]$ in frame k are only to be served by $\boldsymbol{S}^f[k,m] \triangleq \{S^f_m[k], S^f_{(m-1)}[k+1], \ldots, S^f_1[k+m-1]\}$ before their expiry. Based on this observation, we consider a class of policies that make decision on $\boldsymbol{S}^f[k,m]$ for all flows f in each frame k based on current and past information, which, to be revealed later, contains the EDF Policy as a special case.

We define the group of variables $\boldsymbol{G}[k]$ observed by the policy at frame k as

$$\boldsymbol{G}[k] \triangleq \left\{ \begin{array}{c} A^f[k-m+1], \ldots, A^f[k], Y^f[k] \\ \boldsymbol{S}^f[k-m+1,m], \ldots, \boldsymbol{S}^f[k-1,m] \end{array} \right\}_f,$$

based on which the policy chooses $\{\boldsymbol{S}^f[k,m]\}_f$. Note that by iteratively substituting (2) into (3), we can express the value of all Q^f_r and the deficit counters in terms of A^f and S^f_r. In particular, we have

$$\begin{aligned} Y^f[k+1] &= (Y^f[k] - a^f p^f)^+ + A^f[k-m+1] \\ &\quad -\Sigma\boldsymbol{S}^f[k-m+1,m], \end{aligned} \tag{7}$$

where

$$\Sigma\boldsymbol{S}^f[k,m] = \sum_{i=0}^{m-1} S^f_{m-i}[k+i],$$

i.e., it is the total service allocated to $A^f[k]$ in the vector $\boldsymbol{S}[k,m]$. Thus, given the information in $\boldsymbol{G}[k]$, the value of $\{Y^f[k+1], \ldots, Y^f[k+m-1]\}_f$ can be calculated. It can be verified that under this particular class of policies $\boldsymbol{G}[k]$ evolves as a Markov chain. In Fig. 4 we show an example of the Markov Chain state $\boldsymbol{G}[k]$ and the service vector $\boldsymbol{S}^f[k,m]$ for the arrivals $A^f[k]$, where the variables within the thick boundary are in the state $\boldsymbol{G}[k]$, and the ones with red shading are in $\boldsymbol{S}^f[k,m]$.

We then approach the problem by defining a quadratic Lyapunov function and minimizing its drift. Note that this type of Lyapunov drift approach is extensively utilized in many other works (e.g., [20], [13], [24], [25]) in different contexts. Since stabilizing the deficit queues in our model implies that the delivery ratio constraint are satisfied, we define the following Lyapunov function:

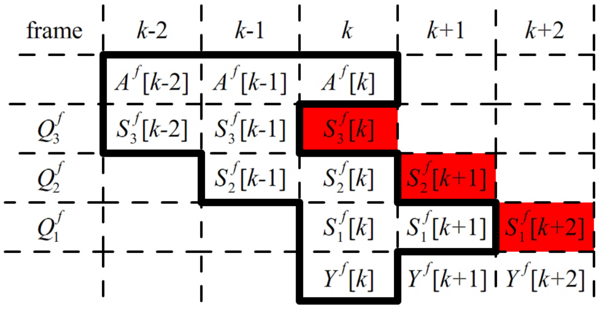


Fig. 4. An example showing $\boldsymbol{G}[k]$ and $\boldsymbol{S}^f[k,m]$ with $m=3$, where $\boldsymbol{G}[k]$ includes all the variables inside the thick edges, and $\boldsymbol{S}^f[k,m]$ are the states with red shading.

$$V(\boldsymbol{G}[k]) \triangleq \frac{1}{2}\sum_f (Y^f[k+m-1])^2, \tag{8}$$

and try to minimize its drift given by

$$\triangle V(\boldsymbol{G}[k]) \triangleq \mathbb{E}[V(\boldsymbol{G}[k+1]) - V(\boldsymbol{G}[k]) | \boldsymbol{G}[k]]$$

under the capacity constraint (1). The problem is formulated as:

$$\begin{aligned} \min_{\{\boldsymbol{S}^f[k,m]\}_f} \quad & \triangle V(\boldsymbol{G}[k]) & (9) \\ s.t. \quad & \sum_f \sum_{r=1}^{m^f} S^f_r[k] \le T, \forall k \\ & S^f_r[k] \le Q^f_r[k], \forall f, r, k \end{aligned}$$

Proposition 1: For all frames k, the following policy yields a feasible solution to the Lyapunov drift minimization problem (9) with a guaranteed maximal difference $B_1 < \infty$ from its minimum, where B_1 is independent of $\boldsymbol{Y}[k]$.

EDF for Multiple-Flow (EDF-M) Policy:

In each frame k, the EDF-M Policy observes $\boldsymbol{G}[k]$ and makes decision for $\{\boldsymbol{S}^f[k,m]\}_f$.

- **Initialization:** Let $\boldsymbol{S}^f[k,m] = \boldsymbol{0}$ for all flows f, i.e., let $S^f_{m-i}[k+i] = 0$ for all $i \in \{0, 1, \ldots, m-1\}$ and $f \in \mathcal{F}$.
- **Scheduling:** Calculate $Y^f[k+m-1]$ for all flows. For all flows f, in the order of decreasing $Y^f[k+m-1]$, do the following:
  For $i = 0$ up to $m-1$, let
  $$\begin{aligned} S^f_{m-i}[k+i] &= \min\Bigg(A^f[k] - \sum_{j=0}^{i-1} S^f_{m-j}[k+j], \\ &\qquad T - \sum_{\tilde{f}\in\mathcal{F}} \sum_{r=1}^{m-i-1} S^{\tilde{f}}_r[k+i] \Bigg), \end{aligned} \tag{10}$$
  i.e., assign as much as possible service to $S^f_{m-i}[k+i]$ as long as there are unserved packets in $A^f[k]$ and capacity available.

Proof: The detailed proof is provided in [26]. ■

Remark 1: As revealed in the proof, the way to allocate service among the indices of $\boldsymbol{S}^f[k,m]$ for a chosen flow f to achieve the minimum is not unique. We make our particular choice which, under a single-flow setup, is equivalent to the EDF Policy. This equivalence can be argued as follows. Consider the service $S_r^f[k]$, $r = 1, \ldots, m^f$, they are assigned in frame $k-m+1, \ldots, k$ respectively, thus the more urgent packets get their service assigned before the less urgent ones. Also, we allocate as much service as possible to them when assigning it, giving the more urgent packets strict priority over less urgent ones. This equivalence is also illustrated in Fig. 6 in Section V as part of our numerical results. The simulation also shows that by giving priority to the flows with larger deficit counter values when assign services, the EDF-M Policy can intelligently adapt to the different drop ratio requirements.

Remark 2: In the derivation of the EDF-M Policy, we allow this class of policies to assign (or reserve) service for future frames without knowing or estimating the arrivals that to occur. This type of decision may cause some performance loss. And as a policy that needs to consider all the service that will be allocated to the current arrivals, it is very complicated to extend to a multi-hop flow setup. These limitations motivate us to find an alternative way to approach the problem. Next we present such an approach with a formulation more suitable to extend to multi-hop scenario.

B. Lyapunov-Minimization-Based Solution

In this subsection, we consider a different class of policies that observes the queue vector $\boldsymbol{Q}[k] \triangleq \{Q_r^f[k]\}_{f,r}$, the arrival vector $\boldsymbol{A}[k] \triangleq \{A^f[k]\}_f$ and the deficit counter vector $\boldsymbol{Y}[k] \triangleq \{Y^f[k]\}_f$ and makes decision for the vector of service $\boldsymbol{S}[k] \triangleq \{S_r^f[k]\}_{f,r}$ at the beginning of frame k.

We define a quadratic Lyapunov function by

$$V(\boldsymbol{Q},\boldsymbol{Y},k) \triangleq \frac{1}{2}\sum_f \Big(\sum_{r=1}^{m^f}(Q_r^f[k])^2 + (Y^f[k])^2\Big), \tag{11}$$

and try to minimize its drift given by

$$\begin{aligned}&\triangle V(\boldsymbol{Q},\boldsymbol{Y},\boldsymbol{A})\\ &\quad\triangleq \mathbb{E}\left[V(\boldsymbol{Q},\boldsymbol{Y},k+1) - V(\boldsymbol{Q},\boldsymbol{Y},k)|\boldsymbol{Q}[k],\boldsymbol{Y}[k],\boldsymbol{A}[k]\right],\end{aligned}$$

under the capacity constraint (1). The problem is formulated as:

$$\begin{aligned}\min_{\boldsymbol{S}[k]} \quad & \triangle V(\boldsymbol{Q},\boldsymbol{Y},\boldsymbol{A}) \qquad (12)\\ s.t. \quad & \sum_f\sum_{r=1}^{m^f} S_r^f[k] \le T,\ \forall l \in \mathcal{L},\\ & S_r^f[k] \le Q_r^f[k],\ \forall l,r,k.\end{aligned}$$

Proposition 2: For all frames k, the following policy yields a feasible solution to the Lyapunov drift minimization problem (12) with a guaranteed finite maximal difference from its minimum, where the difference is independent of $\boldsymbol{Y}[k]$.

Quadratic Drift Minimizing (QDM) Policy for Single-Hop:

- **Queue Evolution:** The queues evolve as in (2) and (3).
- **Scheduling:** We calculate the "weight" $w_r^f[k]$ for packets of flow f with r frames-to-go in frame k as

$$w_r^f[k] = \begin{cases} Q_r^f[k] & , r > 1;\\ Y^f[k] + Q_1^f[k] & , r = 1.\end{cases}$$

In each frame k, we assign $S_r^f[k] = \min\{T, Q_r^f[k]\}$ if $w_r^f[k] = \max_{f',r'}\{w_{r'}^{f'}[k]\}$. If the total assigned service is less than T, repeat the above step for the remaining unassigned $S_r^f[k]$'s. In the case of multiple maxima in the set $\{w_r^f[k]\}$, choose the one with the largest $Q_r^f[k]$.

Proof: This proof is a simplified special case of the proof of Proposition 5 in Section IV-A, and thus is omitted for brevity. ■

Remark: Unlike the EDF and the EDF-M Policy, the QDM Policy does not give priority to the packets that have less time-to-go. Instead, it assigns service according to the weights for different queues. It can be observed from Fig. 6 and Fig. 7 in Section V that the QDM Policy outperforms the EDF-M Policy in the sense that it drops less packets to the deficit counters under the same realization of the arrivals. The formulation of the QDM Policy also has the nice property that it can be easily extended to operate in a multi-hop scenario, which will be discussed in Section IV.

C. Performance of the EDF-M and QDM Policies

To characterize the performance of the EDF-M and QDM Policies, we first introduce the set of *Requirement Satisfying Arrival Processes* in the single-hop scenario as follows.

Definition 1: Given a loss probability upper-bound vector $\boldsymbol{p} = \{p^f\}_f$ and a deadline constraint of m frames for all flows, the set of *Requirement Satisfying Arrival Processes* for policy H, denoted by $\mathcal{A}_m^H(\boldsymbol{p})$, contains all the arrival processes $\boldsymbol{A}[k] = \{A^f[k]\}_f$ such that the policy H can satisfy the delivery ratio for all flows. In particular, we denote such sets for EDF-M and QDM policy by $\mathcal{A}_m^{\text{EDF-M}}(\boldsymbol{p})$ and $\mathcal{A}_m^{\text{QDM}}(\boldsymbol{p})$ respectively. ◇

We study the performance of the EDF-M and the QDM policies in two extreme regimes, where the deadline m is set to 1 or approaches infinity respectively.

1) Performance when $m = 1$: In the case where $m = 1$, i.e., each flow's arrivals must be served during the frame they arrive. In this case, we always have $Q_1^f[k] = A^f[k]$, thus the queue evolution (2) and the deficit counter evolution (3) can be combined as

$$Y^f[k+1] = \left(Y^f[k] - p^f a^f\right)^+ + (A^f[k] - S^f[k]). \tag{13}$$

Thus, the EDF-M and QDM Policies are simplified to choose the service vector $\boldsymbol{S}[k]$ in each frame to maximize $\sum_f S^f[k]Y^f[k]$ and $\sum_f S^f[k](Y^f[k] + A^f[k])$, respectively.

For comparison, we consider the class of stationary policies $\mathcal{H}$ that observes $(\boldsymbol{Y}[k], \boldsymbol{A}[k])$ in each frame k and chooses a service vector $\boldsymbol{S}[k]$. We define the set $\mathcal{A}_1^{\mathcal{H}}(\boldsymbol{p})$ as the set of arrival processes whose loss probability upper-bound $\boldsymbol{p}$ can be satisfied by some stationary policy $H_0 \in \mathcal{H}$, i.e., $\mathcal{A}_1^{\mathcal{H}}(\boldsymbol{p}) = \bigcup_{H_0\in\mathcal{H}} \mathcal{A}_1^{H_0}(\boldsymbol{p})$.

For the performance of the EDF-M and QDM Policies when $m = 1$, we have the following proposition:

Proposition 3: The EDF-M and QDM policies are optimal in the single-hop setup when $m = 1$ in the sense that for any arrival process $\boldsymbol{A}[k]$ that lies strictly within $\mathcal{A}_1^{\mathcal{H}}(\boldsymbol{p})$, we have $\boldsymbol{A}[k] \in \mathcal{A}_1^{\text{EDF-M}}(\boldsymbol{p})$ and $\boldsymbol{A}[k] \in \mathcal{A}_1^{\text{QDM}}(\boldsymbol{p})$.

Proof: Due to the space limitations, here we provide an outline of the proof, and the detailed proof is provided in [26].

To prove Proposition 3, we first characterize the set of $\mathcal{A}_1^{\mathcal{H}}(\boldsymbol{p})$ by looking at the steady-state operation of some stationary policy that can satisfy the requirements of the arrival processes $\boldsymbol{A}[k]$. It can be shown that for any arrival process $\boldsymbol{A}[k] \in \mathcal{A}_1^{H}(\boldsymbol{p})$, there exists a set of $\alpha(\tilde{\boldsymbol{s}}, \boldsymbol{a})$ such that

$$\sum_{\tilde{\boldsymbol{s}} \in \mathcal{S}} \alpha(\tilde{\boldsymbol{s}}, \boldsymbol{a}) = 1,$$

and

$$\sum_{\boldsymbol{a}} P(\boldsymbol{A} = \boldsymbol{a}) \sum_{\tilde{\boldsymbol{s}} \in \mathcal{S}} \tilde{s}^f \alpha(\tilde{\boldsymbol{s}}, \boldsymbol{a}) > (1 - p^f) a^f, \, \forall f \in \mathcal{F},$$

where $\alpha(\tilde{\boldsymbol{s}}, \boldsymbol{a})$ can be interpreted as the probability of some stationary policy H_0 choosing a particular service vector $\tilde{\boldsymbol{s}}$, given the arrival vector is $\boldsymbol{a}$. Thus, for any arrival process $\boldsymbol{A}$ that lies strictly inside $\mathcal{A}_1^{\mathcal{H}}(\boldsymbol{p})$, we can find $\epsilon > 0$ such that for each flow $f \in \mathcal{F}$,

$$(1 - p^f) a^f \leq \sum_{\boldsymbol{a}} P(\boldsymbol{A} = \boldsymbol{a}) \sum_{\tilde{\boldsymbol{s}} \in \mathcal{S}} \tilde{s}^f \alpha(\tilde{\boldsymbol{s}}, \boldsymbol{a}) - \epsilon.$$

Also note the fact that given $\boldsymbol{A}[k] = \boldsymbol{a}$, $\boldsymbol{Y}[k] = \boldsymbol{y}$, the service vector $\boldsymbol{s}^*_{\text{EDF-M}}(\boldsymbol{a}, \boldsymbol{y})$ chosen by the EDF-M policy satisfies

$$\boldsymbol{s}^*_{\text{EDF-M}}(\boldsymbol{a}, \boldsymbol{y}) = \underset{\boldsymbol{s} \in \mathcal{S}}{\operatorname{argmax}} \sum_f s^f y^f,$$

when $m = 1$. Similarly for QDM, we have

$$\boldsymbol{s}^*_{\text{QDM}}(\boldsymbol{a}, \boldsymbol{y}) = \underset{\boldsymbol{s} \in \mathcal{S}}{\operatorname{argmax}} \sum_f s^f (a^f + y^f),$$

when $m = 1$. Utilizing the above facts, we study the expected drift of the Lyapunov function defined as $V(\boldsymbol{Y}[k]) = \frac{1}{2} \sum_f (Y^f[k])^2$, and proved that it is negative expect for a finite set of $\boldsymbol{Y}$ values. Thus by Foster-Lyapunov theorem, the Markov chain $\boldsymbol{Y}[k]$ is positive recurrent and has a finite mean. Thus the QoS requirement is satisfied, i.e., $\forall \boldsymbol{A}[k] \in \mathcal{A}_1^{\mathcal{H}}(\boldsymbol{p})$, we have $\boldsymbol{A}[k] \in \mathcal{A}_1^{\text{EDF-M}}(\boldsymbol{p})$ and $\boldsymbol{A}[k] \in \mathcal{A}_1^{\text{QDM}}(\boldsymbol{p})$. ■

2) Asymptotic Performance as $m \to \infty$: To study the asymptotic performance of the EDF-M and QDM policies, we compare their set of requirement satisfying arrival processes with the *capacity region* as m goes to infinity. In our model, the capacity region $\Lambda(\boldsymbol{p})$ of a single-hop network with loss probability $\boldsymbol{p}$ can be defined as

$$\Lambda(\boldsymbol{p}) = \left\{ \boldsymbol{A} : \sum_f (1 - p^f) a^f < T, \, \mathbb{E}[(A^f)^2] < \infty \right\}.$$

We have the following proposition:

Proposition 4:

$$\lim_{m \to \infty} \mathcal{A}_m^{\text{EDF-M}}(\boldsymbol{p}) = \Lambda(\boldsymbol{p}),$$
$$\lim_{m \to \infty} \mathcal{A}_m^{\text{QDM}}(\boldsymbol{p}) = \Lambda(\boldsymbol{p}).$$

Proof: The detailed proof is provided in Appendix A. ■

Remark: Proposition 4 establishes the asymptotic optimality of the EDF-M and QDM Policy in a single-hop, multi-flow setup. As we relax the deadline constraint (i.e., letting $m \to \infty$), both policies can support any arrival process that lies inside the capacity region. We note that the EDF policy does not possess such asymptotic optimality characteristics in the multi-flow environment. In comparison, our proposed algorithms achieve such asymptotic optimality due to their intelligent packet dropping characteristics.

IV. Multi-Hop Scenario

In this section we present two different approaches to our objective under multi-hop setup. The first one is the extension of the QDM Policy to the multi-hop scenario, and the latter one is a new approach based on optimization.

A. The QDM Policy in Multi-Hop Scenario

Unlike the EDF-M Policy that requires the knowledge of the past information, the QDM formulation uses only the current information, which has a structure that can be extended to multi-hop scenario easily. In the multi-hop scenario, the Lyapunov function is extended to include the queues over the whole route as follows:

$$V(\boldsymbol{Q}, \boldsymbol{Y}, k) \triangleq \frac{1}{2} \sum_f \sum_{i=1}^{h^f} \Big(\sum_{r=1}^{m^f} (Q^f_{[i],r}[k])^2 + (Y^f_{[i]}[k])^2 \Big), \quad (14)$$

and we try to minimize its drift. In the multi-hop scenario, the decision made in each frame also includes the drop ratio upper-bound vector $\boldsymbol{P} \triangleq \{P^f_{[i]}[k]\}_{f,i}$. The problem is formulated as:

$$\begin{aligned} \min_{\{\boldsymbol{S}[k], \boldsymbol{P}[k]\}} \quad & \triangle V(\boldsymbol{Q}, \boldsymbol{Y}, \boldsymbol{A}) \qquad (15) \\ s.t. \quad & \sum_f \sum_{r=1}^{m^f} S^f_{l,r}[k] \mathcal{R}^f[l] \leq T, \, \forall l \in \mathcal{L}, \\ & \sum_{i=1}^{h^f} P^f_{[i]}[k] \leq p^f, \, \forall f \in \mathcal{F}. \\ & S^f_{l,r}[k] \leq Q^f_{l,r}[k], \, \forall l, r, k. \end{aligned}$$

Similarly to the single-hop scenario, this minimization problem can be solved using the following weight based algorithm.

Proposition 5: For all frames k, the following policy provides a feasible solution to the Lyapunov drift minimization problem (15) with a guaranteed maximal difference $B_2 < \infty$ from its minimum, where B_2 is independent of $\boldsymbol{Y}[k]$.

QDM Policy for Multi-Hop:

- **Queue Evolution:** The queues evolve as in (2) and (3).

- **Scheduling:** We calculate the "weight" $w^f_{l,r}[k]$ for packet of flow f with r frames-to-go on link l in frame k as

$$\begin{aligned} & w^f_{l,r}[k] \\ &= \begin{cases} Q^f_{l,r}[k] - Q^f_{N[l],r}[k] & , r > 1; \\ Y^f_l[k] - Y^f_{N[l]}[k] + Q^f_{l,1}[k] - Q^f_{N[l],1}[k] & , r = 1. \end{cases} \end{aligned}$$

For each link l, we assign $S^f_{l,r}[k] = \min\{T, Q^f_{l,r}[k]\}$ if $w^f_{l,r}[k] = \max_{f',r'}\{w^{f'}_{l,r'}[k]\}$. If the total assigned service is less than T, repeat the above step for the remaining unassigned $S^f_{l,r}[k]$'s. In the case of multiple maxima in the set $\{w^f_{l,r}[k]\}$, choose the one with the largest $Q^f_{l,r}[k]$.

- **Deficit Queue Service Update:** For each flow f at frame k, update the service of the deficit queues by

$$P^f_{[i]}[k] = p^f \mathbf{1}(Y^f_{[i]}[k] = \max_{i'}\{Y^f_{[i']}[k]\}),$$

where $\mathbf{1}(\cdot)$ is the indicator function. The deficit queue service policy chooses to serve $a^f p^f$ packets from the deficit queue with the maximum length. In the case of multiple maxima in the set $\{Y^f_{[i]}[k]\}$, divide p^f equally among them.

Proof: The detailed proof is provided in [26]. ■

Remark: Although the calculation of the weights $w^f_{l,r}[k]$ appears similar to the back-log difference, it has a different nature: the packets that depart $Q^f_{l,r}$ do not (and will never) enter $Q^f_{N[l],r}$ but instead enter $Q^f_{N[l],r-1}$. This difference results in the unawareness of this policy to the downstream conditions when assigning services. This leads to possible service of packets that cannot reach the destination within their deadlines. This phenomenon is also observable in the simulation results for the QDM Policy in Section V, which has motivated us to seek other approaches to this problem. We present one such approach based on a relaxed optimization formulation in the following subsections.

B. The Stochastic Optimization Formulation

In the following subsections, we formulate the objective as a new optimization problem and solve it using a primal-dual approach. We assume throughout this section that the loss probability upper bounds on all links are pre-determined and fixed, i.e., $P^f_l[k] = p^f_l$ for all k. By fixing p^f_l, we can force the algorithm to drop packets at the source node to prevent intermediate nodes from wasting resources by serving packets that persistently expire before they reach their destination.

We reformulate this problem by first noticing that in our model, the packets $Q^f_{[i],r}[k]$ are either being served or remain unserved at the end of frame k. Thus we have

$$Q^f_{[i],r}[k] = S^f_{[i],r}[k] + D^f_{[i],r}[k], \forall k \tag{16}$$

Combining this with the queue evolution (2), we get

$$\begin{aligned} & S^f_{[i],r}[k+1] + D^f_{[i],r}[k+1] \\ &\quad = S^f_{[i+1],(r+1)}[k] + D^f_{[i],(r+1)}[k], \end{aligned} \tag{17}$$

which can be interpreted as a version of the flow conservation constraint in our model[1]. We can then formulate the problem as the following constrained optimization problem:

$$\begin{aligned} \max_{\{\boldsymbol{S}[k],\boldsymbol{P}[k]\}} \quad & 0 \qquad\qquad (18) \\ s.t. \quad & \text{Flow conservation constraint as in (17)}, \\ & \forall f \in \mathcal{F}, r = 1, \ldots, m^f, i = 1, \ldots, h^f, \\ & \text{Capacity constraint as in (1)}, \\ & \text{Delivery ratio requirements as in (6)}, \\ & 0 \le S^f_{[i],r}[k],\ 0 \le D^f_{[i],r}[k], \\ & \forall f \in \mathcal{F}, r = 1, \ldots, m^f, i = 1, \ldots, h^f. \end{aligned}$$

We attack this problem by first analyzing a relaxed deterministic model in Section IV-C. The solution to this relaxed model will help in exposition as well as in providing insights on the solution of the above more complex problem. Then, in Section IV-D we use the developed insights to implement the algorithm in the stochastic system.

C. Relaxed Optimization Formulation

In the relaxed scenario (also called the fluid model), all the randomness and dynamics are ignored, and the stochastic constraints are replaced by static ones. Note that (20) is a relaxed version of the flow conservation constraint (17), since it only requires the equality to hold on average rather than frame-by-frame. To compensate for this relaxation, we choose $\hat{m}^f$ for each flow f such that $h^f \le \hat{m}^f \le m^f$. By doing this, the system treats each flow f as if it has a more stringent deadline constraint of $\hat{m}^f$ frames. With the above changes, problem (18) reduces to the following problem in this scenario.

$$\begin{aligned} \max_{\boldsymbol{s},\boldsymbol{d}} \quad & 0 & (19) \\ s.t. \quad & s^f_{[i],r} + d^f_{[i],r} = s^f_{[i+1],(r+1)} + d^f_{[i],(r+1)}, & \\ & \forall f \in \mathcal{F}, r = 1, \ldots, \hat{m}^f, i = 1, \ldots, h^f, & (20) \\ & \sum_f \sum_{r=1}^{\hat{m}^f} s^f_{l,r} \mathcal{R}^f[l] \le T,\ \forall l \in \mathcal{L}, & \\ & s^f_{[i+1],1} + d^f_{[i],1} \le p^f_{[i]} a^f, & \\ & \forall f \in \mathcal{F}, i \in \{1, 2, \ldots, h^f\}, & (21) \\ & 0 \le s^f_{[i],r},\ 0 \le d^f_{[i],r}, & \\ & \forall f \in \mathcal{F}, r = 1, \ldots, \hat{m}^f, i = 1, \ldots, h^f, & \end{aligned}$$

where $s^f_{[i],r}$ and $d^f_{[i],r}$ are average service and drop rates as defined in (4) and (5).

Proposition 6: The relaxed optimization problem (19) can be solved by the following policy.

Optimization-Based Policy (OB Policy):

- **Price Evolution:** We introduce prices $\lambda^f_{l,r}(t)$ and $\mu^f_l(t)$ that evolve in continuous-time t.[2] The following differential equations describe their evolutions, where we omit

[1] Unlike the common flow conservation constraint that imposed on the average, this constraint must be strictly satisfied in each frame.

[2] A discrete-time version of this algorithm can be derived using the technique as in [2], [21]

their dependence on t for brevity:

$$\begin{aligned}\dot{\lambda}^f_{l,r} &= -\left(s^f_{[i],r} + d^f_{[i],r} - s^f_{[i+1],(r+1)} - d^f_{[i],(r+1)}\right),\\ \dot{\mu}^f_l &= \left(s^f_{[i+1],1} + d^f_{[i],1} - p^f_{[i]} a^f\right)^+_{\mu^f_l},\end{aligned}$$

where $\dot{x} = \frac{\mathrm{d}x}{\mathrm{d}t}$, $(x)^+_{\mu^f_l} = 0$ if $\mu^f_l = 0$ and $x < 0$, and x otherwise.

- **Scheduling:** For each link l let

$$\begin{aligned}u^f_r &= \begin{cases} \lambda^f_{l,r} - \lambda^f_{N[l],(r-1)}, & \text{if } 1 < r \leq \hat{m}^f \\ \lambda^f_{l,1} - \mu^f_{N[l]}, & r = 1 \end{cases},\\ v^f_r &= \begin{cases} \lambda^f_{l,r} - \lambda^f_{l,(r-1)}, & \text{if } 1 < r \leq \hat{m}^f \\ \lambda^f_{l,1} - \mu^f_{l}, & r = 1 \end{cases},\end{aligned}$$

and the service is allocated according to the following:

$$\begin{aligned}s^f_{l,r} &= T\mathbf{1}((f,r) = \operatorname*{argmax}_{f',r'}\{u^{f'}_{r'}\}),\\ d^f_{l,r} &= Y_{max}\mathbf{1}(v^f_r > 0),\end{aligned}$$

where Y_{max} is some fixed positive parameter.

Proof: The detailed proof is provided in [26]. ■

Remark 1: The parameter Y_{max} is the upper bound for the rate of drop in this deterministic system. It is imposed to make sure the problem has a finite solution.

Remark 2: Note that the OB Policy possesses a back-pressure like decision mechanism, where the differences of prices are utilized in the scheduling decisions. Yet, it is also significantly different from BP (see Section II-B) since, instead of a measure of congestion levels, the prices contain information on: (i) both the end-to-end delay that the packets will experience at each node, captured by $\boldsymbol{\lambda}$; (ii) and the violation of the delivery ratio requirements, captured by $\boldsymbol{\mu}$. Consequently, the OB Policy continuously measures the end-to-end performance through this pricing scheme, and has the ability to react to it to satisfy the QoS requirements. This differentiates the OB Policy from the QDM Policy, whose decisions were based on more local measures. This far-sighted nature of the OB Policy allows us to select the loss probability vector $\{p^f_{[i]}\}_i$ for each flow f to drop packets at the source that cannot reach the destination before their expiry.

Next, we study the performance of the OB Policy by running it in a linear network with a single flow to validate these characteristics, and also to demonstrate its dynamic service discipline selection nature. Specifically, we assume $T = 9$ and consider a flow f with an arrival rate of $a^f = 10$ packets/frame and an end-to-end deadline of $\hat{m}^f = m^f = 12$ frames that is routed over $h^f = 8$ hops. We assume $p^f = 0.25$, and let $p^f_{[8]} = 0.25$, i.e., that all the packets that cannot be served should to be dropped at the source to avoid unnecessary resource consumption.

Fig. 5 shows the service rate of packets with different remaining times along the route. The thickness of the lines indicates the service rate for a clearer view. It can be observed that the service rate is higher for the packets that have more time left towards the source, and for those with more stringent ones towards the destination. This demonstrates the ability of the OB Policy to dynamically change its service discipline from the source to the destination along the route. This transition is intuitive since at the source, packets with longer time to go have a higher chance to get to the destination, while at the last hop to the destination it is better to serve the most urgent packets.

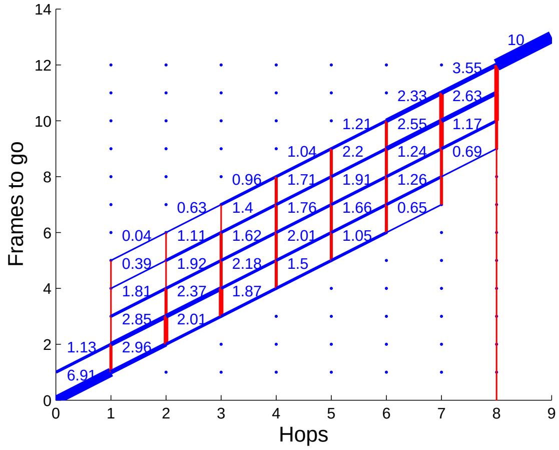


Fig. 5. OB Policy runs in a 8-hop network with a 12-frame deadline.

D. Implementation in the Stochastic System

The OB Policy above is derived based on a relaxation where the constraint (20) is relaxed by associating a price to it. This relaxation enables us to solve the dual problem, but unlike prior works (e.g., [2], [21]), the policy cannot be directly used in the stochastic system. The reason is that in the stochastic system of our model, the number of served packets $S^f_{l,r}[k]$ and the number of dropped packets $D^f_{l,r}[k]$ are tightly related to each other by $Q^f_{l,r}[k]$ as in (16). However, in the relaxed model, these two quantities are only loosely coupled via the pricing mechanism. In other words, the flow conservation constraint (17) holds by definition for each frame k in the stochastic system, but only in an average sense in the relaxed model. Thus the queue evolution in the stochastic system is different from the price evolution in the fluid model.

In order to implement the OB Policy in the stochastic system, we need to modify the definition of $Q^f_{l,r}[k]$. Originally, it represents the number of flow f packets on link l that have r frames-to-go at frame k. Here we redefine it as a queue $\hat{Q}^f_{l,r}$ on link l *at level* r for flow f. Accordingly, the packets counted in $\hat{Q}^f_{l,r}$ are no longer forced to be served or dropped, but they can be queued up and form proper prices. Hence the queue at level r holds packets that have *at most* r frames-to-go. This new evolution for $\hat{Q}^f_{[i],r}$ is captured by

$$\begin{aligned}\hat{Q}^f_{[i],r}[k+1] &= \left(\hat{Q}^f_{[i],r}[k] - (S^f_{[i],r}[k] + D^f_{[i],r}[k])\right)^+\\ &\quad + S^f_{[i+1],r+1}[k] + D^f_{[i],r+1}[k], \qquad (22)\end{aligned}$$

where the strict flow conservation constraint (17) is relaxed. As a consequence to this relaxation, the delivered packets are no longer guaranteed to be unexpired as they were under QDM

Policy. A more stringent deadline of $\hat{m}^f$ frames is chosen for flow f as in the OB Policy to alleviate this affect.

The newly defined queueing architecture is the same as shown in Fig. 3, besides that all the queues are able to build up and there are a total of $\hat{m}^f$ level queues for each flow f in a node instead of m^f queues. With this new queueing architecture, the OB Policy can be implemented in the stochastic system by running it in a separate virtual layer to calculate $S^f_{l,r}[k]$, which is used in the actual layer, where the packets are actually served or dropped. The implementation of the OB Policy in the stochastic system gives us the following policy:

Discrete Optimization-Based Policy (DOB Policy):

- **Price Evolution:** The prices $\lambda^f_{l,r}$ and μ^f_l in the virtual layer evolve as:

$$\begin{aligned}\lambda^f_{l,r}[k+1] &= \lambda^f_{l,r}[k] - \alpha\Big(S^f_{[i],r}[k] + D^f_{[i],r}[k] \\ &\quad - S^f_{[i+1],r+1}[k] - D^f_{[i],r+1}[k]\Big),\\ \mu^f_l[k+1] &= \Big(\mu^f_l[k] + \alpha\big(S^f_{[i+1],1}[k] \\ &\quad + D^f_{[i],1}[k] - p^f_{[i]}a^f\big)\Big)^+,\end{aligned}$$

  where α is the step-size of the gradient algorithm.
- **Scheduling:** For each link l let

$$u^f_r = \begin{cases}\lambda^f_{l,r}[k] - \lambda^f_{N[l],r-1}[k] & \text{if } 1 < r \le \hat{m}^f\\ \lambda^f_{l,1}[k] - \mu^f_{N[l]}[k] & \text{otherwise}\end{cases},$$

$$v^f_r = \begin{cases}\lambda^f_{l,r}[k] - \lambda^f_{l,r}[k] & \text{if } 1 < r \le \hat{m}^f\\ \lambda^f_{l,1}[k] - \mu^f_l[k] & \text{otherwise}\end{cases},$$

  and the service is allocated according to the following:

$$S^f_{l,r}[k] = \begin{cases} T & \text{if } (f,r) = \underset{f,r}{\operatorname{argmax}}\{u^f_r\}\\ 0 & \text{otherwise}\end{cases},$$

$$D^f_{l,r}[k] = \Big(\hat{Q}^f_{l,r}[k] - S^f_{l,r}[k]\Big)^+ \mathbf{1}(v^f_r > 0).$$

- **Queue Evolution:** In the actual layer the level queues evolve as in (22) and the deficit queues evolve as in (3).

Remark: This stochastic implementation inherits the main characteristics of the OB Policy, where prices serve as a measure of the end-to-end delay performance. However, instead of a design parameter Y_{max}, we use the service $S^f_{l,r}[k]$ provided by the virtual layer and the actual queue-length $\hat{Q}^f_{l,r}[k]$ to dynamically determine the maximum number of drops $D^f_{l,r}[k]$. This is because the performance of the DOB Policy is sensitive to the choice of Y_{max} due to the stochastic nature of the arrivals. Note that whether to drop the packets to a lower layer queue is still determined by the virtual layer, thus the level queues $\hat{\boldsymbol{Q}}[k]$ are not flushed at the end of each frame, as opposed to (2) in the QDM Policy. As a consequence, the delivered packets under DOB Policy are not guaranteed to be unexpired as they were under QDM Policy. However, we shall see in our numerical studies in Section V by appropriately choose a more stringent constraint $\hat{m}^f$ for each flow f, we can still meet the deadline requirements.

V. Numerical Results

In this section, we present the numerical results of our policies. The purpose of these simulations is to understand the performance of the policies as well as to get insights on how the service discipline affects the performance.

A. Single-Hop Setup

In these simulations, we run the EDF, EDF-M and QDM policies in a single-hop network with a frame length $T = 10$ and a fixed deadline of $m^f = 10$ frames. We compare their performance by the deficit counter values, where a smaller deficit counter value implies less dropped packets. The arrival processes are independent and identically distributed (i.i.d.) over time with uniform distribution from 0 to $2a^f$ in each frame. In each comparison, we test our algorithms using the same realization of the arrival process. The simulations run for ten thousand frames, and we plot the first two thousand frames for clarity.

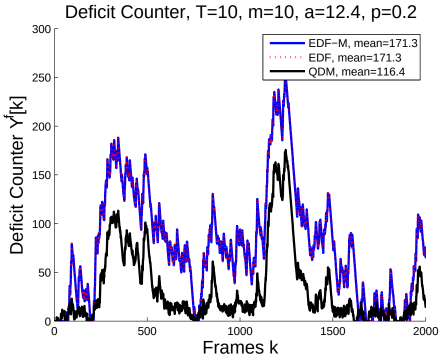


Fig. 6. The evolution of the deficit counter $Y^f[k]$ for a single flow

Fig. 6 shows the evolution of the deficit counter for a single flow under the three different policies. It can be observed that when serving a single flow, the EDF and EDF-M policies have the exactly same performance, which is slightly outperformed by the QDM Policy in the sense that the QDM Policy drops less packets. This observation implies that in the setup where packet expiry is allowed, the EDF Policy may not be strictly optimal.

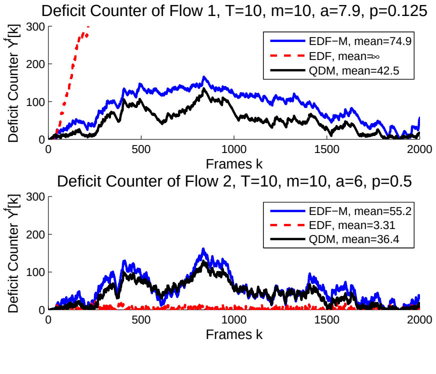


Fig. 7. The evolution of the deficit counter $Y^f[k]$ for two flows with heterogeneous arrival rates and delivery ratio requirements

Fig. 7 illustrates the deficit counters' evolution for two flows with different arrival rates and delivery ratio requirements. Flow 1 has a higher arrival rate as well as a higher delivery ratio requirement compared to Flow 2. By simply serving the most urgent packet in the system, the EDF Policy is ignorant to this difference in the delivery ratio, resulting in allocating insufficient service to Flow 1, whose deficit counter blows up under the EDF Policy. On contrast, the EDF-M Policy uses the deficit counter values to capture the violation of the long-term delivery ratio requirements and tries to balance them for all flows. The QDM Policy utilizes the deficit counter value information to schedule service in a different way, which again gives it slightly better performance than the EDF-M Policy.

B. Multi-Hop Setup

In the multi-hop scenario, the simulations are run for a single flow f in a 4-hop tandem network with frame length $T = 12$. Flow f has an i.i.d. arrivals with uniform distribution from 0 to $2a^f$ in each frame. The loss probability upper bound p^f for flow f is set to 0.25, and, for the DOB Policy, we assume that $p^f_{[4]} = 0.25$.

Service Discipline Choice: We first present the service discipline chosen by the QDM and the DOB policies. In this simulation, we choose $a^f = 10$ with the aforementioned distribution, and the flow has a deadline constraint of 6 frames. It can be verified that this arrival process is supportable under this QoS requirement by a simple (but not necessarily optimal) policy. In each frame k, this simple policy serves as many as possible of the newly arrived packets $A^f[k]$ at the source $f[4]$, while the unserved packets are dropped and never get served again at the source.

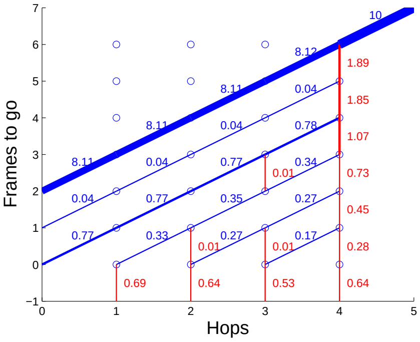


Fig. 8. The average service rates (blue) and drop rates (red) of the QDM Policy of packets with different time to go

Fig. 8 shows the average service rates and drop rates of the QDM Policy in this setup for the packets with different frames-to-go, where the rates shown from 0 frame-to-go downward represents the service rate of the deficit queues. It can be observed that this policy gives higher priority to packets that have longer remaining time throughout the whole route. We can also observe here that packets with 2 and 3 frames-to-go are sent by the source at a non-negligible rate. However those packets have no chance of reaching the destination before their deadlines, thus must be dropped along the route which results in the resource allocated to them wasted. This waste of resources may have an impact on the performance of the policy when the QoS requirements are more stringent.

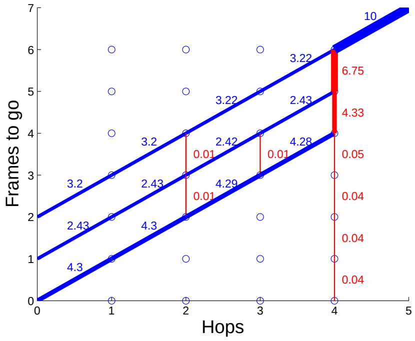


Fig. 9. The average service rates (blue) and drop rates (red) of the DOB Policy of packets with different time to go

As a comparison, Fig. 9 shows the service rates and drop rates for packets at each level queue along the route under the DOB Policy. Note that the back-pressure like pricing mechanism used by the DOB Policy contains information along the downstream link. As a result, the DOB Policy does not send packets that have no chance to reach the destination through the network, saving some resources that would be wasted by the QDM Policy. On the other hand, since the price evolution (22) allows packets to stay at the same level for more than one slot, the received packets may exceed the strict deadline constraint when they reach their destination.

The Affect of $\hat{m}^f$ in the DOB Policy: To show the affect of the parameter $\hat{m}^f$ in the DOB Policy, we now assume flow f has a deadline of 25 frames which gives us a wider range to choose $\hat{m}^f \in [4, 25]$. We simulated for four different arrival rates where $a^f = 10, 11, 11.99$ and 12.5 respectively.

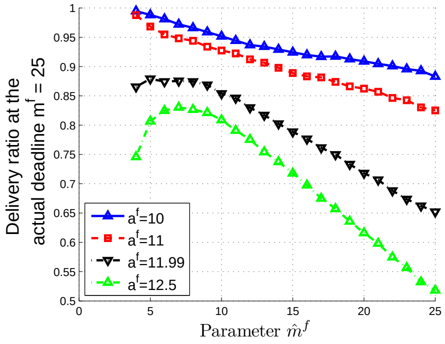
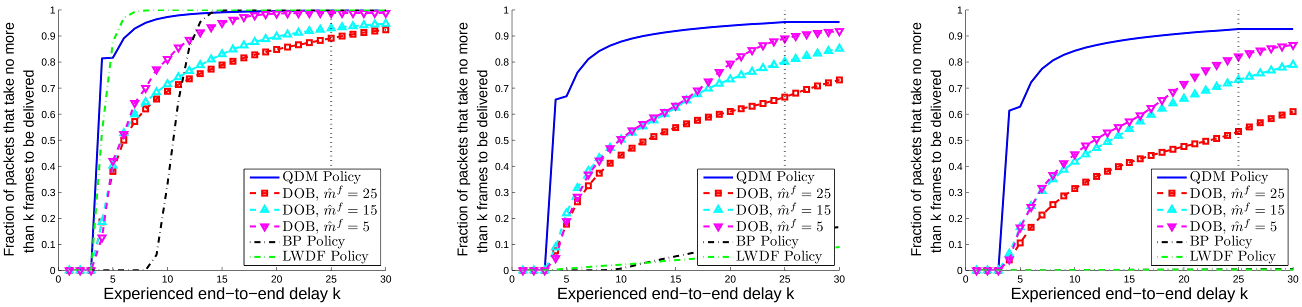


Fig. 10. The achieved delivery ratio when varying $\hat{m}^f$ in $[4, 25]$.

Fig. 10 illustrates the achieved delivery ratio for different $\hat{m}^f$ values. As we can observe, when $\hat{m}^f = m^f = 25$, the delivery ratio performance can be poor especially when the arrival rate is high. It also shows that we by decreasing $\hat{m}^f$, i.e. letting the system assume the flow has a more stringent deadline constraint, we can generally achieve an increasing delivery ratio performance. We can also observe that although



(a)

(b)

(c)

Fig. 11. The QoS performance comparison for (a) $a^f = 10$, (b) $a^f = 11.99$, (c) $a^f = 12.5$

the delivery ratio is less than the required 0.75 when a^f is close to or over the capacity and $\hat{m}^f = m^f = 25$ due to the relaxation of constraint (17), the DOB Policy can achieve the delivery ratio requirement by choosing proper $\hat{m}^f$. It also can be noticed that when the arrival rate is large (for instance, $a^f = 12.5$), choosing a $\hat{m}^f$ that is too stringent (i.e., very close to h^f) may result in degradation in the delivery ratio performance. This is because a $\hat{m}^f$ very close to h^f leaves the system little freedom on choosing the service discipline. Based on these observations, a adaptive policy may be developed which dynamically update the value of $\hat{m}^f$ to achieve the delivery ratio requirements.

QoS Performance: We compare the QoS performance of the proposed policies with the BP and the LWDF policies (please refer to Section II-B for the descriptions of these policies) under different loads.

Fig. 11 shows the distribution of the end-to-end delay experienced by the received packets under different policies and increasing arrival rates. First, we note that in terms of the delivery ratio, both the QDM and DOB Policies significantly outperform the back-pressure algorithm, which does not distinguish the difference in service discipline, in Fig. 11(b) and (c), where the network is critically loaded and overloaded respectively. It can also be observed that without using a more stringent parameter the DOB Policy has worse delay performance than the QDM Policy for low end-to-end delay levels due to the building of the prices along the route, which results in queueing delay in the level queues. However when using a more stringent $\hat{m}^f$ value, the performance of the DOB Policy can increase significantly.

As the arrival rate increases from $a^f = 10$ in Fig. 11(a) to the heavily loaded level of $a^f = 11.99$ in Fig. 11(b), the QoS performance degrades for all policies, but with different degrees. Especially, while the LWDF policy performs the best when $a^f = 10$, it has similar poor performance as the BP policy under the considered heavily loaded scenario. On the other hand, by dropping more packets, both of our proposed policies alleviate the degradation in the end-to-end delay performance caused by the increased arrival rate. When the arrival rate increases to $a^f = 12.5$ which is outside the capacity region of the BP and LWDF policies, their performance degrades further since they do not consider dropping packets. On the contrary, our policy still manages to meet the delivery ratio requirements by actively dropping more packets. This demonstrates the adaptive nature of both policies to the traffic requirements and the network conditions, a property lacking by the BP and LWDF policies.

VI. Conclusion

In this work, we considered the problem of serving flows with fixed end-to-end delay constraint in a multiple-hop network, which is a challenging extension to the previous works. We proposed a novel queueing architecture which allows us to exploit the freedom of choosing service discipline when serving packets. Based on this new model, we developed three algorithms using different approaches, namely the EDF-M Policy, the QDM Policy and the DOB Policy. We performed extensive numerical studies to illustrate the advantages and disadvantages to all of the algorithms.

We proposed three different algorithms with different complexity and characteristics, however the analysis of their performance still remains a challenging and open question to study. Other future research of this topic includes the extension to the flow with multiple routes, as well as the extension to the wireless networks.

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] 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.

[3] 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.

[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] 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.

[6] 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.

[7] 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.

[8] 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.

[9] I.-H. Hou, V. Borkar, and P. R. Kumar, "A theory of qos for wireless," in *INFOCOM 2009, IEEE*, 2009, pp. 486 –494.

[10] I.-H. Hou and P. R. Kumar, "Admission control and scheduling for qos guarantees for variable-bit-rate applications on wireless channels," in *MobiHoc '09: Proceedings of the tenth ACM international symposium on Mobile ad hoc networking and computing*. ACM, 2009, pp. 175–184.

[11] ——, "Scheduling heterogeneous real-time traffic over fading wireless channels," in *INFOCOM, 2010 Proceedings IEEE*, 2010, pp. 1 –9.

[12] J. Jaramillo and R. Srikant, "Optimal scheduling for fair resource allocation in ad hoc networks with elastic and inelastic traffic," in *INFOCOM, 2010 Proceedings IEEE*, 2010, pp. 1 –9.

[13] M. Neely, "Delay-based network utility maximization," in *INFOCOM, 2010 Proceedings IEEE*, 2010, pp. 1 –9.

[14] P. Bhattacharya and A. Ephremides, "Optimal scheduling with strict deadlines," *Automatic Control, IEEE Transactions on*, vol. 34, no. 7, pp. 721 –728, jul 1989.

[15] Z.-X. Zhao, S. Panwar, and D. Towsley, "Queueing performance with impatient customers," in *INFOCOM '91. Proceedings. Tenth Annual Joint Conference of the IEEE Computer and Communications Societies. Networking in the 90s., IEEE*, 1991, pp. 400 –409 vol.1.

[16] A. C. Dalal and S. Jordan, "Optimal scheduling in a queue with differentiated impatient users," *Perform. Eval.*, vol. 59, no. 1, pp. 73–84, 2005.

[17] D. G. Pandelis and D. Teneketzis, "Stochastic scheduling in priority queues with strict deadlines," *Probability in the Engineering and Informational Sciences*, vol. 7, no. 02, pp. 273–289, 1993.

[18] J. Xu and D. Parnas, "Scheduling processes with release times, deadlines, precedence and exclusion relations," *IEEE Trans. Softw. Eng.*, vol. 16, pp. 360–369, March 1990. [Online]. Available: http://portal.acm.org/citation.cfm?id=78266.78285

[19] A. Stoyenko and L. Georgiadis, "On optimal lateness and tardiness scheduling in real-time systems," *Computing*, vol. 47, pp. 215–234, 1992, 10.1007/BF02320193. [Online]. Available: http://dx.doi.org/10.1007/BF02320193

[20] A. Stolyar, "Maximizing queueing network utility subject to stability: Greedy primal-dual algorithm," *Queueing Systems*, vol. 50, no. 4, pp. 401–457, 2005.

[21] 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.

[22] A. Stolyar, "Control of end-to-end delay tails in a multiclass network: Lwdf discipline optimality," *Annals of Applied Probability*, vol. 13, no. 3, pp. 1151–1206, 2003.

[23] 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://www2.ece.ohio-state.edu/∼lir/xiolieryeki10_report.pdf.

[24] M. J. Neely, E. Modiano, and C.-P. Li, "Fairness and optimal stochastic control for heterogeneous networks," *IEEE/ACM Trans. Netw.*, vol. 16, no. 2, pp. 396–409, 2008.

[25] M. J. Neely, "Super-fast delay tradeoffs for utility optimal fair scheduling in wireless networks," in *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, april 2006, pp. 1 –13.

[26] R. Li and A. Eryilmaz, "Scheduling for end-to-end deadline-constrained traffic with reliability requirements in multi-hop networks technical report." [Online]. Available: http://somewhere

Appendix A
Proof of Proposition 4

To prove Proposition 4, we consider our one-hop system as a black box, and compare it to an infinite buffer queue. More specifically, we view our system as a single server queue, who has an arrival rate of $a_\Sigma \triangleq \sum_f a^f$ and a service capacity of T packets per frame. Any packets that spend more than m frames waiting in the queue are removed. We refer to this model as a m-frame deadline queue, and denote the departure rate of the system under a work conserving policy as c_m. As a comparison, we look at a infinite buffer queue with the same arrival rate a_Σ and service capacity T, whose departure rate under a work conserving policy is denoted by c.

We first prove the following lemma:

Lemma 1: $\lim_{m\to\infty} c_m = c$ for all $\boldsymbol{A} \in \Lambda(\boldsymbol{p})$.

Proof: For all $\boldsymbol{A} \in \Lambda(\boldsymbol{p})$, we have $\sum_f (1-p^f)a^f < T$. We consider the following three cases:

Case 1: $a_\Sigma < T$. In this case, the infinite buffer queue is stable, and its departure rate $c = a_\Sigma$. Since it is stable, the random waiting time W in the system has a finite mean, i.e. $\mathbb{E}[W] = \bar{w} < \infty$. By Markov's Inequality, we have

$$P\{W > m\} < \frac{\bar{w}}{m}.$$

Note that $P\{W > m\}$ is the probability of dropping expired packets in the m-frame deadline system. By choosing m large enough, we can make this probability arbitrarily small. Since the departure rate of the m-frame deadline system can be written as $c_m = a_\Sigma(1 - P\{W > m\})$, we have

$$\lim_{m\to\infty} c_m = a_\Sigma = c.$$

Case 2: $a_\Sigma = T$. In this case, assume we "throw away" an arbitrary small fraction $\delta_1 > 0$ of the arrival packets. Use the same argument above, we can choose a m large enough to achieve a departure rate arbitrarily close to $(1-\delta_1)T$. Since we can make δ_1 arbitrarily small, we have in this case

$$\lim_{m\to\infty} c_m = a_\Sigma = T.$$

Case 3: $a_\Sigma > T$ and $\sum_f (1-p^f)a^f < T$. In this case, the infinite buffer queue becomes unstable, thus its departure rate is its serving capacity T. We need to prove that in the m-frame deadline system, the departure rate can approach T as m goes to infinity. Since we consider work conserving policies, the number of departures in each frame is T unless there are less than T packets in the queue. Thus we just need to show that the m-frame deadline system is very likely to have enough packets in the queue despite the fact that the packets waiting more than m frames are removed from the queue.

Since $a_\Sigma > T$, we can find some constant $\delta_2 > 0$ such that $T + \delta_2 < a_\Sigma$. Let $\epsilon > 0$ be an arbitrary positive constant, from the weak law of large numbers, we have for sufficiently large w,

$$P\left\{\left|\frac{\sum_{i=1}^{w}\sum_f A^f[i]}{w} - a_\Sigma\right| > a_\Sigma - (T+\delta_2)\right\} < \epsilon.$$

Thus,

$$P\left\{\frac{\sum_{i=1}^{w}\sum_f A^f[i]}{w} \geq T + \delta_2\right\} > 1 - \epsilon$$

for sufficiently large w, i.e., with probability $1-\epsilon$, more than $(T+\delta_2)w$ packets will arrival during this window of w frames. At least $\delta_2 w$ packets arrived during this window will not be served, since at most wT packets can be served in w frames. Hence, if we choose $m > \frac{T}{\delta_2}w + 1$, i.e., the deadline is longer than the duration of $\frac{T}{\delta_2}$ such windows plus an extra frame, at least wT unexpired packets will remain in the system with probability $(1-\epsilon)^{T/\delta_2}$. This implies if we look at any frame

k, the unserved packets in the previous $\frac{T}{\delta_2}w$ frames are not expired by our choice of m, and with probability $(1-\epsilon)^{T/\delta_2}$, we have at least wT such packets waiting. Thus, for each frame k, it is very likely to have more than wT packets in the system. Since this holds for any $\epsilon > 0$, we have $c_m \to T$ as $m \to \infty$. ■

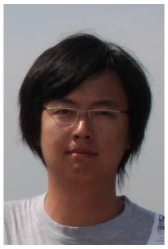
Using this lemma, we can prove Proposition 4 by noticing both EDM-M and QDM Policy are work conserving policies in the single-hop topology. We will prove it for QDM Policy, and the EDF-M Policy follows exactly the same argument.

For all $\boldsymbol{A} \in \Lambda(\boldsymbol{p})$, we have $\sum_f (1-p^f)a^f < \min\{a_\Sigma, T\}$. We can write $\sum_f (1-p^f)a^f = \min\{a_\Sigma, T\} - \epsilon_1$, where $\epsilon_1 > 0$. By the previous lemma, QDM Policy can provide a departure rate of $\min\{a_\Sigma, T\} - \epsilon_1/2$ when m is large enough.

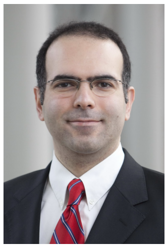
Assume for some $\boldsymbol{A} \in \Lambda(\boldsymbol{p})$, QDM cannot satisfy the delivery ratio requirements for some flows for all m, i.e. $\boldsymbol{A} \notin \mathcal{A}_m^{\text{QDM}}(\boldsymbol{p})$ for all m. That means for some flows, their deficit counters will not be stable under any m. However, according to the operation of the QDM policy, this means all deficit counters will be unstable, since it always serves the flow with the largest deficit counter value. Hence for all flow f, we have $a^f - c^f_{\text{QDM}} > p^f a^f$, where c^f_{QDM} is the average departure rate for flow f under the QDM Policy. This implies

$$\sum_f c^f_{\text{QDM}} < \sum_f (1-p^f)a^f$$

for all m values. This contradicts the fact that the QDM Policy can provide a departure rate of $\min\{a_\Sigma, T\} - \epsilon_1/2 > \sum_f (1-p^f)a^f$ when m is large enough. Thus for all $\boldsymbol{A} \in \Lambda(\boldsymbol{p})$, $\boldsymbol{A} \in \mathcal{A}_m^{\text{QDM}}(\boldsymbol{p})$ for some m.



Ruogu Li (S '10) received his B.S. degree in Electronic Engineering from Tsinghua University, Beijing, in 2007. He is currently a PhD student in Electrical and Computer Engineering at the Ohio State University. His research interests include optimal network control, wireless communication networks, low-delay scheduling scheme design and cross-layer algorithm design.



Atilla Eryilmaz (S '00-M '06) received his M.S. and Ph.D. degrees in Electrical and Computer Engineering from the University of Illinois at Urbana-Champaign in 2001 and 2005, respectively. Between 2005 and 2007, he worked as a Postdoctoral Associate at the Laboratory for Information and Decision Systems at the Massachusetts Institute of Technology. He is currently an Assistant Professor of Electrical and Computer Engineering at the Ohio State University. His research interests include communication networks, optimal control of stochastic networks, optimization theory, distributed algorithms, stochastic processes and network coding. He received the NSF CAREER Award in 2010.