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

Haozhi Xiong, Ruogu Li, Atilla Eryilmaz and Eylem Ekici

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

I. INTRODUCTION

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

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

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

• We propose a novel design paradigm that decouples the objectives of long-term utility maximization from the end-to-end delay-aware resource allocation. This framework is expected to enable the systematic development of future schemes, in addition to the one we develop in this paper.

• We reveal the solution given by the back-pressure policy has a unique structure which facilitate us to establish loop-free multi-path routes that guarantee long-term network utility maximization. This new routing-scheme not only inherits the adaptive and optimal nature of the long-term optimal algorithms, but also eliminates the unnecessary loops to reduce the end-to-end delay without sacrificing from throughput.

• We combine this loop-free route construction strategy with a token-based scheduling discipline that regulates the higher-order statistics of service processes to achieve drastic reductions in the end-to-end delay performance, while guaranteeing long-term optimality characteristics.

The above points also constitute the main differences of this work from the recent efforts in the design of algorithms with low end-to-end delay performance (e.g. [9], [10]). We also note that there has been recent interest in deriving fundamental bounds on the delay performance ([11], [12], [13]). In our future work, we are interested in utilizing and extending these results to study the gap between these fundamental bounds and the delay performance of our algorithms.
The remainder of the paper is organized as follows. Section II introduces our system model and objectives together with the description of several existing algorithms. In Section III, we propose our delay-aware design framework that describes the desired characteristics and interconnections of the routing, scheduling, and flow control components of the network algorithm. In Section IV, we build upon the proposed design framework to construct a delay-aware cross-layer algorithm that performs the desired tasks. The numerical results of the policy and our concluding remarks are provided in Section V and VI respectively.

II. SYSTEM MODEL AND OBJECTIVE

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

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

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

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

A. Objective

Our goal is to develop a joint flow-control, scheduling, and routing algorithm that not only optimizes long-term network utilization, but also provides desirable delay characteristics. We discuss these two goals next.

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

$$\max_{\{X[t], R[t]\}} \sum_{s,d} U_{sd}(x_{sd}^d)$$

s.t.  
$$X_{sd}^d[t] \geq 0, \forall s, d \in \mathcal{N}, \forall t \geq 0,$$
$$R_{ij}^d[t] \geq 0, \forall (i,j) \in \mathcal{L}, \forall t \geq 0,$$
$$\sum_d R_{ij}^d[t] + \sum_d R_{ji}^d[t] \leq c_{ij},$$
$$\forall (i,j) \in \mathcal{L}, \forall d \in \mathcal{D}, \forall t \geq 0,$$
$$x_{i}^d + \sum_{(m,i) \in \mathcal{L}} r_{mi}^d \leq \sum_{(i,j) \in \mathcal{L}} r_{ij}^d,$$
$$\forall i \in \mathcal{N}, \forall d \in \mathcal{D}, i \neq d. \tag{5}$$

We define the feasible solution as:

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

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

$$x^* = \lim_{t \to \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} X^* [\tau], \tag{6}$$
$$r^* = \lim_{t \to \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} R^* [\tau]. \tag{7}$$

Note that under Assumption 1, $x^*$ is unique while $r^*$ generally belongs to a set of optimal link rates denoted by $\mathcal{R}^*$. In this paper, we define the stability as follows:

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

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

B. Background

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

The Discrete Time Back-Pressure (DTBP) Policy:

- Queue (Price) Evolution:
  Each queue evolves as
  $$P_{ij}^d[t+1] = \left( P_{ij}^d[t] - \sum_{(i,j) \in \mathcal{L}} R_{ij}^d[t] \right)^+ + X_{ij}^d[t] + \sum_{(m,i) \in \mathcal{L}} R_{mi}^d[t] \tag{8}$$
• Rate control:
In time slot $t$, the flow controller determines the mean of the injection as
$$E(X_s^d[t] | P_s^d[t]) = \min \{U_{sd}^{-1}(P_s^d[t]/K), X_{\text{max}} \},$$
where $X_{\text{max}}$ is some finite upper bound for the arrival rate.

• Scheduling and Routing:
Each link $l = (i, j)$ selects
$$d^*_l[t] = \arg\max_d |P_i^d[t] - P_j^d[t]|, \ l \in \mathcal{L},$$
and assign
$$R_{ij}^d[t] = \begin{cases} c_{ij} \text{ if } d = d^*_l[t] \text{ and } P_i^d[t] - P_j^d[t] > 0; \\ 0 \text{ otherwise.} \end{cases}$$

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

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

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

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

Remark 3: The price evolution in Equation (8) implies that in the case of a node has less packets than scheduled service, dummy packets are transmitted. Thus the allocated service $R_{ij}^d[t]$ always equals to number of packets transmitted over link $(i, j)$ at time $t$ as suggested in Equation [5].

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

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

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

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

![Fig. 1. The distribution of hop-count for flow 1 packets indicates the presence of unnecessary loops in their routes](image)

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

As a simple example, it is well-known ([15]) that in a G/G/1 queueing system, the mean waiting time $W$ is bounded by
$$W \leq \frac{\sigma^2 + \sigma^2_0}{2\bar{t}(1 - \rho)},$$
where $\sigma^2_0$ is the variance of inter-arrival time, $\sigma^2$ is the variance of service time, $\bar{t}$ is the mean of inter-arrival time, and $\rho$ is the utility factor. This formulation suggests that reducing the variance of the arrival and service processes is crucial in reducing the delay.

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

C. Proposed multi-layer design framework

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

![Multi-Layer Design Framework](image)

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

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

IV. DELAY-AWARE ROUTING AND SCHEDULING

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

A. Delay-aware route construction

To overcome the disadvantages observed in the back-pressure algorithm, a natural idea is to restrict the direction in which packets can be transmitted over a link for a certain commodity. To achieve this, we first introduce a fluid approximation associated with $\tilde{R}_{ij}^d[t]$ for the following discussion.

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

s.t. $x_s^d \geq 0, \forall s, d \in \mathcal{N}$,  
$r_{ij}^d \geq 0, \forall (i, j) \in \mathcal{L}$,  
$\sum_d r_{ij}^d + \sum_d r_{ji}^d \leq c_{ij}, \forall (i, j) \in \mathcal{L}, d \in \mathcal{D}$,  
$x_i^d + \sum_{(m,i) \in \mathcal{L}} r_{mi}^d \leq \sum_{(i,j) \in \mathcal{L}} r_{ij}^d, \forall i \in \mathcal{N}, \forall d \in \mathcal{D}, i \neq d$. (16)

Note that if an optimal solution of (11) is $\{\tilde{X}^*[t], \tilde{R}^*[t]\}$, it is shown in previous works (e.g. [14], [5]) that the associated $(x^*, r^*)$ defined in (6) and (7) is an optimal solution of (12).

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

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

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

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

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

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

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

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

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

**Remark:** Note that under this link rate assignment, for each $(i,j) \in \mathcal{L}$, at least one of $\hat{r}_{ij}^d$ and $\hat{r}_{ij}^d$ equals to zero as if the corresponding bidirectional link becomes a unidirectional link. Clearly, those links with equal rates on both directions will have a net rate of 0. Also, note that this link rate

2 More precisely, $(x^*, r^*)$ is within $O(1/K)$ of the optimal solution of (12). Nevertheless, we shall call $(x^*, r^*)$ an optimal solution of (12) since the converge can be precise if a diminishing step-size (in our notation, if $K$ grows linearly with time $t$) is used.
mapping preserves the optimality of \((\hat{x}^*, \hat{r}^*)\) when applied to any optimal solution \((x^*, r^*)\) to problem (1) which is not necessarily given by the DTBP algorithm.

**The Steady-State Behavior of the DTBP Algorithm:** Previous works (e.g. [12]) also show the stability of the DTBP algorithm. In other words, the Markov chain with the queue-lengths \(P = (P^d_{ij})\) being its states is positive recurrent and ergodic under the DTBP algorithm. We use \(\pi(P)\) to denote the steady state distribution under the DTBP algorithm, i.e., the probability of the queue-lengths being \(P\) is \(\pi(P)\) after the convergence of the DTBP algorithm and \(\sum_P \pi(P) = 1\). Also, for queue \(P^d_{ij}\), the following holds:

\[
\lim_{t \to \infty} \frac{1}{t} \sum_{\tau=0}^{t} \mathbb{E}(P^d_{ij}[\tau]) = \sum_P P^d_{ij} \pi(P),
\]

and we use

\[
P^d_{ij} = \sum_P P^d_{ij} \pi(P)
\]

to denote the optimal average queue-length of queue \(P^d_{ij}\).

Intuitively, the link rate assignment equation (10) of the DTBP algorithm implies that if there is a positive net flow rate from node \(i\) to node \(j\) for destination \(d\), then the average queue-lengths should be strictly decreasing, i.e., \(P^d_{ij} > P^d_{ji}\), since \(i\) can send a packet to \(j\) only when \(P^d_{ij}[t] > P^d_{ji}[t]\).

In certain simple topologies, this statement can be proven rigorously. For a tandem network (a network whose adjacency matrix has non-zero elements only on the first diagonal above the main diagonal and the first diagonal below the main diagonal, see Fig. 3 for an example) with unit link capacities and one end-to-end flow, we have the following results:

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

**Proof:** The detailed proof is provided in Appendix B.

![Fig. 3. A tandem network with \(N - 1\) hops.](image)

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

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

**Proof:** The detailed proof is provided in Appendix C.

Due to the complex interactions in the dynamics of the queue-length in the neighboring nodes in a general network, a rigorous proof of a generalization of Lemma 2 remains an open research problem. However it can be observed from numerical studies of more general networks that, under the DTBP algorithm, the average queue-lengths are strictly decreasing over links with a positive net flow rate. Fig. 4 shows a typical example in a \(6 \times 6\) grid network. The arrows in the figure indicate the direction of the net flow, and the average queue-length of the flow in each node is shown by the number besides the node.

![Fig. 4. The net flow rate and the average queue-length in a \(6 \times 6\) grid network. The average queue-lengths are strictly decreasing in the direction of the net flow.](image)

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

**Assumption 2:** Under our system model, the DTBP algorithm converges to a optimal solution of (1) \(\{\bar{X}^*, \bar{R}^*\}\) and the corresponding queue evolution \(\bar{P}^*\) as defined in (8). The optimal solutions satisfy the following property:

If \(r_{ij}^d > r_{ji}^d\) then \(\bar{P}_{ij}^d > \bar{P}_{ji}^d\), \(\forall (i,j) \in L, d \in D\),

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

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

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

\[
\tilde{N}^d = N; \quad \tilde{\mathcal{L}}^d = \mathcal{L} \setminus \{(i,j) \in \mathcal{L} : l_{ij}^d = 0\}; \quad \tilde{c}_{ij}^d = \begin{cases} 0 & \text{if } (i,j) \notin \tilde{\mathcal{L}}^d \text{ and } (j,i) \notin \tilde{\mathcal{L}}^d; \\ c_{ij} & \text{otherwise}. \end{cases}
\]

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

Then we have the following proposition:

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

**Proof:** Assume \(\tilde{G}^d\) has a cycle that is composed of links \(\{(n_{c_1}, n_{c_2}), (n_{c_2}, n_{c_3}), \ldots, (n_{c_{k-1}}, n_{c_k}), (n_{c_k}, n_{c_1})\} \subseteq \tilde{\mathcal{L}}^d\). By the definition of subgraph \(\tilde{G}^d\), we can assume without loss of generality that \(\tilde{r}_{c_i,c_{i+1}}^d > 0\) for \(0 \leq i \leq k - 1\) and \(\tilde{r}_{c_k,c_1}^d > 0\).
Then by Assumption 2 we have
\[ \hat{P}_{e_1} > \hat{P}_{e_2} > \cdots > \hat{P}_{e_k} > \hat{P}_{e_1}, \]
which is impossible. Therefore \( \hat{d} \) is loop-free.

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

**Example:** We use the topology shown in Fig. 5(a) as an example to illustrate how the delay-aware link rate assignment works with the optimal solution given by DTBP. Each link in the network is assumed to have unit capacity. Node 1 is the source and Node 3 is the destination.

![](fig5.png)

**Fig. 5.** An example to illustrate the DTBP solution with delay-aware link rate assignment.

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

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

**B. Delay-aware scheduler design**

Using the delay-aware routing of Section [IV-A] utility maximization can be achieved while avoiding loops, which leads to lower average end-to-end delays. Here, we seek to improve the delay performance further through delay-aware scheduling at each link. As argued in Section III-B significant per hop delay improvement can be achieved by reshaping the service distribution through regulating the scheduler operation. In previous works [16], the notion of regulators were introduced to help stabilize the network with fixed single-path routing. Adopting a similar approach, we propose a delay-aware scheduler with regulated service that stabilizes \((\mathcal{N}, \mathcal{L}, c)\) when there exists a loop-free \( R[t] \) such that \( X[t] \) satisfies \( x \in \text{int}(\Lambda(\mathcal{N}, \mathcal{L}, c)) \). We assume that the arrivals take place after the service in each time slot, i.e., the arrivals cannot be served in the arriving time slot. We further assume that from now on all links have the same link capacity denoted by \( c \).

\[
A_{n}^{d}[t] = \frac{\tilde{R}_{n,j}^{d}[t]}{\sum_{m \in \mathcal{N}} R_{n,m}^{d}[t]},
\]

where \( \mathcal{N}_{\text{source}}^{d} \) denotes the source node set with respect to commodity \( d \). Let \( A_{n,j}^{d}[t] \) denote the number of packets of commodity \( d \) arriving at node \( n \) in slot \( t \). Mathematically,

\[
A_{n}^{d}[t] = X_{n}^{d}[t] \mathbb{1}_{\{n \in \mathcal{N}_{\text{source}}^{d}\}} + \sum_{k \in \mathcal{N}} D_{kn}^{d}[t],
\]

where \( N_{\text{source}}^{d} \) denotes the source node set with respect to commodity \( d \). Let \( A_{n,j}^{d}[t] \) denote the number of packets of commodity \( d \) arriving at node \( n \) in slot \( t \) and \( A_{n}^{d}[t] = \sum_{j \in \mathcal{N}} A_{n,j}^{d}[t] \). The queue dynamics for each queue is given as:

\[
Q_{n,j}^{d}[t+1] = \left( Q_{n,j}^{d}[t] - D_{r,n,j}^{d}[t] \right)^{+} + A_{n,j}^{d}[t],
\]

\( \forall n \neq d, \) \hspace{1cm} (22)

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

- **Arrival Splitting:**
  Each commodity \( d \) packet arriving at node \( n \) goes into the per-neighbor-per-commodity queue \( Q_{n,j}^{d} \) with probability \( \frac{R_{n,j}^{d}[t]}{\sum_{m \in \mathcal{N}} R_{n,m}^{d}[t]} \), resulting in

\[
\mathbb{E} \left[ A_{n,j}^{d}[t] \big| A_{n}^{d}[t] \right] = A_{n}^{d}[t] \frac{\tilde{R}_{n,j}^{d}[t]}{\sum_{m \in \mathcal{N}} R_{n,m}^{d}[t]},
\]

where \( \tilde{R}_{n,j}^{d}[t] = \frac{1}{\mathcal{L}} \sum_{\tau=0}^{t-1} R_{n,j}^{d}[\tau] \). This splitting preserves the same mean link rates as the original \( R[t] \). As an example, the arrival splitting can be done using a token-based scheme.

- **Token Generation:**
  For each link \( (i,j) \in \mathcal{L} \), the system maintains token counters \( m_{i,j}^{d} \) for each regulator queue \( Q_{n,j}^{d} \) as shown in Fig. 6. As the input to the delay-aware scheduler is
loop-free, the transmissions can only be unidirectional for certain commodity \( d \) over link \((i, j)\). Hence \( m_{nj}^d \) and \( m_{jn}^d \) cannot be non-zero simultaneously. Then we define 
\[
M_{nj}^d = m_{nj}^d + m_{jn}^d .
\]
The token arrivals at each counter with rate 
\[
S_{nj}^d = E[A_n^d[t]] \sum_{m \in N} R_{nm}^d[t] + \delta ,
\]  
where \( \delta > 0 \). Note that the token arrival rates are enlarged by \( \delta \) which is allowed by the excess capacity from the utility maximizing solution for \((N, L, c - \epsilon)\).

- **Service:**
  In each time slot \( t \), the token-based server chooses a winner commodity \( d_{nj}^* \) with a non-zero backlog for each link \((n,j)\) as follows:
  \[
d_{nj}^*[t] = \begin{cases} 
\arg\max_d (M_{nj}^d[t] - c) , & \text{if } \max_d (M_{nj}^d[t] - c) > 0 ; \\
\emptyset , & \text{otherwise}. 
\end{cases}
\]  
  The server serves the commodity with the largest token count that exceeds the link capacity \( c \). Note that the link capacity \( c \) is shared by transmissions on link \((n,j)\) in both directions. Then
  \[
  D_{nj}^d[t] = \begin{cases} 
c , & \text{if } d = d_{nj}^* ; \\
0 , & \text{if } d \neq d_{nj}^* .
\end{cases}
\]
  In case of insufficient packets in queue, the server sends out dummy packets to ensure a departure of \( c \) packets. After service, the token counter \( M_{nj}^d \) for the commodity \( d_{nj}^* \) is decreased by \( c \).

The above algorithm does not require the knowledge of \( R[t] \) as long as \( R[t] := (R_{nj}^d[t])_{n,j,d} \) is provided.

**Definition 5:** (Token count process) We call \( M[t] \) a token count process if:

\[
M[t + 1] = \begin{cases} 
M[t] + \nu , & \text{if } M[t] + \nu < c^{th} ; \\
M[t] + \nu - c^{th} , & \text{if } M[t] + \nu \geq c^{th} .
\end{cases}
\]  
\[
Z[t + 1] = \begin{cases} 
0 , & \text{if } M[t] + \nu < c^{th} ; \\
c^{th} , & \text{if } M[t] + \nu \geq c^{th} .
\end{cases}
\]

where \( Z[t] \) is the associated token process, \( 0 < \nu < c^{th} \) is the token generation rate and \( c^{th} \) is the decision threshold.

Following our definitions of delay-aware scheduler, it is clear that the service process of each scheduler is a token count process. We have the following lemma showing important properties of the token count process, that is, given a finite initial state, the token count process evolves towards a bounded invariant region.

**Lemma 3:** Let \( M[t] = (M_1[t], \ldots, M_n[t]) \) be the token count vector with token generation rate \( \nu = (\nu_1, \ldots, \nu_n) \) and decision threshold \( c^{th} \). Let \( D_i[t] \) be the departure from token counter \( i \) at time \( t \), \( \forall 1 \leq i \leq n \). The delay-aware scheduler serves \( M[t] \) with the following service discipline:

\[
i^*[t] = \begin{cases} 
\arg\max_i (M_i[t] - c^{th}) , & \text{if } \max_i (M_i[t] - c^{th}) > 0 ; \\
\emptyset , & \text{otherwise}. 
\end{cases}
\]  
\[
D_i^*[t] = \begin{cases} 
c^{th} , & \text{if } i = i^* ; \\
0 , & \text{if } i \neq i^* .
\end{cases}
\]

The token count dynamics is as follows:

\[
M_i[t + 1] = M_i[t] - D_i^*[t] + \nu_i , \quad \forall 1 \leq i \leq n .
\]

Let \( M = \{ M \in \mathbb{R}^n : \sum_{i=1}^n M_i < (n + 1)c^{th} \} \). Suppose \( \sum_{i=1}^n \nu_i < c^{th} \), then \( \forall t > 0 \) such that \( \forall M[t_0] \in \mathbb{R}^n \), there exists \( T < \infty \), such that \( M[t_0 + T] \in M \). Moreover, \( M[t] \in M \), \( \forall t \geq t_0 + T \).

**Proof:** The detailed proof is provided in Appendix [D1].

Intuitively, Lemma [3] reveals the fact that the token count process keeps track of the assigned token generation rate and the deviation is always bounded. Lemma [3] will help us in proving the network stability when the delay-aware schedulers are used. Next we define the network capacity region for multi-hop networks.

**Definition 6:** (Network capacity region) The network capacity region is \( \Lambda(N, L, c) = Co\{x\} \), where \( x \) is the exogenous arrival rate vector such that \( x \geq 0 \) and such that there exists a link rate vector \( r \) satisfying

\[
r \geq 0 ,
\]  
\[
r_{nj}^d = 0 , \forall (n,j) \notin L ,
\]  
\[
r_{nn}^d = 0 , \forall n \in N ,
\]  
\[
r_{nj}^d = 0 , \forall n = d ,
\]  
\[
x_n^d \leq \sum_j r_{nj}^d - \sum_n r_{mn}^d , \forall n \neq d ,
\]  
\[
\sum_d r_{nj}^d + \sum_d r_{jn}^d \leq c_{nj} , \forall (n,j) \in L .
\]

Notice that the solution \( (X[t], R[t]) \) is feasible if \( (x, r) \) satisfies Definition [6] such that \( x \in \Lambda(N, L, c) \), where \( x = \lim_{t \to \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} X[\tau], r = \lim_{t \to \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} R[\tau] \).

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

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

**Proof:** The detailed proof is provided in Appendix [D2].

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

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

Delay-Aware Cross-Layer Policy

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

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

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

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

V. Numerical Results

In this section, we present numerical results for our proposed delay-aware cross-layer policy. The cross-layer policy is an online policy starting to function as DTBP is converging. All layers run different algorithms parallely and perform periodical solution updates in implementation. We compare the delay performance of our policy with the back-pressure policy, which has long term optimality guarantees, and the more recent min-resource algorithm (see Section II-B), which has superior routing characteristics over back-pressure policy.

As a representative topology, simulations are carried out in a 6 × 6 grid network. Simulation duration is 30000 time slots. Moving average method is used to update \(\hat{R}[t]\) as follows:

\[
\hat{R}_{nij}[t] = \left[ \frac{1}{W} \sum_{\tau = t_0}^{t_0 + W} (R_{nij}[\tau] - R_{nj}[\tau]) \right]^{+}, \\
t_0 \leq t < t_0 + T,
\]

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

Two representative scenarios are considered to demonstrate the performance difference of the three algorithms. **Scenario 1** has parallel traffics with two commodities. **Scenario 2** contains cross traffics where each source node maintains throughput up to 3 compared to 2 in Scenario 1. The links in the center 4 × 4 square are fully utilized. Scenario 2 has less excess capacity compared to Scenario 1.

For better demonstration, the routing information is embedded in delay distribution graph. Since the source-destination pairs are symmetric, results are given for one commodity only. Squares and circles indicate source and destination locations for commodity 1 and 2, respectively, both flowing from top to bottom. In Scenario 1, as is shown in Fig. 7(a), the back-pressure policy results in loops. The delay distribution is heavy-tailed which also indicates that packets may experience looping. Fig. 7(b) demonstrates that the min-resource algorithm utilizes minimal number of links, thus attaining shortest paths from source to destination for each commodity. Since loops are avoided, the delay distribution is improved with the delay peaks moved towards lower values. Fig. 7(c) shows the network topology for delay-aware cross-layer policy. The selected paths are guaranteed to be loop-free. The same figure also shows that the delay distribution is multimodal. From this distribution, one observes that the majority of packets follow the shortest path and the others are routed over longer paths. The mean delay performance is improved over other two solutions.

In Scenario 2, we demonstrate a situation where the network is heavily utilized. The existence of excess capacity is a necessary condition for back-pressure policy to form loops. Within a network with less excess capacity, back-pressure policy has less resources for loops, which leaves less chance for routing improvement. Fig. 8(a) shows the back-pressure result for such a network with increased throughput. In Fig. 8(b), we observe no improvement for min-resource algorithm since the network is already heavily loaded. Min-resource algorithm cannot further minimize resource usage without sacrificing throughput. In Fig. 8(c), we still observe significant delay improvements in the mean and variance of delay. The result suggests that the delay-aware cross-layer policy is effective in reducing delay regardless of network utilization.

We further study the effect of the design parameter \(K\) used in the DTBP policy. It is known that the allocated flow rates approach the optimal rate \(x^*\) as \(K \to \infty\). However, this also increases the equilibrium queue lengths in the DTBP and causes delay degradation (17), (18), etc.)

The delay-aware cross-layer policy offers a solution to this dilemma. In multi-layer architecture, the delay-aware cross-layer policy keeps real layer queue length separated from back-pressure layer queue length. Then the dependence on \(K\) is relaxed. Fig. 9 shows that in Scenario 1, the sensitivity of the delay on \(K\) is significantly reduced with delay-aware cross-layer policy. As expected, the delay under back-pressure
VI. CONCLUSION

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

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

REFERENCES

[1] L. Tassiulas and A. Ephremides, “Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks,” IEEE Transactions on Automatic Control, vol. 36, pp. 1936–1948, December 1992.
[2] X. Lin and N. Shroff, “Joint rate control and scheduling in multihop wireless networks,” in Proceedings of IEEE Conference on Decision and Control, Paradise Island, Bahamas, December 2004.
[3] A. Stolyar, “Maximizing queueing network utility subject to stability: Greedy primal-dual algorithms,” Queueing Systems, vol. 50, no. 4, pp. 401–457, 2005.
Noticing that for any real number $Z$, $(Z)^+ = \frac{1}{2}(Z + |Z|)$, Equations (18) and (19) can be rewritten as:

$$\hat{r}_{ij}^{sd} = \frac{1}{2} \left( (r_{ij}^{sd} - r_{ij}^d) + |r_{ij}^{sd} - r_{ij}^d| \right);$$

$$\hat{r}_{ji}^{sd} = \frac{1}{2} \left( (r_{ji}^{sd} - r_{ji}^d) + |r_{ji}^{sd} - r_{ji}^d| \right).$$

Subtracting the above two equations, we have

$$\hat{r}_{ij}^{sd} - \hat{r}_{ji}^{sd} = r_{ij}^{sd} - r_{ji}^{sd}. \quad (28)$$

Thus, the following equation holds:

$$\hat{x}_i^{sd} = x_i^{sd} + \sum_{(m,s) \in \mathcal{L}} \hat{x}_{mi}^{sd} - \sum_{(s,j) \in \mathcal{L}} \hat{x}_{ij}^{sd} \quad (a)$$

$$\implies \hat{x}_i^{sd} + \sum_{j} (\hat{r}_{ij}^{sd} - \hat{r}_{ji}^{sd}) \quad (b)$$

$$x_i^{sd} + \sum_{j} (r_{ij}^{sd} - r_{ji}^{sd}) \leq 0,$$

where (a) follows from (28) and (b) follows from (27).

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

### B. Proof of Lemma 7

Lemma 1 can be proven by constructing the queue-evolution Markov chain under the DTBP algorithm and studying the steady-state behavior of the queue-length evolution from zero state.

To simplify the notation, we denote the nodes as node 0 to node $n$ in a $n$-hop tandem network, where node 0 is the destination and node $n$ is the source. Since only one flow is assumed to exist, we omit the superscript $d$ indicating the destination of the flow.

**Case 1:** $n = 1$. In this case, node 1 receives a packet with probability $a$ in each time-slot. Whenever node 1 has one packet, it sends the packet to the destination. Thus $Prob\{P_1 = 1\} = a$, and $Prob\{P_1 = 0\} = 1 - a$, thus

$$P_1^* = a > 0 = \bar{P}_0^*$$

when $a \in (0, 1)$.

**Case 2:** $n = 2$. For brevity, we denote the state of the Markov chain $(P_2, P_1)$ as in the Table I. Since the Markov chain itself is messy to plot, we only present its state transition matrix

$$A = \begin{bmatrix}
    b & a & \cdot & \cdot & \cdot \\
    \cdot & b & a & \cdot & \cdot \\
    \cdot & b & a & \cdot & \cdot \\
    \cdot & b & a & \cdot & \cdot \\
    \cdot & b & a & \cdot & \cdot 
\end{bmatrix}$$
where \( b = 1 - a \), and we use “\( \cdot \)" to represent 0 for clarity. Solve the equation

\[
A\pi = \pi
\]

and the probability constraint

\[
\sum_{i} \pi_{i} = 1
\]

to get the steady-state distribution of this Markov chain, and the average queue-length can be calculated by

\[
\bar{P}_{1}^{*} = \pi_{2} + \pi_{4} + \pi_{6},
\]
\[
\bar{P}_{2}^{*} = \pi_{3} + \pi_{4} + 2(\pi_{5} + \pi_{6}).
\]

It can be verified that

\[
\bar{P}_{2}^{*} > \bar{P}_{1}^{*} > \bar{P}_{0}^{*} = 0
\]

when \( a \in (0, 1) \). Shown in Fig. 11 is the average queue-lengths as a function of \( a \) in a 2-hop tandem network.

Fig. 10. The average queue-length in a 2-hop tandem network under Bernoulli(\( a \)) arrival. The queue-lengths are strictly decreasing from source to destination.

**Case 3:** \( n = 3 \). For brevity, we denote the state of the Markov chain \((P_{3}, P_{2}, P_{1})\) as in the Table II. With this numbering of the states, the state transition matrix is given by

\[
A = \begin{bmatrix}
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & b & a & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & b & a & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & b & a & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & b & a & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & b & a & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & b & a & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & b & a & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & b & a & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & b & a & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & b & a & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & b & a & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & b & a & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & b & a & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & b & a \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & b
\end{bmatrix},
\]

where \( b = 1 - a \), and we use “\( \cdot \)" to represent 0 for clarity. Solve the equation

\[
A\pi = \pi
\]

and the probability constraint

\[
\sum_{i} \pi_{i} = 1
\]

to get the steady-state distribution of this Markov chain, and the average queue-length can be calculated by

\[
\bar{P}_{1}^{*} = \pi_{3} + \pi_{4} + \pi_{5} + \pi_{6} + \pi_{8} + \pi_{10} + \pi_{12},
\]
\[
\bar{P}_{2}^{*} = \pi_{2} + \pi_{3} + \pi_{4} + \pi_{5} + \pi_{6} + \pi_{9} + \pi_{10} + \pi_{12} + \pi_{13} + \pi_{14} + \pi_{15},
\]
\[
\bar{P}_{3}^{*} = \pi_{1} + \pi_{2} + \pi_{3} + \pi_{4} + \pi_{5} + \pi_{6} + \pi_{9} + \pi_{10} + \pi_{12} + \pi_{13} + \pi_{14} + \pi_{15}.
\]

It can be verified analytically or numerically that

\[
\bar{P}_{3}^{*} > \bar{P}_{2}^{*} > \bar{P}_{1}^{*} > \bar{P}_{0}^{*} = 0
\]

when \( a \in (0, 1) \). Fig. 11 shows the average queue-lengths as a function of \( a \) in a 3-hop tandem network.

**Remark:** As we can see from the above analysis, the dimension and the number of states in the Markov chain grow fast as \( n \) increases. It is cumbersome and impractical to precisely analysis the Markov chain to prove the generalization of Lemma 1.

**C. Proof of Lemma 2**

To simplify the notation, we denote the nodes as node 0 to node \( n \) in a \( n \)-hop tandem network, where node 0 is the destination, and node \( n \) is the source. Since only one flow is assumed to exist, we omit the superscript \( d \) indicating the destination of the flow.

Let \( \epsilon \) be defined as \( \epsilon = 1 - a \). By the feasibility constraint (15) and the flow conservation constraint (16), we have

\[
r_{i,i-1}^{*} + r_{i-1,i}^{*} \leq 1
\]
\[
r_{i,i-1}^{*} - r_{i-1,i}^{*} \geq 1 - \epsilon
\]
for all $i = 1, 2, \ldots, n$. Thus we have $r_{i-1,i} \leq \epsilon/2$ for all $i$.

Also, note that under DTBP algorithm, $|P_i[t] - P_{i-1}[t]| \leq 3$ for all $i = 1, 2, \ldots, n-1$ and any $t$. This is because each node $i$ can get at most 2 packets from its neighbors (2 is the degree of all intermediate nodes in a tandem network), while its neighboring nodes can send out at most 1 more packet to other nodes. To receive one packet from both of its neighbor, the queue length at node $i$ is at least one less than that of its neighbor. This implies that the steady state distribution satisfies

$$\pi(P)1_{\{P_i-P_{i-1}>3\}}(P) = 0, \forall i = 1, 2, \ldots, n-1,$$

where $1_{\{A\}}(P)$ is the indicator function of a set $A$ and $1_{\{A\}}(P) = 1$ if $P \in A$ and 0 otherwise. Note that $P_{i-1}[t] - P_i[t] \leq 3$ also holds for the source node $n$.

Thus, we have

$$\pi^* = \sum P \pi(P)(P_1 - P_{i-1})$$

$$= \sum P \pi(P)1_{\{P_i>P_{i-1}\}}(P)(P_1 - P_{i-1})$$

$$- \sum P \pi(P)1_{\{P_i<P_{i-1}\}}(P)(P_1 - P_{i-1})$$

$$\geq_a \sum P \pi(P)1_{\{P_i>P_{i-1}\}}(P)$$

$$\geq -3 \sum P \pi(P)1_{\{P_i<P_{i-1}\}}(P)$$

$$= (b) r_{i-1,i} - 3r_{i-1,i}$$

$$\geq 1 - \epsilon - 2r_{i-1,i}$$

$$\geq 1 - 2\epsilon$$

$$> 0 \text{ if } \epsilon < \frac{1}{2}.$$
where $Q_n^j[t]$ is the queue for commodity $j$ transmissions from node $n$ to node $j$ at time $t$.

For any $T > \left(\frac{\|D\| + 1}{c}\right)$, we consider the $T$ step drift of the Lyapunov function candidate,
\[
\Delta V(Q[t]) = \mathbb{E}\left[ V(Q[t + T]) - V(Q[t]) \mid Q[t], M[t] \right].
\]

The following analysis before (32) is similar to the analysis in [14]. The queuing dynamics over time $T$ satisfies
\[
Q_n^j[t + T] \leq \left( Q_n^j[t] - \sum_{\tau = t}^{t+T-1} D_n^j[\tau] \right)^+ + \sum_{\tau = t}^{t+T-1} A_n^j[\tau],
\]
where $A_n^j[\tau] = X_n^j[t] \mathbb{I}_{\{n \in N^j_{source}\}} + \sum_{k \in N} D_k^j[\tau]$. Squaring both sides of (31) and moving terms to the left side, we have
\[
\begin{align*}
&\left( Q_n^j[t + T] \right)^2 - \left( Q_n^j[t] \right)^2 \\
\leq &\left( \sum_{\tau = t}^{t+T-1} D_n^j[\tau] \right)^2 + \left( \sum_{\tau = t}^{t+T-1} A_n^j[\tau] \right)^2 - 2 Q_n^j[t] \sum_{\tau = t}^{t+T-1} D_n^j[\tau] \\
&+ 2 \sum_{\tau = t}^{t+T-1} A_n^j[\tau] (Q_n^j[t] - \sum_{\tau = t}^{t+T-1} D_n^j[\tau]) \\
&\leq \sum_{\tau = t}^{t+T-1} D_n^j[\tau] \sum_{\tau = t}^{t+T-1} A_n^j[\tau] - 2 Q_n^j[t] \sum_{\tau = t}^{t+T-1} D_n^j[\tau] \\
&+ 2 \sum_{\tau = t}^{t+T-1} A_n^j[\tau] (Q_n^j[t] - \sum_{\tau = t}^{t+T-1} D_n^j[\tau]) \\
&\quad + 2 Q_n^j[t] \left( \sum_{\tau = t}^{t+T-1} A_n^j[\tau] - \sum_{\tau = t}^{t+T-1} D_n^j[\tau] \right).
\end{align*}
\]

Then the $T$ step drift of the Lyapunov function satisfies
\[
\begin{align*}
\Delta V(Q[t]) &= \mathbb{E}\left[ V(Q[t + T]) - V(Q[t]) \mid Q[t], M[t] \right] \\
&= \sum_{n \in N} \sum_{j \in J} \sum_{\tau = t}^{t+T-1} (Q_n^j[t + T])^2 - \sum_{n \in N} \sum_{j \in J} (Q_n^j[t])^2 \mid Q[t], M[t] \\
&\leq \sum_{n \in N} \sum_{j \in J} \mathbb{E}\left[ (\sum_{\tau = t}^{t+T-1} D_n^j[\tau])^2 + (\sum_{\tau = t}^{t+T-1} A_n^j[\tau])^2 \mid M[t] \right] \\
&+ 2 \mathbb{E}\left[ \sum_{n \in N} \sum_{j \in J} Q_n^j[t] \right] \\
&\quad \cdot \left( \sum_{\tau = t}^{t+T-1} A_n^j[\tau] - \sum_{\tau = t}^{t+T-1} D_n^j[\tau] \right) \mid M[t] \\
&\leq B + 2 \sum_{n \in N} \sum_{j \in J} Q_n^j[t] \\
&\quad \cdot \mathbb{E}\left[ \sum_{\tau = t}^{t+T-1} A_n^j[\tau] - \sum_{\tau = t}^{t+T-1} D_n^j[\tau] \mid M[t] \right].
\end{align*}
\]

From (29), we have
\[
\begin{align*}
&\mathbb{E}\left[ \sum_{\tau = t}^{t+T-1} A_n^j[\tau] \mid M[t] \right] \\
&= \sum_{\tau = t}^{t+T-1} \mathbb{E}\left[ A_n^j[\tau] \sum_{m \in N} \bar{R}_{nm}^d \mid M[t] \right] \\
&= \sum_{\tau = t}^{t+T-1} \mathbb{E}\left[ X_n^d[\tau] \sum_{m \in N} \bar{R}_{nm}^d \mathbb{I}_{\{m \in N^d_{source}\}} \right] \\
&+ \sum_{\tau = t}^{t+T-1} \sum_{k \in N} \mathbb{E}\left[ D_k^d[\tau] \mid M[t] \right] \sum_{m \in N} \bar{R}_{nm}^d. \tag{33}
\end{align*}
\]

We know that $\sum_d S_{nj}^d < 1$ and $\sum_{d \in D} M_n^d[0] < (|D| + 1)c$. By Lemma [3], we have $\sum_{d \in D} M_n^d[t] < (|D| + 1)c$, $\forall t > 0$. For each $D_n^j[t]$, we have
\[
\sum_{\tau = t}^{t+T-1} D_n^j[\tau] = \sum_{\tau = t}^{t+T-1} S_{nj}^d + M_n^d[t + T - 1] - M_n^d[t].
\]

Moving terms to the left side and taking the absolute value of both sides,
\[
\begin{align*}
&\left| \sum_{\tau = t}^{t+T-1} D_n^j[\tau] \right| - \sum_{\tau = t}^{t+T-1} S_{nj}^d \\
&= \left| M_n^d[t + T - 1] - M_n^d[t] \right| \\
&\leq (|D| + 1)c.
\end{align*}
\]

As a result of sending out dummy packets in case of insufficient packets in $Q_n^j[t]$, the departure process $D_n^j[t]$ is deterministic given its associate token counts before time $t$. So we have
\[
\begin{align*}
\mathbb{E}\left[ \sum_{\tau = t}^{t+T-1} D_n^j[\tau] \mid M[t] \right] &= \sum_{\tau = t}^{t+T-1} D_n^j[\tau].
\end{align*}
\]

Therefore
\[
\begin{align*}
&\mathbb{E}\left[ \sum_{\tau = t}^{t+T-1} D_n^j[\tau] \mid M[t] \right] - \sum_{\tau = t}^{t+T-1} S_{nj}^d \\
&\leq (|D| + 1)c. \tag{34}
\end{align*}
\]
Substituting (33) into (32), we have
\[
\Delta V(Q[t]) \leq B + 2 \sum_{d} \sum_{n} \sum_{j} Q_{nj}^d[t].
\]
\[
\{ \sum_{\tau=t}^{t+T-1} \mathbb{E} \left[ X_n^d[\tau] \right] \sum_{m \in N} R_{nm}^{\text{d}} \mathbb{I}_{\{n \in N \text{source}^d_{\tau} \}} \\
\sum_{\tau=t}^{t+T-1} \sum_{k \in N} D_{kn}^d[\tau] \mathbb{E} \left[ M[t] \right] \sum_{m \in N} R_{nm}^{\text{d}} \\
- \sum_{\tau=t}^{t+T-1} \mathbb{E} \left[ D_{nj}^d[\tau] \mathbb{I}_{\{n \in N \text{source}^d_{\tau} \}} \right] \}
\]
\[
= B + 2 \sum_{d} \sum_{n} \sum_{j} Q_{nj}^d[t].
\]
Applying (34) and (30) to (35), then
\[
\Delta V(Q[t]) \leq B + 2 \sum_{d} \sum_{n} \sum_{j} Q_{nj}^d[t].
\]
\[
\{ \sum_{\tau=t}^{t+T-1} \mathbb{E} \left[ X_n^d[\tau] \right] \sum_{m \in N} R_{nm}^{\text{d}} \mathbb{I}_{\{n \in N \text{source}^d_{\tau} \}} \\
\sum_{\tau=t}^{t+T-1} \sum_{k \in N} D_{kn}^d[\tau] \mathbb{E} \left[ M[t] \right] \sum_{m \in N} R_{nm}^{\text{d}} \\
- \sum_{\tau=t}^{t+T-1} \mathbb{E} \left[ D_{nj}^d[\tau] \mathbb{I}_{\{n \in N \text{source}^d_{\tau} \}} \right] \}
\]
\[
\leq B + 2 \sum_{d} \sum_{n} \sum_{j} Q_{nj}^d[t].
\]
Hence for any \( T > \frac{(|D|+1)c}{\delta} \), the T step drift satisfies
\[
\Delta V(Q[t]) \leq B - 2T\delta' \sum_{d} \sum_{n} \sum_{j} Q_{nj}^d[t],
\]
where \( \delta' = T\delta - (|D|+1)c > 0 \). Applying Lemma 2 in [14] to (36), the proposition is proved.