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 an alternative model that enables us to exploit the degree of freedom in choosing appropriate service discipline. Based on the new model, we propose two alternative solutions, first based on a Lyapunov-drift minimization approach, and second based on a novel relaxed optimization-formulation. We provide extensive numerical results to compare the performance of both 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.**

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.

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.

This work was supported in part by DTRA Grant HDTRA 1-08-1-0016, and NSF Awards: CAREER-CNS-0953515 and CCF-0916664.

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. Moreover, the extension to the multi-hop scenario renders some more sophisticated scheduling policies, such as *the Earliest Deadline First* (EDF), inapplicable. In particular, the EDF policy requires the calculation of a threshold level at each hop on the route, which quickly becomes intractable, even with a centralized controller, in the setup of end-to-end deadline constraints.

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:

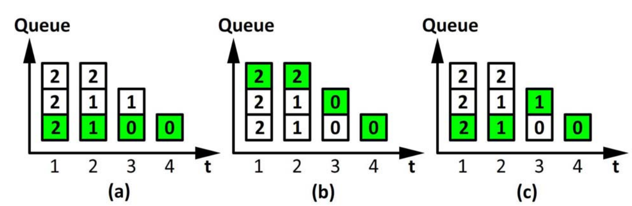


Fig. 1. A example showing the gain we can get by exploiting the service discipline

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%.

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 two different ways, namely the Lyapunov drift minimization approach and a pricing scheme based on a novel optimization formulation out of which two different algorithms are developed.
- We perform extensive numerical studies to compare the performance of both of these solutions. 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. In Section III and IV, we present the Lyapunov drift minimization approach and the optimization based approach we use to solve this problem and the corresponding algorithms respectively. 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 single route $\mathcal{R}^f$ with h^f hops, where $\mathcal{R}^f$ 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 network. 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 is denoted by $A^f[k]$. We make the following standard 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 second moment for all $f \in \mathcal{F}$ and $k \in \mathbb{N}$.

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).

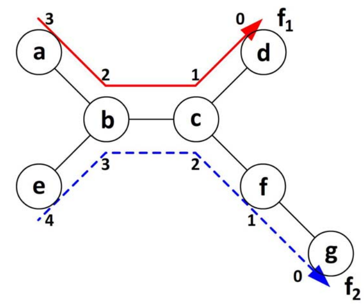


Fig. 2. A simple example: $h_{f_1} = 3$, $h_{f_2} = 4$, $(b, c) = f_1[2] = f_2[3]$

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 $f[i]$, $i = 1, \cdots, h^f$ has its loss probability upper bound of $p^f_{f[i]}$ which, for brevity, we write as $p^f_{[i]}$. Thus, 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 $i = 1, \cdots, h^f$ be the total loss probability upper bound from the source of flow f up to and including link $f[i]$.

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

Assumption 2: The arrival rate of the flows satisfies

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

where $q^f_l = q^f_{[i]}$ if $l = f[i]$. This assumption suggests that there exists a loss probability vector $\{p^f_{[i]}\}_{f,i}$ such that the corresponding minimal delivery ratio requirements on each link l is supportable by the network.

Queueing Architecture: Each link l maintains one queue Q_l^f for each flow f that traverses it. The number of packets in Q_l^f at the beginning of a frame k is denoted by $Q_l^f[k]$. We also define $Q_{l,r}^f[k]$ to be the number of packets in $Q_l^f[k]$ 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

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

In each frame k, a set of decisions $S_l^f[k] = \{S_{l,r}^f[k]\}_r$ is made by link l for each flow f that traverse it, where $S_{l,r}^f[k]$ is the number of packets being served that have r frames-to-go. We have the following capacity constraint on the service:

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

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

$$S_{l,r}^f[k] \le Q_{l,r}^f[k], \forall l, r, k.$$

At the end of the k^{th} frame, we denote $D_{l,r}^f[k]$ to be the number of unserved packets of flow f that has r frames-to-go on link l. It, then, follows from above that $D_{l,r}^f[k] = Q_{l,r}^f[k] - S_{l,r}^f[k]$. By definition, those unserved packets that have r frames-to-go will have $(r-1)$ frames-to-go in the $(k+1)^{st}$ frame, thus we will also refer to $D_{l,r}^f[k]$ as the packets that are being dropped to a lower layer queue. 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 $f[i]$ as

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

which can be concisely written as

$$Q_{[i],r}^f[k+1] = D_{[i],(r+1)}^f[k] + S_{[i+1],(r+1)}^f[k], \tag{2}$$

where

$$\begin{aligned} D_{[i],m^f+1}^f[k] &= 0, \forall i = 1, 2, \cdots, h^f, \forall k, \\ S_{[h^f+1],(r+1)}^f[k] &= 0, \forall r < m^f, \forall k, \\ S_{[h^f+1],(m^f+1)}^f[k] &= A^f[k], \forall k. \end{aligned}$$

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

For flow f, the unserved packets that have 1 frame-to-go at link $f[i]$ are placed into a *deficit queue* $Y_{[i]}^f$, the length of which reflects the amount of missed service for flow f on link $f[i]$. The deficit queue $Y_{[i]}^f$ has a service of $a^f P_{[i]}^f[k]$ at frame k, where $P_{[i]}^f[k]$ is the loss probability upper bound to link $f[i]$ in frame k. It can be either fixed for each frame k, or dynamically or probabilistically chosen, and it to satisfy

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

By stabilizing these deficit queues, we can satisfy the loss probability upper bounds. The deficit queue $Y_{[i]}^f$ evolves as

$$\begin{aligned} &Y_{[i]}^f[k+1] \\ &= \left(Y_{[i]}^f[k] - a^f P_{[i]}^f[k]\right)^+ + D_{[i],1}^f[k] + S_{[i+1],1}^f[k]. \end{aligned} \tag{3}$$

The queueing architecture is illustrated in Fig. 3 for a flow with a deadline of 3 frames traversing a 2-hop network.

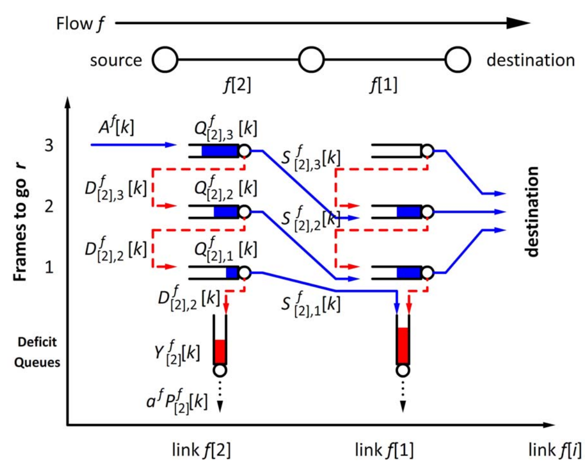


Fig. 3. An example showing the queueing architecture for the stochastic implementation in a 2-hop network with one flow whose deadline is 3 frames.

Remark: Although we refer to the evolution of $Q_{l,r}^f$ as queue evolution, $Q_{l,r}^f$'s are conceptual. In the actual implementation, the "queues" $Q_{l,r}^f$ 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 $Q_{l,r}^f$ cannot grow unboundedly. Thus the $Q_{l,r}^f$'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

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

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

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

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

B. Background

In this subsection, we introduce two well-known scheduling policies, namely the back-pressure (BP) policy (e.g. [1], [18], [19], [4]) and the Largest Weighted Delay First (LWDF) policy (see [20]), that we will use as comparison benchmarks in our numerical studies.

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., 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], [21]). 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 [20] 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 both 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. Lyapunov-Minimization-Based Solution

The above BP policy can be developed by minimizing the drift of a quadratic Lyapunov function of the form $V = \sum_{i,f} (Q^f_i[t])^2$ (e.g., [1], [22]). By minimizing the drift, the BP algorithm is proven to stabilize the queues in the network whenever they are supportable by any other policy. This Lyapunov drift approach is extensively utilized in many other works (e.g., [18], [13], [23], [24]) in different contexts. Since stabilizing the deficit queues in our model can satisfy the delivery ratio constraint, a natural approach to achieve our goal is to define a similar Lyapunov function and minimize its drift.

We define a quadratic Lyapunov function by

$$V(\boldsymbol{Q}, \boldsymbol{Y}, k) \triangleq \frac{1}{2} \sum_{f} \sum_{i=1}^{h^f} \left(\sum_{r=1}^{m^f} (Q^f_{[i],r}[k])^2 + (Y^f_{[i]}[k])^2 \right), \quad (7)$$

where $\boldsymbol{Q}$ is the vector of all queues and $\boldsymbol{Y}$ is the vector of all deficit queues. We try to minimize its drift given by

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

under the capacity constraint (1), where $\boldsymbol{A}$ is the vector of all arrivals. The problem is formulated as:

$$\min_{\{\boldsymbol{S}[k], \boldsymbol{P}[k]\}} \quad \triangle V(\boldsymbol{Q}, \boldsymbol{Y}, \boldsymbol{A}) \quad (8)$$

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

$$\sum_{i=1}^{h^f} P^f_{[i]}[k] \le p^f, \forall f \in \mathcal{F}.$$

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

Proposition 1: The Lyapunov drift minimization problem (8) can be solved by the following policy.

Quadratic Drift Minimizing Policy (QDM Policy):

- **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

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

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 $S_{l,r}^f[k]$'s. In the case of multiple maxima in the set $\{w_{l,r}^f[k]\}$, choose the one with the largest $Q_{l,r}^f[k]$.

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

$$P_{[i]}^f[k] = p^f \mathbf{1}(Y_{[i]}^f[k] = \max_{i'}\{Y_{[i']}^f[k]\}),$$

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 $\{Y_{[i]}^f[k]\}$, divide p^f equally among them.

Proof: The minimization problem (8) can be solved by observing that by swapping the order of summation, $V(\boldsymbol{Q}, \boldsymbol{Y}, k)$ can be rewritten as

$$\begin{aligned} &V(\boldsymbol{Q}, \boldsymbol{Y}, k) \\ =\ & \frac{1}{2}\sum_f \left[\sum_{i=1}^{h^f} \Big(\sum_{r=2}^{m^f} (Q_{[i],r}^f[k])^2 + (Y_{[i]}^f[k])^2 \Big) \right] \\ =\ & \frac{1}{2}\sum_{l\in\mathcal{L}} \left[\sum_{f:l\in\mathcal{R}^f} \Big(\sum_{r=1}^{m^f} (Q_{l,r}^f[k])^2 + (Y_l^f[k])^2 \Big) \right], \end{aligned} \tag{9}$$

and similar for $V(\boldsymbol{Q}, \boldsymbol{Y}, k+1)$.

By substituting (9) into (8), we can expand it and upper-bound the quadratic terms of $Q_{l,r}^f[k]$ and $A^f[k]$ by constants by Assumption 1. Quadratic terms of $S_{l,r}^f[k]$ can be upper bounded by T^2 since $S_{l,r}^f[k] \leq T$ and quadratic terms between $S_{l,r}^f[k]$ and $P_l^f[k]$ can also be bounded similarly. Thus we can simplify (8) to the form of

$$\begin{aligned} &\triangle V(\boldsymbol{Q}, \boldsymbol{Y}, \boldsymbol{A}) \\ \leq\ & B - \sum_{l\in\mathcal{L}} \sum_{f:l\in\mathcal{R}^f} \left[\sum_{r=1}^{m^f} w_{l,r}^f[k] S_{l,r}^f[k] \right] && (10) \\ & - \sum_{f\in\mathcal{F}} \Big(\sum_{i=1}^{h^f} Y_{[i]}^f[k] P_{[i]}^f[k] \Big). && (11) \end{aligned}$$

Note that (10) and (11) are linear functions of $S_{l,1}^f[k]$ and $P_{[i]}^f[k]$, respectively. Thus, minimizing $\triangle V$ can be achieved by assigning the largest possible value to these $S_{l,r}^f[k]$ and $P_{[i]}^f[k]$ with the largest coefficients. ∎

Remark: Although the calculation of the weights $w_{l,r}^f[k]$ appears similar to the back-log difference, it has a different nature: the packets that depart $Q_{l,r}^f$ do not (and will never) enter $Q_{N[l],r}^f$ but instead enter $Q_{N[l],r-1}^f$. 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 next section.

IV. Relaxed-Optimization-Based Solution

In this section, 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., $P_l^f[k] = p_l^f$ for all k. By fixing p_l^f, 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 $Q_{[i],r}^f[k]$ are either being served or remain unserved at the end of the frame k. Thus we have

$$Q_{[i],r}^f[k] = S_{[i],r}^f[k] + D_{[i],r}^f[k],\ \forall k \tag{12}$$

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

$$\begin{aligned} &S_{[i],r}^f[k+1] + D_{[i],r}^f[k+1] \\ &\quad = S_{[i+1],(r+1)}^f[k] + D_{[i],(r+1)}^f[k], \end{aligned} \tag{13}$$

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

$$\max_{\{\boldsymbol{S}[k],\boldsymbol{P}[k]\}} \quad A \tag{14}$$

$$\begin{aligned} s.t. \quad & \text{Flow conservation constraint as in (13)}, \\ & \forall f\in\mathcal{F}, r=1,\cdots,m^f, i=1,\cdots,h^f, \\ & \text{Capacity constraint as in (1)}, \\ & \text{Delivery ratio requirements as in (6)}, \\ & 0 \leq S_{[i],r}^f[k],\ 0 \leq D_{[i],r}^f[k], \\ & \forall f\in\mathcal{F}, r=1,\cdots,m^f, i=1,\cdots,h^f. \end{aligned}$$

We attack this problem by first analyzing a relaxed deterministic model in Section IV-A. 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-B we use the developed insights to implement the algorithm in the stochastic system.

A. 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 constraints. Then, problem (14) reduces to the following problem in this scenario.

$$\max_{\boldsymbol{s},\boldsymbol{d}} \quad A \tag{15}$$

$$\begin{aligned} s.t. \quad & s_{[i],r}^f + d_{[i],r}^f = s_{[i+1],(r+1)}^f + d_{[i],(r+1)}^f, \\ & \forall f\in\mathcal{F}, r=1,\cdots,m^f, i=1,\cdots,h^f, && (16) \\ & \sum_f \sum_{r=1}^{m^f} s_{l,r}^f \mathcal{R}^f[l] \leq T,\ \forall l\in\mathcal{L}, \\ & s_{[i+1],1}^f + d_{[i],1}^f \leq p_{[i]}^f a^f, \\ & \forall f\in\mathcal{F}, i\in\{1,2,\cdots,h^f\}, && (17) \\ & 0 \leq s_{[i],r}^f,\ 0 \leq d_{[i],r}^f, \\ & \forall f\in\mathcal{F}, r=1,\cdots,m^f, i=1,\cdots,h^f, \end{aligned}$$

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

Proposition 2: The relaxed optimization problem (15) 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[1]. The following differential equations describe their evolutions, where we omit 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

$$\begin{aligned} u^f_r &= \begin{cases} \lambda^f_{l,r} - \lambda^f_{N[l],(r-1)}, & \text{if } r > 1 \\ \lambda^f_{l,1} - \mu^f_{N[l]}, & \text{otherwise} \end{cases}, \\ v^f_r &= \begin{cases} \lambda^f_{l,r} - \lambda^f_{l,(r-1)}, & \text{if } r > 1 \\ \lambda^f_{l,1} - \mu^f_l, & \text{otherwise} \end{cases}, \end{aligned}$$

  and the service is allocated according to the following:

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

  where Y_{max} is some fixed positive parameter.

Proof: To solve problem (15), we associate a set of Lagrangian multipliers $\boldsymbol{\lambda} = \{\lambda^f_{[i],r}\}_{f,r,i}$ to the constraint (16) and $\boldsymbol{\mu} = \{\mu^f_{[i]}\}_{f,i}$ to the constraint (17) and construct a Lagrange function as

$$\begin{aligned} &L(\boldsymbol{s}, \boldsymbol{d}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \\ &= A + \sum_{f,i,r} \lambda^f_{[i],r}(s^f_{[i],r} + d^f_{[i],r} - s^f_{[i+1],(r+1)} - d^f_{[i],(r+1)}) \\ &\quad - \sum_{f,i} \mu^f_{[i]}(s^f_{[i+1],1} + d^f_{[i],1} - p^f_{[i]}a^f) \\ &= A + \sum_{f,i,r} \lambda^f_{[i],r}(s^f_{[i],r} - s^f_{[i+1],(r+1)}) - \sum_{f,i} \mu^f_{[i]} s^f_{[i+1],1} \\ &\quad + \sum_{f,i,r} \lambda^f_{[i],r}(d^f_{[i],r} - d^f_{[i],(r+1)}) - \sum_{f,i} \mu^f_{[i]} d^f_{[i],1} \\ &\quad + \sum_{f,i} \mu^f_{[i]} p^f_{[i]} a^f. \end{aligned}$$

Since problem (15) satisfies Slater's condition ([25]) due to Assumption 2, the strong duality holds. Thus we can solve the dual problem

$$\begin{aligned} \max_{\boldsymbol{\lambda},\boldsymbol{\mu}\succeq 0} \quad & \min_{\boldsymbol{s},\boldsymbol{d}} L(\boldsymbol{s}, \boldsymbol{d}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \qquad (18) \\ s.t. \quad & \sum_f \sum_{r=1}^{m^f} s^f_{l,r} \mathcal{R}^f[l] \le T, \forall l \in \mathcal{L}, \\ & 0 \le s^f_{[i],r},\ 0 \le d^f_{[i],r} \\ & \forall f \in \mathcal{F}, r = 1, \cdots, m^f, i = 1, \cdots, h^f. \end{aligned}$$

to get the optimal solution to the primal problem.

Note that by switching the order of summation, we have

$$\begin{aligned} &\sum_{f,i,r} \lambda^f_{[i],r}(s^f_{[i],r} - s^f_{[i+1],(r+1)}) \\ &\qquad = \sum_{l\in\mathcal{L}} \sum_{f:l\in\mathcal{R}^f} \sum_{r=1}^{m^f} s^f_{l,r}(\lambda^f_{l,r} - \lambda^f_{N[l],(r-1)}), \end{aligned}$$

and similarly

$$\begin{aligned} &\sum_{f,i,r} \lambda^f_{[i],r}(d^f_{[i],r} - d^f_{[i],(r+1)}) \\ &\qquad = \sum_{l\in\mathcal{L}} \sum_{f:l\in\mathcal{R}^f} \sum_{r=1}^{m^f} d^f_{l,r}(\lambda^f_{l,r} - \lambda^f_{l,(r-1)}). \end{aligned}$$

Thus, the Lagrange function can be written as

$$\begin{aligned} &L(\boldsymbol{s}, \boldsymbol{d}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \\ &= A + \sum_{l\in\mathcal{L}} \Bigg(\sum_{f:l\in\mathcal{R}^f} \sum_{r=2}^{m^f} s^f_{l,r}(\lambda^f_{l,r} - \lambda^f_{N[l],(r-1)}) \\ &\quad + \sum_{f:l\in\mathcal{R}^f} s^f_{l,1}(\lambda^f_{l,1} - \mu^f_{N[l]}) \Bigg) \\ &\quad + \sum_{l\in\mathcal{L}} \Bigg(\sum_{f:l\in\mathcal{R}^f} \sum_{r=2}^{m^f} d^f_{l,r}(\lambda^f_{l,r} - \lambda^f_{l,(r-1)}) \\ &\quad + \sum_{f:l\in\mathcal{R}^f} d^f_{l,1}(\lambda^f_{l,1} - \mu^f_l) \Bigg) + \sum_{f,i} \mu^f_{[i]} p^f_{[i]} a^f. \end{aligned}$$

Hence, the decision for each link l can be made separately. Using the gradient algorithm ([25], [26]) to solve the dual problem (18), we obtain the above policy. ■

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 requirement, 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

[1]A discrete-time version of this algorithm can be derived using the technique as in [2], [19]

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 $m^f = 12$ 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.

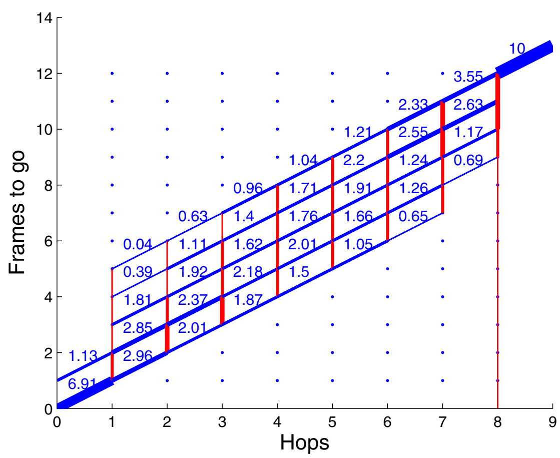


Fig. 4. OB Policy runs in a 8-hop network with a 12-frame deadline.

Fig. 4 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.

B. Implementation in the Stochastic System

The OB Policy above is derived based on a relaxation where the constraint (16) is relaxed by associating a price to it. This relaxation enables us to solve the dual problem, but unlike prior works (e.g., [2], [19]), 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 (12). However, in the relaxed model, these two quantities are only loosely coupled via the pricing mechanism. In other words, the flow conservation constraint (13) 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. The newly defined queueing architecture is the same as shown in Fig. 3, but unlike those in the original model, all the queues are able to build up.

The queue-length evolution for $\hat{Q}^f_{[i],r}$ is then given by

$$\begin{aligned}\hat{Q}^f_{[i],r}[k+1] &= \Big(\hat{Q}^f_{[i],r}[k] - (S^f_{[i],r}[k] + D^f_{[i],r}[k])\Big)^+ \\ &\quad + S^f_{[i+1],r+1}[k] + D^f_{[i],r+1}[k]. \qquad (19)\end{aligned}$$

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

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

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

  and the service is allocated according to the following:

$$\begin{aligned}S^f_{l,r}[k] &= \begin{cases} T & \text{if } (f,r) = \operatorname{argmax}_{f,r}\{u^f_r\} \\ 0 & \text{otherwise} \end{cases}, \\ D^f_{l,r}[k] &= \Big(\hat{Q}^f_{l,r}[k] - S^f_{l,r}[k]\Big)^+ \mathbf{1}(v^f_r > 0).\end{aligned}$$

- **Queue Evolution:** In the actual layer the level queues evolve as in (19) 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 $\hat{Q}_{l,r}^{f}[k]$ to dynamically determine the maximum number of drops $D_{l,r}^{f}[k]$. 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. We shall see in our numerical studies in Section V that the fraction of delivered but expired packets varies in different cases.

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.

The simulations are run for a single flow f in a 4-hop tandem network with frame length $T = 12$. Flow f has a 6-frame end-to-end deadline requirement and independent and identically distributed (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 $p_{[4]}^{f} = 0.25$.

A. Service Discipline Choice

We first present the service discipline chosen by the QDM the DOB Policies. In this simulation, we choose $a^f = 10$ with the aforementioned distribution. 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.

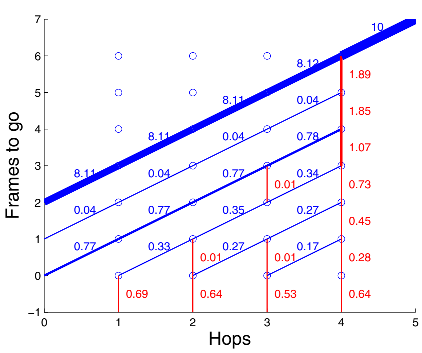


Fig. 5. The average service rate of the QDM Policy of packets with different time to go

Fig. 5 shows the average service 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 are 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.

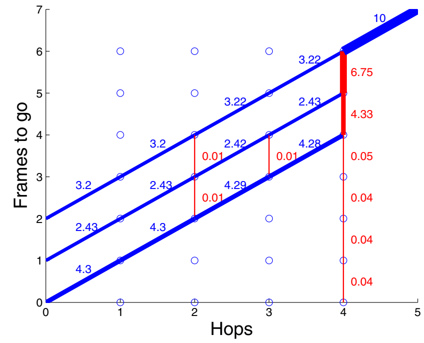


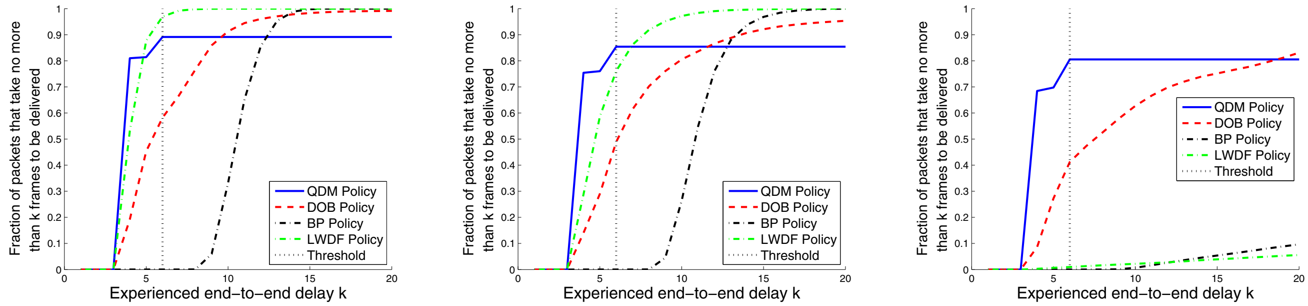
Fig. 6. The average service rate of the DOB Policy of packets with different time to go

As a comparison, Fig. 6 shows the service 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 (19) 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.

B. QoS Performance

In this subsection, 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. 7 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 in all three cases, which does not distinguish the difference in service discipline. It can also be observed that 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. As the arrival rate increases from $a^f = 10$ in Fig. 7(a) to the heavily loaded level of $a^f = 11.99$ in Fig. 7(c), 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. This demonstrates the adaptive nature of both policies to the traffic requirements and the network conditions, a property lacking by the BP policy and the LWDF policy.



(a) (b) (c)

Fig. 7. The QoS performance comparison for (a) $a^f = 10$, (b) $a^f = 11$, (c) $a^f = 11.99$

VI. Conclusion

In this work, we consider 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 propose a novel queueing architecture which allows us to exploit the freedom of choosing service discipline when serving packets. Based on this new model, we develop two algorithms using different approaches. The first is based on a Lyapunov-drift minimization method, and the second is based on an optimization approach. We perform extensive numerical studies to illustrate the advantages and disadvantages to both of the algorithms.

Future research of this topic includes the following: (i) As we have observed in our numerical studies, the QDM Policy and the DOB Policy have their advantages and disadvantages respectively. It needs more work to develop an algorithm that combines both advantages. (ii) Both of our algorithms treat a frame as a whole rather than utilize the slots in a frame separately and make scheduling decisions accordingly. Fully utilizing this setup may enable our algorithms to achieve better performance. (iii) In this work we assume each flow to be associated with a single route. We can relax this assumption by studying the case when the flows have multiple routes.

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

[19] 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.

[20] 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.

[21] 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.

[22] M. Neely, E. Modiano, and C. Rohrs, "Dynamic power allocation and routing for time varying wireless networks," in *Proceedings of IEEE Infocom*, April 2003, pp. 745–755.

[23] 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.

[24] 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.

[25] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge: Cambridge University Press, 2004.

[26] R. Srikant, *The Mathematics of Internet Congestion Control*. Boston, MA: Birkhäuser, 2004.