Improving Backpressure-based Adaptive Routing via Incremental Expansion of Routing Choices

Ping Yin
University of California
San Diego
La Jolla, California, USA
piyin@ucsd.edu

Sen Yang
Georgia Institute
of Technology
Atlanta, Georgia, USA
sen.yang@gatech.edu

Jun Xu
Georgia Institute
of Technology
Atlanta, Georgia, USA
jun.xu@cc.gatech.edu

Jim Dai
Cornell University
Ithaca, New York, USA
jd694@cornell.edu

Bill Lin
University of California
San Diego
La Jolla, California, USA
billlin@ucsd.edu

ABSTRACT

Backpressure-based adaptive routing algorithms have been studied extensively in the literature. Although backpressure-based adaptive routing algorithms have been shown to be network-wide throughput optimal, they typically have poor delay performance under light or moderate loads because packets may be sent over unnecessarily long routes. Further, backpressure-based algorithms have required every node to compute differential backlogs for every destination queue with the corresponding destination queue at every adjacent node. This computation is expensive given the large number of possible pairwise differential backlogs and requires many exchanges of backlog information between adjacent nodes. In this paper, we propose new backpressure-based adaptive routing algorithms that only use shortest-path routes to destinations when they are sufficient to accommodate the given traffic load, but the proposed algorithms will incrementally expand routing choices as needed to accommodate increasing traffic loads. We show analytically by means of fluid analysis that the proposed algorithms retain network-wide throughput optimality, and we show empirically by means of simulations that our proposed algorithms provide substantial improvements in delay performance. Our evaluations further show that in practice, our approach dramatically reduces the number of pairwise differential backlogs that have to be computed and the amount of corresponding backlog information that has to be exchanged because routing choices are only incrementally expanded as needed.

Keywords

Backpressure; adaptive routing; delay reduction

ANCS '17 May 18-19, 2017, Beijing, China
© 2017 ACM. ISBN 123-4567-24-567/08/06…$15.00
DOI: 10.475/123_4

1. INTRODUCTION

The backpressure algorithm first introduced in [16] has been extensively studied in the literature [4, 19, 12, 13, 1, 14, 2]. It was initially introduced in the context of wireless radio networks, but it can be easily adapted to wireline networks as well, for example for packet routing in backbone networks for the Internet. Although backpressure-based adaptive routing algorithms have been shown to be network-wide throughput optimal [16], they have been rarely used in practice due to several shortcomings.

First, backpressure-based algorithms typically have poor end-to-end delay performance under light or moderate loads because packets may be sent over unnecessarily long routes, potentially even traversing routing loops. The original backpressure algorithm allows the routing of a packet to any adjacent node as the next-hop, even if the routing decisions will cause a packet to take long detours.

Second, backpressure algorithms typically maintain per-destination queues, and the routing and scheduling decisions are based on maintaining differential backlogs for every destination queue with the corresponding destination queue at every adjacent node. Although the implementation of per-destination queues has often been cited as a concern, we note that significant advances have been made in memory architectures since the original backpressure routing work for implementing huge packet buffers at line rates that support a very large number of logical queues [10, 15, 18, 17][1].

Despite these advances that address the practical implementation of per-destination queues, the need remains for backpressure-based algorithms to compute differential backlogs for every destination queue with the corresponding destination queue at every adjacent node. This computation is expensive given the large number of possible pairwise differential backlogs. Further, the computation requires many exchanges of backlog information between every pair of adjacent nodes for every pair of destination queues. The substantial amount of computations and associated information

[1] Some of these memory architectures have been in commercial use in modern Internet routers to support per-class queuing or per-flow queuing, where the number of logical queues far exceeds the number of destinations needed for backpressure routing.

978-1-5090-6386-4/17 $31.00 © 2017 IEEE
DOI 10.1109/ANCS.2017.11

exchanges remain significant impediments for practical implementations.

To address the poor delay performance concern, the backpressure idea can be applied to a fixed routing problem, where packets are forced to use shortest paths [4]. However, limiting routing choices shrinks the network stability region and is thus not throughput optimal. As we shall later see in the evaluation section, limiting the routing choices to just shortest paths will cause the network to saturate much earlier than if all routing choices are permitted.

Several prior works [12, 11, 8, 19] have recognized the importance of favoring shorter paths instead of only considering shortest paths. However, all these approaches still require the computation of differential backlogs for all destination queues between every pair of neighboring nodes and the associated exchanges of backlog information. Although the approach studied in [19] further offers provably minimal-hop routing, their solution dramatically increases the number of queues that each node needs to maintain since their approach requires per-hop queues for each destination. The dramatic increase in the number of queues makes the associated differential backlog computation problem even more difficult.

Finally, [4, 2] introduced the idea of shadow queues for making adaptive routing decisions. Their idea is to create a shadow network in which a backpressure algorithm is used to make routing decisions. Although their approach does not require per-destination queuing of packets, their solution still incurs the same calculation complexity as the original backpressure algorithm for the shadow queues in that the same computation of differential backlogs for all destinations between every pair of neighboring nodes and the same associated exchanges of backlog information are still required. Further, although their approach stores the actual packets in per-neighbor queues instead of per-destination queues, the amount of packet buffer storage that each node needs remains the same[2]. We believe that our solution is complementary to [4, 2] in that the algorithms described in this paper can be used as the shadow queue algorithm in their solution framework.

1.1 Our approach

In this paper, we propose several modified backpressure-based algorithms that address the aforementioned concerns. Our work applies to both wirelined and wireless networks. In particular, our modified backpressure-based algorithms are based on the idea that routing choices should be limited to next-hops that are along shortest path routes by default. This approach significantly reduces the amount of differential backlog calculations and associated information exchanges as each node only has to consider a subset of next hops for each destination. In addition, this approach addresses the delay performance concern by only routing packets along shortest path routes when the traffic load is light or moderate.

We propose to detect congestion by monitoring destination queue lengths or the waiting times of packets in the destination queues. When the length of a destination queue or the waiting time of a packet at the head of a destination queue exceeds some threshold, the routing choices for the corresponding destination queue get *expanded* to include next hops that are not along shortest path routes. This expansion of routing choices is on a per-destination queue and a per-node basis. Although a packet may be forwarded to a next hop that is not along a shortest path route to the destination, the packet may still be forwarded along a shortest path route from this next hop to the destination if the corresponding destination queue at this next hop is not yet congested. This way, routing choices are *incrementally* expanded at different nodes in the network as needed with increasingly longer paths considered. In effect, a packet can take a detour whenever it encounters congestion along the way to the destination. When a node expands its routing choices for packets for a particular destination, it notifies other adjacent nodes to begin providing backlog information, and it expands its differential backlog calculations with those adjacent nodes as well.

1.2 Contributions of the paper

The main contributions of this paper are as follows:

- We propose two modified backpressure-based algorithms, called L-BP and A-BP, that are based on the incremental expansion of routing choices in response to congestion at a node on a per-destination basis. L-BP detects congestion by monitoring destination queue lengths, whereas A-BP detects congestion by monitoring the waiting times of packets at the heads of destination queues. We refer to these algorithms as *hybrid* backpressure algorithms since some destination queues are in shortest-path mode while others are allowed to be forwarded to any neighbor node.

- We prove theoretically that both algorithms are network-wide throughput optimal (i.e., the proposed algorithms can explore the same network stability region as the original backpressure algorithm). In particular, we use a fluid model for our proofs, which models well the system dynamics of our modified algorithms.

- Our proposed algorithms can be applied to both wireline and wireless networks. In particular, we extensively evaluate our proposed algorithms on the adaptive Internet routing problem. We show our evaluations on the Abilene network [9], a public PoP-level academic network in the US, using actual traffic profiles measured on the network. Our simulation results show that our proposed algorithms indeed provide substantial improvements in delay performance. Our simulation results further show that in practice, our approach dramatically reduces the number of pairwise differential backlogs that have to be computed and the amount of corresponding backlog information that has to be exchanged because routing choices are only incrementally expanded as needed. That is, only a subset of destination queues in a subset of nodes need to consider expanded routing choices even for traffic loads that approach the edge of the network stability region.

The rest of the paper is organized as follows: In Section 2, we present the basic network model and summarize the original backpressure algorithm. In Section 3, we present our hybrid backpressure-based adaptive routing algorithms. In Section 4, we use fluid analysis to prove that both of these algorithms are throughput optimal. In Section 5, we

[2] The state-of-the-art DRAM-based packet buffers [10, 15, 18, 17] can store a huge number of packets, tens of gigabytes, and support a very large number of logical queues.

describe our experimental setup and the simulation results. We conclude our paper in Section 6.

2. BACKGROUND

2.1 The network model

We consider a multi-hop network represented by a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ is the set of nodes, and $\mathcal{L}$ is the set of directed links. All packets that enter the network are associated with a particular *commodity* that corresponds to the packet *destination*. A packet that is destined for node c is regarded as a commodity c packet, $c = 1, \ldots, N$. We use $\mathcal{L}_c$ to denote the routing restrictions for commodity c, which is the set of all links (a, b) that a commodity c packet is allowed to use. Obviously, if there is no routing restriction for commodity c packets, then $\mathcal{L}_c = \mathcal{L}$. The link capacity $\mu_{ab}(t)$ for link (a, b) is defined to be the maximum number of packets that can be transmitted over link (a, b) in one timeslot[3]. In general, multiple commodities might be transmitted over this link during a single timeslot, but the total rate cannot exceed the link capacity $\mu_{ab}(t)$.

Each node i maintains a set of internal queues for storing network layer packets according to their commodity. Let $A_n^{(c)}(t)$ represent the *cumulative* amount of new commodity c packets that exogenously arrives to source node n by timeslot t (since time 0). Assume these arrival processes are *admissible*. Let $D_{ab}^{(c)}(t)$ be the *cumulative* amount of commodity c packets sent from node a to node b via link (a, b) by timeslot t (since time 0), $a, b, c = 1, \ldots, N$.

Let $Q_n^{(c)}$ denote the internal queue in node n that stores packets destined for node c. With a slight abuse of notation, let $Q_n^{(c)}(t)$ represent the current backlog of commodity c packets stored in an internal queue at node n. The queue backlog $Q_n^{(c)}(t)$ contains packets that arrived exogenously by $A_n^{(c)}(t)$ as well as packets that arrived endogenously from other nodes by $D_{an}^{(c)}(t)$, $a = 1, \ldots, N$. We define $Q_c^{(c)}(t) = 0$ and $D_{cn}^{(c)}(t) = 0$ for all t, $c = 1, \ldots, N$ and $n = 1, \ldots, N$, so that any packet that has been delivered to its destination is assumed to exit the network right away. The queue backlogs then satisfy the following equation for all $n = 1, \ldots, N$ and $c = 1, \ldots, N$ such that $n \neq c$.

$$Q_n^{(c)}(t) = Q_n^{(c)}(0) - \sum_{b=1}^{N} D_{nb}^{(c)}(t) + \sum_{a=1}^{N} D_{an}^{(c)}(t) + A_n^{(c)}(t) \quad (1)$$

2.2 The backpressure algorithm

The original backpressure algorithm was first introduced in [16] in the context of wireless radio networks. It has been shown to achieve optimal throughput [16] and can be served as a solution to certain multi-commodity flow problems [3].

For each link (a, b), the algorithm defines the *optimal commodity* $c_{ab}^*(t)$ as the commodity that maximizes the differential backlog (ties broken arbitrarily):

$$c_{ab}^*(t) \triangleq \arg \max_{\{c | (a,b) \in \mathcal{L}_c\}} \left[Q_a^{(c)}(t) - Q_b^{(c)}(t) \right], \quad (2)$$

and defines $W_{ab}^*(t)$ as the corresponding optimal weight:

$$W_{ab}^*(t) \triangleq \max \left[Q_a^{(c_{ab}^*(t))}(t) - Q_b^{(c_{ab}^*(t))}(t), 0 \right]. \quad (3)$$

If $W_{ab}^*(t) > 0$, then the internal commodity $c_{ab}^*(t)$ queue is scheduled to be served , and the packets will be transmitted over link (a, b) during timeslot t. Otherwise, no packets will be transmitted over link (a, b) during timeslot t.

It is common in wireless networks that only a subset of all links, referred to as a *schedule*, can transmit packets simultaneously due to interference. Let $\mathcal{S}$ be the set of all possible schedules. The original backpressure algorithm finds the *optimal schedule*, $S^*(t) \in \mathcal{S}$ as an optimization problem as follows:

$$S^*(t) = \arg \max_{S \in \mathcal{S}} \sum_{(a,b) \in S} W_{ab}^*(t) \mu_{ab}(t) \quad (4)$$

At each timeslot t, for each link $(a, b) \in S^*(t)$, $\mu_{ab}(t)$ packets are removed from $Q_a^{(c_{ab}^*)}$ and transmitted to $Q_b^{(c_{ab}^*)}$. If $Q_a^{(c_{ab}^*)}$ does not have $\mu_{ab}(t)$ packets, then all packets will leave $Q_a^{(c_{ab}^*)}$. For *wireline* networks, $\mu_{ab}(t)$ is a typically constant (e.g., one packet per timeslot), and $\mathcal{S}$ is always the set of all links since all links can be activated without interfering with each other.

The intuition behind the backpressure algorithm is that packets may not be transmitted if the differential backlog is non-negative, which indicates a congestion at the downstream node. The original backpressure algorithm considers $\mathcal{L}_c = \mathcal{L}$ for any commodity c packet. That is, any packet in node a, no matter what commodity it belongs to, can transmit to *any* neighbor node of node a, as long as Equation 3 is satisfied. This feature essentially exploits all feasible paths in the network for any commodity packet, and as a result, stabilizes the network under heavy traffic loads. However, this feature also incurs large end-to-end packet delays when the network is only lightly or moderatey loaded because packets unnecessarily explore and traverse long paths.

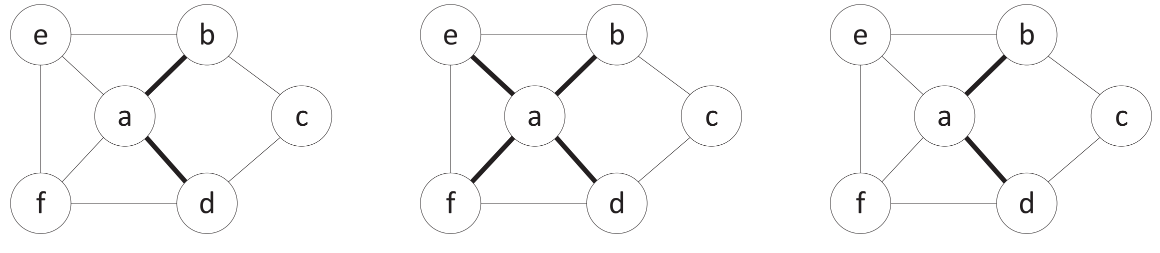
3. HYBRID BP ALGORITHMS

As it has been mentioned in Section 2.2, the original backpressure algorithm assumes that $\mathcal{L}_c$ contains all of the links of the network, $\mathcal{L}$. This unconstrained routing may introduce large delays when the traffic load is light, as a packet can unnecessarily explore long paths.

One way to reduce the end-to-end delay is to restrict $\mathcal{L}_c$ to only shortest paths. We call this Shortest-Path Backpressure algorithm (SPBP). Assume node a has two neighbors, b and c. Neighbor c is on the shortest path for commodity d packets, while neighbor b is not. In this case, commodity d packets in node a can only transmit to node c on condition that d is the optimal commodity for link (a, c) and its weight, computed by Equation 3, is positive. In comparison, the original backpressure algorithm allows commodity d packets to transmit to both node b and c, as long as d is the optimal commodity for each link and the weight is positive.

While SPBP can reduce the delay, it also shrinks the network stability region, as it limits the routing choices compared with the original backpressure algorithm. On the other hand, our hybrid backpressure algorithms can retain the same stability region as the original backpressure and reduce delay for light or moderate traffic loads by incremental

[3] Although we define $\mu_{ab}(t)$ here in terms of number of packets, our algorithms and results are applicable to any unit of data as appropriate for the intended application. For example, the unit of data can just be bits or be a rate.



(a) Queue $Q_a^{(c)}$ is initially in Phase I. (b) Queue $Q_a^{(c)}$ switches over to Phase II. (c) Queue $Q_a^{(c)}$ returns back to Phase I.

Figure 1: A simple network showing how $\mathcal{L}_c$ changes.

expansion of routing choices. It starts with the shortest-path routing choices as described above for SPBP. To overcome its shortcomings, a dynamic change of routing choices is introduced.

Each internal queue in a node n has two phases of routing. A queue in Phase I can switch over to Phase II when a *transition criterion* is satisfied. A Phase II queue can also return Phase I when the transition criterion is no longer met.

Similar to the SPBP, packets in a Phase I queue can only go to a subset of the neighbor nodes, which are on the shortest paths from current node to the destination. In Phase II, similar to the original backpressure algorithm, packets in that queue can be transmitted to **any** neighbor of the current node. The rest of the backpressure scheduling rules are the same.

For each link (a, b), the algorithm defines the *optimal commodity* $c^*_{ab}(t)$ as the commodity that maximizes the differential backlog (ties broken arbitrarily):

$$c^*_{ab}(t) \triangleq \arg \max_{\{c|(a,b)\in\mathcal{L}_c\}} \Big[Q_a^{(c)}(t) - Q_b^{(c)}(t)\Big], \tag{5}$$

and define $W^*_{ab}(t)$ as the corresponding optimal weight:

$$W^*_{ab}(t) \triangleq \max \Big[Q_a^{(c^*_{ab}(t))}(t) - Q_b^{(c^*_{ab}(t))}(t), 0\Big]. \tag{6}$$

Solve the optimization problem

$$S^*(t) = \arg\max_{S\in\mathcal{S}} \sum_{(a,b)\in S} W^*_{ab}(t)\mu_{ab}(t) \tag{7}$$

The routing choices $\mathcal{L}_c$ are changing dynamically. In the beginning, all internal per-destination queues are in Phase I. This is equivalent to restricting $\mathcal{L}_c$ to allow only links on the shortest paths. When a transition criterion is satisfied, queue $Q_n^{(c)}$ switches over to Phase II, and we add all (n, k) to set $\mathcal{L}_c$ for any neighbor k of node n. When the transition criterion is no longer satisfied, $Q_n^{(c)}$ returns back to Phase I, and we remove those added links from $\mathcal{L}_c$. When all queues are in Phase II, the $\mathcal{L}_c$ becomes $\mathcal{L}$, and this is equivalent to the original backpressure algorithm.

Consider the network shown in Figure 1 as an example and consider queue $Q_a^{(c)}$. In the beginning, $(a, b), (a, d) \in \mathcal{L}_c$, because node b and node c are on shortest paths to node c. When $Q_a^{(c)}$ switches over to Phase II, $(a, e), (a, f)$ are added to $\mathcal{L}_c$. When $Q_a^{(c)}$ returns back to Phase I, $(a, e), (a, f)$ are then removed from $\mathcal{L}_c$.

We propose two transition criteria, a length-based criterion and an age-based criterion, which we refer to the corresponding hybrid backpressure algorithms as L-BP and A-BP, respectively:

- **L-BP**: Let a constant L_{max} to be the maximum backlog that a queue $Q_n^{(c)}$ can stay in Phase I. Whenever $Q_n^{(c)}(t) > L_{max}$, the queue $Q_n^{(c)}$ switches over to Phase II. Whenever $Q_n^{(c)}(t) \leq L_{max}$, it returns back to Phase I.

- **A-BP**: Consider the head packet of queue $Q_n^{(c)}$. Let $E_n^{(c)}(t)$ represent the *age* of the head packet, which is the period from the timeslot that the head packet enters the queue until current timeslot t. Let A_{max} to be the maximum *age* of the head packet for its queue to stay in Phase I. Whenever $E_n^{(c)}(t) > A_{max}$, the queue switches over to Phase II. Whenever $E_n^{(c)}(t) \leq A_{max}$, it returns back to Phase I.

4. THROUGHPUT OPTIMALITY

In this section, we use fluid model to prove that both the L-BP and A-BP hybrid backpressure algorithms are throughput optimal.

4.1 Modeling and assumptions

At each timeslot, each node needs to make a schedule to transmit data in the network. Let $\mathcal{S}$ be the set of all possible schedules. Each schedule $\beta = (\beta_{ab}^{(c)} : a, b, c = 1, \ldots, N) \in \mathcal{S}$ is a vector in $\mathbb{Z}^{N\times N\times N}$, where $\beta_{ab}^{(c)}$ gives amount of commodity c data sent from node a to node b via link (a, b) under schedule β. It is assumed that $\beta_{cn}^{(c)} = 0$ for all $\beta \in \mathcal{S}$, $c = 1, \ldots, N$ and $n = 1, \ldots, N$. It is also assumed that $\mathcal{S}$ is *monotone* in the following sense: if $\beta \in \mathbb{Z}^{N\times N\times N}$ and there exists $\beta' \in \mathcal{S}$ such that $\beta \leq \beta'$, then $\beta \in \mathcal{S}$. This is because if β' is a valid schedule and we decrease the amount of data sent in some of the links, the resulted schedule must also be a valid one.

We define $\mathcal{S}(t) \subset \mathcal{S}$ as the set of valid schedules given the scheduling strategy and the systems status at timeslot t. We assume that $\mathcal{S}(t)$ maintains the monotonicity of $\mathcal{S}$, i.e., $\beta \in \mathcal{S}(t)$ if there exists $\beta' \in \mathcal{S}(t)$ such that $\beta \leq \beta'$. A necessary constraint on $\mathcal{S}(t)$ is that we must have

$$\sum_{b=1}^{N} \beta_{nb}^{(c)} \leq Q_n^{(c)}(t) \tag{8}$$

for all $\beta \in \mathcal{S}(t)$ and $n, c = 1, \ldots, N$. For each schedule $\beta \in \mathcal{S}$, we define a "collapsed" schedule $\gamma(\beta) = (\gamma_n^{(c)}(\beta) : n, c = 1, \ldots, N) \in \mathbb{Z}^{N\times N}$ where $\gamma_n^{(c)}(\beta) = \sum_{b=1}^{N} \beta_{nb}^{(c)} - \sum_{a=1}^{N} \beta_{an}^{(c)}$

is the speed that schedule β empties the backlog in queue $Q_n^{(c)}$. Let $\mathit{\Gamma}_{\mathcal{S}} \subset \mathbb{Z}^{N\times N}$ be the set of all possible collapsed schedules given $\mathcal{S}$. Let $< \mathit{\Gamma}_{\mathcal{S}} >$ be the convex hull of $\mathit{\Gamma}_{\mathcal{S}}$. Assume each link (a,b), $a,b = 1,\ldots,N$, has a finite maximal transmission speed, i.e., $\exists\ R \geq 0$ such that $\beta_{ab}^{(c)} \leq R$ for all $\beta \in \mathcal{S}$ and $a,b,c = 1,\ldots,N$. Then both $\mathcal{S}$ and $\mathit{\Gamma}_{\mathcal{S}}$ are finite sets.

To analyze the stability of our schemes, we first define a family of generalized max-weighted scheduling schemes as follows. Define the weight $W(\beta, Q(t))$ of schedule β given queue length $Q(t)$ as

$$W(\beta, Q(t)) \triangleq \sum_{a=1}^{N}\sum_{b=1}^{N}\sum_{c=1}^{N}\beta_{ab}^{(c)}[Q_a^{(c)}(t) - Q_b^{(c)}(t)]$$

At each timeslot, a schedule $\beta \in \mathcal{S}$ that solves the following optimization problem will be selected for activation.

$$\begin{aligned}\max_{\beta} \quad & W(\beta, Q(t)) \\ s.t. \quad & \beta \in \mathcal{S}(t)\end{aligned}$$

Both the baseline backpressure scheme and our hybrid backpressure schemes belong to this family of generalized max-weighted scheduling schemes. The only difference among these schemes is the definition of the valid schedule set $\mathcal{S}(t)$ at each timeslot.

Let $\mathcal{S}^{BP}(t)$ be the set of valid schedules at timeslot t for the baseline backpressure scheme. Let $\mathcal{S}^{L}(t)$ be the set of valid schedules at timeslot t for the L-BP hybrid backpressure scheme. Let $\mathcal{S}^{A}(t)$ be the set of valid schedules at timeslot t for the A-BP hybrid backpressure scheme. We have

$$\mathcal{S}^{BP}(t) = \left\{\beta \in \mathcal{S} \,\middle|\, \sum_{b=1}^{N}\beta_{nb}^{(c)} \leq Q_n^{(c)}(t) \text{ for all } n,c = 1,\ldots,N\right\}$$

$$\mathcal{S}^{L}(t) \subseteq \mathcal{S}^{BP}(t)$$

$$\mathcal{S}^{A}(t) \subseteq \mathcal{S}^{BP}(t)$$

$$\mathcal{S}^{L}(t) \supseteq \left\{\beta \in \mathcal{S}^{BP}(t) \,\middle|\, \beta_{nb}^{(c)} = 0 \text{ if } Q_n^{(c)}(t) < L_{max},\ n,b,c = 1,\ldots,N\right\} \qquad (9)$$

$$\mathcal{S}^{A}(t) \supseteq \left\{\beta \in \mathcal{S}^{BP}(t) \,\middle|\, \triangle\right\} \qquad (10)$$

$\triangle$: $\beta_{nb}^{(c)} = 0$ if at the begining of timeslot t the head packet in $Q_n^{(c)}$ has waited for more than A_{max} timeslots, $n,b,c = 1,\ldots,N$.

Let $T_\beta(t)$, $\beta \in \mathcal{S}$, be the cumulative number of time slots that schedule β was employed by timeslot t. From (8) , we have

$$D_{ab}^{(c)}(t) = \sum_{\beta\in\mathcal{S}}\sum_{\ell=1}^{t}\beta_{ab}^{(c)}\cdot(T_\beta(\ell) - T_\beta(\ell-1)) \qquad (11)$$

To formulate the fluid model, we extend the above discrete time functions to the continuous time domain. Specifically, for $t \in [0,+\infty)$, we define

$$A_n^{(c)}(t) = A_n^{(c)}(\lfloor t \rfloor) \qquad n,c = 1,\ldots,N$$

$$Q_n^{(c)}(t) = Q_n^{(c)}(\lfloor t \rfloor) \qquad n,c = 1,\ldots,N$$

$$D_{ab}^{(c)}(t) = D_{ab}^{(c)}(\lfloor t \rfloor) + (t - \lfloor t \rfloor)(D_{ab}^{(c)}(\lceil t \rceil) - D_{ab}^{(c)}(\lfloor t \rfloor)), \qquad a,b,c = 1,\ldots,N$$

$$T_\beta(t) = T_\beta(\lfloor t \rfloor) + (t - \lfloor t \rfloor)(T_\beta(\lceil t \rceil) - T_\beta(\lfloor t \rfloor)) \qquad \beta \in \mathcal{S}$$

where $\lceil t \rceil$ is the largest integer that is smaller than or equal to t and $\lfloor t \rfloor$ is the smallest integer that is larger than or equal to t.

Assume the arrival process $A(t)$ satisfies a strong law of large numbers (SLLN), i.e., there exists a constant arrival rate matrix $\lambda = (\lambda_n^{(c)} : n,c = 1,\ldots,N)$, such that, with probability one,

$$\lim_{t\to\infty}\frac{A_n^{(c)}(t)}{t} = \lambda_n^{(c)} \qquad \forall\ n,c = 1,\ldots,N \qquad (12)$$

Without loss of generality, we assume that $\lambda_c^{(c)} = 0$ for all $c = 1,\ldots,N$ for simplicity.

4.2 Basic fluid model equations

We now investigate the stochastic process $(Q(t), D(t), T(t))$, where

$$Q(t) \triangleq \left\{Q_n^{(c)}(t) \,\middle|\, n,c = 1,2,\ldots,N\right\}$$

$$D(t) \triangleq \left\{D_{ab}^{(c)}(t) \,\middle|\, a,b,c = 1,2,\ldots,N\right\}$$

$$T(t) \triangleq \left\{T_\beta(t) \,\middle|\, \beta \in \mathcal{S}\right\}$$

Let $(\mathit{\Omega}, \mathcal{F}, \mathbb{P})$ be the probability space that this stochastic process is defined on, where $\mathit{\Omega}$ is the sample space, $\mathcal{F}$ is a σ-field on $\mathit{\Omega}$, and $\mathbb{P}$ is the probability measure on $(\mathit{\Omega}, \mathcal{F})$. We shall sometimes use the notations $Q(\cdot,\omega)$, $D(\cdot,\omega)$ and $T(\cdot,\omega)$ to explicitly denote the dependency on the sample path $\omega \in \mathit{\Omega}$.

Now, for each $r > 0$, we define fluid scaled processes

$$\left(\hat{Q}^r(t,\omega), \hat{D}^r(t,\omega), \hat{T}^r(t,\omega)\right) \triangleq \frac{1}{r}\left(Q(rt,\omega), D(rt,\omega), T(rt,\omega)\right)$$

Proposition 1 (Fluid Model). *For each sample path $\omega \in \mathit{\Omega}$ satisfying (12) and any sequence $\{r_n\}$ with $r_n \to \infty$, there exists a subsequence $\{r_{n_k}\}$ and continuous functions $(\hat{Q}, \hat{D}, \hat{T})$ with $\hat{Q}(0) = 0$, such that*

$$\left(\hat{Q}^{r_{n_k}}(\cdot,\omega), \hat{D}^{r_{n_k}}(\cdot,\omega), \hat{T}^{r_{n_k}}(\cdot,\omega)\right) \to \left(\hat{Q}, \hat{D}, \hat{T}\right) \quad u.o.c \quad as\ k \to \infty \qquad (13)$$

where the convergence is uniform on compact sets (u.o.c). The three-tuple $(\hat{Q}, \hat{D}, \hat{T})$ is said to be a fluid limit path of the system. It satisfies the following fluid model equations

$$\hat{Q}_n^{(c)}(t) = \lambda_n^{(c)}t + \sum_{a=1}^{N}\hat{D}_{an}^{(c)}(t) - \sum_{b=1}^{N}\hat{D}_{nb}^{(c)}(t) \qquad n,c = 1,\ldots,N \ and\ n \neq c \qquad (14)$$

$$\hat{Q}_c^{(c)}(t) = 0 \qquad c = 1,\ldots,N \qquad (15)$$

$$\hat{D}_{ab}^{(c)}(t) = \sum_{\beta\in\mathcal{S}}\beta_{ab}^{(c)}\hat{T}_\beta(t) \qquad a,b,c = 1,\ldots,N \qquad (16)$$

$$\sum_{\beta\in\mathcal{S}}\hat{T}_\beta(t) = t \qquad (17)$$

$$\hat{Q}_n^{(c)}(t) \geq 0 \qquad n,c = 1,\ldots,N$$

$$\hat{T}_\beta(0) = 0, \quad \hat{T}_\beta(\cdot)\ is\ non\text{-}decreasing \qquad \beta \in \mathcal{S}$$

$$\hat{T}_\beta(t) - \hat{T}_\beta(s) \leq t - s\ for\ 0 \leq s < t \qquad \beta \in \mathcal{S}$$

The proof of Proposition 1 is somewhat standard. We refer the reader to [5].

4.3 Main results

We first give the formal definition of throughput optimilities.

Definition 1 (Rate Stability). *We say the system is rate stable, if with probability one,*

$$\lim_{t\to\infty} \frac{\sum_{n=1}^N D_{nc}^{(c)}(t)}{t} = \sum_{n=1}^N \lambda_n^{(c)} \qquad c = 1, \ldots, N \tag{18}$$

for any arrival process satisfying (12).

Note that the left-hand-side of (18) is actually the long-run average rate of commodity c packets that *depart* from the network, while the right-hand-side is the long-run average rate of commodity c packets that *arrive* to the network. In other words, the system is guaranteed to achieve 100% throughput whenever it is rate stable.

Definition 2 (Throughput Optimal). *We say an arrival process $A(t)$ is admissible if its arrival rate matrix λ belongs to $< \mathit{\Gamma}_{\mathcal{S}} >$. The system is said to be throughput optimal if it is rate stable under any admissible arrival process.*

The main results of this section are stated as follows.

Theorem 1. *The system is throughput optimal when working under the L-BP hybrid backpressure scheme.*

Theorem 2. *The system is throughput optimal when working under the A-BP hybrid backpressure scheme.*

We'll give the proof of these results using fluid model in the rest of this section. More specifically, we'll first prove that their fluid model is weakly stable as defined in Definition 3 and the stability of the original system is then guaranteed by Proposition 2.

Definition 3 (Weak Fluid Stability). *The fluid model is said to be weakly stable if for each fluid limit path with $\hat{Q}(0) = 0$ we have $\hat{Q}(t) = 0$ for all $t \geq 0$.*

Proposition 2. *For a given arrival rate matrix λ, the system is rate stable if the corresponding fluid model is weakly stable.*

Proof. From our assumptions in Section 4.1, we know that $\beta_{cn}^{(c)} = 0$ for all $\beta \in \mathcal{S}$ and $n, c = 1, \ldots, N$. Then by (16), we know that $\hat{D}_{cn}^{(c)}(t) = 0$ for all $n, c = 1, \ldots, N$. This equality will be used in justifying Equation (19) below.

Denote $\mathcal{N} = \{1, \ldots, N\}$ as the set of nodes in the network. From (14) and (15), for each commodity c, we have

$$\begin{aligned}
\sum_{n=1}^N \hat{Q}_n^{(c)}(t) &= \sum_{n\in\mathcal{N}\setminus\{c\}} \hat{Q}_n^{(c)}(t) \\
&= \sum_{n\in\mathcal{N}\setminus\{c\}} \left(\lambda_n^{(c)} t + \sum_{a=1}^N \hat{D}_{an}^{(c)}(t) - \sum_{b=1}^N \hat{D}_{nb}^{(c)}(t)\right) \\
&= \sum_{n\in\mathcal{N}\setminus\{c\}} \lambda_n^{(c)} t + \sum_{n\in\mathcal{N}\setminus\{c\}} \sum_{a=1}^N \hat{D}_{an}^{(c)}(t) \\
&\quad - \sum_{n\in\mathcal{N}\setminus\{c\}} \sum_{b=1}^N \hat{D}_{nb}^{(c)}(t) \\
&= \sum_{n=1}^N \lambda_n^{(c)} t + \sum_{n\in\mathcal{N}\setminus\{c\}} \sum_{a=1}^N \hat{D}_{an}^{(c)}(t) - \sum_{n=1}^N \sum_{b=1}^N \hat{D}_{nb}^{(c)}(t) && (19) \\
&= \sum_{n=1}^N \lambda_n^{(c)} t - \sum_{c=1}^N \hat{D}_{nc}^{(c)}(t) && (20)
\end{aligned}$$

Equation (19) stands because (1) $\lambda_c^{(c)} = 0$ for $c = 1, \ldots, N$ from our assumptions in in Section 4.1; and (2) $\hat{D}_{cn}^{(c)}(t) = 0$ for $n, c = 1, \ldots, N$.

If the fluid model is weakly stable, by Definition 3, we know that $\sum_{n=1}^N \hat{Q}_n^{(c)}(t) = 0$ for all $t \geq 0$. Then by (20), we have

$$\sum_{c=1}^N \hat{D}_{nc}^{(c)}(t) = \sum_{n=1}^N \lambda_n^{(c)} t$$

Thus

$$\sum_{c=1}^N \hat{D}_{nc}^{(c),r}(t, \omega) \to \sum_{n=1}^N \lambda_n^{(c)} t \quad \textit{u.o.c} \quad \textit{as } r \to \infty$$

In particular, $\sum_{c=1}^N \hat{D}_{nc}^{(c),r}(1, \omega) \to \sum_{n=1}^N \lambda_n^{(c)}$ as $r \to \infty$ or

$$\lim_{r\to\infty} \frac{\sum_{n=1}^N D_{nc}^{(c)}(r)}{r} = \sum_{n=1}^N \lambda_n^{(c)}$$

which is actually the same to (18), proving the theorem. □

4.4 Throughput Optimality of L-BP

We first present a lemma that will be used in the Proof of Theorem 1.

Lemma 1. *Each of the fluid limit paths satisfies the following fluid equation if the system works under the L-BP hybrid backpressure scheme.*

For each $\beta \in \mathcal{S}$, $\dfrac{d}{dt}\hat{T}_\beta(t) = 0$ *if* $W(\beta, \hat{Q}(t)) < \max_{\alpha\in\mathcal{S}} W(\alpha, \hat{Q}(t))$

$$\tag{21}$$

Proof. The proof is similar to the proof of Lemma 4 in [6]. For completeness, we produce a full proof here.

Suppose $(\hat{Q}, \hat{D}, \hat{T})$ is a fluid limit path. Fix a sample path $\omega \in \mathit{\Omega}$ such that (12) and (13) hold. There exists a sequence $\{r_k\}$ with $r_k \to \infty$ as $k \to \infty$, such that

$$\begin{aligned}
&\left(\hat{Q}^{r_k}(\cdot, \omega), \hat{D}^{r_k}(\cdot, \omega), \hat{T}^{r_k}(\cdot, \omega)\right) \\
&\qquad \to \left(\hat{Q}, \hat{D}, \hat{T}\right) \quad \textit{u.o.c} \quad \textit{as } k \to \infty
\end{aligned} \tag{22}$$

Fix a time $t \geq 0$. Define $\mathcal{A}(\hat{Q}(t)) = \{(a, c) \,|\, \hat{Q}_a^{(c)}(t) > 0\}$. Let $\tilde{\alpha} = \mathrm{argmax}_{\alpha\in\mathcal{S}} W(\alpha, \hat{Q}(t))$. Define $\tilde{\beta}$ via

$$\tilde{\beta}_{ab}^{(c)} = \begin{cases} \tilde{\alpha}_{ab}^{(c)} & \text{if } (a, c) \in \mathcal{A}(\hat{Q}(t)) \\ 0 & \text{otherwise} \end{cases}$$

We have $\tilde{\beta} \in \mathcal{S}$ since $\tilde{\beta} \leq \tilde{\alpha}$. We now prove that $W(\tilde{\beta}, \hat{Q}(t)) = \max_{\alpha\in\mathcal{S}} W(\alpha, \hat{Q}(t))$. Note that

$$\begin{aligned}
W(\tilde{\alpha}, \hat{Q}(t)) &= \sum_{a=1}^{N}\sum_{b=1}^{N}\sum_{c=1}^{N} \tilde{\alpha}_{ab}^{(c)}[\hat{Q}_a^{(c)}(t) - \hat{Q}_b^{(c)}(t)] \\
&= \sum_{a=1}^{N}\sum_{b=1}^{N}\sum_{c=1}^{N} \tilde{\alpha}_{ab}^{(c)}[\hat{Q}_a^{(c)}(t) - \hat{Q}_b^{(c)}(t)] \cdot 1_{\hat{Q}_a^{(c)}(t)=0} \\
&+ \sum_{a=1}^{N}\sum_{b=1}^{N}\sum_{c=1}^{N} \tilde{\alpha}_{ab}^{(c)}[\hat{Q}_a^{(c)}(t) - \hat{Q}_b^{(c)}(t)] \cdot 1_{\hat{Q}_a^{(c)}(t)>0} \\
&= \sum_{a=1}^{N}\sum_{b=1}^{N}\sum_{c=1}^{N} \tilde{\alpha}_{ab}^{(c)}[0 - \hat{Q}_b^{(c)}(t)] \cdot 1_{\hat{Q}_a^{(c)}(t)=0} \\
&+ \sum_{a=1}^{N}\sum_{b=1}^{N}\sum_{c=1}^{N} \tilde{\alpha}_{ab}^{(c)}[\hat{Q}_a^{(c)}(t) - \hat{Q}_b^{(c)}(t)] \cdot 1_{\hat{Q}_a^{(c)}(t)>0} \\
&\le \sum_{a=1}^{N}\sum_{b=1}^{N}\sum_{c=1}^{N} \tilde{\alpha}_{ab}^{(c)}[\hat{Q}_a^{(c)}(t) - \hat{Q}_b^{(c)}(t)] \cdot 1_{\hat{Q}_a^{(c)}(t)>0} \\
&= \sum_{a=1}^{N}\sum_{b=1}^{N}\sum_{c=1}^{N} \tilde{\beta}_{ab}^{(c)}[\hat{Q}_a^{(c)}(t) - \hat{Q}_b^{(c)}(t)] \cdot 1_{\hat{Q}_a^{(c)}(t)>0} \\
&= \sum_{a=1}^{N}\sum_{b=1}^{N}\sum_{c=1}^{N} \tilde{\beta}_{ab}^{(c)}[\hat{Q}_a^{(c)}(t) - \hat{Q}_b^{(c)}(t)]
\end{aligned}$$

Then we must have

$$W(\tilde{\beta}, \hat{Q}(t)) = W(\tilde{\alpha}, \hat{Q}(t)) = \max_{\alpha \in \mathcal{S}} W(\alpha, \hat{Q}(t)).$$

Fix a schedule $\beta \in \mathcal{S}$ with $W(\beta, \hat{Q}(t)) < W(\tilde{\beta}, \hat{Q}(t))$. There exists a constant $\epsilon > 0$ such that

$$W(\tilde{\beta}, \hat{Q}(t)) - W(\beta, \hat{Q}(t))\rangle \ge \epsilon, \hat{Q}_a^{(c)}(t) > \epsilon \text{ for } (a,c) \in \mathcal{A}(\hat{Q}(t))$$

By the continuity of $\hat{Q}(\cdot)$, there exists $\tau > 0$ such that for each $s \in [t-\tau, t+\tau]$

$$\begin{aligned}
&W(\tilde{\beta}, \hat{Q}(t)) - W(\beta, \hat{Q}(t)) \ge \frac{\epsilon}{2} \\
&\hat{Q}_a^{(c)}(s) > \frac{\epsilon}{2} \qquad \text{for } (a,c) \in \mathcal{A}(\hat{Q}(t))
\end{aligned}$$

Let R be the maximal link speed all over the network as defined in Section 4.1. By (22), there exists $K > 0$ such that, for any $k > K$ we have $\frac{\epsilon}{4} r_k > \max(L_{max}, NR)$ and for each $s \in [t-\tau, t+\tau]$

$$\begin{aligned}
&\Bigg| \Big(W(\tilde{\beta}, \hat{Q}^{r_k}(s)) - W(\beta, \hat{Q}^{r_k}(s)) \Big) \\
&\quad - \Big(W(\tilde{\beta}, \hat{Q}(s)) - W(\beta, \hat{Q}(s)) \Big) \Bigg| \le \frac{\epsilon}{4}
\end{aligned}$$

$$\left| \hat{Q}_a^{(c),r_k}(s) - \hat{Q}_a^{(c)}(s) \right| \le \frac{\epsilon}{4}, \text{ for } (a,c) \in \mathcal{A}(\hat{Q}(t))$$

Thus for $k > K$ and each $s \in [t-\tau, t+\tau]$, we have

$$\begin{aligned}
&W(\tilde{\beta}, \hat{Q}^{r_k}(s)) - W(\beta, \hat{Q}^{r_k}(s)) \ge \frac{\epsilon}{4} \\
&\hat{Q}_a^{(c),r_k}(s) \ge \frac{\epsilon}{4} \qquad \text{for } (a,c) \in \mathcal{A}(\hat{Q}(t))
\end{aligned}$$

Therefore, for each time $s \in [(t-\tau)r_k, (t+\tau)r_k]$, we have

$$W(\tilde{\beta}, Q(s,\omega)) > W(\beta, Q(s,\omega)) \tag{23}$$

$$Q_a^{(c)}(s,\omega) \ge \frac{\epsilon}{4} r_k > \max(L_{max}, NR) \quad \text{for } (a,c) \in \mathcal{A}(\hat{Q}(t)) \tag{24}$$

Condition (24) implies that schedule $\tilde{\beta}$ only serves queue that has queue backlog larger than $\max(L_{max}, NR)$ throughout time interval $[(t-\tau)r_k, (t+\tau)r_k]$ and the queues it serves all have sufficient backlog to send. By (9), it must be a valid schedule under the L-BP hybrid backpressure scheme throughout time interval $[(t-\tau)r_k, (t+\tau)r_k]$. By (23), the weight of schedule β is always less than that of $\tilde{\beta}$, and thus should never be employed throughout time interval $[(t-\tau)r_k, (t+\tau)r_k]$. Therefore, for any $u_1 \le u_2$, $u_1, u_2 \in [(t-\tau)r_k, (t+\tau)r_k]$ we have

$$T_\beta(u_2, \omega) - T_\beta(u_1, \omega) = 0$$

Thus, for any $u_1, u_2 \in [(t-\tau), (t+\tau)]$ with $u_1 \le u_2$, we have

$$T_\beta(u_2 r_k, \omega) - T_\beta(u_1 r_k, \omega) = 0$$

i.e.,

$$\hat{T}_\beta^{r_k}(u_2, \omega) - \hat{T}_\beta^{r_k}(u_1, \omega) = 0$$

Taking the limit as $k \to \infty$, we have

$$\hat{T}_\beta(u_2) - \hat{T}_\beta(u_1) = 0$$

for any $u_1, u_2 \in [(t-\tau), (t+\tau)]$ with $u_1 \le u_2$, from which (21) follows, proving the lemma. □

Proof of Theorem 1.

PROOF. The proof is similar to the proof of Theorem 1 in [5]. For completeness, we produce a full proof here.

It's sufficient to show that the fluid model is weakly stable for any arrival rate matrix λ that belongs to $< \mathit{\Gamma}_{\mathcal{S}} >$. Fix an arrival rate matrix λ that belongs to $< \mathit{\Gamma}_{\mathcal{S}} >$. Suppose $(\hat{Q}, \hat{D}, \hat{T})$ is a fluid limit path with $|\hat{Q}(0)| = 0$. Define a Lyapunov function

$$f(t) = \sum_{n=1}^{N}\sum_{c=1}^{N} \left(\hat{Q}_n^{(c)}(t) \right)^2$$

We have $f(0) = 0$, $f(t) \ge 0$ and $f(t) = 0 \Leftrightarrow \hat{Q}(t) = 0$ for all $t > 0$. It's then sufficient to prove that such that $f(t) = 0$ for all $t \ge 0$.

Since λ belongs to $< \mathit{\Gamma}_{\mathcal{S}} >$, there exist constants $p_\beta \in [0,1]$, $\beta \in \mathcal{S}$, such that

$$\begin{aligned}
&\lambda_n^{(c)} = \sum_{\beta \in \mathcal{S}} p_\beta \cdot \gamma_n^{(c)}(\beta) = \sum_{\beta \in \mathcal{S}} p_\beta \left(\sum_{b=1}^{N} \beta_{nb}^{(c)} - \sum_{a=1}^{N} \beta_{an}^{(c)} \right), \\
&\qquad\qquad\qquad\qquad c = 1, \ldots, N \\
&\sum_{\beta \in \mathcal{S}} p_\beta \le 1
\end{aligned} \tag{25}$$

Let $W_{max}(\hat{Q}(t)) = \max_{\beta \in \mathcal{S}} W(\beta, \hat{Q}(t))$. For $t \ge 0$ we have

$$\begin{aligned}
W_{max}(\hat{Q}(t)) &\ge \sum_{\beta \in \mathcal{S}} p_\beta W(\beta, \hat{Q}(t)) \\
&= \sum_{\beta \in \mathcal{S}} p_\beta \sum_{a=1}^{N}\sum_{b=1}^{N}\sum_{c=1}^{N} \beta_{ab}^{(c)} [\hat{Q}_a^{(c)}(t) - \hat{Q}_b^{(c)}(t)] \\
&= \sum_{\beta \in \mathcal{S}} p_\beta \sum_{n=1}^{N}\sum_{c=1}^{N} \hat{Q}_n^{(c)}(t) \left(\sum_{b=1}^{N} \beta_{nb}^{(c)} - \sum_{a=1}^{N} \beta_{an}^{(c)} \right)
\end{aligned}$$

$$= \sum_{n=1}^{N}\sum_{c=1}^{N} \hat{Q}_n^{(c)}(t) \cdot \sum_{\beta\in\mathcal{S}} p_\beta \left(\sum_{b=1}^{N} \beta_{nb}^{(c)} - \sum_{a=1}^{N} \beta_{an}^{(c)} \right)$$

$$= \sum_{n=1}^{N}\sum_{c=1}^{N} \hat{Q}_n^{(c)}(t) \cdot \lambda_n^{(c)} \qquad (26)$$

Let t be a fixed value such that $\hat{Q}(\cdot)$ is differentiable at t. Let $\mathcal{S}'$ be the set of schedules β such that $W(\beta, \hat{Q}(t)) = W_{max}(\hat{Q}(t))$. Then we have $\frac{d}{dt}W(\beta, \hat{Q}(t)) = \frac{d}{dt}W_{max}(\hat{Q}(t))$ for $\beta \in \mathcal{S}'$ (see proof of Lemma 3.2 of [7]). By (17) and Lemma 1, we have

$$\sum_{\beta\in\mathcal{S}'} \frac{d}{dt}\hat{T}_\beta(t) = 1$$

It follows that

$$\begin{aligned}
&\sum_{n=1}^{N}\sum_{c=1}^{N} \hat{Q}_n^{(c)}(t) \left(\sum_{b=1}^{N} \frac{d}{dt}\hat{D}_{nb}^{(c)}(t) - \sum_{a=1}^{N} \frac{d}{dt}\hat{D}_{an}^{(c)}(t) \right) \\
&= \sum_{n=1}^{N}\sum_{c=1}^{N} \hat{Q}_n^{(c)}(t) \Bigg(\sum_{b=1}^{N}\sum_{\beta\in\mathcal{S}'} \beta_{nb}^{(c)} \frac{d}{dt}\hat{T}_\beta(t) \\
&\qquad\qquad - \sum_{a=1}^{N}\sum_{\beta\in\mathcal{S}'} \beta_{an}^{(c)} \frac{d}{dt}\hat{T}_\beta(t) \Bigg) \\
&= \sum_{\beta\in\mathcal{S}'} \frac{d}{dt}\hat{T}_\beta(t) \sum_{n=1}^{N}\sum_{c=1}^{N} \hat{Q}_n^{(c)}(t) \left(\sum_{b=1}^{N} \beta_{nb}^{(c)} - \sum_{a=1}^{N} \beta_{an}^{(c)} \right) \\
&= \sum_{\beta\in\mathcal{S}'} \frac{d}{dt}\hat{T}_\beta(t) \sum_{a=1}^{N}\sum_{b=1}^{N}\sum_{c=1}^{N} \beta_{ab}^{(c)} [\hat{Q}_a^{(c)}(t) - \hat{Q}_b^{(c)}(t)] \\
&= \sum_{\beta\in\mathcal{S}'} \frac{d}{dt}\hat{T}_\beta(t) \cdot W(\beta, \hat{Q}(t)) \\
&= W_{max}(\beta, \hat{Q}(t)) \sum_{\beta\in\mathcal{S}'} \frac{d}{dt}\hat{T}_\beta(t) \\
&= W_{max}(\beta, \hat{Q}(t))
\end{aligned}$$

Thus,

$$\begin{aligned}
&\frac{d}{dt}f(t) = 2\sum_{n=1}^{N}\sum_{c=1}^{N} \hat{Q}_n^{(c)}(t) \cdot \frac{d}{dt}\hat{Q}_n^{(c)}(t) \\
&= 2\sum_{n=1}^{N}\sum_{c=1}^{N} \hat{Q}_n^{(c)}(t) \left(\lambda_n^{(c)} + \sum_{a=1}^{N} \frac{d}{dt}\hat{D}_{an}^{(c)}(t) - \sum_{b=1}^{N} \frac{d}{dt}\hat{D}_{nb}^{(c)}(t) \right) \\
&= 2\sum_{n=1}^{N}\sum_{c=1}^{N} \hat{Q}_n^{(c)}(t)\lambda_n^{(c)} \\
&\quad - 2\sum_{n=1}^{N}\sum_{c=1}^{N} \hat{Q}_n^{(c)}(t) \left(\sum_{b=1}^{N} \frac{d}{dt}\hat{D}_{nb}^{(c)}(t) - \sum_{a=1}^{N} \frac{d}{dt}\hat{D}_{an}^{(c)}(t) \right) \\
&= 2\left(\sum_{n=1}^{N}\sum_{c=1}^{N} \hat{Q}_n^{(c)}(t)\lambda_n^{(c)} - W_{max}(\hat{Q}(t)) \right) \\
&\le 0
\end{aligned}$$

In other words, we have $\frac{d}{dt}f(t) \le 0$ for almost every t such that f is differentiable at t. Since $f(0) = 0$, it follows that $f(t) = 0$ for all $t \ge 0$ (see, for example, the proof of Lemma 1 of [5]), proving the theorem. □

4.5 Throughput Optimality of A-BP

Proof of Theorem 2.

Proof. It's sufficient to prove that each of the fluid limit paths satisfies the fluid equation (21) if the system works under the A-BP hybrid backpressure scheme. The proof is quite similar to that of Lemma 1.

Similarly, suppose $(\hat{Q}, \hat{D}, \hat{T})$ is a fluid limit path. Fix a sample path $\omega \in \Omega$ such that (12) and (13) hold. There exists a sequence $\{r_k\}$ with $r_k \to \infty$ as $k \to \infty$, such that

$$\begin{aligned}
&\left(\hat{Q}^{r_k}(\cdot,\omega), \hat{D}^{r_k}(\cdot,\omega), \hat{T}^{r_k}(\cdot,\omega) \right) \\
&\qquad \to \left(\hat{Q}, \hat{D}, \hat{T} \right) \quad u.o.c \quad as\ k \to \infty \qquad (27)
\end{aligned}$$

Fix a time $t \ge 0$. Define $\mathcal{A}(\hat{Q}(t)) = \{(a,c) \,|\, \hat{Q}_a^{(c)}(t) > 0\}$. Let $\tilde{\alpha} = \mathrm{argmax}_{\alpha\in\mathcal{S}} W(\alpha, \hat{Q}(t))$. Define $\tilde{\beta}$ via

$$\tilde{\beta}_{ab}^{(c)} = \begin{cases} \tilde{\alpha}_{ab}^{(c)} & \text{if } (a,c) \in \mathcal{A}(\hat{Q}(t)) \\ 0 & \text{otherwise} \end{cases}$$

We have $\tilde{\beta} \in \mathcal{S}$ and $W(\tilde{\beta}, \hat{Q}(t)) = \max_{\alpha\in\mathcal{S}} W(\alpha, \hat{Q}(t))$.

Fix a schedule $\beta \in \mathcal{S}$ with $W(\beta, \hat{Q}(t)) < W(\tilde{\beta}, \hat{Q}(t))$. There exists a constant $\epsilon > 0$ such that

$$W(\tilde{\beta}, \hat{Q}(t)) - W(\beta, \hat{Q}(t)))\rangle \ge \epsilon$$

and $\hat{Q}_a^{(c)}(t) > \epsilon$ for $(a,c) \in \mathcal{A}(\hat{Q}(t))$. By the continuity of $\hat{Q}(\cdot)$, there exists $\tau > 0$ such that for each $s \in [t-\tau, t+\tau]$

$$\begin{aligned}
W(\tilde{\beta}, \hat{Q}(t)) - W(\beta, \hat{Q}(t)) &\ge \frac{\epsilon}{2} \\
\hat{Q}_a^{(c)}(s) &> \frac{\epsilon}{2} \qquad \text{for } (a,c) \in \mathcal{A}(\hat{Q}(t))
\end{aligned}$$

Let R be the maximal link speed all over the network as defined in Section 4.1. By (27), there exists $K > 0$ such that, for any $k > K$ we have $\frac{\epsilon}{4}r_k > \max(RA_{max}, RN)$, $\frac{\tau}{2}r_k > A_{max}$ and for each $s \in [t-\tau, t+\tau]$

$$\begin{aligned}
&\Bigg| \left(W(\tilde{\beta}, \hat{Q}^{r_k}(s)) - W(\beta, \hat{Q}^{r_k}(s)) \right) \\
&\qquad - \left(W(\tilde{\beta}, \hat{Q}(s)) - W(\beta, \hat{Q}(s)) \right) \Bigg| \le \frac{\epsilon}{4}
\end{aligned}$$

$$\left| \hat{Q}_a^{(c),r_k}(s) - \hat{Q}_a^{(c)}(s) \right| \le \frac{\epsilon}{4} \qquad \text{for } (a,c) \in \mathcal{A}(\hat{Q}(t))$$

Thus for $k > K$ and each $s \in [t-\tau, t+\tau]$, we have

$$\begin{aligned}
W(\tilde{\beta}, \hat{Q}^{r_k}(s)) - W(\beta, \hat{Q}^{r_k}(s)) &\ge \frac{\epsilon}{4} \\
\hat{Q}_a^{(c),r_k}(s) &\ge \frac{\epsilon}{4} \qquad \text{for } (a,c) \in \mathcal{A}(\hat{Q}(t))
\end{aligned}$$

Therefore, for each time $s \in [(t-\tau)r_k, (t+\tau)r_k]$, we have

$$W(\tilde{\beta}, Q(s)) > W(\beta, Q(s)) \qquad (28)$$

$$Q_a^{(c)}(s) \ge \frac{\epsilon}{4}r_k > \max(RA_{max}, RN), \text{ for } (a,c) \in \mathcal{A}(\hat{Q}(t)) \qquad (29)$$

Condition (29) implies that for $(a,c) \in \mathcal{A}(\hat{Q}(t))$, the length of queue $Q_a^{(c)}$ is always larger than RA_{max}. Thus the delay of the head-of-line packet in $Q_a^{(c)}$ is always larger than A_{max} throughout time interval $[(t-\tau)r_k + A_{max}, (t+\tau)r_k]$. Since

$\frac{\tau}{2}r_k > A_{max}$, we have $[(t-\frac{\tau}{2})r_k, (t+\frac{\tau}{2})r_k] \subset [(t-\tau)r_k + A_{max}, (t+\tau)r_k]$. In other words, schedule $\tilde{\beta}$ only serves queue with delay of head-of-line packets larger than A_{max} throughout time interval $[(t-\frac{\tau}{2})r_k, (t+\frac{\tau}{2})r_k]$ and the queues it serves all have sufficient backlog to send. By (10), it must be a valid schedule under the A-BP hybrid backpressure scheme throughout time interval $[(t-\frac{\tau}{2})r_k, (t+\frac{\tau}{2})r_k]$. By (28), the weight of schedule β is always less than that of $\tilde{\beta}$, and thus should never be employed throughout time interval $[(t-\frac{\tau}{2})r_k, (t+\frac{\tau}{2})r_k]$. Therefore, for any $u_1 \leq u_2$, $u_1, u_2 \in [(t-\frac{\tau}{2})r_k, (t+\frac{\tau}{2})r_k]$, we have

$$T_\beta(u_2, \omega) - T_\beta(u_1, \omega) = 0$$

Therefore, for any $u_1, u_2 \in [(t-\frac{\tau}{2}), (t+\frac{\tau}{2})]$ with $u_1 \leq u_2$, we have

$$T_\beta(u_2 r_k, \omega) - T_\beta(u_1 r_k, \omega) = 0$$

i.e.,

$$\hat{T}_\beta^{r_k}(u_2, \omega) - \hat{T}_\beta^{r_k}(u_1, \omega) = 0$$

Taking the limit as $k \to \infty$, we have

$$\hat{T}_\beta(u_2) - \hat{T}_\beta(u_1) = 0$$

for any $u_1, u_2 \in [(t-\frac{\tau}{2}), (t+\frac{\tau}{2})]$ with $u_1 \leq u_2$, from which (21) follows, proving the theorem. □

5. EVALUATIONS

In this section, we present evaluations of our two proposed hybrid backpressure algorithms. These algorithms can be applied to both wireline and wireless networks. To evaluate these algorithms, we focus on the adaptive routing problem for the wireline case. In particular, we present our evaluations using a real, large PoP-level backbone network, namely the Abilene[9] network. The Abilene network has been studied and discussed in the research literature. Its network topology, traffic dataset, and routing information are available in the public domain [20]. In the following, we first describe our experimental setup and then present our simulation results.

5.1 Experiment setup

The Abilene network is a public academic network in the U.S. with 12 nodes interconnected by OC192, 9.92 Gbits/s links. We use the traffic matrices obtained in [20] in the experiments. Each traffic matrix consists of the demand rate of every source destination pair within five minutes. Therefore, these traffic matrices provide a snapshot of real total demand offerings between each source-destination pair in the Abilene network every five minutes. The actual dataset spans from March 1, 2004 to September 4, 2004. As the traffic matrices indicate, the Abilene network is underutilized. To demonstrate that our hybrid backpressure algorithm improves delay performance while retaining optimal throughput, we selected the traffic matrix with the highest traffic load, and scaled it by different factors. We incrementally increased the scaling factor until the resulting arrival rates exceed the network's stability region. Then we normalized that largest scaling factor.

We implemented our simulator in C++. Traffic generation follows a Bernoulli arrival process with probability

$$p = \frac{\text{traffic demand}}{\text{link capacity}}.$$

We assume $\mu_{ab}(t) = 1$. That is, at each timeslot, at most one packet may be transmitted over each link. The end-to-end delay is measured by the time period from the timeslot when a packet enters the network by the traffic generation function to the timeslot when the packet arrives at its destination and thus leaves the network. The packets generated are put into internal commodity queues directly based on their destinations. To get reliable results, the simulation time should be long enough so that the network is in its stable status. We simulated for 40 million timeslots for each scaling factor. For the results presented in this section, $L_{max} = 10$ was used for the L-BP algorithm, and $A_{max} = 2$ was used for the A-BP algorithm.

5.2 Experiment results

5.2.1 *Delay performance*

In this section, we present and compare the end-to-end delay performance in the simulation results for original backpressure algorithm(BP), Open Shortest Path First (OSPF), Equal-cost multi-path routing (ECMP), Shortest-path backpressure algorithm(SPBP), L-BP hybrid backpressure (L-BP), and A-BP hybrid backpressure (A-BP) in Figure 2. From these results, we can observe the following:

- **Original BP**: We observe in Figure 2a that under Original BP, the delay first decreases and then increases with increasing traffic loads. This phenomenon validates that original BP incurs large delays under light or moderate traffic loads, because packets explore unnecessary long paths.

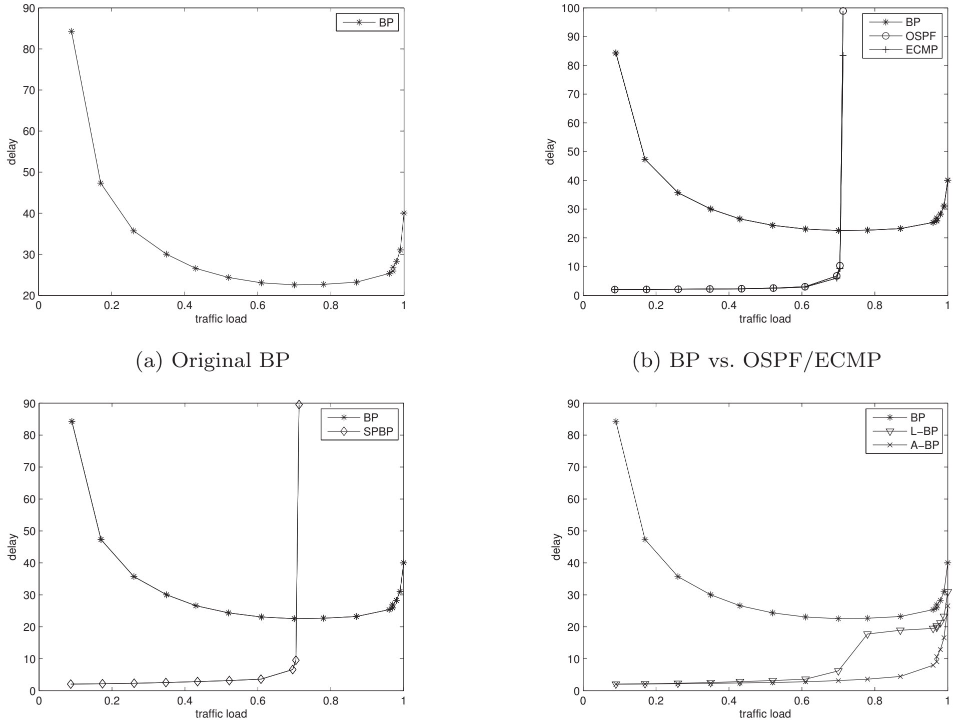
- **OSPF and ECMP**: OSPF and ECMP are popular routing algorithms in the industry. From Figure 2b, we observe that OSPF and ECMP have low average end-to-end delay compared with Original BP. However, they both saturate the network early. That means, OSPF and ECMP shrinks the network stability region to only about 75% of that under original BP.

- **SPBP**: Figure 2c illustrates that although SPBP reduces delay, it is not throughput optimal because it saturates the network early compared with the original backpressure algorithm. This is to be expected since reducing routing choices shrinks the network stability region.

- **L-BP and A-BP**: Figure 2d shows that both the L-BP and A-BP hybrid backpressure algorithms achieve the same optimal throughput as the original backpressure algorithm. Besides, they offer much lower delay under light or moderate traffic loads. We can further observe that A-BP has even better delay performance than L-BP.

5.2.2 *Backlog information exchanges*

In this section, we examine two metrics: the *backlog information exchange frequency* and the *percentage of queues* in the network that are in Phase II. A *backlog information exchange* is recognized when an internal per-destination queue needs its neighbor's corresponding per-destination queue backlog information to compute the backpressure. For example, according to the original backpressure algorithm, if a node has three neighbors, then each internal per-destination queue has to know the backlog information from its three



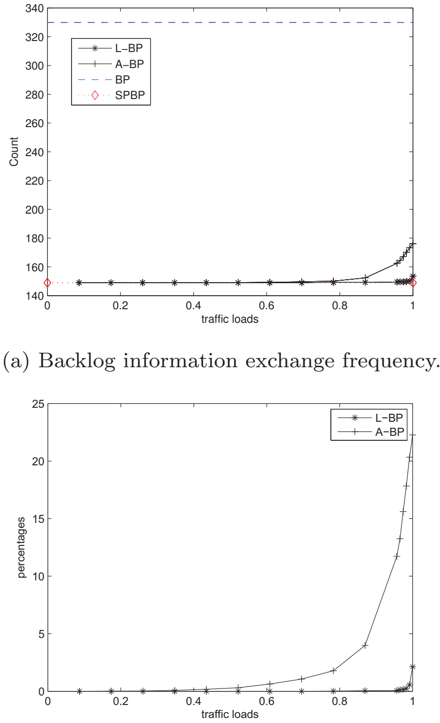
(a) Original BP

(b) BP vs. OSPF/ECMP

(c) BP vs. SPBP

(d) BP vs. L-BP/A-BP

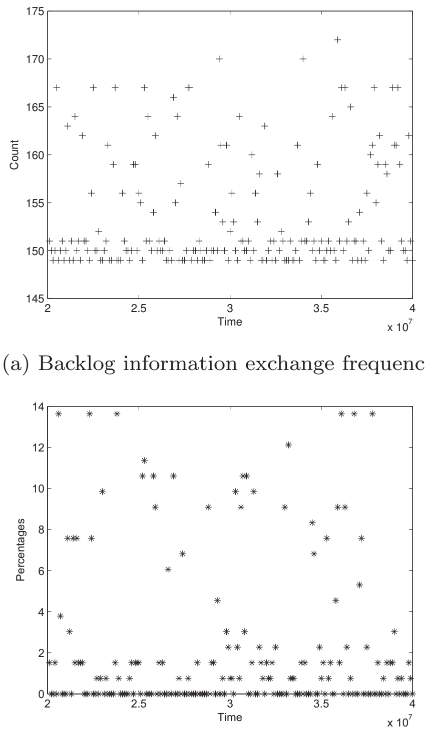
Figure 2: Delay comparison under different traffic loads.



(a) Backlog information exchange frequency.

(b) Percentages of Phase II queues.

Figure 3: Backlog information exchange frequency and percentages of Phase II queues under all traffic loads.



(a) Backlog information exchange frequency.

(b) Percentages of Phase II queues.

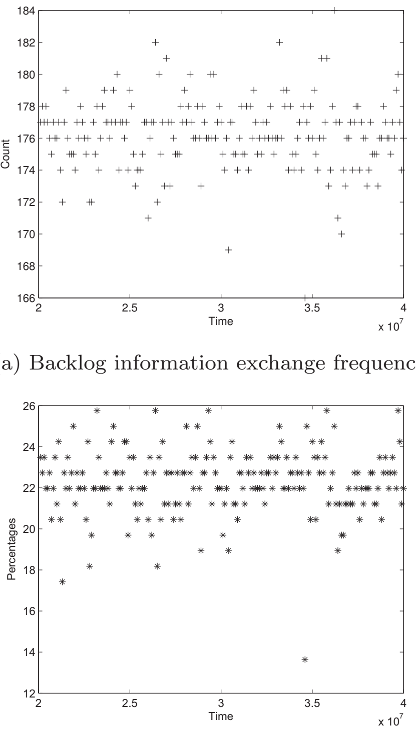
Figure 4: Backlog information exchange frequency and percentages of Phase II queues for L-BP under the maximum traffic load.

neighbors, and the corresponding differential backlogs need to be computed for these three neighbors. However, in our proposed algorithms, if a queue is in Phase I, then the backlog information only needs to be exchanged with a subset of neighbor nodes that are a part of some shortest path routes, and the differential backlog calculations only need to be computed with respect to these nodes.

It should be noted that, among BP, SPBP, L-BP, and A-BP algorithms, BP has the maximum number of backlog information exchanges, which is the upper bound. SPBP has the least and thus is the lower bound. For the L-BP/A-BP algorithms, whether their backlog information exchange frequencies are closer to BP or SPBP depends on how many queues in L-BP/A-BP are in Phase II. If no queue in L-BP/A-BP is in Phase II, then L-BP/A-BP is the same as SPBP. If all queues in L-BP/A-BP are in Phase II, then L-BP/A-BP is the same as BP.

Compared with SPBP, which is **not** throughput optimal, L-BP/A-BP adaptively adds the number of queues that switch over to Phase II from Phase I as needed to achieve the optimal throughput. The more queues are in Phase II, the more bandwidth is taken by exchanging the backlog information, and the more problems need to be considered for the exchange (e.g., the backlog information is not up-to-date). To some extend, compared with SPBP, the hybrid backpressure algorithms sacrifice the frequency of backlog information exchanges for the optimal throughput. In the following experiments, we would like to explore how much the *cost* is.

Figure 3 shows that the cost for the optimal throughput is very small when our proposed hybrid backpressure algorithms are used. Figure 3a shows the backlog information exchange frequency for BP, SPBP, L-BP, and A-BP algorithms, under different traffic loads. As expected, BP has the maximum number of exchanges, which is a constant at 330. On the other hand, SPBP has the least number of exchanges, which is also a constant at 149. What is surprising is that the L-BP/A-BP algorithms only have slightly more backlog information exchanges compared with SPBP, even at very high traffic loads. This means that, by adding a bit more backlog information exchanges, L-BP/A-BP algorithms can achieve the optimal throughput, which is a much higher throughput than SPBP. In Figure 3b we can see an increasing number of queues are switched over to Phase II when the traffic load increases. However, the change is negligible for L-BP. For A-BP, even under very high traffic loads, the percentage is still below 23%. All of this translates to much lower computational requirements for calculating the necessary different backlogs. We simulate the network for 40 million timeslots for each traffic load, though the network becomes stable enough to measure the performance after 20 million timeslots. From timeslots 20 million to 40 million, we sample for every 100,000 timeslots, and thus we have 200 data points for each traffic load. We finally average over these 200 data points to obtain Figure 3.



(a) Backlog information exchange frequency.

(b) Percentages of Phase II queues.

Figure 5: Backlog information exchange frequency and percentages of Phase II queues for A-BP under maximum traffic load.

Figure 4 and Figure 5 take the maximum traffic load as an example, and show the number of backlog information exchanges and the percentages of Phase II queues from timeslot 20 million to 40 million for L-BP and A-BP, respectively. In Figure 4a, most data points for the backlog information exchanges are between 149 and 151. Recall that the lower bound from SPBP for the number of the backlog information exchange frequency is 149. So Figure 4a shows that most of the time the backlog information exchange frequency of L-BP is comparable to the SPBP algorithm. As can be seen from Figure 4b, most of the time the majority of queues are in Phase I.

In Figure 5, most of the 200 data points are between 176 and 178. Recall that the upper bound from BP for the number of the backlog information exchange is 330. Figure 5a indicates that most of the time the addition of the backlog information exchange is limited, and Figure 5b shows that most of the time only less than 24% queues switch over to Phase II. We also observe by comparing Figure 4b and Figure 5b that more queues switch over to Phase II for A-BP. The reason is that we use a smaller number as A_{max} for A-BP. Therefore, the queues in A-BP can more easily satisfy the transition condition.

6. CONCLUSION

In this paper, we proposed two new hybrid backpressure-based adaptive routing algorithms, called L-BP and A-BP, that are based on the incremental expansion of routing choices in response to congestion at a node on a per-destination basis. Both variants are shown to be throughput optimal by means of a fluid model analysis. Simulation results using actual traffic profiles on a public network demonstrate that our proposed algorithms indeed provide substantial improvements in delay performance. The simulation results further show that in practice, our approach dramatically reduces the number of pairwise differential backlogs that have to be computed and the amount of backlog information that has to be exchanged because routing choices are only incrementally expanded as needed.

7. ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation under CNS-1248117, CNS-1422286, and CNS-1423182.

8. REFERENCES

[1] M. Alresaini, K. L. Wright, B. Krishnamachari, and M. J. Neely. Backpressure delay enhancement for encounter-based mobile networks while sustaining throughput optimality. *IEEE/ACM Transactions on Networking*, 24(2):1196–1208, April 2016.

[2] E. Athanasopoulou, L. X. Bui, T. Ji, R. Srikant, and A. Stolyar. Back-pressure-based packet-by-packet adaptive routing in communication networks. *IEEE/ACM Transactions on Networking (TON)*, 21(1):244–257, 2013.

[3] B. Awerbuch and T. Leighton. A simple local-control approximation algorithm for multicommodity flow. In *Proceedings of 1993 IEEE 34th Annual Foundations of Computer Science*, pages 459–468, Nov 1993.

[4] L. Bui, R. Srikant, and A. Stolyar. Novel architectures and algorithms for delay reduction in back-pressure scheduling and routing. In *INFOCOM 2009, IEEE*, pages 2936–2940. IEEE, 2009.

[5] J. Dai and B. Prabhakar. The throughput of data switches with and without speedup. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 556–564. IEEE, 2000.

[6] J. G. Dai and W. Lin. Maximum pressure policies in stochastic processing networks. *Operations Research*, 53(2):197–218, 2005.

[7] J. G. Dai and G. Weiss. Stability and instability of fluid models for reentrant lines. *Mathematics of Operations Research*, 21(1):115–134, 1996.

[8] A. Eryilmaz and R. Srikant. Joint congestion control, routing, and mac for stability and fairness in wireless networks. *IEEE Journal on Selected Areas in Communications*, 24(8):1514–1524, 2006.

[9] Internet2. Advanced networking for leading-edge research and education. http://www.internet2.edu/.

[10] S. Iyer, R. R. Kompella, and N. McKeown. Designing packet buffers for router linecards. *IEEE/ACM Transactions on Networking (TON)*, 16(3):705–717, 2008.

[11] M. J. Neely, E. Modiano, , and C. Li. Fairness and optimal stochastic control for heterogeneous networks. In *Proceedings of IEEE INFOCOM*. IEEE Press, 2005.

[12] M. J. Neely, E. Modiano, and C. E. Rohrs. Dynamic power allocation and routing for time-varying wireless networks. *IEEE Journal on Selected Areas in Communications*, 23(1):89–103, 2005.

[13] J. Ryu, L. Ying, and S. Shakkottai. Back-pressure routing for intermittently connected networks. In *Proceedings of the 29th Conference on Information Communications*, INFOCOM'10, pages 306–310, Piscataway, NJ, USA, 2010. IEEE Press.

[14] H. Seferoglu and E. Modiano. Separation of routing and scheduling in backpressure-based wireless networks. *IEEE/ACM Transactions on Networking*, 24(3):1787–1800, June 2016.

[15] G. Shrimali and N. McKeown. Building packet buffers using interleaved memories. In *Proceedings of the Workshop on High Performance Switching and Routing (HPSR)*. IEEE Press, May 2005.

[16] 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*, 37(12):1936–1948, Dec 1992.

[17] H. Wang and B. Lin. Block-based packet buffer with deterministic packet departures. In *Proceedings of the 11th International Conference on High Performance Switching (HPSR)*. IEEE Press, June 2010.

[18] H. Wang, H. Zhao, B. Lin, and J. Xu. Robust pipelined memory system with worst case performance guarantee for network processing. *IEEE/ACM Transactions on Computers (TC)*, 61(10):1386–1400, 2012.

[19] L. Ying, S. Shakkottai, A. Reddy, and S. Liu. On combining shortest-path and back-pressure routing over multihop wireless networks. *IEEE/ACM Transactions on Networking (TON)*, 19(3):841–854, 2011.

[20] Y. Zhang. Abilene traffic matrices. http://www.cs.utexas.edu/~yzhang/research/AbileneTM/.