# Spatial-Temporal Routing for Supporting End-to-End Hard Deadlines in Multi-hop Networks

Xin Liu, Weichang Wang and Lei Ying

*School of Electrical, Computer and Energy Engineering,*
*Arizona State University, Tempe, AZ, United States, 85287.*
*Email: {xliu272, wwang195, lei.ying.2}@asu.edu*

**Abstract**

We consider the problem of routing packets with end-to-end hard deadlines in multihop communication networks. This is a challenging problem due to the complex spatial-temporal correlation among flows with different deadlines. To tackle this problem, we introduce the concepts of virtual links and virtual routes to incorporate end-to-end deadline constraints into routing and propose a novel virtual queue architecture to guide the spatial-temporal routing where the routing algorithm specifies where and when a packet should be routed. For the frame-based periodic traffic, the proposed policies can support any arrival within throughput region when the ratio between the link capacity and the packet size is sufficiently large. For the general traffic, we integrate a constrained resource-pooling heuristic into the spatial-temporal routing, which improves the delivery ratio and performs well. Our extensive simulations show that the policies outperform traditional policies such as backpressure and earliest-deadline-first (EDF) for more general traffic flows in multihop communication networks.

*Keywords:* Spatial-temporal routing; end-to-end deadline; throughput optimality.

## 1. Introduction

Network applications, such as emergency messages, voice calls and video streaming, demand reliable and predictable transmissions over multi-hop communication networks. Despite remarkable progress on the design of communication networks with maximum throughput and low delays, routing packets with end-to-end hard deadlines in multi-hop communication networks remains to be a challenging problem. When end-to-end hard deadlines are imposed on packets, routing decisions at each hop are coupled with the decisions in the sequential hops. Each link faces the dilemma of allocating its capacity to flows with large backlogs or to flows with short deadlines.

This challenging problem has been studied under several settings in the literature. [1] studied communication with per-packet deadline constraints in a multi-hop network and proposed a novel queue structure to dynamically adjust the service discipline to meet end-to-end deadlines. However, [1] does not establish any performance guarantee of the proposed resource allocation policies. [2] proposed a throughput optimal policy for tree networks with a single sink node (i.e., a single destination) and frame-based traffic flows. [3] presented an online algorithm with throughput guarantees for general traffic flows and networks, where the competitive ratio is inversely proportional to the hops of the longest route. More recently, [4] developed a maximal throughput scheduling policy for multi-hop wireless networks in which links do not interfere with each other and the transmit power can be adjusted. A common assumption made in these work is that they all consider static routing and assume each flow is associated with a prefixed route, so routing packets with the end-to-end deadlines remains to be a problem that has not been well studied in the literature. We also would like to point out that [5] established a trade off between throughput and the end-to-end delay bound in multi-hop networks using the Lyapunov drift analysis, where the end-to-end delay bound is decomposed into the worst-case delay at each hop along a route which is different from hard deadline constraints we consider in this work.

This paper considers dynamic routing and focuses on routing packets with end-to-end hard deadlines in multi-hop communication networks. In particular, no prefixed route is given and a packet is allowed to use any route along which it can reach its destination before the deadline expires. In order to guarantee end-to-end hard deadlines, a routing algorithm not only needs to decide "where" to send the packet (the next hop) but also "when" to send the packet (the time slot the packet should be transmitted). A simple example of three-node network is presented in Figure 1. The "where" question has been extensively studied in the literature. For example, the celebrated back-pressure algorithm is throughput optimal. The "when" question has not been well understood and there is little work in the literature addressing both problems to support end-to-end deadlines. To answer the "when" question, a routing algorithm needs to control the timing of transmitting each packet. To incorporate this temporal perspective, we introduce the concept of *virtual links*, where each virtual link represents the use of a physical link at a specific time slot (within a frame, the concept of frame will be introduced later). In a network consisting of virtual links, a route (called virtual route) for a packet specifies both the spatial path (the sequence of nodes to traverse) to the destination but also the time slot each transmission should occur so that the packet will reach the destination before the deadline expires. These two novel concepts, virtual links and virtual routes, enable us to characterize the complicated spatial-temporal correlation of packets from different flows and with different deadlines. Based on the concepts, we propose a new queueing architecture which maintains per-destination, per-deadline and per-time-slot virtual queues on each node to keep track of the congestion levels of each virtual link with respect to different flows. Based on that, a spatial-temporal routing policy, which shares the same spirit of the back-pressure algorithm, is developed to dynamically balance traffic flows among all possible virtual routes. The main contributions of this paper are summarized below.

- We introduce two novel concepts of *virtual links* and *virtual routes*, which generalize the traditional concepts of links and routes to the temporal domain. Assuming frame-based periodic traffic pattern (to be defined in Section 2), the concepts enable us to explicitly characterize the network throughput region under end-to-end deadline constraints.
- Based on the network utility maximization framework, we develop two distributed routing policies, called spatial-temporal backpressure and spatial-temporal water-filling, for traffic flows with end-to-end deadlines. For frame-based periodic traffic, our proposed policies can support any such traffic that is within the network throughput region when the ratio between the link capacity and the packet size is sufficiently large. The routing is fully distributed and only requires information exchange among neighboring nodes.
- To tackle general traffic, we propose a constrained resource-pooling heuristic to enhance spatial-temporal routing, which pools link resource allocated to the virtual commodities with the same destination to smooth stochastic arrivals across time horizon and improves the delivery ratio.
- We evaluate the performance of the proposed algorithms by numerical simulations. From the simulations, the proposed policies significantly outperform backpressure and EDF for frame-based periodic traffic flows and the heuristic helps support a higher delivery ratio under general traffic flows.

## 2. Network Model

We consider a multi-hop communication network denoted by a graph $\mathcal{G} = (\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{L}$ is the set of links. The network is assumed to be a time-slotted system, where arrivals occur at the beginning of a time slot and departures occur at the end of a time slot. Denote by $(a, b)$ the link from node $a$ to node $b$ and $C_{(a,b)}$ the capacity of the link (i.e., the number of packets that can be transmitted in one time slot). In this paper, we assume transmissions at different links do not interfere with each other. So the communication network is either a wired network or a wireless network in which neighboring links operate over orthogonal channels. We further assume the capacity of a link is time invariant.

Packets with hard deadlines are injected into the network. We assume a frame structure for traffic flows such that every $T$ consecutive time slots are grouped into a frame. Each incoming packet is associated with

a destination and a deadline, where the deadline is the time slot (within a frame) by which the packet should be delivered to its destination otherwise the packet will be dropped. The deadline of a packet expires within the same frame in which the packet arrives. This frame-based traffic structure has been commonly assumed in the literature [2, 6, 7]. Traditionally, in a multi-commodity flow problem, a flow is defined by its source and destination. In this paper, we define a flow using 4-tuple: its source, destination, the relative arrival time in a frame and the relative expiration time in a frame. In particular, for flow $f$, let $s(f)$ denote its source, $d(f)$ denote its destination, $t_b(f)$ the relative arrival time (beginning time) of the packets belonging to flow $f$ and $t_e(f)$ the relative expiration time (end time) of the packets of flow $f$. Therefore, a flow is not only defined in space (by its source and destination) and also defined over time (by the beginning and end time slots). Denote by $\mathcal{F}$ the collection of flows.

Consider a simple example in Fig. 1, where the frame size is two. Two flows are defined in the figure, where packets of flow $f_1$ arrive at node $a$ at the first time slot of a frame and need to be delivered to node $c$ at the end of the frame, and packets of $f_2$ arrive at node $a$ at the first time slot of a frame and needs to be delivered to node $b$ at the end of the frame.

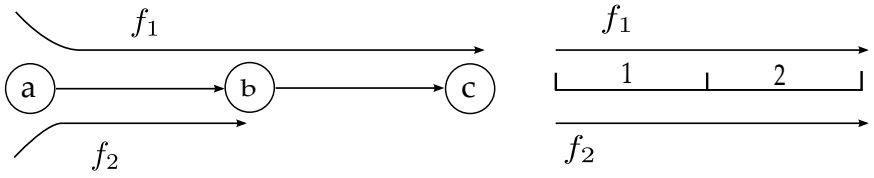



Figure 1: Illustration of Flows

We further assume periodic traffic arrivals such that for each flow, the arrival pattern within each frame remains the same across frames. This models applications such as real-time surveillance systems where remote sensors collect and report data periodically. The periodic traffic model also serves as a stepstone to study general traffic, which we will discuss after introducing the spatial-temporal routing policies for the periodic traffic model.

## 3. Virtual Links/Routes and Spatial-Temporal Routing

Traditionally, the throughput region of a communication network is defined based on link capacity, and a set of flows are said to be within the network throughput region if there exists a resource allocation algorithm under which the long-term average throughput is equal to the arrival rate of each flow. So the traditional resource allocation in communication networks concerns the "spatial allocation" (the allocation of link capacity across flows). For packets with hard deadlines, resource allocation across time becomes critical. In this paper, we aim at incorporating hard deadline constraints into the characterization of the throughput region. One key step is to introduce two novel ideas: virtual links and virtual routes. The concept of a virtual route expands the traditional route to the temporal domain and is defined to be an $N \times T$ matrix $\mathbf{R}$ such that $R(n,t) = 1$ denotes the route traverses node $n$ at the $t$th time slot of a frame. A virtual route specifies not only which links to use but when to use them, so can be used to control the end-to-end deadline of transmitting a packet.

Like a virtual route, a virtual link expands the resource of a physical link to a spatial-temporal domain. Denote by $\{(a,b),t\}$ a virtual link, which represents link $(a,b)$ at the $t$th time slot of a frame. Fig. 2 illustrates the concepts of virtual links and virtual routes using the toy example in the previous section. Since the frame size is two, each physical link is represented by two virtual links. For packets of flow $f_1$, it has to take the virtual route

$$\{(a,b),1\} \rightarrow \{(b,c),2\}$$

to reach destination $c$ by the end of time slot 2. For packets belonging to flow $f_2$, both virtual routes $\{(a,b),1\}$ and $\{(a,b),2\}$ are feasible routes.

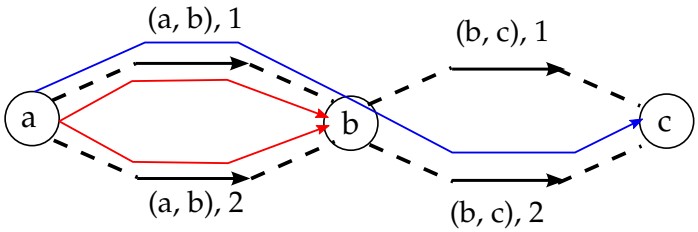
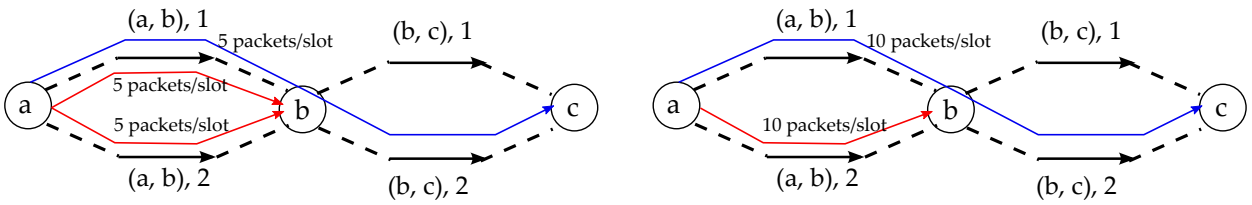



Figure 2: Illustration of virtual routes and virtual links. Blue and red lines represent virtual routers for flows $f_1$ and $f_2$, respectively. Black lines represent virtual links.

Now to achieve the full throughput region in the network, a resource allocation algorithm needs to 1) identify the set of feasible virtual routes for each flow and 2) allocate the load appropriately on the set of routes under the network capacity constraints. Let us assume the capacity of both links are 10 packets/time slot in the toy example. We consider the following two different cases.

- Case 1: Flow $f_1$ has a periodic arrival of 5 packets at the first time slot of each frame; and flow $f_2$ has a periodic arrival of 10 packets at the first time slot of each frame. One of the feasible routing solutions is shown in Fig. 3a, where flow $f_2$ splits its traffic evenly among the two virtual routes $\{(a,b),1\}$ and $\{(a,b),2\}$.
- Case 2: Flow $f_1$ has a periodic arrival of 10 packets at the first time slot of each frame; and flow $f_2$ has a periodic arrival of 10 packets at the first time slot of each frame. In this case, the unique routing solution to support the traffic is to have flow $f_2$ use the unique virtual route $\{(a,b),2\}$ and flow $f_1$ use virtual route $\{(a,b),1\} \rightarrow \{(b,c),2\}$ as shown in Fig. 3b.




(a) Case 1

(b) Case 2

Figure 3: Illustration of two feasible virtual routes.

The challenge of routing packets with end-to-end hard deadline constraints is that even without introducing virtual routes, the number of possible routes in a network is exponential in the number of links. After representing each physical link with $T$ virtual links, the number of virtual routes becomes even larger. To overcome this difficulty, we exploit the idea in [8], which includes both dynamical routing and load balancing in a communication network without requiring per route information.

One of the algorithms proposed in [8] achieves throughput optimality while guaranteeing hop constraints (a constraint on the number of hops a packet is allowed to travel before reaching its destination). A critical component of the algorithm is to maintain per-destination and per-hop queues and to only allow packets to be transmitted to neighboring nodes who can meet the hop constraints. For packets with hard deadlines, with the concept of virtual links and virtual routes, our proposed policies will maintain per-destination, per-deadline and per-time-slot virtual queues, and only allow a packet to be transmitted to virtual links that guarantees the feasibility of delivering the packet before its deadline. Furthermore, our policies are completely distributed. The design of the policies and the performance analysis will be presented in the following sections.

## 4. Throughput Region for communication networks with Hard Deadlines

For traffic flows with hard deadlines, the network throughput region depends on both link capacity and the distributions of traffic arrivals and deadlines. Assuming periodic traffic flows with a frame structure, we next characterize the network throughput region using flow conservation constraints and virtual link capacity constraints based on the necessary conditions for virtual commodities. We define $\mathcal{D}_{\{d,t_e\}} = \{f : d(f) = d, t_e(f) = t_e\}$ to be virtual commodity $\{d, t_e\}$ such that a packet of virtual commodity $\{d, t_e\}$ has to reach the destination $d$ by the end of the $t_e$th time slot in a frame. The following necessary condition has to be satisfied for each virtual commodity, each node and each time slot in a frame.

$$\sum_{f\in\mathcal{D}_{\{d,t_e\}}} a_f 1_{s(f)=n,t_b(f)=t} + \sum_{a:(a,n)\in\mathcal{L}} \sum_{i=1}^{t-1} u_{\{d,t_e\}}^{\{(a,i)\to(n,t-1)\}}$$

$$\leq \sum_{b:(n,b)\in\mathcal{L}} \sum_{i=t}^{t_e-h_{b\to d}^{\min}} u_{\{d,t_e\}}^{\{(n,t)\to(b,i)\}}, \forall d, n \in \mathcal{N},\ t_e, t \in [1,T]. \tag{1}$$

In the condition above, $1_{s(f)=n,t_b(f)=t}=1$ indicates flow $f$ is injected into node $n$ at the beginning of the $t$th time slot and $a_f$ is the arrival rate of flow $f$; $h_{b\to d}^{\min}$ denotes the minimum number of hops from node $b$ to node $d$, so $t_e - h_{b\to d}^{\min}$ is the maximum number of time slots a packet can be hold at node $n$ before being transmitted to node $b$ so that the packet is still feasible to reach its destination before the deadline expires; $u_{\{d,t_e\}}^{\{(a,i)\to(n,t)\}}$ denotes the number of packets that arrive at node $a$ at the beginning of the $i$th time slot and are transmitted to node $n$ at the end of the $t$th time slot (Note $u_{\{d,t_e\}}^{\{(a,i)\to(n,t)\}}$ could be non-integers and the fraction of packet indicates the possible link multiplexing among virtual flows). The flow conservation constraint (1) basically states that the incoming packets of virtual commodity $\{d, t_e\}$ at node $n$ at the beginning of the $t$th time slot should be sent to node $n$'s neighbors in the subsequent time slots while guaranteeing the feasibility of delivering them before their deadlines.

We further have the following necessary condition due to link capacity constraint. Note that all traffics $\{(a, i) \to (n, t)\}$ uses virtual link $\{(a, n), t\}$, so we have the following capacity constraint

$$\sum_{\{d,t_e\}\in\mathcal{D}} \sum_{i=1}^{t} u_{\{d,t_e\}}^{\{(a,i)\to(n,t)\}} \leq C_{\{(a,n),t\}},\ (a,n) \in \mathcal{L},\ t \in [1,T], \tag{2}$$

where $C_{\{(a,n),t\}} = C_{(a,n)}$ for any $t$. Moreover, we define the link capacity region to be $\mathcal{C} = \{\mathbf{u} \mid \mathbf{u}$ satisfies (2) $\}$, where $\mathbf{u}$ is the vector version of transmission rates.

Now given traffic $\mathbf{A} = \{a_f\}_{f\in\mathcal{F}}$ with deadline constraint $\mathbf{D} = \{t_b(f), t_e(f)\}_{f\in\mathcal{F}}$, we can define the throughput region

$$\Omega = \{(\mathbf{A}, \mathbf{D}) \mid \text{there exists } \mathbf{u} \text{ that satisfies both (1) and (2)}\}.$$

Then we define a routing policy as follows.

**Definition 1.** *A routing policy is a set of rates allocation for each virtual commodity on each link given the past and current state information.*

Further, we give the following definition of and theorem.

**Definition 2.** *The arrival traffic* $(\mathbf{A}, \mathbf{D})$ *is* supportable *by a routing policy if the packet dropping rate converges to zero as* $t \to \infty$.

**Theorem 1.** *No routing policy can support an arrival traffic* $(\mathbf{A}, \mathbf{D}) \notin \Omega$.

*Proof.* See Appendix Appendix A. □

Before going forward, we summarize the notations in Table 1.

| Notation | Description |
|---|---|
| $f$, $s(f)$ and $d(f)$ | the flow $f$, source of flow $f$, and destination of flow $f$ |
| $a_f$ | the arrival rate of flow $f$ |
| $\mathcal{D}_{\{d,t_e\}}$ | the virtual commodity where packets have to reach the destination $d$ within time $t_e$ |
| $h_{a\to b}^{\min}$ | the minimum number of hops from node $a$ to node $b$ |
| $(a,b)$ and $\{(a,b),t\}$ | the link $(a,b)$ and the link $(a,b)$ at time $t$ |
| $C_{(a,b)}$ and $C_{\{(a,b),t\}}$ | the capacity of link $(a,b)$ and the capacity of link $(a,b)$ at time $t$ |
| $u_{\{d,t_e\}}^{\{(a,i)\to(b,j)\}}$ | the number of (virtual) packets belonging to the commodity $\{d,t_e\}$ that arrive at node $a$ at the beginning of the $i$th time slot and are transmitted to node $b$ at the end of the $j$th time slot. |

Table 1: Notations

## 5. Optimization Framework

After defining throughput region $\Omega$, we consider the following optimization problem whose goal is to reduce the average latency while guaranteeing hard deadline constraints

$$\min \sum_{\{d,t_e\}\in\mathcal{D}} \sum_{a=1}^{N}\sum_{b=1}^{N}\sum_{t=1}^{T}\sum_{i=t}^{T} V\cdot(i-t)U\left(u_{\{d,t_e\}}^{\{(a,t)\to(b,i)\}}\right) \tag{3a}$$

$$\text{s.t. } \mathbf{u}\in\Omega. \tag{3b}$$

In the optimization problem, $V\cdot(i-t)U(\cdot)$ is a convex utility function of transmission rate $u_{\{d,t_e\}}^{\{(a,t)\to(b,i)\}}$ and $V$ is a positive constant. The problem above (3) is a convex optimization since both objective function and $\Omega$ are convex. When $U(\cdot)=0$, the optimization problem (3) is formulated to find a feasible routing policy to support end-to-end deadlines. When $U(\cdot)$ is a strictly increasing and convex function, the optimization problem (3) has a unique optimal solution. In our optimization framework, we add a weight $(i-t)$ to the utility on each intermediate transmission, $u_{\{d,t_e\}}^{\{(a,t)\to(b,i)\}}$, along the route, where $i-t$ can be interpreted as the waiting time at the intermediate hop. Therefore, the utility function is designed to penalize large latency in each hop in order to further reduce the end-to-end latency. As we will see in Section 6.3, $V$ is the tuning parameter in the routing policy to control the tradeoff between the optimality of the resource allocation solution and the convergence rate. Specifically, a large $V$ will let packets go to the routes with small end-to-end latency to achieve better utility and might delay sending packets to other possible routes.

### 5.1. Dual Decomposition

To derive a distributed routing policy from the optimization problem (3), we keep link constraint (2) and define $\lambda_{\{n,t,d,t_e\}}$ to be the Lagrangian multiplier associated with flow conservation constraint (1). We obtain the following dual problem:

$$\max_{\boldsymbol{\lambda}\geq 0} D(\boldsymbol{\lambda}) = \max_{\boldsymbol{\lambda}\geq 0}\min_{\mathbf{u}\in\Omega}\mathcal{L}(\boldsymbol{\lambda},\mathbf{u}) \tag{4}$$

where $\mathcal{L}(\boldsymbol{\lambda},\mathbf{u})$ is the partial Lagrangian function. Since (3) is a convex problem and satisfies Slater's condition, the duality gap is zero [9]. We aim to find the optimal $\mathbf{u}$ by the dual decomposition on $\mathcal{L}(\boldsymbol{\lambda},\mathbf{u})$:

$$\mathcal{L}(\boldsymbol{\lambda},\mathbf{u}) = \sum_{\{d,t_e\}\in\mathcal{D}}\sum_{n=1}^{N}\sum_{t=1}^{T}\lambda_{\{n,t,d,t_e\}}\Bigg(\sum_{f\in\mathcal{D}_{\{d,t_e\}}} a_f 1_{s(f)=n,t_b(f)=t} + \sum_{a:(a,n)\in\mathcal{L}}\sum_{i=1}^{t-1} u_{\{d,t_e\}}^{\{(a,i)\to(n,t-1)\}}$$
$$- \sum_{b:(n,b)\in\mathcal{L}}\sum_{i=t}^{t_e-h_{b\to d}^{\min}} u_{\{d,t_e\}}^{\{(n,t)\to(b,i)\}}\Bigg) + \sum_{\{d,t_e\}\in\mathcal{D}}\sum_{a=1}^{N}\sum_{b=1}^{N}\sum_{t=1}^{T}\sum_{i=t}^{T} V\cdot(i-t)U\left(u_{\{d,t_e\}}^{\{(a,t)\to(b,i)\}}\right)$$

$$= \underbrace{\sum_{\{d,t_e\}\in\mathcal{D}} \sum_{n=1}^{N}\sum_{t=1}^{T} \lambda_{\{n,t,d,t_e\}} \sum_{f\in\mathcal{D}_{\{d,t_e\}}} a_f 1_{s(f)=n,t_b(f)=t}}_{X}$$

$$+ \underbrace{\sum_{\{d,t_e\}\in\mathcal{D}} \sum_{a=1}^{N}\sum_{b=1}^{N}\sum_{t=1}^{T}\sum_{i=t}^{T} V\cdot(i-t)U\left(u_{\{d,t_e\}}^{\{(a,t)\to(b,i)\}}\right)}_{Y}$$

$$\underbrace{-\sum_{\{d,t_e\}\in\mathcal{D}} \sum_{n=1}^{N}\sum_{t=1}^{T} \lambda_{\{n,t,d,t_e\}}\left(\sum_{b:(n,b)\in\mathcal{L}}\sum_{i=t}^{t_e-h_{b\to d}^{\min}} u_{\{d,t_e\}}^{\{(n,t)\to(b,i)\}} - \sum_{a:(a,n)\in\mathcal{L}}\sum_{i=1}^{t-1} u_{\{d,t_e\}}^{\{(a,i)\to(n,t-1)\}}\right)}_{Z}$$

When $\boldsymbol{\lambda}$ is given, term $X$ is fixed, and we decompose term $Z$ as follows:

$$\begin{aligned}
Z &\overset{(a)}{=} \sum_{n=1}^{N}\sum_{b=1}^{N}\sum_{\{d,t_e\}\in\mathcal{D}}\sum_{t=1}^{T}\lambda_{\{n,t,d,t_e\}}\sum_{i=t}^{t_e-h_{b\to d}^{\min}} u_{\{d,t_e\}}^{\{(n,t)\to(b,i)\}} - \sum_{a=1}^{N}\sum_{n=1}^{N}\sum_{\{d,t_e\}\in\mathcal{D}}\sum_{t=1}^{T}\lambda_{\{n,t,d,t_e\}}\sum_{i=1}^{t-1} u_{\{d,t_e\}}^{\{(a,i)\to(n,t-1)\}} \\
&\overset{(b)}{=} \sum_{a=1}^{N}\sum_{b=1}^{N}\sum_{\{d,t_e\}\in\mathcal{D}}\left(\sum_{t=1}^{T}\lambda_{\{a,t,d,t_e\}}\sum_{i=t}^{t_e-h_{b\to d}^{\min}} u_{\{d,t_e\}}^{\{(a,t)\to(b,i)\}} - \sum_{t=1}^{T}\lambda_{\{b,t,d,t_e\}}\sum_{i=1}^{t-1} u_{\{d,t_e\}}^{\{(a,i)\to(b,t-1)\}}\right) \\
&= \sum_{a=1}^{N}\sum_{b=1}^{N}\sum_{\{d,t_e\}\in\mathcal{D}}\left(\sum_{t=1}^{T}\sum_{i=t}^{t_e-h_{b\to d}^{\min}}\lambda_{\{a,t,d,t_e\}} u_{\{d,t_e\}}^{\{(a,t)\to(b,i)\}} - \sum_{t=1}^{T}\sum_{i=1}^{t-1}\lambda_{\{b,t,d,t_e\}} u_{\{d,t_e\}}^{\{(a,i)\to(b,t-1)\}}\right) \\
&\overset{(c)}{=} \sum_{a=1}^{N}\sum_{b=1}^{N}\sum_{\{d,t_e\}\in\mathcal{D}}\left(\sum_{t=1}^{t_e-h_{b\to d}^{\min}}\sum_{i=t}^{t_e-h_{b\to d}^{\min}}\lambda_{\{a,t,d,t_e\}} u_{\{d,t_e\}}^{\{(a,t)\to(b,i)\}} - \sum_{t=2}^{t_e-h_{b\to d}^{\min}+1}\sum_{i=1}^{t-1}\lambda_{\{b,t,d,t_e\}} u_{\{d,t_e\}}^{\{(a,i)\to(b,t-1)\}}\right) \\
&\overset{(d)}{=} \sum_{a=1}^{N}\sum_{b=1}^{N}\sum_{\{d,t_e\}\in\mathcal{D}}\left(\sum_{i=1}^{t_e-h_{b\to d}^{\min}}\sum_{t=1}^{i}\lambda_{\{a,t,d,t_e\}} u_{\{d,t_e\}}^{\{(a,t)\to(b,i)\}} - \sum_{t=1}^{t_e-h_{b\to d}^{\min}}\sum_{i=1}^{t}\lambda_{\{b,t+1,d,t_e\}} u_{\{d,t_e\}}^{\{(a,i)\to(b,t)\}}\right) \\
&\overset{(e)}{=} \sum_{a=1}^{N}\sum_{b=1}^{N}\sum_{\{d,t_e\}\in\mathcal{D}}\left(\sum_{t=1}^{t_e-h_{b\to d}^{\min}}\sum_{i=1}^{t}\lambda_{\{a,i,d,t_e\}} u_{\{d,t_e\}}^{\{(a,i)\to(b,t)\}} - \sum_{t=1}^{t_e-h_{b\to d}^{\min}}\sum_{i=1}^{t}\lambda_{\{b,t+1,d,t_e\}} u_{\{d,t_e\}}^{\{(a,i)\to(b,t)\}}\right) \\
&= \sum_{a=1}^{N}\sum_{b=1}^{N}\sum_{\{d,t_e\}\in\mathcal{D}}\sum_{t=1}^{t_e-h_{b\to d}^{\min}}\sum_{i=1}^{t} u_{\{d,t_e\}}^{\{(a,i)\to(b,t)\}}\left(\lambda_{\{a,i,d,t_e\}} - \lambda_{\{b,t+1,d,t_e\}}\right) \qquad (5)
\end{aligned}$$

where (a) is obtained by interchanging the summations of commodities and links; (b) holds by changing notation $n$ to $a$ and $b$ in the first and second terms respectively; (c) holds by restricting $t$ to the feasible values for both terms; (d) holds by interchanging the summations of $t$ and $i$ in the first term and changing variable $t-1$ to $t$ in the second term; (e) holds by exchanging notation $t$ and $i$ in the first term;

Similar to steps (a), (c) and (d) above, by restricting $t$ to feasible values and interchanging the summations of $t$ and $i$ in term $Y$, we obtain

$$\begin{aligned}
Y &= \sum_{a=1}^{N}\sum_{b=1}^{N}\sum_{\{d,t_e\}\in\mathcal{D}}\sum_{t=1}^{T}\sum_{i=t}^{T} V\cdot(i-t)U\left(u_{\{d,t_e\}}^{\{(a,t)\to(b,i)\}}\right) \\
&= \sum_{a=1}^{N}\sum_{b=1}^{N}\sum_{\{d,t_e\}\in\mathcal{D}}\sum_{i=1}^{t_e-h_{b\to d}^{\min}}\sum_{t=1}^{i} V\cdot(i-t)U\left(u_{\{d,t_e\}}^{\{(a,t)\to(b,i)\}}\right)
\end{aligned}$$

$$= \sum_{a=1}^{N}\sum_{b=1}^{N}\sum_{\{d,t_e\}\in\mathcal{D}}\sum_{t=1}^{t_e-h_{b\to d}^{\min}}\sum_{i=1}^{t} V\cdot(t-i)U\left(u_{\{d,t_e\}}^{\{(a,i)\to(b,t)\}}\right) \tag{6}$$

Given $\boldsymbol{\lambda}$, we then consider the following problem:

$$\min\ \mathcal{L}(\boldsymbol{\lambda},\mathbf{u}) \tag{7a}$$
$$\text{s.t. } \mathbf{u}\in\mathcal{C}. \tag{7b}$$

After substituting (5) and (6) into (7a), we decouple objective (7a) into individual optimization problems for each link $(a,b)$ as follows:

$$\max \sum_{\{d,t_e\}\in\mathcal{D}}\sum_{t=1}^{t_e-h_{b\to d}^{\min}}\sum_{i=1}^{t}\left(\lambda_{\{a,i,d,t_e\}}-\lambda_{\{b,t+1,d,t_e\}}\right)u_{\{d,t_e\}}^{\{(a,i)\to(b,t)\}} + V\cdot(i-t)U\left(u_{\{d,t_e\}}^{\{(a,i)\to(b,t)\}}\right) \tag{8a}$$

$$\text{s.t.} \sum_{\{d,t_e\}\in\mathcal{D}}\sum_{i=1}^{t} u_{\{d,t_e\}}^{\{(a,i)\to(n,t)\}} \le C_{\{(a,n),t\}},\ \forall\ (a,b)\in\mathcal{L},\ t\in[1,T]. \tag{8b}$$

Further, (8) is decoupled into sub-optimization problems at each time slot in a frame

$$\max \sum_{\{d,t_e\}\in\mathcal{D}}\sum_{i=1}^{t}\left(\lambda_{\{a,i,d,t_e\}}-\lambda_{\{b,t+1,d,t_e\}}\right)u_{\{d,t_e\}}^{\{(a,i)\to(b,t)\}} + V\cdot(i-t)U\left(u_{\{d,t_e\}}^{\{(a,i)\to(b,t)\}}\right) \tag{9a}$$

$$\text{s.t.} \sum_{\{d,t_e\}\in\mathcal{D}}\sum_{i=1}^{t} u_{\{d,t_e\}}^{\{(a,i)\to(b,t)\}} \le C_{\{(a,b),t\}},\ \forall\ (a,b)\in\mathcal{L},\ t\in[1,T]. \tag{9b}$$

Now the original problem (3) has been decoupled into individual sub-optimization problem (9) for each link at each time slot in a frame. Based on the reduced formulation (9), we then develop two fully-distributed routing policies for different choices of $U(\,\cdot\,)$.

## 6. Spatial-Temporal Routing

Motivated by the connection between Lagrangian dual variables and queue lengths (page 145 to 149) [10], we first introduce the virtual queue architecture.

### *6.1. Virtual Queue Architecture*

Let $d_{\{n,t,d,t_e\}}[k]$ denote virtual queue associated with the Lagrangian dual variable $\lambda_{\{n,d,t_e,t\}}$ which updates as

$$d_{\{n,t,d,t_e\}}[k+1] = \Bigg(d_{\{n,t,d,t_e\}}[k] + \sum_{f\in\mathcal{D}_{\{d,t_e\}}} A_f[k]1_{s(f)=n,t_b(f)=t} + \sum_{a:(a,n)\in\mathcal{L}}\sum_{i=1}^{t-1} u_{\{d,t_e\}}^{\{(a,i)\to(n,t-1)\}}[k] - \sum_{b:(n,b)\in\mathcal{L}}\sum_{i=t}^{t_e-h_{b\to d}^{\min}} u_{\{d,t_e\}}^{\{(n,t)\to(b,i)\}}[k]\Bigg)^{+}. \tag{10}$$

Note that $d_{\{n,t,d,t_e\}}[k]$ measures the congestion level of transmitting the commodity $\{d,t_e\}$ at node $n$ at the $t$th time slot during frame $k$. We remark that $k$ is the frame index, so the virtual queues are updated once every frame (instead of every time slot). If $d_{\{n,t,d,t_e\}}[k]$ is large, packets belonging to $\{d,t_e\}$ are less likely to get through if being routed to node $n$ at the $t$th time slot, so should be routed to a different time slot $t$ in the frame. As convention, we assume $d_{\{n,\cdot,n,\cdot\}}[k]=0$ for $\forall n\in\mathcal{N}$ since packets will be moved to the upper layer after arriving at their destinations. We still use a toy example to illustrate the virtual queue architecture in Fig. 4, where packets of commodities $\{c,2\}$ (blue) and $\{b,2\}$ (red) are injected into the common source $a$ at the beginning of the first time slot, so node $a$ maintains virtual queue $d_{\{a,1,b,2\}}$ for $\{b,2\}$ and $d_{\{a,1,c,2\}}$ for $\{c,2\}$. The virtual queues are counters and do not hold real packets. Node $b$ maintains $d_{\{b,2,b,2\}}$ and $d_{\{b,3,b,2\}}$ for $\{b,2\}$ and $d_{\{b,2,c,2\}}$ for $\{c,2\}$, since for $\{b,2\}$, packets can be routed along either $d_{\{a,1,b,2\}}\to d_{\{b,2,b,2\}}$ or $d_{\{a,1,b,2\}}\to d_{\{b,3,b,2\}}$; and for $\{c,2\}$, $d_{\{a,1,c,2\}}\to d_{\{b,2,c,2\}}$ is the only feasible route in its first hop. Node $c$ maintains $d_{\{c,3,c,2\}}$ for commodity $\{c,2\}$. As defined, $d_{\{b,2,b,2\}}$, $d_{\{b,3,b,2\}}$ and $d_{\{c,3,c,2\}}$ are always 0.

Note in (10), we use a perturbed version of the real periodic arrival $A_f[k]=(1-\theta)a_f+\theta A_f^{\theta}[k], 0<\theta<1$, where $A_f^{\theta}[k]$ is a Poisson random variable with unit mean. This perturbed version, called the virtual arrival, becomes the arrival into virtual queues, which makes the Markov chain aperiodic. Under this perturbation, virtual queues $d_{\{n,t,d,t_e\}}[k],\forall n,t,d,t_e$ form an aperiodic Markov chain. The reason to add this "perturbation" is to guarantee the existence and uniqueness of the stationary distribution of the virtual queues.

We analyze the complexity of virtual queues and service rates per node as follows: Consider virtual queue $d_{\{n,t,d,t_e\}}$ at node $n$ and denote by $h^{min}_{a\to b}$ the minimum hop-length between $a$ and $b$. We know $d_{\{n,t,d,t_e\}}=0$ for $t$ outside the range $\left[h^{min}_{s\to n},t_e-h^{min}_{n\to d}\right]$, because the packets with deadline $t$ cannot reach the destination $d$ before the deadline. Therefore, the maximum number of virtual queues at node $n$ for virtual commodity $(d,t_e)$ is $t_e$, and the maximum number of virtual queues at node $n$ for the destination $d$ is $T\times t_e$. Therefore, the total number of virtual queues will be much better than the worst complexity of $O(T^2N)$ per node. The same argument works for the service rate variables. Note that $\mu^{\{(n,i)\to(b,t)\}}_{\{d,t_e\}}=0$ for all $i>t,t_e>t$ and $b\notin\mathcal{N}(n)$, where $\mathcal{N}(n)$ is the neighborhood of node $n$ and $|\mathcal{N}(n)|$ is usually $O(1)$. Therefore, the maximum number of service variables at node $n$ for the destination $d$ is $T\times t_e^2$.

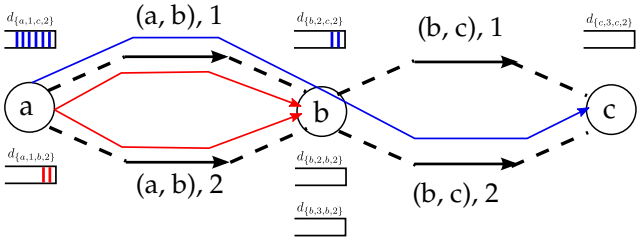



Figure 4: Illustration of the Virtual Queue Architecture

Based on the virtual queue architecture, we propose two spatial-temporal routing policies in the following subsections. Before jumping into the details, we illustrate the main workflow in Fig. 5. Two types of queues exist in spatial-temporal routing polices, real queues (real packets) and virtual queues (counters). $A_f^{real}$ and $A_f^{virtual}$ are arrivals into real queues and virtual queues, respectively, where $A_f^{virtual}$ is a perturbed version of $A_f^{real}$. Spatial-temporal routing returns virtual service rates $\bar{\mathbf{u}}$ and real service rates are obtained by rounding operation $\hat{\mathbf{u}}=\lfloor\bar{\mathbf{u}}\rfloor$.

### *6.2. Spatial-Temporal Backpressure Policy*

Assuming $U(\,\cdot\,)=0$ in (9), we propose the following spatial-temporal backpressure policy in Policy 1.

Denote $Q^t_{\{d,t_e\}}[k]$ to be the number of real packets of $\{d,t_e\}$ at $t$th time slot in frame $k$. Spatial-temporal backpressure $w^{\{(a,i)\to(b,t+1)\}}_{\{d,t_e\}}[k]$ is the difference between the values of virtual queues of $(a,i)$ and $(b,t+1)$ for commodity $\{d,t_e\}$, which extends the traditional backpressure to the spatial-temporal domain. In policy 1, virtual link $\{(a,b),t\}$ increases the capacity allocation to the tuple $\{d,t_e,i\}^*$ with the maximum backpessure

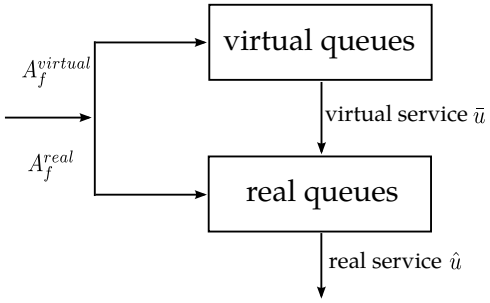



Figure 5: Main Workflow in Spatial-temporal Routing

$w_{\{d,t_e\}^*}^{\{(a,i^*)\to(b,t+1)\}}$ (breaking ties arbitrarily). If $Q_{\{d,t_e\}}^t[k] < \hat{u}_{\{d,t_e\}}^{\{(a,b),t\}}[k]$, then all real packets are transmitted. Again, we consider the example in Fig. 4. Suppose $a_{f_1} = a_{f_2} = 10$, $d_{\{a,1,c,2\}}[k] = 30$, $d_{\{b,2,c,2\}}[k] = 10$ and $d_{\{a,1,b,2\}}[k] = 10$, then $w_{\{c,2\}}^{\{(a,1)\to(b,2)\}}[k] = 20$, $w_{\{b,2\}}^{\{(a,1)\to(b,2)\}}[k] = 10$ and $w_{\{b,2\}}^{\{(a,1)\to(b,3)\}}[k] = 10$. For ease of exposition, we assume a non-perturbed version of arrival $A_{f_1}[k] = a_{f_1}$ and $A_{f_2}[k] = a_{f_2}$. During the first time slot at the $k$th frame, as $w_{\{c,2\}}^{\{(a,1)\to(b,2)\}}[k] > w_{\{b,2\}}^{\{(a,1)\to(b,2)\}}[k]$, virtual link $\{(a,b),1\}$ will increases its capacity allocation to the commodity $\{c,2\}$. Similarly in the second time slot, virtual link $\{(a,b),2\}$ and $\{(b,c),2\}$ will increases its capacity allocation to $\{b,2\}$ and $\{c,2\}$, respectively. Then at the beginning of the $k+1$th frame, the values of virtual queues are updated to be $d_{\{a,1,c,2\}}[k+1] = 30$, $d_{\{b,2,c,2\}}[k+1] = 10$ and $d_{\{a,1,b,2\}}[k+1] = 10$, which remain unchanged in the $k$th frame and real transmission rates $\hat{\mathbf{u}}$ converges to a feasible solution for $a_{f_1} = a_{f_2} = 10$ as shown in Fig. 3b.

In Policy 1, steps 1, 2 and 5 are related to the virtual queue system and generate the virtual transmission rates; steps 3 and 4 transfer the virtual transmission rates into the real transmission rates by "averaging", "aggregation" and "rounding" operations. Note "average" in step 3 is to avoid oscillation and guarantee the real transmission rates converge to a unique solution. Denote $\eta$ to be the ratio between the link capacity and the packet size. We are ready to present the following theorem.

**Theorem 2.** *Given any traffic with end-to-end deadline constraints such that* $\big((1+\varepsilon+\delta)\mathbf{A}, \mathbf{D}\big) \in \Omega$ *for some* $\varepsilon > 0$ *and* $\delta = O\left(\frac{1}{\eta}\right)$, $\hat{\mathbf{u}}[k]$ *generated by Policy 1 satisfies both conditions (1) and (2) for the given traffic. In other words, Policy 1 is near throughput optimal when* $\eta$ *becomes sufficiently large.*

*Proof.* See Appendix Appendix B. □

Spatial-temporal backpressure policy will result in a feasible routing solution whenever possible. The policy exploits all feasible virtual routes. As a result, packets might traverse routes with unnecessary large end-to-end delay. To further improve delay performance, we propose a water-filling policy to reduce end-to-end latency while guaranteeing throughput performance.

### *6.3. Spatial-Temporal Water-filling Policy*

To solve (9), we define $v$ to be the Lagrange dual variable of (9b) and derive its Lagrange function

$$L(\mathbf{u}, v) = C_{\{(a,b),t\}} \cdot v + \sum_{\{d,t_e\}\in\mathcal{D}} \sum_{i=1}^{t} u_{\{d,t_e\}}^{\{(a,i)\to(b,t)\}} \left(\lambda_{\{a,i,d,t_e\}} - \lambda_{\{b,t+1,d,t_e\}} - v\right) + V \cdot (i-t) U\left(u_{\{d,t_e\}}^{\{(a,i)\to(b,t)\}}\right)$$

According to the KKT conditions [9], the optimal solution of the problem satisfies

$$\lambda_{\{a,i,d,t_e\}} - \lambda_{\{b,t+1,d,t_e\}} - v + V(i-t)U'\left(u_{\{d,t_e\}}^{\{(a,i)\to(b,t)\}}\right) = 0,$$

**Algorithm 1** Spatial-Temporal Backpressure Policy

1: For virtual link $\{(a,b),t\}$ at the beginning of the $k$th frame, calculate spatial-temporal backpressure

$$w_{\{d,t_e\}}^{\{(a,i)\to(b,t+1)\}}[k] = \left(d_{\{a,i,d,t_e\}}[k] - d_{\{b,t+1,d,t_e\}}[k]\right)^+$$

2: Compute virtual transmission rate at the beginning of the $k$th frame

$$u_{\{d,t_e\}}^{\{(a,i)\to(b,t)\}}[k] \in \arg\max_{\mathbf{u}\in\mathcal{C}} \sum_{\{d,t_e\}\in\mathcal{D}} \sum_{i} w_{\{d,t_e\}}^{\{(a,i)\to(b,t+1)\}}[k] \cdot u_{\{d,t_e\}}^{\{(a,i)\to(b,t)\}}$$

3: Compute aggregated transmission rate for $\{d,t_e\}$ on virtual link $\{(a,b),t\}$ by

$$\bar{u}_{\{d,t_e\}}^{\{(a,b),t\}}[k] = \sum_{i=1}^{t} \bar{u}_{\{d,t_e\}}^{\{(a,i)\to(b,t)\}}[k],$$

where $\bar{u}_{\{d,t_e\}}^{\{(a,i)\to(b,t)\}}[k]$ is the time average of the virtual transmission rate up to the $k$th frame,

$$\bar{u}_{\{d,t_e\}}^{\{(a,i)\to(b,t)\}}[k] = \frac{1}{k}\sum_{i=1}^{k} u_{\{d,t_e\}}^{\{(a,i)\to(b,t)\}}[i].$$

4: Transmit packets of virtual commodity $\{d,t_e\}$ at node $a$ with $\hat{u}_{\{d,t_e\}}^{\{(a,b),t\}}[k] = \left\lfloor \bar{u}_{\{d,t_e\}}^{\{(a,b),t\}}[k] \right\rfloor$ to node $b$ at the $t$th time slot of the $k$th frame.

5: Update virtual queues at the end of the $k$th frame as (10).

$$\sum_{\{d,t_e\}\in\mathcal{D}} \sum_{i=1}^{t} u_{\{d,t_e\}}^{\{(a,i)\to(b,t)\}} = C_{\{(a,b),t\}},$$

from which, we derive spatial-temporal water-filling routing in Policy 2.

As suggested in (11), virtual transmission rates are allocated in a "water-filling" way, $v$ is regarded to "water level". This policy scales backpressure $w_{\{d,t_e\}}^{\{(a,i)\to(b,t+1)\}}$ with inverse of holding time $t-i$ on link $(a,b)$. Since $U'^{-1}(\cdot)$ is a non-decreasing function, link (a,b) is likely to increase capacity allocation to $\bar{u}_{\{d,t_e\}}^{\{(a,i)\to(b,t)\}}$ rather than $\bar{u}_{\{d,t_e\}}^{\{(a,j)\to(b,t)\}}$, $j<i$, which reduces holding time for $\{d,t_e\}$ on link $(a,b)$, and thus decreases the end-to-end latency for $\{d,t_e\}$. Again, we prove the optimality of spatial-temporal water-filling policy in the following theorem.

**Theorem 3.** *Given any traffic with end-to-end deadline constraints such that* $\big((1+\varepsilon+\delta)\mathbf{A},\mathbf{D}\big) \in \Omega$ *for some* $\varepsilon>0$ *and* $\delta = O\left(\frac{1}{\eta}\right)$, $\hat{\mathbf{u}}[k]$ *generated by Policy 2 satisfies both conditions (1) and (2) for the given traffic. Policy 2 achieves the trade-off* $[O(1/V), O(V)]$ *between the gap to the network utility and average virtual queue length.*

*Proof.* See Appendix Appendix C. □

We now analyze the complexity of spatial-temporal routing policy: Denote by $T_d$ the length of the maximum end-to-end deadline. For node $n$ at time slot $t$, only virtual commodities $\{d,t_e\}$, where $t_e$ belongs to $[t-T_d, t+T_d]$, are "activated" and required to be considered at time slot $t$. The number of "active" virtual commodities is $O(T_d)$ at time slot $t$, and the computation complexity of node $n$ at time slot $t$ in spatial-temporal routing policy is $O(VT_d^2)$.

**Algorithm 2** Spatial-Temporal Water-filling Policy

1: For virtual link $\{(a,b),t\}$ at the beginning of the $k$th frame, calculate virtual transmission rates

$$u_{\{d,t_e\}}^{\{(a,i)\to(b,t)\}}[k] = U^{'-1}\left(\frac{\left(w_{\{d,t_e\}}^{\{(a,i)\to(b,t+1)\}}[k]-v\right)^+}{V\cdot(t-i)}\right) \tag{11}$$

where $v$ satisfies

$$\sum_{\{d,t_e\}\in\mathcal{D}}\sum_{i=1}^{t} u_{\{d,t_e\}}^{\{(a,i)\to(b,t)\}}[k] = C_{\{(a,b),t\}}$$

2: Compute aggregated transmission rate for $\{d,t_e\}$ on virtual link $\{(a,b),t\}$ by

$$\bar{u}_{\{d,t_e\}}^{\{(a,b),t\}}[k] = \sum_{i=1}^{t}\bar{u}_{\{d,t_e\}}^{\{(a,i)\to(b,t)\}}[k]$$

where $\bar{u}_{\{d,t_e\}}^{\{(a,i)\to(b,t)\}}[k]$ is time average of virtual transmission rate up to the $k$th frame as

$$\bar{u}_{\{d,t_e\}}^{\{(a,i)\to(b,t)\}}[k] = \frac{1}{k}\sum_{i=1}^{k} u_{\{d,t_e\}}^{\{(a,i)\to(b,t)\}}[i]$$

3: Transmit packets of virtual commodity $\{d,t_e\}$ at node $a$ with $\hat{u}_{\{d,t_e\}}^{\{(a,b),t\}}[k] = \left\lfloor \bar{u}_{\{d,t_e\}}^{\{(a,b),t\}}[k]\right\rfloor$ to node $b$ at the $t$th time slot of the $k$th frame.

4: Update virtual queues at the end of the $k$th frame as (10).

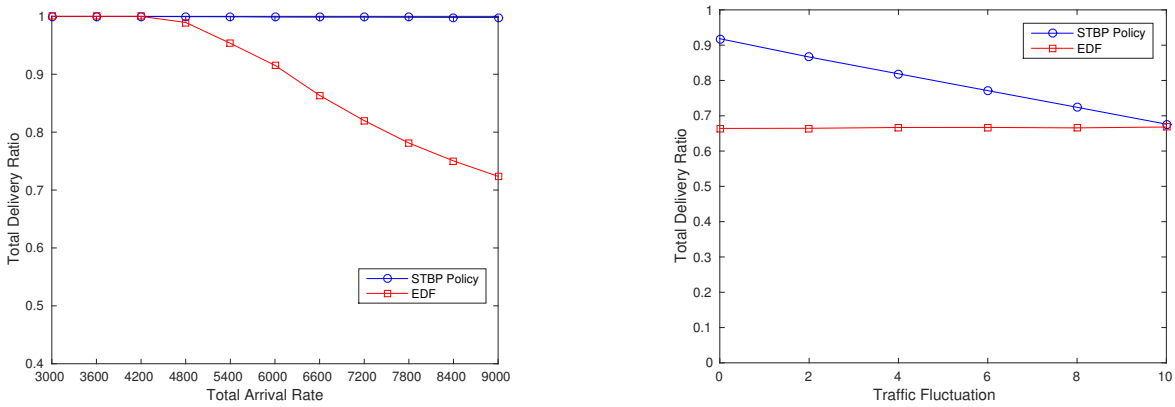
Theorem 2 and Theorem 3 prove 1) spatial-temporal backpressure policy and spatial-temporal water-filling policy can support any periodic arrival within throughput region when $\eta$ is sufficiently large, and imply 2) both policies achieve high delivery ratio when randomness in traffic is low. We verify these two arguments with simulations in Fig. 6a) and 6b), respectively, where the detailed parameters can be found in Section 8.

To tackle general traffic, we propose a constrained resource-pooling heuristic to improve the original spatial-temporal routing policies in the next section.

## 7. A constrained resource-pooling heuristic

In this section, we enhance spatial-temporal policies by a constrained resource-pooling heuristic (CRPH) imposed on real packets transmission, i.e. step 4 in Policy 1 and step 3 in Policy 2.

The motivation is to guarantee spatial-temporal policies to be "work-conserving" with general traffic by pooling the link capacity allocated to virtual commodities with the same destination. We explain how the heuristic works. At the $t$th time slot in frame $k$, two virtual commodities $\{d,t_e\}$ and $\{d,t_e+1\}$ with the same destination $d$ share the virtual link $\{(a,b),t)\}$ and the corresponding numbers of packets are $m$ and $n$. Spatial-temporal routing Policy 1 or Policy 2 computes the service rates allocated to these two virtual commodities are $\hat{u}_{\{d,t_e\}}^{\{(a,b),t)\}}[k]$ and $\hat{u}_{\{d,t_e+1\}}^{\{(a,b),t)\}}[k]$. Suppose $m < \hat{u}_{\{d,t_e\}}^{\{(a,b),t)\}}[k]$ and $n > \hat{u}_{\{d,t_e+1\}}^{\{(a,b),t)\}}[k]$, the proposed heuristic suggests to shift $\min\left(n-\hat{u}_{\{d,t_e+1\}}^{\{(a,b),t)\}}[k], \hat{u}_{\{d,t_e\}}^{\{(a,b),t)\}}[k]-m\right)$ packets from $\{d,t_e+1\}$ to $\{d,t_e\}$ and mark them belonging to $\{d,t_e\}$. Note the shifted packets can be delivered successfully since $\hat{u}_{\{d,t_e\}}^{\{(a,b),t)\}}[k]$ are the guaranteed service rate obtained from spatial-temporal routing.




(a) A periodic traffic pattern

(b) A general traffic pattern

Figure 6: Throughput performance of spatial-temporal routing under a periodic and general traffic pattern. (a) verifies spatial-temporal routing is "near optimal" in a periodic traffic; (b) verifies spatial-temporal routing performs well when the traffic fluctuation is low in a general traffic.

We further illustrate CRPH with a toy example. Consider a physical link $(a, b)$ with capacity=10 and frame size of 3 time slots in Fig. 7. Two virtual commodities $\{b, 2\}$ and $\{b, 3\}$ share a virtual links $\{(a, b), 2\}$. The arrivals for both virtual commodities are random integer variables from 14 to 16, equally likely. Assume

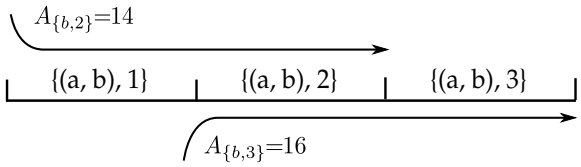



Figure 7: Illustration of a constrained resource-pooling heuristic

spatial-temporal routing returns a feasible solution: $\hat{u}_{\{b,2\}}^{\{(a,b),1)\}} = 10$, $\hat{u}_{\{b,2\}}^{\{(a,b),2)\}} = 5$, $\hat{u}_{\{b,3\}}^{\{(a,b),2)\}} = 5$, and $\hat{u}_{\{b,3\}}^{\{(a,b),3)\}} = 10$. Consider a traffic pattern with $A_{\{b,2\}} = 14$ and $A_{\{b,3\}} = 16$ in Fig. 7. If we follow spatial-temporal routing $\hat{\mathbf{u}}$, one packet of $\{b, 3\}$ would have to be dropped. However, by integrating CRPH into $\hat{\mathbf{u}}$, one packet of $\{b, 3\}$ is shifted to $\{b, 2\}$ at time slot 2 and it will be delivered successfully. As we can see, the heuristic smoothes the stochastic arrival across time horizon to enhance the performance of the network.

Note CRPH shares link resource among virtual commodities with the same destination, one may consider a different heuristic, which shares link resource among all virtual commodities regardless of their destinations. However, we will see this "intuitive" heuristic can degrade the performance. Consider three-node network with three links $(a, b)$, $(b, c)$ and $(a, c)$ in Fig. 8. Link capacity is fixed to be $C_{a,b} = 10$, $C_{b,c} = 5$ and $C_{a,c} = 5$, respectively. The frame size consists of two time slots and there are two virtual flows $(b, 1)$ and $(c, 2)$ arriving at the beginning of the frame with mean $E[A_{(b,1)}] = 15$ and $E[A_{(c,2)}] = 5$. By spatial temporal routing, we have $\hat{u}_{\{b,1\}}^{\{(a,b),1)\}} = 5$, $\hat{u}_{\{c,2\}}^{\{(a,b),1)\}} = 5$, $\hat{u}_{\{c,2\}}^{\{(b,c),2)\}} = 5$, $\hat{u}_{\{c,2\}}^{\{(a,c),1)\}} = 5$, and $\hat{u}_{\{c,2\}}^{\{(a,c),2)\}} = 5$. One virtual route is marked with red color for $(b, 1)$ and three virtual routes are marked with blue color for $(c, 2)$ as shown in Fig. 8. Suppose in frame $k$, we have $A_{(b,1)}[k] = 2$ and $A_{(c,2)}[k] = 16$. If following the "intuitive" heuristic, the virtual link $\{(a, b), 1\}$ will transmit 8 packets of $\{c, 2\}$ and 2 packets of $\{b, 1\}$, resulting in 3 packets of $\{c, 2\}$ dropped due to the bottle link $C_{\{b,c\}} = 5$. In fact, by allocating 5 packets of $\{c, 2\}$ according to spatial-temporal routing, we have only 1 packet of $\{c, 2\}$ dropped in this example. Therefore, it is important to restrict resource pooling to virtual commodities with the same destination to guarantee "work-conserving".

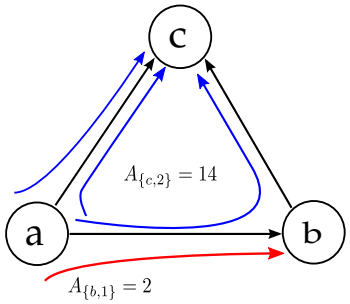



Figure 8: Illustration of downside on an "intuitive" heuristic

The spatial-temporal policies with CRPH are presented in Policy 3.

**Algorithm 3** ST Policy with resource-pooling heuristic

1: Follow step 1, 2 and 3 in STBP or step 1 and 2 STWF to obtain $\{\hat{u}\}$.
2: Perform the constrained resource-pooling heuristic for $\{d, t_e\}$ on $\{(a,b),t\}$

$$
\begin{aligned}
r_{\{d,t_e\}}^{\{(a,b),t\}} &= \min\left\{\hat{u}_{\{d,t_e\}}^{\{(a,b),t\}}[k], Q_{\{d,t_e\}}^{t}[k]\right\},\\
l_{\{d,t_e\}}^{\{(a,b),t\}} &= \left(\hat{u}_{\{d,t_e\}}^{\{(a,b),t\}}[k] - Q_{\{d,t_e\}}^{t}[k]\right)^{+},\\
\nu_{\{d,t_e+j\}}^{\{(a,b),t\}} &= \left(Q_{\{d,t_e+j\}}^{t}[k] - \hat{u}_{\{d,t_e+j\}}^{\{(a,b),t\}}[k]\right)^{+}, j \geq 1,
\end{aligned}
$$

and allocate real packet transmission rates to be

$$
r_{\{d,t_e\}}^{\{(a,b),t\}}[k] + \min\left\{ l_{\{d,t_e\}}^{\{(a,b),t\}}[k], \sum_{j\geq 1} \nu_{\{d,t_e+j\}}^{\{(a,b),t\}}[k] \right\}
$$

3: Update virtual queues at the end of the $k$th frame as in ST policy.

In step 2 in CRPH, $l_{\{d,t_e\}}^{\{(a,b),t\}}$ is the redundant resource of $\{d, t_e\}$ and $\nu_{\{d,t_e+j\}}^{\{(a,b),t\}}$ is the number of packets of $\{d, t_e + j\}$ that exceeds the rate allocated to the corresponding virtual commodity. It is worth to mentioning that CRPH improves the robustness to traffic fluctuation without affecting the convergence of the original spatial-temporal policies.

## 8. Simulations

In this section, we evaluate the performance of the proposed policies via simulations in two scenarios. The first scenario is dedicated to confirming the theoretical results in Section 6 and to verify the effectiveness of CRPH proposed in Section 7; and the second scenario is to demonstrate spatial-temporal routing policies with CRPH is effective in a practical scenario, in particular, for video transmission in the Abilene network [11].

### *8.1. First scenario: Symmetric topology*

We consider a network topology shown in Fig. 9, where all links have the same capacity of $C = 10^3$ packets/time slot and we have three flows in the network. Flow 1 is from node 1 to 6, flow 2 is from node 2 to 7, and flow 3 is from node 3 to 5. We call our spatial-temporal backpressure policy "STBP Policy" and spatial-temporal waterfilling policy "STWF Policy" in which we choose the utility function $U(x) = x^{1.1}$

and trade-off factor $V = 10^4$. We choose $x^{1.1}$ so that the optimization problem has a unique solution, and it is close to $x$ so that the object is to minimize the average latency in addition to guarantee end-to-end deadlines. We compared them with the backpressure and EDF policies. In our simulations, EDF adopts randomized routing to route each packet to a randomized chosen neighbor, and backpressure adopts random packet scheduling for the selected commodity.

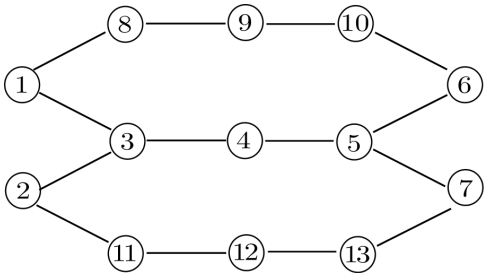
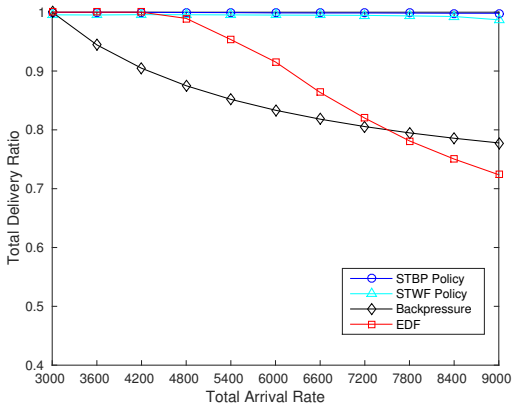
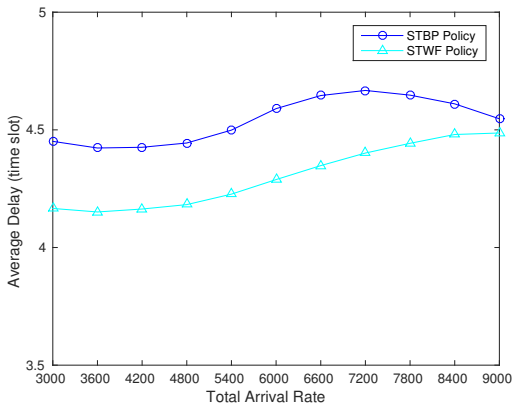



Figure 9: Symmetric Topology

*Periodic traffics*: We assume frame size $T = 6$. Packets of flow 1 and flow 2 arrive at the beginning of the frame and expire at the end of the frame. Packets of flow 3 arrive at time slot 2 and expire at time slot 4 of a frame. Packet arrivals per frame increase from $(2000, 1000, 0)$ to $(4000, 3000, 2000)$ with the step size $(200, 200, 200)$. We observe in Fig. 10a that two proposed spatial-temporal policies achieve the same throughput and outperform both backpressure and EDF at all incoming rates. EDF has a higher throughput than backpressure at the low traffic regime and backpressure outperforms EDF at the high traffic regime. It can also be easily verified that (4000, 3000, 2000) is at the boundary of the network capacity given the periodic traffic pattern. The results in Fig. 10a verifies ST policies can support (4000,3000,2000) as we proved in Theorem 2 and Theorem 3. As for the average delay performance of delivery packets, we only compare two proposed ST policies because of their significant advantage of throughput performance against EDF and backpressure. Fig. 10b shows STWF policy dominates STBP policy on the average delay performance and demonstrates STWF policy can further reduce the average delay by routing packets on these virtual routes with small delay.




(a) Throughput




(b) Average delay

Figure 10: Throughput and Delay Performance of ST routing policies for a periodic traffic pattern

Besides, "Backpressure-type" could be regarded as a dual sub-gradient algorithm. However, STBP could converge fast by incorporating time information into backpressure because the real-time constraint help bluid "backpressure" and feasible routes quickly, and avoid exploring unnecessary long routes (as

traditional backpressure usually does), which is one main reason traditional backpressure suffering from slow convergence rate. We investigated the convergence rate of STBP (Policy 1) under three workloads $(0.5, 0.8, 1.0)$, which are corresponding to total arrival rates $(4500, 7200, 9000)$ in Fig.10a in the paper. From Fig. 11, the results show the delivery ratio approaches to be 100% fast (even for large workload).

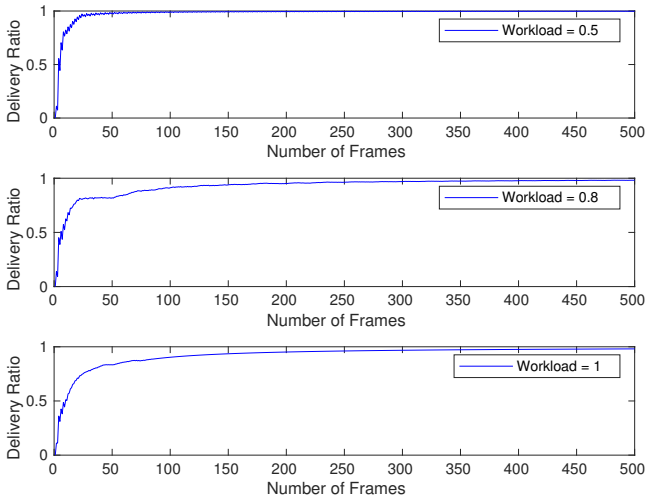


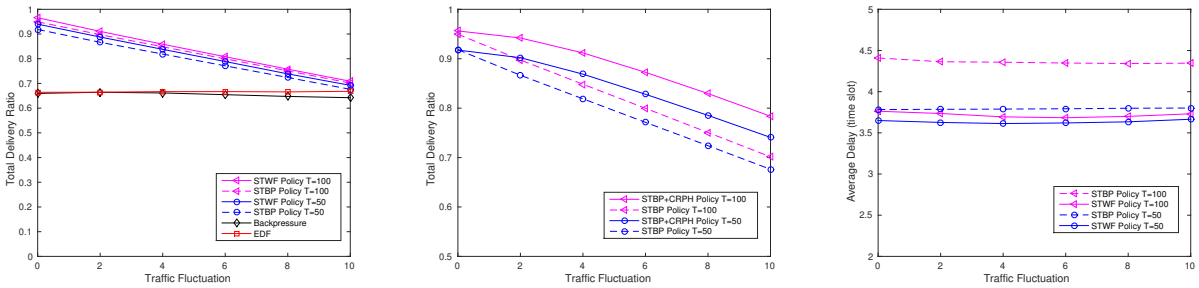

Figure 11: Convergence rate of STBP under various workloads.

*More General traffic*: We consider a traffic pattern such that each of three flows has packet arrivals at every time slot. Each packet has a deadline of 6 time slots. The number of packet arrivals for each flow at each time slot is a random variable that takes integer values from $10^3 \cdot (10 - i)$ to $10^3 \cdot (10 + i)$, equally likely. We varied $i$ from 0 to 10. A larger value of $i$ implies larger variance of the incoming traffic. We impose virtual frame structure on this general traffic pattern and evaluated the ST policies as the frame size increases, $T = 50$ and 100. From Fig. 12a, we can see that the delivery ratio of both ST policies increases as frame $T$ increases and STWF policy (solid line) has better throughput performance than STBP policy (dash line) especially for small $T = 50$. Furthermore, ST policies outperform both backpressure and EDF for almost all values of $i$ even when $T$ is small. The delivery ratio increases as the variance of the incoming traffic decreases. This is intuitive since it is more difficult to guarantee end-to-end deadlines when the randomness in the traffic increases. Then we consider STBP with CRPH heuristic to tackle the traffic randomness. In Fig. 12b, it is clearly seen STBP+CRPH (solid line) improves STBP (dash line) by a large margin and CRPH heuristic becomes more beneficial as the randomness increase (e.g. $> 6.5\%$ gain for $i = 10$). In Fig. 12c, we evaluate the average delay performance. Increasingly, we can see that the average delay under STWF decreases as $T$ increases while it increases under STBP. The reason we believe is that STBP is designed for guaranteeing end-to-end deadlines while STWF further attempts to minimize end-to-end latency.

### *8.2. Second scenario: Video transmission in the Abilene network*

In this simulations, we consider a video transmission in a real-world network, Abilene network in Fig. 13, where three video streaming flows exist, flow 1 is from node 1 to node 8, flow 2 is from node 6 to node 5 and flow 3 is from node 7 to node 11.

We used packet size with 1250 bytes and Ethernet interface with capacity 10G bits/s. Assume that the length of a time slot is 10ms, then link capacity in terms of packets per time slot is $10^4$packets/time slot. According to the requirements of interactive Video, one-way latency should be less than 150ms, so we choose the deadline to be 100ms (10 time slots) and frame size $T = 100$. The number of packets arrival




(a) Throughput (b) Throughput with STBP+CRPH (c) Average delay

Figure 12: Throughput and Delay Performance of ST routing policies for a general traffic pattern

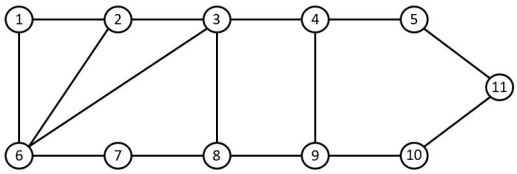



Figure 13: Abilene Topology

for each video streaming is a random variable that takes integer values from $10^3 \cdot (10 - i)$ to $10^3 \cdot (10 + i)$, equally likely, where $i$ is varied from 0 to 10. The throughput performance is shown in Fig. 14. We can see both STBP and STBP+CRPH outperform EDF and Backpressure significantly for small $i$. As $i$ increases ("wild" traffic), STBP degrades. However, STBP+CRPH is robust to traffic fluctuation. Specifically, for $i = 10$, STBP+CRPH outperform EDF by a large margin, 79.8% v.s. 70.2%.

## 9. Conclusions

In this paper, we developed spatial-temporal routing for supporting end-to-end hard deadlines in communication networks. We first introduced the concepts of virtual links and virtual routes by modeling each physical link as multiple virtual links across time slots for each time frame, by which we were able

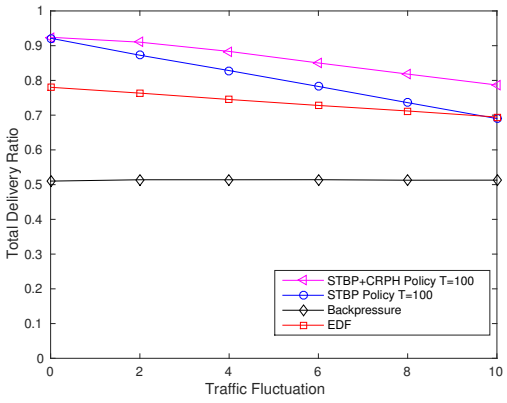



Figure 14: Throughput performance of ST routing policies for Abilene Network

to characterize the deadline constrained network throughput region with some additional assumption on the incoming traffic. Then we proposed a virtual queue architecture and developed two spatial-temporal routing policies: spatial-temporal backpressure and spatial-temporal water-filling. Our policies can support any periodic traffic within the network throughput region. For the general traffic, we proposed a constrained resource-pooling heuristic to enhance the spatial-temporal routing. Numerical simulations verified that the proposed policies outperform existing routing policies.

**Acknowledgment**

This work was supported in part by the U.S. Office of Naval Research (ONR Grant No. N00014-15-1-2169) and by the National Science Foundation (NSF) under Grants CNS-1264012 and CNS-1262329.

## Appendix A. Proof of Theorem 1

We consider the policies under which, the time-average transmission rates converge for ease of exposition. When this assumption does not hold, the strict separate theorem can be used to prove this theorem by following the idea on page 118 in [10]. Assume the service rate of a policy $\pi$ converges to $\mathbf{u}$ when arrival $(\mathbf{A}, \mathbf{D}) \notin \Omega$ and the long-term average packet dropping rates are zero for all flows. According to the definition of $\Omega$, $(\mathbf{A}, \mathbf{D}) \notin \Omega$ implies that either flow conservation constraint (1) or link capacity constraint (2) does not hold. First, we show that any violation of a link capacity constraints can always be transferred to a violation of a flow conservation constraint without affecting the other constraints. Without loss generality, we assume flow conservation constraints hold but link capacity constraints violate for the commodity $\{d, t_e\}$ such that

$$\begin{aligned}
&\sum_{f\in\mathcal{D}_{\{d,t_e\}}} a_f 1_{s(f)=n,t_b(f)=t} + \sum_{a:(a,n)\in\mathcal{L}} \sum_{i=1}^{t-1} u_{\{d,t_e\}}^{\{(a,i)\to(n,t-1)\}} \\
&\leq \sum_{b:(n,b)\in\mathcal{L}} \sum_{i=t}^{t_e-h_{b\to d}^{\min}} u_{\{d,t_e\}}^{\{(n,t)\to(b,i)\}}, \forall d, n\in\mathcal{N},\ t_e, t\in[1,T], \\
&\sum_{\{d,t_e\}\in\mathcal{D}} \sum_{i=1}^{t} u_{\{d,t_e\}}^{\{(n,i)\to(b,t)\}} + \varepsilon \geq C_{\{(n,b),t\}}, \{(n,b),t\}\in\mathcal{I},
\end{aligned}$$

where $\varepsilon > 0$ and $\mathcal{I}$ denotes the set of virtual links (including both incoming links and outgoing links) such that the link capacity constraints do not hold. We construct a policy $\bar{\pi}$ that converges to $\bar{\mathbf{u}}$ such that

$$\begin{aligned}
\bar{u}_{\{d,t_e\}}^{\{(n,i)\to(b,t)\}} &\leq u_{\{d,t_e\}}^{\{(n,i)\to(b,t)\}} - \varepsilon,\ \{(n,b),t\}\in\mathcal{I}, \\
\bar{u}_{\{d,t_e\}}^{\{(n,i)\to(b,t)\}} &\leq u_{\{d,t_e\}}^{\{(n,i)\to(b,t)\}},\qquad \{(n,b),t\}\in\mathcal{I}^c,
\end{aligned}$$

such that link constraints hold

$$\sum_{\{d,t_e\}\in\mathcal{D}} \sum_{i=1}^{t} \bar{u}_{\{d,t_e\}}^{\{(n,i)\to(b,t)\}} \leq C_{\{(n,b),t\}}, \forall (n,b)\in\mathcal{L},\ t\in[1,T].$$

but flow conservation constraints with $\bar{u}_{\{d,t_e\}}^{\{(n,t)\to(b,i)\}}$ do not hold

$$\sum_{f\in\mathcal{D}_{\{d,t_e\}}} a_f 1_{s(f)=n,t_b(f)=t} + \sum_{a:(a,n)\in\mathcal{L}} \sum_{i=1}^{t-1} \bar{u}_{\{d,t_e\}}^{\{(a,i)\to(n,t-1)\}} \geq \sum_{b:(n,b)\in\mathcal{L}} \sum_{i=t}^{t_e-h_{b\to d}^{\min}} \bar{u}_{\{d,t_e\}}^{\{(n,t)\to(b,i)\}} + \varepsilon. \tag{A.1}$$

Denote by $\{\bar{\mathbf{u}}[k]\}$ by a sample path of the service rate vector generated by policy $\bar{\pi}$ we have that with probability 1

$$\bar{u}_{\{d,t_e\}}^{\{(a,i)\to(b,t)\}} = \frac{1}{K}\sum_{k=0}^{K-1} \bar{u}_{\{d,t_e\}}^{\{(a,i)\to(b,t)\}}[k]. \tag{A.2}$$

Combining (A.1) and (A.2), we further have

$$\begin{aligned}
&\sum_{f\in\mathcal{D}_{\{d,t_e\}}} a_f 1_{\substack{s(f)=n,\\ t_b(f)=t}} + \sum_{a:(a,n)\in\mathcal{L}} \sum_{i=1}^{t-1} \lim_{K\to\infty} \frac{1}{K}\sum_{k=0}^{K-1} \bar{u}_{\{d,t_e\}}^{\{(a,i)\to(n,t-1)\}}[k] \\
&\geq \sum_{b:(n,b)\in\mathcal{L}} \sum_{i=t}^{t_e-h_{b\to d}^{\min}} \lim_{K\to\infty} \frac{1}{K}\sum_{k=0}^{K-1} \bar{u}_{\{d,t_e\}}^{\{(n,t)\to(b,i)\}}[k] + \varepsilon,
\end{aligned}$$

which means packets belonging to the commodity $\{d, t_e\}$ have to be dropped at node $n$ and dropping rate is $\varepsilon > 0$, and $(\mathbf{A}, \mathbf{D}) \notin \Omega$ is not supportable by the policy $\bar{\pi}$. It further implies $(\mathbf{A}, \mathbf{D}) \notin \Omega$ is not supportable by the policy $\pi$ since for any given policy $\pi$, we can always find a corresponding policy $\bar{\pi}$. It contradicts the assumption that under $\pi$, the packet dropping rates are zero for all flows.

## Appendix B. Proof of Theorem 2

We prove Theorem 2 by proving two following lemmas.

**Lemma 1.** *Virtual queue process* $(\mathbf{d}[k])$ *is positive recurrent under spatial-temporal backpressure policy for any arrival strictly in* $\Omega$*, i.e.* $((1+\varepsilon)\mathbf{A}, \mathbf{D}) \in \Omega$ *for* $\varepsilon > 0$.

*Proof.* Since spatial-temporal backpressure is a special case of spatial-temporal waterfilling when $U(\cdot) = 0$. The proof of this lemma follows that of Lemma 4 by letting $U(\cdot) = 0$. The detailed proof can also be found in the attached double-column version. □

**Lemma 2.** *Denote by* $\{\mathbf{u}[k] | k \in \mathbb{N}\}$ *a sample path generated from policy 1, its time average* $\bar{\mathbf{u}} = \lim_{K\to\infty} \frac{1}{K}\sum_{k=1}^{K-1} \mathbf{u}[k]$ *is a feasible rate allocation* $\mathbf{u}^*$ *given any arrival strictly in* $\Omega$.

*Proof.* Define a stochastic process $(\mathbf{d}[k], \mathbf{u}[k])$, it is irreducible (ergodic) and positive recurrent since $(\mathbf{d}[k])$ is irreducible and positive recurrent under Policy 1. Pick any fixed state as a regeneration point, and denote $K$ frames as the random generation of a regeneration cycle, $\mathbf{S}[K] = \sum_{k=0}^{K-1} \mathbf{u}[k]$ as the total service rate during $K$ frames. Since the process is ergodic, $E[K] < \infty$, then

$$\bar{\mathbf{u}} \overset{(a)}{=} \frac{E[\mathbf{S}[K]]}{E[K]} = \mathbf{u}^*, \text{ w.p.1}$$

where (a) is derived via Renewal Reward Theorem [12]. $\mathbf{u}^*$ is a feasible point due to the fact $(\mathbf{d}[k])$ is positive recurrent. □

Combining lemma 1 and lemma 2, we prove $\bar{\mathbf{u}}$ generated by spatial-temporal backpresure policy is a feasible allocation for any arrival strictly in $\Omega$ if fractional packets can be transmitted. In practice, the link capacity (e.g. G bits/s) is much larger than a packet size (e.g. 1500 bytes). Therefore rounding $\bar{\mathbf{u}}$ returns a near optimal solution when the link capacity is sufficiently large. Recall $\eta$ to be the ratio between the link capacity and the packet size.

**Lemma 3.** *Given any traffic with end-to-end deadline constraints such that* $((1+\varepsilon+\delta)\mathbf{A},\mathbf{D}) \in \Omega$*, where* $\varepsilon$ *is a positive number and* $\delta = O\left(\frac{1}{\eta}\right)$*. The solution* $\hat{\mathbf{u}} = \lfloor \bar{\mathbf{u}} \rfloor$ *satisfies both conditions (1) and (2) for the given traffic.*

*Proof.* From Lemma 2, we have $\bar{\mathbf{u}}$ satisfying both the flow conservation constraints (1) and the link capacity constraints (2) for any given traffic $((1+\varepsilon)\mathbf{A},\mathbf{D}) \in \Omega$,. Since $\hat{\mathbf{u}}$ is a floor version of $\bar{\mathbf{u}}$, (2) still hold.

Define the residual $\check{\mathbf{u}} = \bar{\mathbf{u}} - \lfloor \bar{\mathbf{u}} \rfloor$, which is $O\left(\frac{1}{\eta}\right)$. Now we check the flow conservation constraints (1),

$$\sum_{f\in\mathcal{D}_{\{d,t_e\}}} a_f 1_{s(f)=n,t_b(f)=t} - \delta_f + \sum_{a:(a,n)\in\mathcal{L}} \sum_{i=1}^{t-1} \hat{u}_{\{d,t_e\}}^{\{(a,i)\to(n,t-1)\}} = \sum_{b:(n,b)\in\mathcal{L}} \sum_{i=t}^{t_e-h_{b\to d}^{\min}} \hat{u}_{\{d,t_e\}}^{\{(n,t)\to(b,i)\}},$$

where $\delta_f$ is less than $\left|\sum_{a:(a,n)\in\mathcal{L}} \sum_{i=1}^{t-1} \check{u}_{\{d,t_e\}}^{\{(a,i)\to(n,t-1)\}} - \sum_{b:(n,b)\in\mathcal{L}} \sum_{i=t}^{t_e-h_{b\to d}^{\min}} \check{u}_{\{d,t_e\}}^{\{(n,t)\to(b,i)\}}\right|$ and $\delta_f = O\left(\frac{1}{\eta}\right)$. Then we conclude that $a_f - \delta_f$ is supportable with $\hat{\mathbf{u}}$ and it is near optimal solution as $\eta$ is sufficiently large. □

Based on Lemma 1, 2 and 3, we prove Theorem 2.

## Appendix C. Proof of Theorem 3

Similarity to the proof of Theorem 2, we first prove $(\mathbf{d}[k])$ is positive recurrent under spatial-temporal water-filling policy, and use that to quantify the performance of the algorithm by lemma 2 and 3.

**Lemma 4.** *Virtual queue process* $(\mathbf{d}[k])$ *is positive recurrent under spatial-temporal water-filling policy for any arrival strictly in* $\Omega$*, i.e.* $((1+\varepsilon)\mathbf{A},\mathbf{D}) \in \Omega$ *for* $\varepsilon > 0$*..*

*Proof.* Without loss of generality, we assume initial value of $\mathbf{d}[0] = 0$ and consider the state space $\mathcal{M}$ of Markov chain that is reachable from $\mathbf{d}[0] = 0$. Given the perturbed arrival, when all virtual queues are empty, they remain to be empty with a positive probability, so the Markov chain constrained to state space $\mathcal{M}$ is irreducible and aperiodic.

Define Lyapunov function

$$\mathbf{L}[k] = \frac{1}{2} \sum_{\{d,t_e\}\in\mathcal{D}} \sum_{n=1}^{N} \sum_{t=1}^{T} \left(d_{\{n,t,d,t_e\}}[k]\right)^2$$

Lyapunov Drift is

$$\begin{aligned}
&\delta[k]\\
=&E\big[\mathbf{L}[k+1] - \mathbf{L}[k] | \mathbf{d}[k] = \mathbf{d}\big]\\
=&\frac{1}{2}E\big[\sum_{\{d,t_e\}\in\mathcal{D}} \sum_{n=1}^{N} \sum_{t=1}^{T} \left(d_{\{n,t,d,t_e\}}[k+1]\right)^2 - \left(d_{\{n,t,d,t_e\}}[k]\right)^2 | \mathbf{d}[k] = \mathbf{d}\big]\\
\le& B_0 + E\Big[\sum_{\{d,t_e\}\in\mathcal{D}} \sum_{n=1}^{N} \sum_{t=1}^{T} d_{\{n,t,d,t_e\}} \sum_{f\in\mathcal{D}_{\{d,t_e\}}} A_f[k] 1_{s(f)=n,t_b(f)=t}\\
&- \sum_{\{d,t_e\}\in\mathcal{D}} \sum_{n=1}^{N} \sum_{t=1}^{T} \Big( \sum_{b:(n,b)\in\mathcal{L}} \sum_{i=t}^{t_e-h_{b\to d}^{\min}} u_{\{d,t_e\}}^{\{(n,t)\to(b,i)\}}[k] - \sum_{a:(a,n)\in\mathcal{L}} \sum_{i=1}^{t-1} u_{\{d,t_e\}}^{\{(a,i)\to(n,t-1)\}}[k] \Big) d_{\{n,t,d,t_e\}} \Big]\\
=& B_0 + \sum_{\{d,t_e\}\in\mathcal{D}} \sum_{n=1}^{N} \sum_{t=1}^{T} d_{\{n,t,d,t_e\}} \sum_{f\in\mathcal{D}_{\{d,t_e\}}} a_f 1_{s(f)=n,t_b(f)=t}
\end{aligned}$$

$$
\begin{aligned}
&+E\Bigg[\sum_{a=1}^{N}\sum_{b=1}^{N}\sum_{\{d,t_e\}\in\mathcal{D}}\sum_{t=1}^{T}\sum_{i=t}^{T}V(i-t)U\Big(u_{\{d,t_e\}}^{\{(a,t)\to(b,i)\}}[k]\Big)\\
&-\sum_{\{d,t_e\}\in\mathcal{D}}\sum_{n=1}^{N}\sum_{t=1}^{T}\Bigg(\sum_{b:(n,b)\in\mathcal{L}}\sum_{i=t}^{t_e-h_{b\to d}^{\min}}u_{\{d,t_e\}}^{\{(n,t)\to(b,i)\}}[k]-\sum_{a:(a,n)\in\mathcal{L}}\sum_{i=1}^{t-1}u_{\{d,t_e\}}^{\{(a,i)\to(n,t-1)\}}[k]\Bigg)d_{\{n,t,d,t_e\}}\\
&-\sum_{a=1}^{N}\sum_{b=1}^{N}\sum_{\{d,t_e\}\in\mathcal{D}}\sum_{t=1}^{T}\sum_{i=t}^{T}V(i-t)U\Big(u_{\{d,t_e\}}^{\{(a,t)\to(b,i)\}}[k]\Big)\Bigg]\\
\overset{(a)}{=}&B_0+\sum_{\{d,t_e\}\in\mathcal{D}}\sum_{n=1}^{N}\sum_{t=1}^{T}d_{\{n,t,d,t_e\}}\sum_{f\in\mathcal{D}_{\{d,t_e\}}}a_f1_{s(f)=n,t_b(f)=t}\\
&+E\Bigg[\sum_{a=1}^{N}\sum_{b=1}^{N}\sum_{\{d,t_e\}\in\mathcal{D}}\sum_{t=1}^{t_e-h_{b\to d}^{min}}\sum_{i=1}^{t}V(t-i)U\Big(u_{\{d,t_e\}}^{\{(a,t)\to(b,i)\}}[k]\Big)\\
&-\sum_{a=1}^{N}\sum_{b=1}^{N}\sum_{\{d,t_e\}\in\mathcal{D}}\sum_{t=1}^{t_e-h_{b\to d}^{\min}}\sum_{i=1}^{t}u_{\{d,t_e\}}^{\{(a,i)\to(b,t)\}}[k]\Big(d_{\{a,i,d,t_e\}}-d_{\{b,t+1,d,t_e\}}\Big)\\
&-\sum_{a=1}^{N}\sum_{b=1}^{N}\sum_{\{d,t_e\}\in\mathcal{D}}\sum_{t=1}^{t_e-h_{b\to d}^{min}}\sum_{i=1}^{t}V(t-i)U\Big(u_{\{d,t_e\}}^{\{(a,t)\to(b,i)\}}[k]\Big)\Bigg]\\
\overset{(b)}{\leq}&B_0+E\Bigg[\sum_{\{d,t_e\}\in\mathcal{D}}\sum_{n=1}^{N}\sum_{t=1}^{T}d_{\{n,t,d,t_e\}}\sum_{f\in\mathcal{D}_{\{d,t_e\}}}a_f1_{s(f)=n,t_b(f)=t}\\
&-\sum_{a=1}^{N}\sum_{b=1}^{N}\sum_{\{d,t_e\}\in\mathcal{D}}\sum_{t=1}^{t_e-h_{b\to d}^{\min}}\sum_{i=1}^{t}u_{\{d,t_e\}}^{\{(a,i)\to(b,t)\}}\Big(d_{\{a,i,d,t_e\}}-d_{\{b,t+1,d,t_e\}}\Big)\\
&+\sum_{a=1}^{N}\sum_{b=1}^{N}\sum_{\{d,t_e\}\in\mathcal{D}}\sum_{t=1}^{t_e-h_{b\to d}^{\min}}\sum_{i=1}^{t}V(t-i)U\Big(u_{\{d,t_e\}}^{\{(a,t)\to(b,i)\}}\Big)-V(t-i)U\Big(u_{\{d,t_e\}}^{\{(a,t)\to(b,i)\}}[k]\Big)\Bigg]\\
\overset{(c)}{=}&B_0+\sum_{\{d,t_e\}\in\mathcal{D}}\sum_{n=1}^{N}\sum_{t=1}^{T}d_{\{n,f,t\}}\Bigg(\sum_{f\in\mathcal{D}_{\{d,t_e\}}}a_f1_{s(f)=n,t_b(f)=t}\\
&+\sum_{a:(a,n)\in\mathcal{L}}\sum_{i=1}^{t-1}u_{\{d,t_e\}}^{\{(a,i)\to(n,t-1)\}}-\sum_{b:(n,b)\in\mathcal{L}}\sum_{i=t}^{t_e-h_{b\to d}^{\min}}u_{\{d,t_e\}}^{\{(n,t)\to(b,i)\}}\Bigg)\\
&+E\Bigg[\sum_{a=1}^{N}\sum_{b=1}^{N}\sum_{\{d,t_e\}\in\mathcal{D}}\sum_{t=1}^{T}\sum_{i=t}^{T}V(i-t)U\Big(u_{\{d,t_e\}}^{\{(a,t)\to(b,i)\}}\Big)-V(i-t)U\Big(u_{\{d,t_e\}}^{\{(a,t)\to(b,i)\}}[k]\Big)\Bigg] \qquad \text{(C.1)}\\
&\leq B_0+B_1-\varepsilon\sum_{f\in\mathcal{F}}\sum_{n=1}^{N}\sum_{t=1}^{T}a_fd_{\{n,t,d,t_e\}} \qquad \text{(C.2)}\\
&\leq B_0+B_1-\varepsilon a_{\min}\sum_{f\in\mathcal{F}}\sum_{n=1}^{N}\sum_{t=1}^{T}d_{\{n,t,d,t_e\}} \qquad \text{(C.3)}
\end{aligned}
$$

where (a) and (c) hold via similar steps with the derivations of dual decomposition of term Z in (5); (b) holds since spatial-temporal water-filling policy maximizes the term in $E[\cdot]$. $B_0$ is the same constant as the proof of theorem 2; $|B_1| \leq |\mathcal{L}||\mathcal{D}|T^2L_{max}$ and $L_{max}$ is the maximum of $U(\cdot)$; $a_{\min} = \min_{f\in\mathcal{F}} a_f$ is the

minimum mean arrival.

Again according to Foster's theorem [10], we conclude $\mathbf{d}[k]$ is positive recurrent under spatial-temporal water-filling policy for any arrival strictly in $\Omega$.

□

In Lemma 4 and Lemma 2 (Lemma 2 holds for the water-filling policy as well), we have shown $\bar{\mathbf{u}}$ generated by the spatial-temporal water-filling policy is a feasible solution for original problem (3). Combining with Lemma 3, we have proved $\hat{\mathbf{u}} = \lfloor \bar{\mathbf{u}} \rfloor$ can support for given traffic within $((1+\varepsilon+\delta)\mathbf{A}, \mathbf{D}) \in \Omega$, which is the "achievability" part in Theorem 3.

Moreover, based on the bound of drift (C.1), we can characterize the trade-off between the achievable utility and the average virtual queue lengths [13], [14]. Define

$$F_\varepsilon(\mathbf{u}) = \sum_{a=1}^{N}\sum_{b=1}^{N}\sum_{\{d,t_e\}\in\mathcal{D}}\sum_{t=1}^{T}\sum_{i=t}^{T}(i-t)U\left(u_{\{d,t_e\}}^{\{(a,t)\to(b,i)\}}\right),$$

under $\big((1+\varepsilon)\mathbf{A}, \mathbf{D}\big) \in \Omega$, and (C.1) becomes

$$\begin{aligned}\delta[k] &= E\big[\mathbf{L}[k+1] - \mathbf{L}[k] | \mathbf{d}[k] = \mathbf{d}\big] \\ &\le B_0 + VF_\varepsilon(\mathbf{u}) - VE\big[F_\varepsilon(\mathbf{u}[k])\big] - \varepsilon a_{\min}\sum_{f\in\mathcal{F}}\sum_{n=1}^{N}\sum_{t=1}^{T} d_{\{n,t,d,t_e\}}.\end{aligned}$$

Then the telescoping summation of the inequality above across $k \in \{0, 1, \cdots, K-1\}$ is

$$\begin{aligned}&E[\mathbf{L}[K]] - E[\mathbf{L}[0]] \\ &\le KB_0 + VKF_\varepsilon(\mathbf{u}) - V\sum_{i=0}^{K-1}E\big[F_\varepsilon(\mathbf{u}[k])\big] - \varepsilon a_{\min}\sum_{f\in\mathcal{F}}\sum_{n=1}^{N}\sum_{t=1}^{T}\sum_{i=0}^{K-1}E\big[d_{\{n,t,d,t_e\}}[k]\big] \qquad \text{(C.4)}\end{aligned}$$

By rearranging terms in (C.4) and divided $VK$ on both sides, we have

$$\begin{aligned}&\frac{E[\mathbf{L}[K]]}{VK} + \frac{1}{K}\sum_{i=0}^{K-1}E\big[F_\varepsilon(\mathbf{u}[k])\big] \\ &\le \frac{B_0}{V} + F_\varepsilon(\mathbf{u}) + \frac{E[\mathbf{L}[0]]}{VK} - \frac{\varepsilon a_{\min}}{VK}\sum_{f\in\mathcal{F}}\sum_{n=1}^{N}\sum_{t=1}^{T}\sum_{i=0}^{K-1}E\big[d_{\{n,t,d,t_e\}}[k]\big]\end{aligned}$$

ignoring non-negative terms and letting $K \to \infty$, we have

$$\begin{aligned}\bar{F}_\varepsilon = \lim_{K\to\infty}\frac{1}{K}\sum_{i=0}^{K-1}E\big[F_\varepsilon(\mathbf{u}[k])\big] &\le \frac{B_0}{V} + F_\varepsilon(\mathbf{u}), \\ \sum_{f\in\mathcal{F}}\sum_{n=1}^{N}\sum_{t=1}^{T}\bar{d}_{\{n,t,d,t_e\}} &\le \frac{B_0 + V\big(F_\varepsilon(\mathbf{u}) - \bar{F}_\varepsilon\big)}{\varepsilon a_{\min}},\end{aligned}$$

where the average virtual queue length $\bar{d}_{\{n,t,d,t_e\}} = \lim_{K\to\infty}\frac{1}{K}\sum_{i=0}^{K-1}E\big[d_{\{n,t,d,t_e\}}[k]\big]$. By using Jensen's inequality for convex function $F_\varepsilon(\cdot)$, we have $E\big[F_\varepsilon(\bar{\mathbf{u}})\big] \le \bar{F}_\varepsilon$ and we immediately obtain $[O(1/V), O(V)]$ trade-off between the achievable utility and the average virtual queue length:

$$F^*_{\varepsilon\to 0} \le E\big[F_\varepsilon(\bar{\mathbf{u}})\big] \le \frac{B_0}{V} + F^*_{\varepsilon\to 0}, \quad \sum_{f\in\mathcal{F}}\sum_{n=1}^{N}\sum_{t=1}^{T}\bar{d}_{\{n,t,d,t_e\}} \le \frac{B_0 + V\big(F_\varepsilon(\mathbf{u}) - \bar{F}_\varepsilon\big)}{\varepsilon a_{\min}}$$

where $F^*_{\varepsilon\to 0}$ is the optimal value as $\varepsilon \to 0$.

## References


[1] R. Li, A. Eryilmaz, Scheduling for end-to-end deadline-constrained traffic with reliability requirements in multihop networks, IEEE/ACM Trans. Netw. 20 (5) (2012) 1649–1662.

[2] I.-H. Hou, Packet scheduling for real-time surveillance in multihop wireless sensor networks with lossy channels, IEEE Trans. Wireless Commun. 14 (2) (2015) 1071–1079.

[3] Z. Mao, C. E. Koksal, N. B. Shroff, Optimal online scheduling with arbitrary hard deadlines in multihop communication networks, IEEE/ACM Trans. Netw. PP (99).

[4] R. Singh, P. R. Kumar, Decentralized throughput maximizing policies for deadline-constrained wireless networks, in: 2015 54th IEEE Conference on Decision and Control (CDC), 2015, pp. 3759–3766.

[5] M. J. Neely, Opportunistic scheduling with worst case delay guarantees in single and multi-hop networks, in: 2011 Proceedings IEEE INFOCOM, 2011, pp. 1728–1736.

[6] I.-H. Hou, V. Borkar, P. R. Kumar, A theory of QoS for wireless, in: Proc. IEEE Int. Conf. Computer Communications (INFOCOM), Rio de Janeiro, Brazil, 2009, pp. 486–494.

[7] J. J. Jaramillo, R. Srikant, Optimal scheduling for fair resource allocation in ad hoc networks with elastic and inelastic traffic, IEEE/ACM Trans. Netw. 19 (2011) 1125–1136.

[8] L. Ying, S. Shakkottai, A. Reddy, S. Liu, On combining shortest-path and back-pressure routing over multihop wireless networks, IEEE/ACM Trans. Netw. 19 (3) (2011) 841–854.

[9] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge Univ. Press, New York, NY, 2004.

[10] R. Srikant, L. Ying, Communication Networks: An Optimization, Control and Stochastic Networks Perspective, Cambridge University Press, 2014.

[11] S. Knight, H. Nguyen, N. Falkner, R. Bowden, M. Roughan, The internet topology zoo, Selected Areas in Communications, IEEE Journal on 29 (9) (2011) 1765 –1775. doi:10.1109/JSAC.2011.111002.

[12] R. G. Gallager, Discrete Stochastic Processes, The Springer International Series in Engineering and Computer Science, Springer US, New York, NY, 1996.

[13] M. J. Neely, E. Modiano, C.-P. Li, Fairness and optimal stochastic control for heterogeneous networks, IEEE/ACM Trans. Netw. 16 (2008) 396–409.

[14] A. Eryilmaz, R. Srikant, Joint congestion control, routing and MAC for stability and fairness in wireless networks, IEEE J. Sel. Areas Commun. 24 (8) (2006) 1514–1524.