IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 26, NO. 4, PP. 1605-1618, JUNE 2018

A New Backpressure Algorithm for Joint Rate Control and Routing with Vanishing Utility Optimality Gaps and Finite Queue Lengths

Hao Yu and Michael J. Neely

***Abstract*—The backpressure algorithm has been widely used as a distributed solution to the problem of joint rate control and routing in multi-hop data networks. By controlling an algorithm parameter, the backpressure algorithm can achieve an arbitrarily small utility optimality gap. However, this in turn brings in a large queue length at each node and hence causes large network delay. This phenomenon is known as the fundamental utility-delay tradeoff. The best known utility-delay tradeoff for general networks is $[O(\epsilon), O(1/\epsilon)]$ and is attained by a backpressure algorithm based on a drift-plus-penalty technique. This may suggest that to achieve an arbitrarily small utility optimality gap, backpressure-based algorithms must incur arbitrarily large queue lengths. However, this paper proposes a new backpressure algorithm that has a vanishing utility optimality gap, so utility converges to exact optimality as the algorithm keeps running, while queue lengths are bounded throughout by a finite constant. The technique uses backpressure and drift concepts with a new method for convex programming.**

***Index Terms*—backpressure algorithm; rate control; routing; utility-delay tradeoff**

I. INTRODUCTION

In multi-hop data networks, the problem of joint rate control and routing is to accept data into the network to maximize certain utilities and to make routing decisions at each node such that all accepted data are delivered to intended destinations without overflowing any queue in intermediate nodes. The original backpressure algorithm proposed in the seminal work [2] by Tassiulas and Ephremides addresses this problem by assuming that incoming data are given and are inside the network stability region and develops a routing strategy to deliver all incoming data without overflowing any queue. In the context of [2], there is essentially no utility maximization consideration in the network. The backpressure algorithm is further extended by a drift-plus-penalty technique to deal with both utility maximization and queue stability [3], [4], [5]. Alternative extensions for both utility maximization and queue stabilization are developed in [6], [7], [8], [9]. The above extended backpressure algorithms have different dynamics and/or may yield different utility-delay tradeoff results. However, all of them rely on "backpressure" quantities, which are the differential backlogs between neighboring nodes.

It has been observed in [10], [6], [8], [11] that the drift-plus-penalty and other alternative algorithms can be interpreted as first order Lagrangian dual type methods for constrained optimization. In addition, these backpressure algorithms follow certain fundamental utility-delay tradeoffs. For instance, the primal-dual type backpressure algorithm in [6] achieves an $O(\epsilon)$ utility optimality gap with an $O(1/\epsilon^2)$ queue length. That is, a small utility optimality gap (corresponding to a small ϵ) is available only at the cost of a large queue length. The drift-plus-penalty backpressure algorithm [5], which has the best utility-delay tradeoff among all existing first order Lagrangian dual type methods for general networks, can only achieve an $O(\epsilon)$ utility optimality gap with an $O(1/\epsilon)$ queue length. Under certain restrictive assumptions over the network, a better $[O(\epsilon), O(\log(1/\epsilon))]$ tradeoff is achieved via an exponential Lyapunov function in [12], and an $[O(\epsilon), O(\log^2(1/\epsilon))]$ tradeoff is achieved via a LIFO-backpressure algorithm in [13].

Fundamental lower bounds on utility-delay tradeoffs in [14], [15], [16], [17], [12] show that, for various stochastic network settings, a large queue delay is unavoidable if a small utility optimality gap is demanded. These works consider certain hard problems with stochastic behavior. It leaves open the question of whether or not performance can be improved for networks that fall outside these hard cases. The current paper investigates network flow problems that can be written as (deterministic) convex programs, which are not restricted to the prior lower bounds. We pursue the question of whether or not improved tradeoffs are possible. Can optimal utility be approached with constant queue sizes?

Recently, there have been many attempts in obtaining new variations of backpressure algorithms for deterministic network flow problems by applying Newton's method to the Lagrangian dual function. In the recent work [11], the authors develop a Newton's method for joint rate control and routing. However, the utility-delay tradeoff in [11] is still $[O(\epsilon), O(1/\epsilon^2)]$; and the algorithm requires a centralized projection step although Newton directions can be approximated in a distributed manner. Work [18] considers a network flow control problem where the path of each flow is given (and hence there is no routing part in the problem), and proposes a decentralized Newton based algorithm for rate control. Work [19] considers network routing without an end-to-end utility and only shows the stability of the proposed Newton based backpressure algorithm. All of the above Netwon's method based algorithms rely on distributed approximations for the inverse of Hessians, whose computations still require certain coordinations for the local information updates and propagations and do not scale well with the network size. In

The authors are with the Electrical Engineering department at the University of Southern California, Los Angeles, CA, 90007.

This work was presented in part at IEEE International Conference on Computer Communications (INFOCOM), Atlanta, GA, USA, May, 2017 [1].
This work is supported in part by NSF grant CCF-1718477.

contrast, the first order Lagrangian dual type methods do not need global network topology information. Rather, each node only needs the queue length information of its neighbors.

This paper proposes a new first order Lagrangian dual type backpressure algorithm that is as simple as the existing algorithms in [5], [6], [8] but has a better utility-delay tradeoff. The new backpressue algorithm achieves a vanishing utility optimality gap that decays like $O(1/t)$, where t is the number of iterations. It also guarantees that the queue length at each node is always bounded by a fixed constant of the same order as the optimal Lagrange multiplier of the network optimization problem. This improves on the utility-delay tradeoffs of prior work. In particular, it improves the steady-state $[O(\epsilon), O(1/\epsilon^2)]$ utility-delay tradeoff in [6] and the $[O(\epsilon), O(1/\epsilon)]$ utility-delay tradeoff of the drift-plus-penalty algorithm in [5], both of which yield an unbounded queue length to have a vanishing utility optimality gap. Indeed, the steady-state utility-delay tradeoff of our algorithm is $[0, O(1)]$. The convergence time to reach this limiting performance is also faster than prior work. Our algorithm achieves a zero utility gap and $O(1)$ queue lengths with a fast $O(1/t)$ convergence rate.

The new backpressure algorithm differs from existing first order backpressure algorithms in the following aspects:

1) The "backpressure" quantities in this paper are with respect to newly introduced weights. These are different from queues used in other backpressure algorithms, but can still be locally tracked and updated.
2) The rate control and routing decision rule involves a quadratic term that is similar to a term used in proximal algorithms [20].

Note that the benefit of introducing a quadratic term in network optimization has been observed in [21]. Work [21] developed a distributive rate control algorithm for network utility maximization (NUM) problems with given routing paths that can be reformulated as a special case of the problem treated in this paper. The algorithm of [21] considers a fixed set of predetermined paths for each session and does not scale well when treating all (typically exponentially many) possible paths of a general network. The algorithm proposed in [21] is not a backpressure type and and uses virtual queues rather than actual queues. It is interesting to note that the virtual queues in [21] remain $O(1)$.[1] However, $O(1)$ virtual queue size does not imply the achieved utility has a fast convergence (the convergence rate of [21] remains an open question). In contrast, the algorithm in the current paper uses both a quadratic term and a modified weight to get $O(1)$ actual queue size with a fast $O(1/t)$ convergence rate.

In our conference version [1], the proposed backpressure algorithm has a global algorithm parameter α, which is required to be chosen based on the total number of links and sessions in the network. In this paper, the proposed backpressure algorithm allows each node n to locally determine its algorithm parameter α_n based on the number of local link connections.

The network algorithm developed in this paper is based on our recent work [23] that develops a new convex programming method with an $O(1/t)$ error decay for general constrained convex programs (including problems that are nonsmooth, not strongly convex and have nonlinear constraints). After our conference version [1] and the arXiv preprint [24] of this paper, Wang and Shroff in [25] studied the same joint rate control and routing problem considered in this paper and propose another backpressure algorithm by using the generalized alternating direction method of multipliers (ADMM) from [26], [27], [28]. While that ADMM approach is very different from ours, the obtained backpressure algorithm in [25] is remarkably similar to our algorithm in this paper: It uses identical virtual queues (up to a constant scaling) and solves source rate and link rate subproblems with identical structures. One difference is the two algorithms use different weights for these subproblems. The other main difference is that at each iteration the algorithm in [25] requires to update link rates first and then update source rates based on the new link rate values, while our algorithm can update link rates and source rates in parallel since these subproblems are fully decoupled. This is because the algorithm in [25] is restricted to the sequential update procedure of ADMM, while our algorithm uses the parallel methods developed by us in [23] together with the natural separability of the linear node flow balance constraints.

The algorithm in [25] achieves source rates that satisfy $\mathbf{x}[t] \to \mathbf{x}^*$, although a convergence rate for general utility functions is not given. [2] In contrast, our analysis proves an explicit $O(1/t)$ optimality deviation for the time average network utility. This time average starts at time 0 and provides a performance guarantee for online implementation during the entire network operation. However, a remarkable property of the algorithm in [25] is that when the utility is smooth (i.e., differentiable with Lipschitz continuous gradients) and strongly convex, then the algorithm in [25] can ensure the *final value* of the iterate $\mathbf{x}[t]$ converges exponentially fast to an optimal value $\mathbf{x}^*$. It remains a promising research direction to investigate whether the fast final iteration convergence also holds for our algorithm, and to investigate the effect of the parallel source and link rate update property and different update rules for weight parameters in the subproblems.

II. System Model and Problem Formulation

Consider a slotted data network with normalized time slots $t \in \{0, 1, 2, \ldots\}$. This network is represented by a graph $\mathcal{G} = (\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{L} \subseteq \mathcal{N} \times \mathcal{N}$ is the set of directed links. Let $|\mathcal{N}| = N$ and $|\mathcal{L}| = L$. This network is shared by F end-to-end sessions denoted by a set $\mathcal{F}$. For each end-to-end session $f \in \mathcal{F}$, the source node $\mathrm{Src}(f)$

[1]A technique in the proof of Prop 6.3.2 in [22] suggests that in the case of fixed path routing of [21] and under several other assumptions that include the implementation of a closest-to-source priority rule, there exists a constant that bounds the deviation between virtual and actual queues. The technique in [22] relies on a fixed-path assumption and has bounds that can grow at each stage k of the path.

[2]According to [26], the authors in [25] conclude their algorithm has "$O(1/t)$ non-ergodic convergence" defined as $\|\mathbf{x}[t] - \mathbf{x}[t-1]\|^2 \leq O(1/t)$, or equivalently, $\|\mathbf{x}[t] - \mathbf{x}[t-1]\| \leq O(1/\sqrt{t})$, in [26]. However, this unusual definition of non-ergodic convergence rate does not imply any convergence rate of $\|\mathbf{x}[t] - \mathbf{x}^*\|$ or utility optimality gap $U(\mathbf{x}[t]) - U(\mathbf{x}^*)$ and hence does not provide insight into convergence behavior that network optimization concerns. Thus, it has nothing to do with the $O(1/t)$ convergence to optimal utility we show in the current paper.

and destination node $\text{Dst}(f)$ are given but the routes are not specified. Each session f has a continuous and concave utility function $U_f(x_f)$ that represents the "satisfaction" received by accepting x_f amount of data for session f into the network at each slot. Unlike [6], [11] where $U_f(\cdot)$ is assumed to be differentiable and strongly concave, this paper considers general concave utility functions $U_f(\cdot)$, including those that are neither differentiable nor strongly concave. Formally, each utility function U_f is defined over an interval $\text{dom}(U_f)$, called the *domain* of the function.

Denote the capacity of link l as C_l and assume it is a fixed and positive constant.[3] Define $\mu_l^{(f)}$ as the amount of session f's data routed at link l that is to be determined by our algorithm. Note that in general, the network may be configured such that some session f is forbidden to use link l. For each link l, define $\mathcal{S}_l \subseteq \mathcal{F}$ as the set of sessions that are allowed to use link l. The case of unrestricted routing is treated by defining $\mathcal{S}_l = \mathcal{F}$ for all links l.

Note that if $l = (n, m)$ with $n, m \in \mathcal{N}$, then $\mu_l^{(f)}$ and C_l can also be respectively written as $\mu_{(n,m)}^{(f)}$ and $C_{(n,m)}$. For each node $n \in \mathcal{N}$, denote the sets of its incoming links and outgoing links as $\mathcal{I}(n)$ and $\mathcal{O}(n)$, respectively. Note that $x_f, \forall f \in \mathcal{F}$ and $\mu_l^{(f)}, \forall l \in \mathcal{L}, \forall f \in \mathcal{F}$ are the decision variables of a joint rate control and routing algorithm. If the global network topology information is available, the optimal joint rate control and routing can be formulated as the following multi-commodity network flow problem:

$$\max_{x_f, \mu_l^{(f)}} \sum_{f\in\mathcal{F}} U_f(x_f) \tag{1}$$

$$\text{s.t.} \quad x_f \mathbf{1}_{\{n=\text{Src}(f)\}} + \sum_{l\in\mathcal{I}(n)} \mu_l^{(f)} \leq \sum_{l\in\mathcal{O}(n)} \mu_l^{(f)} \quad \forall f \in \mathcal{F}, \forall n \in \mathcal{N} \setminus \{\text{Dst}(f)\} \tag{2}$$

$$\sum_{f\in\mathcal{F}} \mu_l^{(f)} \leq C_l, \forall l \in \mathcal{L}, \tag{3}$$

$$\mu_l^{(f)} \geq 0, \forall l \in \mathcal{L}, \forall f \in \mathcal{S}_l, \tag{4}$$

$$\mu_l^{(f)} = 0, \forall l \in \mathcal{L}, \forall f \in \mathcal{F} \setminus \mathcal{S}_l, \tag{5}$$

$$x_f \in \text{dom}(U_f), \forall f \in \mathcal{F} \tag{6}$$

where $\mathbf{1}_{\{\cdot\}}$ is an indicator function; (2) represents the node flow conservation constraints relaxed by replacing the equality with an inequality, meaning that the total rate of flow f into node n is less than or equal to the total rate of flow f out of the node (since, in principle, we can always send fake data for departure links when the inequality is loose); and (3) represents link capacity constraints. Note that for each flow f, there is no constraint (2) at its destination node $\text{Dst}(f)$ since all incoming data are consumed by this node.

The above formulation includes network utility maximization with fixed paths as special cases. In the case when each session only has one single given path, e.g., the network utility maximization problem considered in [30], we could modify the sets $\mathcal{S}_l$ used in constraints (4) and (5) to reflect this fact. For example, if link l_1 is only used for sessions f_1 and f_2, then $\mathcal{S}_{l_1} = \{f_1, f_2\}$. Similarly, the case [21] where each session is restricted to using links from a set of predefined paths can be treated by modifying the sets $\mathcal{S}_l$ accordingly. See Appendix A for more discussions.

The solution to problem (1)-(6) corresponds to the optimal joint rate control and routing. However, to solve this convex program at a single computer, we need to know the global network topology and the solution is a centralized one, which is not practical for large data networks. As observed in [10], [6], [8], [11], various versions of backpressure algorithms can be interpreted as distributed solutions to problem (1)-(6) from first order Lagrangian dual type methods.

Assumption 1: (Feasibility) Problem (1)-(6) has at least one optimal solution vector $[x_f^*; \mu_l^{(f),*}]_{f\in\mathcal{F}, l\in\mathcal{L}}$.

Assumption 2: (Existence of Lagrange multipliers) Assume the convex program (1)-(6) has Lagrange multipliers attaining the strong duality. Specifically, define convex set $\mathcal{C} = \{[x_f; \mu_l^{(f)}]_{f\in\mathcal{F}, l\in\mathcal{L}} : \text{(3)-(6) hold}\}$. Assume there exists a Lagrange multiplier vector $\boldsymbol{\lambda}^* = [\lambda_n^{(f),*}]_{f\in\mathcal{F}, n\in\mathcal{N}\setminus\{\text{Dst}(f)\}} \geq \mathbf{0}$ such that

$$q(\boldsymbol{\lambda}^*) = \max\{(1) : (2)\text{-}(6)\}$$

where $q(\boldsymbol{\lambda}) = \max_{[x_f;\mu_l^{(f)}]\in\mathcal{C}} \big\{ \sum_{f\in\mathcal{F}} U_f(x_f) - \sum_{f\in\mathcal{F}} \sum_{n\in\mathcal{N}\setminus\{\text{Dst}(f)\}} \lambda_n^{(f)} \big[x_f \mathbf{1}_{\{n=\text{Src}(f)\}} + \sum_{l\in\mathcal{I}(n)} \mu_l^{(f)} - \sum_{l\in\mathcal{O}(n)} \mu_l^{(f)} \big] \big\}$ is the Lagrangian dual function of problem (1)-(6) by treating (3)-(6) as a convex set constraint.

Assumptions 1 and 2 hold in most cases of interest. For example, Slater's condition guarantees Assumption 2. Since the constraints (2)-(6) are linear, Proposition 6.4.2 in [31] ensures that Lagrange multipliers exist whenever constraints (2)-(6) are feasible and when the utility functions U_f are either defined over open sets (such as $U_f(x) = \log(x)$ with $\text{dom}(U_f) = (0, \infty)$) or can be *concavely extended* to open sets, meaning that there is an $\epsilon > 0$ and a concave function $\widetilde{U}_f : (-\epsilon, \infty) \to \mathbb{R}$ such that $\widetilde{U}_f(x) = U_f(x)$ whenever $x \geq 0$.[4]

Fact 1: (Replacing inequality with equality) If Assumption 1 holds, problem (1)-(6) has an optimal solution vector $[x_f^*; \mu_l^{(f),*}]_{f\in\mathcal{F}, l\in\mathcal{L}}$ such that all constraints (2) take equalities.

Proof: Note that each $\mu_l^{(f)}$ can appear on the left side in at most one constraint (2) and appear on the right side in at most one constraint (2). Let $[x_f^*, \mu_l^{(f),*}]_{f\in\mathcal{F}, l\in\mathcal{L}}$ be an optimal solution vector such that at least one inequality constraint (2) is loose. Note that we can reduce the value of $\mu_l^{(f),*}$ on the right side of a loose (2) until either that constraint holds with equality, or until $\mu_l^{(f),*}$ reduces to 0. The objective function value does not change, and no constraints are violated. We can repeat the process until all inequality constraints (2) are tight. ■

[3] As stated in [11], this is a suitable model for wireline networks and wireless networks with fixed transmission power and orthogonal channels.

[4] If $\text{dom}(U_f) = [0, \infty)$, such concave extension is possible if the right-derivative of U_f at $x = 0$ is finite (such as for $U_f(x) = \log(1 + x)$ or $U_f(x) = \min[x, 3]$). Such an extension is impossible for the example $U_f(x) = \sqrt{x}$ because the slope is infinite at $x = 0$. Nevertheless, Lagrange multipliers often exist even for these utility functions, such as when Slater's condition holds [31].

III. The New Backpresure Algorithm

A. Discussion of Various Queueing Models

At each node, an independent queue backlog is maintained for each session. At each slot t, let $x_f[t]$ be the source session rates; and let $\mu_l^{(f)}[t]$ be the link session rates. Some prior work enforces the constraint (2) via virtual queues $Y_n^{(f)}[t]$ of the following form:

$$Y_n^{(f)}[t+1] = \max\Big\{Y_n^{(f)}[t] + x_f[t]\mathbf{1}_{\{n=\text{Src}(f)\}} + \sum_{l\in\mathcal{I}(n)} \mu_l^{(f)}[t] - \sum_{l\in\mathcal{O}(n)} \mu_l^{(f)}[t], 0\Big\}. \quad (7)$$

While this virtual equation is a meaningful approximation, it differs from reality in that new injected data are allowed to be transmitted immediately, or equivalently, a single packet is allowed to enter and leave many nodes within the same slot. Further, there is no clear connection between the virtual queues $Y_n^{(f)}[t]$ in (7) and the actual queues in the network. Indeed, it is easy to construct examples that show there can be an arbitrarily large difference between the $Y_n^{(f)}[t]$ value in (7) and the physical queue size in actual networks (see Appendix B).

An actual queueing network has queues $Z_n^{(f)}[t]$ with the following dynamics:

$$Z_n^{(f)}[t+1] \le \max\Big\{Z_n^{(f)}[t] - \sum_{l\in\mathcal{O}(n)} \mu_l^{(f)}[t], 0\Big\} + x_f[t]\mathbf{1}_{\{n=\text{Src}(f)\}} + \sum_{l\in\mathcal{I}(n)} \mu_l^{(f)}[t]. \quad (8)$$

This is faithful to actual queue dynamics and does not allow data to be retransmitted over multiple hops in one slot. Note that (8) is an inequality because the new arrivals from other nodes may be strictly less than $\sum_{l\in\mathcal{I}(n)}\mu_l^{(f)}[t]$ because those other nodes may not have enough backlog to send. The model (8) allows for any decisions to be made to fill the transmission values $\mu_l^{(f)}[t]$ in the case that $Z_n^{(f)}[t] \le \sum_{l\in\mathcal{O}(n)}\mu_l^{(f)}[t]$, provided that (8) holds.

This paper develops an algorithm that converges to the optimal utility defined by problem (1)-(6), and that produces worst-case bounded queues on the actual queueing network, that is, with actual queues that evolve as given in (8). To begin, it is convenient to introduce the following virtual queue equation

$$Q_n^{(f)}[t+1] = Q_n^{(f)}[t] - \sum_{l\in\mathcal{O}(n)} \mu_l^{(f)}[t] + x_f[t]\mathbf{1}_{\{n=\text{Src}(f)\}} + \sum_{l\in\mathcal{I}(n)} \mu_l^{(f)}[t] \quad (9)$$

where $Q_n^{(f)}[t]$ represents a **virtual queue** value associated with session f at node n. At first glance, this model (9) appears to be only an approximation, perhaps even a worse approximation than (7), because it allows the $Q_n^{(f)}[t]$ values to be negative. Indeed, we use $Q_n^{(f)}[t]$ only as virtual queues to inform the algorithm and do not treat them as actual queues. However, this paper shows that using these virtual queues to choose the $\boldsymbol{\mu}[t]$ decisions ensures not only that the desired constraints (2) are satisfied, but that the resulting $\boldsymbol{\mu}[t]$ decisions create bounded queues $Z_n^{(f)}[t]$ in the **actual network**, where the actual queues evolve according to (8). In short, our algorithm can be faithfully implemented with respect to actual queueing networks, and converges to exact optimality on those networks.

The next lemma shows that if an algorithm can guarantee virtual queues $Q_n^{(f)}[t]$ defined in (9) are bounded, then actual physical queues satisfying (8) are also bounded.

Lemma 1: Consider a network flow problem described by problem (1)-(6). For all $l \in \mathcal{L}$ and $f \in \mathcal{F}$, let $\mu_l^{(f)}[t], x_f[t]$ be decisions yielded by a dynamic algorithm. Suppose $Y_n^{(f)}[t]$, $Z_n^{(f)}[t]$, $Q_n^{(f)}[t]$ evolve by (7)-(9) with initial conditions $Y_n^{(f)}[0] = Z_n^{(f)}[0] = Q_n^{(f)}[0] = 0$. If there exists a constant $B > 0$ such that $|Q_n^{(f)}[t]| \le B, \forall t$, then we also have

1) $Z_n^{(f)}[t] \le 2B + \sum_{l\in\mathcal{O}(n)} C_l$ for all $t \in \{0,1,2,\ldots\}$.
2) $Y_n^{(f)}[t] \le 2B + \sum_{l\in\mathcal{O}(n)} C_l$ for all $t \in \{0,1,2,\ldots\}$.

Proof:

1) Fix $f \in \mathcal{F}, n \in \mathcal{N} \setminus \{\text{Dst}(f)\}$. Define an auxiliary virtual queue $\widehat{Q}_n^{(f)}[t]$ that is initialized by $\widehat{Q}_n^{(f)}[0] = B + \sum_{l\in\mathcal{O}(n)} C_l$ and evolves according to (9). It follows that $\widehat{Q}_n^{(f)}[t] = Q_n^{(f)}[t] + B + \sum_{l\in\mathcal{O}(n)} C_l, \forall t$. Since $Q_n^{(f)}[t] \ge -B, \forall t$ by assumption, we have $\widehat{Q}_n^{(f)}[t] \ge \sum_{l\in\mathcal{O}(n)} C_l \ge \sum_{l\in\mathcal{O}(n)} \mu_l^{(f)}[t], \forall t$. This implies that $\widehat{Q}_n^{(f)}[t]$ also satisfies:
$$\widehat{Q}_n^{(f)}[t+1] = \max\Big\{\widehat{Q}_n^{(f)}[t] - \sum_{l\in\mathcal{O}(n)} \mu_l^{(f)}[t], 0\Big\} + x_f[t]\mathbf{1}_{\{n=\text{Src}(f)\}} + \sum_{l\in\mathcal{I}(n)} \mu_l^{(f)}[t], \forall t \quad (10)$$
which is identical to (8) except the inequality is replaced by an equality. Since $Z_n^{(f)}[0] = 0 < \widehat{Q}_n^{(f)}[0]$; and $\widehat{Q}_n^{(f)}[t]$ satisfies (10), by inductions, $Z_n^{(f)}[t] \le \widehat{Q}_n^{(f)}[t], \forall t$.
Since $\widehat{Q}_n^{(f)}[t] = Q_n^{(f)}[t] + B + \sum_{l\in\mathcal{O}(n)} C_l, \forall t$ and $Q_n^{(f)}[t] \le B, \forall t$, we have $\widehat{Q}_n^{(f)}[t] \le 2B + \sum_{l\in\mathcal{O}(n)} C_l, \forall t$. It follows that $Z_n^{(f)}[t] \le 2B + \sum_{l\in\mathcal{O}(n)} C_l, \forall t$.
2) The proof of part (2) is similar and is in Appendix C.

■

B. The New Backpressure Algorithm

In this subsection, we propose a new backpressure algorithm that yields source session rates $x_f[t]$ and link session rates $\mu_l^{(f)}[t]$ at each slot such that the physical queues for each session at each node are bounded by a constant and the time average utility satisfies

$$\frac{1}{t}\sum_{\tau=0}^{t-1}\sum_{f\in\mathcal{F}} U_f(x_f[t]) \ge \sum_{f\in\mathcal{F}} U_f(x_f^\ast) - O(1/t), \forall t \ge 1$$

where $x_f^\ast$ are from the optimal solution to (1)-(6). Note that Jensen's inequality further implies that

$$\sum_{f\in\mathcal{F}} U_f\Big(\frac{1}{t}\sum_{\tau=0}^{t-1} x_f[\tau]\Big) \ge \sum_{f\in\mathcal{F}} U_f(x_f^\ast) - O(1/t), \forall t \ge 1$$

The new backpressure algorithm is described in Algorithm 1. Similar to existing backpressure algorithms, the updates in Algorithm 1 at each node n are fully distributed and only depend on weights at itself and its neighbor nodes. Unlike existing backpressure algorithms, the weights used to update decision variables $x_f[t]$ and $\mu_l^{(f)}[t]$ are not the virtual queues $Q_n^{(f)}[t]$ themselves, rather, they are augmented values $W_n^{(f)}[t]$ equal to the sum of the virtual queues and the amount of net injected data in the previous slot $t-1$. In addition, the updates involve an additional quadratic term, which is similar to a term used in proximal algorithms [20]. Note that the source rate update (in the second bullet) and the link rate update (in the third bullet) are fully decoupled and hence can be performed in parallel or in a swapped order.

Algorithm 1 The New Backpressure Algorithm

Let $\alpha_n > 0, \forall n \in \mathcal{N}$ be constant parameters. Initialize $x_f[-1] = 0$, $\mu_l^{(f)}[-1] = 0, \forall f \in \mathcal{F}, \forall l \in \mathcal{L}$ and $Q_n^{(f)}[0] = 0, \forall n \in \mathcal{N}, \forall f \in \mathcal{F}$. At each time $t \in \{0, 1, 2, \ldots\}$, each node n does the following:

- For each $f \in \mathcal{F}$, if node n is not the destination node of session f, i.e., $n \neq \text{Dst}(f)$, then define weight $W_n^{(f)}[t]$:
$$\begin{aligned} W_n^{(f)}[t] =& Q_n^{(f)}[t] + x_f[t-1]\mathbf{1}_{\{n=\text{Src}(f)\}} \\ &+ \sum_{l\in\mathcal{I}(n)} \mu_l^{(f)}[t-1] - \sum_{l\in\mathcal{O}(n)} \mu_l^{(f)}[t-1]. \end{aligned} \quad (11)$$
If node n is the destination node, i.e., $n = \text{Dst}(f)$, then define $W_n^{(f)}[t] = 0$. Notify neighbor nodes (nodes k that can send session f data to node n, i.e., $\forall k$ such that $f \in \mathcal{S}_{(k,n)}$) about this new $W_n^{(f)}[t]$ value.
- For each $f \in \mathcal{F}$, if node n is the source node of session f, i.e., $n = \text{Src}(f)$, choose $x_f[t]$ as the solution to
$$\max_{x_f} \quad U_f(x_f) - W_n^{(f)}[t]x_f - \alpha_n\big(x_f - x_f[t-1]\big)^2 \quad (12)$$
$$\text{s.t.} \quad x_f \in \text{dom}(U_f) \quad (13)$$
- For all $(n,m) \in \mathcal{O}(n)$, choose $\{\mu_{(n,m)}^{(f)}[t], \forall f \in \mathcal{F}\}$ as the solution to the following convex program:
$$\begin{aligned} \max_{\mu_{(n,m)}^{(f)}} \quad & \sum_{f\in\mathcal{F}} \big(W_n^{(f)}[t] - W_m^{(f)}[t]\big)\mu_{(n,m)}^{(f)} \\ & -\big(\alpha_n + \alpha_m\big)\sum_{f\in\mathcal{F}}\big(\mu_{(n,m)}^{(f)} - \mu_{(n,m)}^{(f)}[t-1]\big)^2 \end{aligned} \quad (14)$$
$$\text{s.t.} \quad \sum_{f\in\mathcal{F}} \mu_{(n,m)}^{(f)} \leq C_{(n,m)} \quad (15)$$
$$\mu_{(n,m)}^{(f)} \geq 0, \forall f \in \mathcal{S}_{(n,m)} \quad (16)$$
$$\mu_{(n,m)}^{(f)} = 0, \forall f \notin \mathcal{S}_{(n,m)} \quad (17)$$
- For each $f \in \mathcal{F}$, if node n is not the destination of f, i.e., $n \neq \text{Dst}(f)$, update virtual queue $Q_n^{(f)}[t+1]$ by (9).

C. Almost Closed-Form Updates in Algorithm 1

This subsection shows the decisions $x_f[t]$ and $\mu_l^{(f)}[t]$ in Algorithm 1 have either closed-form solutions or "almost" closed-form solutions at each iteration t.

Lemma 2: Let $\hat{x}_f \equiv x_f[t]$ denote the solution to (12)-(13).

1) Suppose $\text{dom}(U_f) = [0, \infty)$ and $U_f(x_f)$ is differentiable. Let $\psi(x_f) = U_f'(x_f) - 2\alpha_n x_f + 2\alpha_n x_f[t-1] - W_n^{(f)}[t]$. If $\psi(0) \leq 0$, then $\hat{x}_f = 0$; otherwise $\hat{x}_f$ is the root to the equation $\psi(x_f) = 0$ and can be found by a bisection search.
2) Suppose $\text{dom}(U_f) = (0, \infty)$ and $U_f(x_f) = w_f \log(x_f)$ for some weight $w_f > 0$. Then:
$$\begin{aligned} \hat{x}_f =& \frac{2\alpha_n x_f[t-1] - W_n^{(f)}[t]}{4\alpha_n} \\ &+ \frac{\sqrt{(W_n^{(f)}[t] - 2\alpha_n x_f[t-1])^2 + 8\alpha_n w_f}}{4\alpha_n} \end{aligned}$$

Proof: Omitted for brevity. ■

The problem (14)-(17) can be represented as follows by eliminating $\mu_{(n,m)}^{(f)}, f \notin \mathcal{S}_{(n,m)}$, completing the square and replacing maximization with minimization. (Note that $K = |\mathcal{S}_{(n,m)}| \leq |\mathcal{F}|$.)

$$\min \quad \frac{1}{2}\sum_{k=1}^{K}(z_k - a_k)^2 \quad (18)$$
$$\text{s.t.} \quad \sum_{k=1}^{K} z_k \leq b \quad (19)$$
$$z_k \geq 0, \forall k \in \{1, 2, \ldots, K\} \quad (20)$$

Lemma 3: The solution to problem (18)-(20) is given by $z_k^* = \max\{0, a_k - \theta^*\}, \forall k \in \{1, 2, \ldots, K\}$ where $\theta^* \geq 0$ can be found either by a bisection search (See Appendix D) or by Algorithm 2 with complexity $O(K \log K)$.

Proof: A similar problem where (19) is replaced with an equality constraint is considered in [32]. The optimal solution to this quadratic program is characterized by its KKT condition and a corresponding algorithm can be developed to obtain its KKT point. A complete proof is presented in Appendix D. ■

Algorithm 2 Algorithm to solve problem (18)-(20)

1) Check if $\sum_{k=1}^{K} \max\{0, a_k\} \leq b$ holds. If yes, let $\theta^* = 0$ and $z_k^* = \max\{0, a_k\}, \forall k \in \{1, 2, \ldots, K\}$ and terminate the algorithm; else, continue to the next step.
2) Sort all $a_k, \in \{1, 2, \ldots, K\}$ in a decreasing order π such that $a_{\pi(1)} \geq a_{\pi(2)} \geq \cdots \geq a_{\pi(K)}$. Define $S_0 = 0$.
3) For $k = 1$ to K
   - Let $S_k = S_{k-1} + a_k$. Let $\theta^* = \frac{S_k - b}{k}$.
   - If $\theta^* \geq 0$, $a_{\pi(k)} - \theta^* > 0$ and $a_{\pi(k+1)} - \theta^* \leq 0$, then terminate the loop; else, continue to the next iteration in the loop.
4) Let $z_k^* = \max\{0, a_k - \theta^*\}, \forall k \in \{1, 2, \ldots, K\}$ and terminate the algorithm.

Note that step (3) in Algorithm 2 has complexity $O(K)$ and hence the overall complexity of Algorithm 2 is dominated by the sorting step (2) with complexity $O(K \log(K))$.

IV. Performance Analysis of Algorithm 1

A. Preliminaries

This subsection introduces facts from convex analysis and some useful additional notation.

Definition 1 (Strongly Concave Functions): Let $\mathcal{Z} \subseteq \mathbb{R}^n$ be a convex set. Function f is said to be strongly concave on

$\mathcal{Z}$ with modulus α if there exists a constant $\alpha > 0$ such that $f(\mathbf{z}) + \frac{1}{2}\alpha\|\mathbf{z}\|^2$ is concave on $\mathcal{Z}$.

By the definition of strongly concave functions, it is easy to show that if $f(\mathbf{z})$ is concave and $\alpha > 0$, then $f(\mathbf{z}) - \alpha\|\mathbf{z} - \mathbf{z}_0\|^2$ is strongly concave with modulus 2α for any constant $\mathbf{z}_0$. The next lemma shows that a strongly concave function deviates at least quadratically from its maximum value.

Lemma 4 (Corollary 1 in [23]): Let $\mathcal{Z} \subseteq \mathbb{R}^n$ be a convex set. Let function f be strongly concave on $\mathcal{Z}$ with modulus α and $\mathbf{z}^{opt}$ be a global maximum of f on $\mathcal{Z}$. Then, $f(\mathbf{z}^{opt}) \geq f(\mathbf{z}) + \frac{\alpha}{2}\|\mathbf{z}^{opt} - \mathbf{z}\|^2$ for all $\mathbf{z} \in \mathcal{Z}$.

Define column vector $\mathbf{y} = [x_f; \mu_l^{(f)}]_{f\in\mathcal{F}, l\in\mathcal{L}}$, which aggregates all optimization variables in problem (1)-(6). For each $f \in \mathcal{F}, n \in \mathcal{N} \setminus \{\text{Dst}(f)\}$, define column vector

$$\mathbf{y}_n^{(f)} = \begin{cases} [x_f; \mu_l^{(f)}]_{l\in\mathcal{I}(n)\cup\mathcal{O}(n)} & \text{if } n = \text{Src}(f), \\ [\mu_l^{(f)}]_{l\in\mathcal{I}(n)\cup\mathcal{O}(n)} & \text{else,} \end{cases} \tag{21}$$

which is composed of the control actions appearing in each constraint (2); and introduce a function with respect to $\mathbf{y}_n^{(f)}$ as

$$g_n^{(f)}(\mathbf{y}_n^{(f)}) = x_f \mathbf{1}_{\{n=\text{Src}(f)\}} + \sum_{l\in\mathcal{I}(n)} \mu_l^{(f)} - \sum_{l\in\mathcal{O}(n)} \mu_l^{(f)} \tag{22}$$

Thus, constraint (2) can be rewritten as

$$g_n^{(f)}(\mathbf{y}_n^{(f)}) \leq 0, \forall f \in \mathcal{F}, \forall n \in \mathcal{N} \setminus \{\text{Dst}(f)\}.$$

Note that each vector $\mathbf{y}_n^{(f)}$ is a subvector of $\mathbf{y}$ and has length $d_n + 1$ where d_n is the degree of node n (the total number of outgoing links and incoming links) if node n is the source of session f; and has length d_n if node n is not the source of session f. Note that components in different vector variables $\mathbf{y}_n^{(f)}$ can overlap. The vector variables $\mathbf{y}$ and $\mathbf{y}_n^{(f)}$ are introduced only to simplify notation.

The virtual queue update equation (9) can be rewritten as:

$$Q_n^{(f)}[t+1] = Q_n^{(f)}[t] + g_n^{(f)}(\mathbf{y}_n^{(f)}[t]). \tag{23}$$

The weight update equation (11) can be rewritten as:

$$W_n^{(f)}[t] = Q_n^{(f)}[t] + g_n^{(f)}(\mathbf{y}_n^{(f)}[t-1]). \tag{24}$$

Define

$$L(t) = \frac{1}{2}\sum_{f\in\mathcal{F}}\sum_{n\in\mathcal{N}\setminus\text{Dst}(f)} \big(Q_n^{(f)}[t]\big)^2 \tag{25}$$

and call it a *Lyapunov function*. In the remainder of this paper, double summations are often written compactly as a single summation, e.g.,

$$\sum_{f\in\mathcal{F}}\sum_{n\in\mathcal{N}\setminus\text{Dst}(f)} (\,\cdot\,) \overset{\Delta}{=} \sum_{\substack{f\in\mathcal{F},\\ n\in\mathcal{N}\setminus\text{Dst}(f)}} (\,\cdot\,).$$

Define the Lyapunov drift as

$$\Delta[t] = L(t+1) - L(t).$$

The following lemma follows directly from equation (23).

Lemma 5: At each iteration $t \in \{0, 1, \ldots\}$ in Algorithm 1, the Lyapunov drift is given by

$$\Delta[t] = \sum_{\substack{f\in\mathcal{F},\\ n\in\mathcal{N}\setminus\text{Dst}(f)}} \Big(Q_n^{(f)}[t]g_n^{(f)}(\mathbf{y}_n^f[t]) + \frac{1}{2}\big(g_n^{(f)}(\mathbf{y}_n^f[t])\big)^2\Big). \tag{26}$$

Proof: Fix $f \in \mathcal{F}$ and $n \in \mathcal{N} \setminus \text{Dst}(f)$, we have

$$\begin{aligned} &\frac{1}{2}\big(Q_n^{(f)}[t+1]\big)^2 - \frac{1}{2}\big(Q_n^{(f)}[t]\big)^2 \\ \overset{(a)}{=}&\frac{1}{2}\big(Q_n^{(f)}[t] + g_n^{(f)}(\mathbf{y}_n^{(f)}[t])\big)^2 - \frac{1}{2}\big(Q_n^{(f)}[t]\big)^2 \\ =&Q_n^{(f)}[t]g_n^{(f)}(\mathbf{y}_n^f[t]) + \frac{1}{2}\big(g_n^{(f)}(\mathbf{y}_n^f[t])\big)^2 \end{aligned} \tag{27}$$

where (a) follows from (23).

By the definition of $\Delta[t]$, we have

$$\begin{aligned} \Delta[t] =& \frac{1}{2}\sum_{\substack{f\in\mathcal{F},\\ n\in\mathcal{N}\setminus\text{Dst}(f)}} \Big(\big(Q_n^{(f)}[t+1]\big)^2 - \big(Q_n^{(f)}[t]\big)^2\Big) \\ \overset{(a)}{=}& \sum_{\substack{f\in\mathcal{F},\\ n\in\mathcal{N}\setminus\text{Dst}(f)}} \Big(Q_n^{(f)}[t]g_n^{(f)}(\mathbf{y}_n^f[t]) + \frac{1}{2}\big(g_n^{(f)}(\mathbf{y}_n^f[t])\big)^2\Big) \end{aligned}$$

where (a) follows from (27). ∎

B. Intuitions of Algorithm 1

Recall $\mathbf{y} = [x_f; \mu_l^{(f)}]_{f\in\mathcal{F}, l\in\mathcal{L}}$. Define

$$h(\mathbf{y}) = \sum_{f\in\mathcal{F}} U_f(x_f). \tag{28}$$

The intuition behind our approach comes from examining the "drift-plus-penalty" expression from prior Lyapunov based backpressure type algorithms as in [5]. With $\Delta[t]$ given in (26) being a change of the Lyapunov function, prior backpressure algorithms would yield

$$\begin{aligned} &Vh(\mathbf{y}[t]) - \Delta[t] \\ =&Vh(\mathbf{y}[t]) - \mathbf{Q}[t]^{\mathsf{T}}\mathbf{g}(\mathbf{y}[t]) - \underbrace{\frac{1}{2}\|\mathbf{g}(\mathbf{y}[t])\|^2}_{\leq B} \end{aligned}$$

where $V > 0$ is an algorithm parameter and $\mathbf{Q}[t]$ and $\mathbf{g}(\mathbf{y}[t])$ are vectors that concatenate the individual $Q_n^{(f)}[t]$ and $g_n^{(f)}(\cdot)$ components. Prior algorithms seek to choose $\mathbf{y}[t]$ to maximize the first two terms on the right-hand-side of this expression over (3)-(6); the quadratic term marked by an underbrace is bounded by a constant B and results in a B/V error (which can be made small by increasing the V parameter). One could eliminate the B term by changing the algorithm to optimize all three terms (including the $\frac{1}{2}\|\mathbf{g}(\mathbf{y}[t])\|^2$ term) but this would be a nonseparable optimization that is as difficult as the original problem (1)-(6). Our insight is to approximate this quadratic term as $[\mathbf{g}(\mathbf{y}[t-1])]^{\mathsf{T}}[\mathbf{g}(\mathbf{y}[t])]$, include this easy-to-optimize term in the maximization step, and then compensate for the approximation error by introducing prox-terms $\|\mathbf{y}_n^{(f)}[t] - \mathbf{y}_n^{(f)}[t-1]\|^2$ together with a strong concavity argument (which we can rigorously establish even though our utility function is not necessarily strongly concave). Since this approach eliminates the B constant, we no longer need the V parameter to be large, and so we use $V = 1$.

At each time t, consider choosing a decision vector $\mathbf{y}[t]$ that includes elements in each subvector $\mathbf{y}_n^{(f)}[t]$ to solve problem (29)-(30). The expression (29) is a *modified drift-plus-penalty expression*. Unlike the standard drift-plus-penalty expressions from [5], the above expression uses weights $W_n^{(f)}[t]$, which augments each $Q_n^{(f)}[t]$ by $g_n^{(f)}(\mathbf{y}_n^{(f)}[t-1])$, rather than virtual queues $Q_n^{(f)}[t]$. It also includes a "prox"-like term that penalizes deviation from the previous $\mathbf{y}[t-1]$ vector. This results in the novel backpressure-type algorithm of Algorithm 1. Indeed, the decisions in Algorithm 1 were *derived* as the solution to problem (29)-(30). This is formalized in the next lemma.

Lemma 6: At each iteration $t \in \{0, 1, \dots\}$, the action $\mathbf{y}[t]$ jointly chosen in Algorithm 1 is the solution to problem (29)-(30).

Proof: The proof involves collecting terms associated with the $x_f[t]$ and $\mu_l^{(f)}[t]$ decisions. See Appendix E for details. ■

Furthermore, the next lemma relates $h(\mathbf{y}^*)$ and $h(\mathbf{y}[t])$ yielded by action $\mathbf{y}[t]$ that aggregates all control actions jointly chosen in Algorithm 1 at each iteration $t \in \{0, 1, \dots\}$.

Lemma 7: Let $\mathbf{y}^* = [x_f^*; \mu_l^{(f),*}]_{f\in\mathcal{F}, l\in\mathcal{L}}$ be an optimal solution to problem (1)-(6) given in Fact 1, i.e., $g_n^{(f)}(\mathbf{y}_n^{(f),*}) = 0, \forall f \in \mathcal{F}, \forall n \in \mathcal{N} \setminus \text{Dst}(f)$. If $\alpha_n \geq \frac{1}{2}(d_n+1), \forall n \in \mathcal{N}$, where d_n is the degree of node n, then the action $\mathbf{y}[t] = [x_f[t]; \mu_l^{(f)}[t]]_{f\in\mathcal{F}, l\in\mathcal{L}}$ jointly chosen in Algorithm 1 at each iteration $t \in \{0, 1, \dots\}$ satisfies

$$h(\mathbf{y}[t]) \geq h(\mathbf{y}^*) + \Phi[t] - \Phi[t-1] + \Delta[t]$$

where $\Phi[t] = \sum_{f\in\mathcal{F}, n\in\mathcal{N}} \big(\alpha_n \mathbf{1}_{\{n\neq \text{Dst}(f)\}} \|\mathbf{y}_n^{(f),*} - \mathbf{y}_n^{(f)}[t]\|^2 + \alpha_n \mathbf{1}_{\{n=\text{Dst}(f)\}} \sum_{l\in\mathcal{I}(n)} (\mu_l^{(f),*} - \mu_l^{(f)}[t])^2\big)$ and $h(\mathbf{y})$ is defined in (28).

Proof: See Appendix F. ■

It remains to show that this modified backpressure algorithm leads to fundamentally improved performance.

C. Utility Optimality Gap Analysis

Define column vector $\mathbf{Q}[t] = \big[Q_n^{(f)}[t]\big]_{f\in\mathcal{F}, n\in\mathcal{N}\setminus\{\text{Dst}(f)\}}$ as the stacked vector of all virtual queues $Q_n^{(f)}[t]$ defined in (9). Note that (25) can be rewritten as $L(t) = \frac{1}{2}\|\mathbf{Q}[t]\|^2$. Define vectorized constraints (2) as $\mathbf{g}(\mathbf{y}) = [g_n^{(f)}(\mathbf{y}_n^{(f)})]_{f\in\mathcal{F}, n\in\mathcal{N}\setminus\text{Dst}(f)}$.

Lemma 8: Let $\mathbf{y}^* = [x_f^*; \mu_l^{(f),*}]_{f\in\mathcal{F}, l\in\mathcal{L}}$ be an optimal solution to problem (1)-(6) given in Fact 1, i.e., $g_n^{(f)}(\mathbf{y}_n^{(f),*}) = 0, \forall f \in \mathcal{F}, \forall n \in \mathcal{N} \setminus \text{Dst}(f)$. If $\alpha_n \geq \frac{1}{2}(d_n+1), \forall n \in \mathcal{N}$ in Algorithm 1, where d_n is the degree of node n, then for all $t \geq 1$,

$$\sum_{\tau=0}^{t-1} h(\mathbf{y}[\tau]) \geq t h(\mathbf{y}^*) - \zeta + \frac{1}{2}\|\mathbf{Q}[t]\|^2.$$

where $\zeta = \Phi[-1] = \sum_{f\in\mathcal{F}, n\in\mathcal{N}} \big(\alpha_n \mathbf{1}_{\{n\neq\text{Dst}(f)\}} \|\mathbf{y}_n^{(f),*}\|^2 + \alpha_n \mathbf{1}_{\{n=\text{Dst}(f)\}} \sum_{l\in\mathcal{I}(n)} (\mu_l^{(f),*})^2\big)$ is a constant.

Proof: By Lemma 7, we have $h(\mathbf{y}[\tau]) \geq h(\mathbf{y}^*) + \Phi[\tau] - \Phi[\tau-1] + \Delta[\tau], \forall \tau \in \{0, 1, \dots, t-1\}$. Recall $\Delta[\tau] = L[\tau+1] - L[\tau]$. Summing over $\tau \in \{0, 1, \dots, t-1\}$ yields

$$\begin{aligned}
&\sum_{\tau=0}^{t-1} h(\mathbf{y}[\tau]) \\
\geq& t h(\mathbf{y}^*) + \sum_{\tau=0}^{t-1} \big(\Phi[\tau] - \Phi[\tau-1]\big) + \sum_{\tau=0}^{t-1} \Delta[\tau] \\
=& t h(\mathbf{y}^*) + \Phi[t-1] - \Phi[-1] + L[t] - L[0] \\
\overset{(a)}{\geq}& t h(\mathbf{y}^*) - \Phi[-1] + \frac{1}{2}\|\mathbf{Q}[t]\|^2
\end{aligned}$$

where (a) follows from the fact that $\Phi[t] \geq 0, \forall t$, $L[t] = \frac{1}{2}\|\mathbf{Q}[t]\|^2$ and $L[0] = 0$. ■

The next theorem summarizes that Algorithm 1 yields a vanishing utility optimality gap that approaches zero like $O(1/t)$.

Theorem 1: Let $\mathbf{y}^* = [x_f^*; \mu_l^{(f),*}]_{f\in\mathcal{F}, l\in\mathcal{L}}$ be an optimal solution to problem (1)-(6) given in Fact 1, i.e., $g_n^{(f)}(\mathbf{y}_n^{(f),*}) = 0, \forall f \in \mathcal{F}, \forall n \in \mathcal{N} \setminus \text{Dst}(f)$. If $\alpha_n \geq \frac{1}{2}(d_n+1), \forall n \in \mathcal{N}$ in Algorithm 1, where d_n is the degree of node n, then for all $t \geq 1$, we have

$$\frac{1}{t}\sum_{\tau=0}^{t-1}\sum_{f\in\mathcal{F}} U_f(x_f[\tau]) \geq \sum_{f\in\mathcal{F}} U_f(x_f^*) - \frac{1}{t}\zeta,$$

where ζ is a constant defined in Lemma 8. Moreover, if we define $\overline{x}_f[t] = \frac{1}{t}\sum_{\tau=0}^{t-1} x_f[\tau], \forall f \in \mathcal{F}$, then

$$\sum_{f\in\mathcal{F}} U_f(\overline{x}_f[t]) \geq \sum_{f\in\mathcal{F}} U_f(x_f^*) - \frac{1}{t}\zeta.$$

Proof: Recall by (28) that $h(\mathbf{y}) = \sum_{f\in\mathcal{F}} U_f(x_f)$. By Lemma 8, we have

$$\begin{aligned}
\sum_{\tau=0}^{t-1}\sum_{f\in\mathcal{F}} U_f(x_f[\tau]) \geq& t\sum_{f\in\mathcal{F}} U_f(x_f^*) - \zeta + \frac{1}{2}\|\mathbf{Q}[t]\|^2 \\
\overset{(a)}{\geq}& t\sum_{f\in\mathcal{F}} U_f(x_f^*) - \zeta.
\end{aligned}$$

where (a) follows from the trivial fact that $\|\mathbf{Q}[t]\|^2 \geq 0$.

Dividing both sides by a factor t yields the first inequality in this theorem. The second inequality follows from the concavity of $U_f(\cdot)$ and Jensen's inequality. ■

D. Queue Stability Analysis

Lemma 9: Let $\mathbf{Q}[t], t \in \{0, 1, \dots\}$ be the virtual queues in Algorithm 1. For any $t \geq 1$,

$$\mathbf{Q}[t] = \sum_{\tau=0}^{t-1} \mathbf{g}(\mathbf{y}[\tau])$$

Proof: This lemma follows directly from the fact that $\mathbf{Q}[0] = \mathbf{0}$ and queue update equation (9) can be written as $\mathbf{Q}[t+1] = \mathbf{Q}[t] + \mathbf{g}(\mathbf{y}[t])$. ■

The next theorem shows the boundedness of all virtual queues $Q_n^{(f)}[t]$ in Algorithm 1.

Theorem 2: Let $\mathbf{y}^* = [x_f^*; \mu_l^{(f),*}]_{f\in\mathcal{F}, l\in\mathcal{L}}$ be an optimal solution to problem (1)-(6) given in Fact 1, i.e., $g_n^{(f)}(\mathbf{y}_n^{(f),*}) = 0, \forall f \in \mathcal{F}, \forall n \in \mathcal{N} \setminus \text{Dst}(f)$, and $\boldsymbol{\lambda}^*$ be a Lagrange multiplier

$$\max_{\mathbf{y}} \ h(\mathbf{y}) - \sum_{\substack{f\in\mathcal{F},\\ n\in\mathcal{N}\setminus\mathrm{Dst}(f)}} \left(W_n^{(f)}[t]g_n^{(f)}(\mathbf{y}_n^{(f)}) + \alpha_n\|\mathbf{y}_n^{(f)} - \mathbf{y}_n^{(f)}[t-1]\|^2\right) - \sum_{\substack{f\in\mathcal{F},\\ n=\mathrm{Dst}(f)}} \alpha_n \sum_{l\in\mathcal{I}(n)} (\mu_l^{(f)} - \mu_l^{(f)}[t-1])^2 \tag{29}$$

$$\text{s.t.} \quad (3)\text{-}(6) \tag{30}$$

vector given in Assumption 2. If $\alpha_n \geq \frac{1}{2}(d_n+1), \forall n \in \mathcal{N}$ in Algorithm 1, where d_n is the degree of node n, then $\forall f \in \mathcal{F}, \forall n \in \mathcal{N} \setminus \{\mathrm{Dst}(f)\}$, we have

$$|Q_n^{(f)}[t]| \leq \|\mathbf{Q}[t]\| \leq 2\|\boldsymbol{\lambda}^*\| + \sqrt{2\zeta}, \forall t \geq 1$$

where ζ is a constant defined in Lemma 8.

Proof: Let $q(\boldsymbol{\lambda}) = \sup_{\mathbf{y}\in\mathcal{C}}\left\{h(\mathbf{y}) - \boldsymbol{\lambda}^{\mathsf{T}}\mathbf{g}(\mathbf{y})\right\}$ be the Lagrangian dual function defined in Assumption 2. For all $\tau \in \{0,1,\ldots,\}$, by Assumption 2, we have

$$h(\mathbf{y}^*) = q(\boldsymbol{\lambda}^*) \overset{(a)}{\geq} h(\mathbf{y}[\tau]) - \boldsymbol{\lambda}^{*,\mathsf{T}}\mathbf{g}(\mathbf{y}[\tau])$$

where (a) follows from the definition of $q(\boldsymbol{\lambda}^*)$. Rearranging terms yields

$$h(\mathbf{y}[\tau]) \leq h(\mathbf{y}^*) + \boldsymbol{\lambda}^{*,\mathsf{T}}\mathbf{g}(\mathbf{y}[\tau]), \forall \tau \in \{0,1,\ldots\}.$$

Fix $t > 0$. Summing over $\tau \in \{0,1,\ldots,t-1\}$ yields

$$\begin{aligned}\sum_{\tau=0}^{t-1} h(\mathbf{y}[\tau]) \leq & t h(\mathbf{y}^*) + \sum_{\tau=0}^{t-1}\boldsymbol{\lambda}^{*,\mathsf{T}}\mathbf{g}(\mathbf{y}[\tau])\\ =& t h(\mathbf{y}^*) + \boldsymbol{\lambda}^{*,\mathsf{T}}\sum_{\tau=0}^{t-1}\mathbf{g}(\mathbf{y}[\tau])\\ \overset{(a)}{=}& t h(\mathbf{y}^*) + \boldsymbol{\lambda}^{*,\mathsf{T}}\mathbf{Q}[t]\\ \overset{(b)}{\leq}& t h(\mathbf{y}^*) + \|\boldsymbol{\lambda}^*\|\|\mathbf{Q}[t]\|\end{aligned}$$

where (a) follows form Lemma 9 and (b) follows from Cauchy-Schwarz inequality.

On the other hand, by Lemma 8, we have

$$\sum_{\tau=0}^{t-1} h(\mathbf{y}[\tau]) \geq t h(\mathbf{y}^*) - \zeta + \frac{1}{2}\|\mathbf{Q}[t]\|^2.$$

Combining the last two inequalities and cancelling the common terms yields

$$\begin{aligned}&\frac{1}{2}\|\mathbf{Q}[t]\|^2 - \zeta \leq \|\boldsymbol{\lambda}^*\|\|\mathbf{Q}[t]\|\\ \Rightarrow&\left(\|\mathbf{Q}[t]\| - \|\boldsymbol{\lambda}^*\|\right)^2 \leq \|\boldsymbol{\lambda}^*\|^2 + 2\zeta\\ \Rightarrow&\|\mathbf{Q}[t]\| \leq \|\boldsymbol{\lambda}^*\| + \sqrt{\|\boldsymbol{\lambda}^*\|^2 + 2\zeta}\\ \overset{(a)}{\Rightarrow}&\|\mathbf{Q}[t]\| \leq 2\|\boldsymbol{\lambda}^*\| + \sqrt{2\zeta}\end{aligned}$$

where (a) follows from the basic inequality $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$ for any $a, b \geq 0$.

Thus, for any $f \in \mathcal{F}$ and $n \in \mathcal{N} \setminus \{\mathrm{Dst}(f)\}$, we have

$$|Q_n^{(f)}[t]| \leq \|\mathbf{Q}[t]\| \leq 2\|\boldsymbol{\lambda}^*\| + \sqrt{2\zeta}.$$

■

This theorem shows that the absolute values of all virtual queues $Q_n^{(f)}[t]$ are bounded by a constant $B = 2\|\boldsymbol{\lambda}^*\| + \sqrt{2\zeta}$ from above. By Lemma 1 and discussions in Section III-A, the actual physical queues $Z_n^{(f)}[t]$ evolving via (8) satisfy $Z_n^{(f)}[t] \leq 2B + \sum_{l\in\mathcal{O}(n)} C_l, \forall t$. This is summarized in the next corollary.

Corollary 1: Let $\mathbf{y}^* = [x_f^*; \mu_l^{(f),*}]_{f\in\mathcal{F}, l\in\mathcal{L}}$ be an optimal solution to problem (1)-(6) given in Fact 1, i.e., $g_n^{(f)}(\mathbf{y}_n^{(f),*}) = 0, \forall f \in \mathcal{F}, \forall n \in \mathcal{N} \setminus \mathrm{Dst}(f)$, and $\boldsymbol{\lambda}^*$ be a Lagrange multiplier vector given in Assumption 2. If $\alpha_n \geq \frac{1}{2}(d_n+1)^2, \forall n \in \mathcal{N}$ in Algorithm 1, where d_n is the degree of node n, then all actual physical queues $Z_n^{(f)}[t], \forall f \in \mathcal{F}, \forall n \in \mathcal{N} \setminus \{\mathrm{Dst}(f)\}$ in the network evolving via (8) satisfy

$$Z_n^{(f)}[t] \leq 4\|\boldsymbol{\lambda}^*\| + 2\sqrt{2\zeta} + \sum_{l\in\mathcal{O}(n)} C_l, \quad \forall t.$$

where ζ is a constant defined in Lemma 8.

Define vector $\mathbf{x}^* = [x_f^*]_{f\in\mathcal{F}}$ and $\overline{\mathbf{x}}[t] = [\overline{x}_f[t]]_{f\in\mathcal{F}}$ where x_f^* and $\overline{x}_f[t]$ are defined in Theorem 1. Note that if each $U_f(\mathbf{x}_f)$ is strongly concave with respect to x_f, then $\mathbf{x}^*$ is unique by strong concavity. (However, $[\mu_l^{(f),*}]_{l\in\mathcal{L}}$ is not necessarily unique.) In this case, Corollary 2 shows $\overline{\mathbf{x}}[t]$ yielded by Algorithm 1 converges to the unique maximizer $\mathbf{x}^*$.

Corollary 2: If the conditions in Theorem 1 hold and each $U_f(x_f)$ is strongly concave with respect to x_f, then Algorithm 1 guarantees $\overline{\mathbf{x}}[t] \to \mathbf{x}^*$ as $t \to \infty$.

Proof: Assume each $U_f(x_f)$ is strongly concave with respect to x_f with modulus c_f. Let $c = \min_{f\in\mathcal{F}}\{c_f\}$. By Assumption 2, we have $\mathbf{y}^* = \operatorname{argmax}_{\mathbf{y}\in\mathcal{C}}\{h(\mathbf{y}) - \boldsymbol{\lambda}^{*,\mathsf{T}}\mathbf{g}(\mathbf{y})\}$. Recall that $h(\mathbf{y}) = \sum_{f\in\mathcal{F}} U_f(x_f)$ and $\mathbf{g}(\mathbf{y})$ are separable since they can be written as the sum of scalar functions in terms of x_f and $\mu_l^{(f)}$. Thus, x_f^* and $[\mu_l^{(f),*}]_{f\in\mathcal{F}}$ appear separably and maximize in the left-side of (31) where each x_f^* satisfying (6) maximizes a strongly concave part and each vector $[\mu_l^{(f),*}]_{f\in\mathcal{F}}$ satisfying (3)-(5) maximizes a concave part. Define $\overline{\mathbf{y}}[t] = \frac{1}{t}\sum_{\tau=0}^{t-1}\mathbf{y}[\tau]$. Note that $\overline{\mathbf{y}}[t]$ satisfies (3)-(6) since each $\mathbf{y}[\tau]$ is generated by Algorithm 1. By Lemma 4, for all $t \geq 1$,

$$\begin{aligned}&\sum_{f\in\mathcal{F}} U_f(x_f^*) - \boldsymbol{\lambda}^{*,\mathsf{T}}\mathbf{g}(\mathbf{y}^*)\\ \geq& \sum_{f\in\mathcal{F}} U_f(\overline{x}_f[t]) - \boldsymbol{\lambda}^{*,\mathsf{T}}\mathbf{g}(\overline{\mathbf{y}}[t]) + \sum_{f\in\mathcal{F}}\frac{c_f}{2}(x_f^* - \overline{x}_f[t])^2\\ \overset{(a)}{\geq}& \sum_{f\in\mathcal{F}} U_f(\overline{x}_f[t]) - \boldsymbol{\lambda}^{*,\mathsf{T}}\mathbf{g}(\overline{\mathbf{y}}[t]) + \frac{c}{2}\|\overline{\mathbf{x}}[t] - \mathbf{x}^*\|^2\\ \overset{(b)}{=}& \sum_{f\in\mathcal{F}} U_f(\overline{x}_f[t]) - \boldsymbol{\lambda}^{*,\mathsf{T}}\frac{1}{t}\sum_{t=0}^{t-1}\mathbf{g}(\mathbf{y}[\tau]) + \frac{c}{2}\|\overline{\mathbf{x}}[t] - \mathbf{x}^*\|^2\\ \overset{(c)}{=}& \sum_{f\in\mathcal{F}} U_f(\overline{x}_f[t]) - \frac{1}{t}\boldsymbol{\lambda}^{*,\mathsf{T}}\mathbf{Q}(t) + \frac{c}{2}\|\overline{\mathbf{x}}[t] - \mathbf{x}^*\|^2\end{aligned} \tag{31}$$

where (a) follows from $c = \min_{f\in\mathcal{F}}\{c_f\}$; (b) follows from the linearity of $\mathbf{g}(\cdot)$ and the definition of $\overline{\mathbf{y}}[t]$; and (c) follows from Lemma 9.

Recall that $\boldsymbol{\lambda}^{*,\mathsf{T}}\mathbf{g}(\mathbf{y}^*) = 0$ by strong duality of convex programs (Assumption 2). Thus, (31) implies

$$\begin{aligned}
&\frac{c}{2}\|\overline{\mathbf{x}}[t]-\mathbf{x}^*\|^2\\
\leq& \sum_{f\in\mathcal{F}} U_f(x_f^*) - \sum_{f\in\mathcal{F}} U_f(\overline{x}_f[t]) + \frac{1}{t}\boldsymbol{\lambda}^{*,\mathsf{T}}\mathbf{Q}(t)\\
\overset{(a)}{\leq}& \sum_{f\in\mathcal{F}} U_f(x_f^*) - \sum_{f\in\mathcal{F}} U_f(\overline{x}_f[t]) + \frac{1}{t}\|\boldsymbol{\lambda}^*\|\|\mathbf{Q}(t)\|\\
\overset{(b)}{\leq}& \frac{1}{t}\zeta + \frac{1}{t}\|\boldsymbol{\lambda}^*\|(2\|\boldsymbol{\lambda}^*\| + \sqrt{2\zeta})
\end{aligned}$$

where (a) follows from the Cauchy-Schwarz inequality; and (b) follows from Theorem 1, which implies $\sum_{f\in\mathcal{F}} U_f(\overline{x}_f[t]) \geq \sum_{f\in\mathcal{F}} U_f(x_f^*) - \frac{1}{t}\zeta, \forall t \geq 1$, and Theorem 2, which implies $\|\mathbf{Q}(t)\| \leq 2\|\boldsymbol{\lambda}^*\| + \sqrt{2\zeta}, \forall t \geq 1$.

Taking limits $t \to \infty$ on both sides yields that $\overline{\mathbf{x}}[t] \to \mathbf{x}^*$ as $t \to \infty$. ■

E. Performance of Algorithm 1

Theorems 1 and 2 together imply that Algorithm 1 with $\alpha_n \geq \frac{1}{2}(d_n + 1), \forall n \in \mathcal{N}$ can achieve a vanishing utility optimality gap that decays like $O(1/t)$, where t is number of iterations, and guarantees the physical queues at each node for each session are always bounded by a constant that is independent of the utility optimality gap.

This is superior to existing backpressure algorithms from [6], [5], [11] that can achieve $O(\epsilon)$ utility gaps only at the cost of $O(1/\epsilon^2)$ or $O(1/\epsilon)$ queue lengths. To obtain a vanishing utility gap, existing backpressure algorithms in [6], [5], [11] necessarily yield unbounded queues. Note that to achieve $O(\epsilon)$ utility gaps, existing backpressure algorithms in [6], [5], [11] need to choose an algorithm parameter $V = O(1/\epsilon)$, which in turn leads to $O(V^2)$ or $O(V)$ queue lengths. In fact, the $O(1/\epsilon^2)$ queue bound in the primal-dual type backpressure algorithm [6] is given by $V^2\|\boldsymbol{\lambda}^*\| + B_1$ where $\boldsymbol{\lambda}^*$ is the Lagrangian multiplier vector attaining strong duality and B_1 is a constant determined by the problem parameters. A recent work [33] also shows that the $O(1/\epsilon)$ queue bound in the backpressure algorithm from drift-plus-penalty is of the order $V\|\boldsymbol{\lambda}^*\| + B_2$ where B_2 is also a constant determined by the problem parameters. Since $\boldsymbol{\lambda}^*$ is a constant vector independent of V and $V = O(1/\epsilon)$, both algorithms are claimed to have $O(1/\epsilon^2)$ or $O(1/\epsilon)$ queue bounds. By Corollary 1, Algorithm 1 guarantees physical queues at each node are bounded by $4\|\boldsymbol{\lambda}^*\| + B_3$, where B_3 is constant given a problem. Thus, the constant queue bound guaranteed by Algorithm 1 is typically smaller than the $O(V^2)$ or $O(V)$ queue bounds from [6] and [33] even for a small V. (A small V corresponds to a poor utility performance for the backpressure algorithms in [6], [5].)

V. Numerical Experiment

In this section, we consider a simple network with 6 nodes and 8 links as described in Figure 1. This network has two sessions: session 1 from node 1 to node 6 has utility function $\log(x_1)$ and session 2 from node 3 to node 4 has utility function $1.5\log(x_2)$. (The log utilities are widely used as metrics of proportional fairness in data networks [29].) The routing path of each session is arbitrary as long as data can be delivered from the source node to the destination node. For simplicity, assume that each link has capacity 1. The optimal source session rate to problem (1)-(6) is $x_1^* = 1.2$ and $x_2^* = 1.8$ and link session rates, i.e., static routing for each session, is drawn in Figure 2.

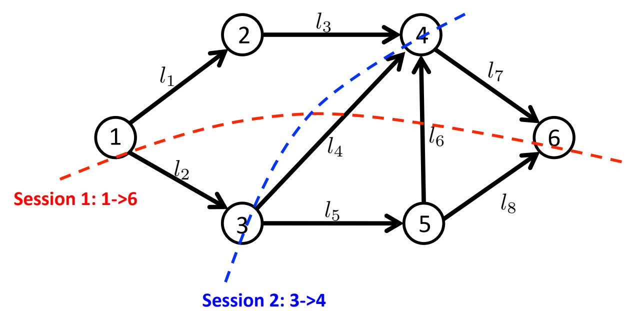

Fig. 1. A simple network with 6 nodes, 8 links and 2 sessions.

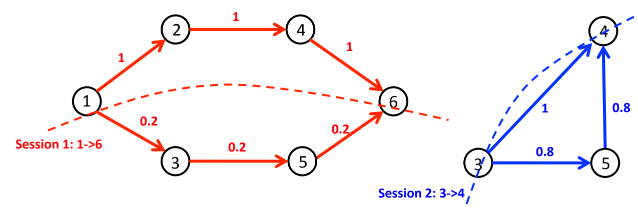

Fig. 2. The optimal routing for the network in Figure 1.

To compare the convergence performance of Algorithm 1 and the backpressure algorithm in [5] (with the best utility-delay tradeoff among all existing backpressure algorithms), we run both Algorithm 1 with $\alpha_n = \frac{1}{2}(d_n + 1), \forall n \in \mathcal{N}$ and the backpressure algorithm in [5] with $V = 500$ (This backpressure algorithm [5] yields $O(1/V)$ utility gaps and $O(V)$ queue lengths, where V is the algorithm parameter) to plot Figure 3. Since each session has a log utility function, which is strongly convex, by Corollary 2, the running averages of source rates yielded by Algorithm 1 converge to the optimal source rates. In fact, our simulation results in Figure 3 show that per iteration source rates $\mathbf{x}[t]$ (without averaging) also converge to the optimal source rates. It can be observed from Figure 3 that Algorithm 1 converges to the optimal source rates faster than the backpressure algorithm in [5]. The backpressure algorithm in [5] with $V = 500$ takes around 2500 iterations to converge to source rates close to $(1.2, 1.8)$ while Algorithm 1 only takes around 400 iterations to converges to $(1.2, 1.8)$ (as shown in the zoom-in subfigure at the top right corner.) In fact, the backpressure algorithm in [5] with $V = 500$ can not converge to the exact optimal source session rate $(1.2, 1.8)$ but can only converge to its neighborhood with a distance gap determined by the value of V. This is an effect from the fundamental $[O(1/V), O(V)]$ utility-delay tradeoff of the backpressure algorithm in [5]. In contrast, Algorithm 1 can eventually converge to the exact optimal source session rate $(1.2, 1.8)$. A zoom-in subfigure at the bottom right corner in Figure 3 verifies this and shows that the source rate for Session 1 in Algorithm 1 converges to 1.2 while the source rate in the backpressure algorithm in [5] with $V = 500$ oscillates around a point slightly larger than 1.2.

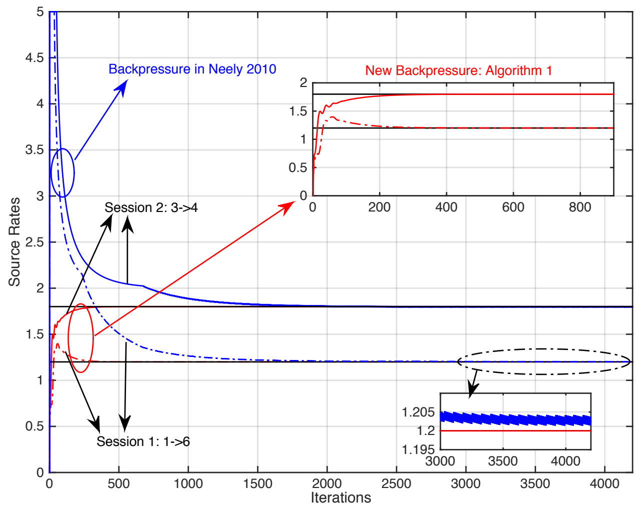

Fig. 3. Convergence performance comparison between Algorithm 1 and the backpressure algorithm in [5] with $V = 500$.

Corollary 1 shows that Algorithm 1 guarantees each actual queue in the network is bounded by constant $4\|\lambda^*\| + 2\sqrt{2\zeta} + \sum_{l\in\mathcal{O}(n)} C_l$. Recall that the backpressure algorithm in [5] can guarantee the actual queues in the network are bounded by a constant of order $V\|\lambda^*\|$. Figure 4 plots the sum of **actual queue length** at each node for Algorithm 1 and the backpressure algorithm in [5] with $V = 10, 100$ and 500. (Recall a larger V in the backpressure algorithm in [5] yields a smaller utility gap but a larger queue length.) It can be observed that Algorithm 1 has the smallest **actual queue length** (see the zoom-in subfigure) and the actual queue length of the backpressure algorithm in [5] scales linearly with respect to V.

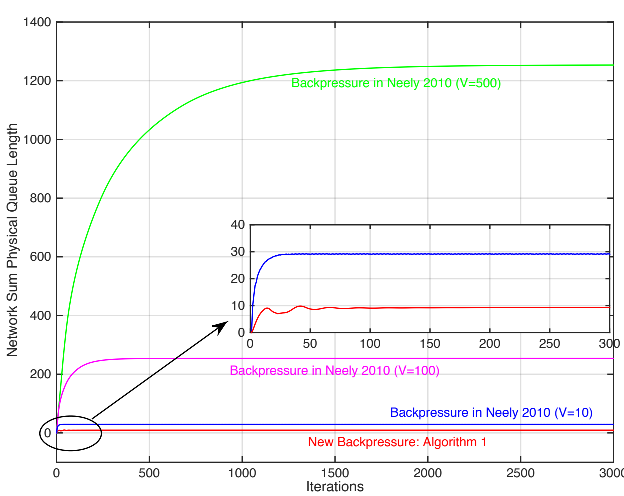

Fig. 4. Actual queue length comparison between Algorithm 1 and the backpressure algorithm in [5].

VI. Conclusion

This paper develops a new first-order Lagrangian dual type backpressure algorithm for joint rate control and routing in multi-hop data networks. The new backpressure algorithm can achieve vanishing utility optimality gaps and finite queue lengths. This improves the state-of-the-art $[O(\epsilon), O(1/\epsilon^2)]$ or $[O(\epsilon), O(1/\epsilon)]$ utility-delay tradeoff attained by existing backpressure algorithms [6], [10], [8], [11].

Appendix A
Network Utility Maximization with Predetermined Multi-Path

Consider the multi-path network utility maximization in [21] where each session has multiple given paths. Let x_f be the total source rate of each session $f \in \mathcal{F}$. Let $\mathcal{P}_f$ be the set of paths for session f. The link session rate $\mu_l^{(f)}$ becomes a vector $\boldsymbol{\mu}_l^{(f)} = [\mu_l^{(f,j)}]_{j\in\mathcal{P}_f}$. (Note that multiple paths for the same session are allowed to overlap.) Define $\mathcal{S}_l^{(f)}$ as the set of paths for session f that are allowed to use link l. Note that $\mathcal{S}_l^{(f)}$ are determined by the given paths for each session. That is, if path j for session f uses link l, then $j \in \mathcal{S}_l^{(f)}$; if no given path for session f uses link l, then $\mathcal{S}_l^{(f)} = \emptyset$. The multi-path network utility maximization problem can be formulated as follows:

$$\begin{aligned}
\max \quad & \sum_{f\in\mathcal{F}} U_f(x_f) \\
\text{s.t.} \quad & x_f \mathbf{1}_{\{n=\text{Src}(f)\}} + \sum_{l\in\mathcal{I}(n)}\sum_{j\in\mathcal{P}_f} \mu_l^{(f,j)} \leq \sum_{l\in\mathcal{O}(n)}\sum_{j\in\mathcal{P}_f} \mu_l^{(f,j)}, \\
& \qquad\qquad \forall f \in \mathcal{F}, \forall n \in \mathcal{N} \setminus \{\text{Dst}(f)\} \\
& \sum_{f\in\mathcal{F}}\sum_{j\in\mathcal{P}_f} \mu_l^{(f,j)} \leq C_l, \forall l \in \mathcal{L}, \\
& \mu_l^{(f,j)} \geq 0, \forall l \in \mathcal{L}, \forall f \in \mathcal{F}, \forall j \in \mathcal{S}_l^{(f)}, \\
& \mu_l^{(f,j)} = 0, \forall l \in \mathcal{L}, \forall f \in \mathcal{F}, \forall j \in \mathcal{P}_f \setminus \mathcal{S}_l^{(f)}, \\
& x_f \in \text{dom}(U_f), \forall f \in \mathcal{F}
\end{aligned}$$

The above formulation is in the form of problem (1)-(6) except that the variable dimension is extended.

In this case, Algorithm 1 developed in Section III-B to solve problem (1)-(6) can be adapted to solve the above multi-path network utility maximization problem by replacing $\mu_l^{(f)}$ with $\sum_{j\in\mathcal{P}_f} \mu_l^{(f,j)}$ in updates (11) and (9); and replacing subproblem (14)-(17) with

$$\begin{aligned}
\max_{\mu_{(n,m)}^{(f)}} \quad & \sum_{f\in\mathcal{F}} \big(W_n^{(f)}[t] - W_m^{(f)}[t]\big) \sum_{j\in\mathcal{P}_f} \mu_{(n,m)}^{(f,j)} \\
& -(\alpha_n + \alpha_m) \sum_{f\in\mathcal{F}}\sum_{j\in\mathcal{P}_f} \big(\mu_{(n,m)}^{(f,j)} - \mu_{(n,m)}^{(f,j)}[t-1]\big)^2 \\
\text{s.t.} \quad & \sum_{f\in\mathcal{F}}\sum_{j\in\mathcal{P}_f} \mu_{(n,m)}^{(f)} \leq C_{(n,m)} \\
& \mu_{(n,m)}^{(f,j)} \geq 0, \forall f \in \mathcal{F}, \forall j \in \mathcal{S}_{(m,n)}^{(f)} \\
& \mu_{(n,m)}^{(f,j)} = 0, \forall f \in \mathcal{F}, \forall j \notin \mathcal{S}_{(m,n)}^{(f)}
\end{aligned}$$

which again has the same structure as subproblem (14)-(17) except that the variable dimension is extended.

Appendix B
An Example Illustrating the Possibly Large Gap between Model (7) and Model (8)

Consider a network example shown in Figure 5. The network has $3k + 1$ nodes. Only node 0 is a destination and only nodes $a_i, b_i, i \in \{1, 2, \ldots, k\}$ can have exogenous arrivals. Assume all link capacities are equal to 1 and the exogenous arrivals are periodic with period $2k$, as follows:

- Time slot 1: One packet arrives at node a_1.
- Time slot 2: One packet arrives at node a_2.
- $\cdots$
- Time slot k: One packet arrives at node a_k.
- Time slot $k+1$: One packet arrives at node b_1.
- Time slot $k+2$: One packet arrives at node b_2.
- $\cdots$
- Time slot $2k$: One packet arrives at node b_k.

Under dynamics (7), each packet arrives on its own slot and traverses all links of its path to exit on the same slot it arrived. The queue backlog in each node is 0 for all time.

Under dynamics (8), the first packet arrives at time slot 1 to node a_1. This packet visits node a_2 at time slot 2, when the second packet also arrives at a_2. One of these packets is delivered to node a_3 at time slot 3, and another packet also arrives to node 3. The nodes $\{1, \ldots, k\}$ do not have any exogenous arrivals and act only to delay the delivery of all packets from the a_i nodes. It follows that the link from node k to node 0 will send exactly one packet over each slot $t \in \{2k+1, 2k+2, \ldots, 2k+k\}$. Similarly, the link from b_k to 0 sends exactly one packet to node 0 over each of these same slots. Thus, node 0 receives 2 packets on each slot $t \in \{2k+1, 2k+2, \ldots, 2k+k\}$, but can only output 1 packet per slot. The queue backlog in this node grows linearly and reaches $k+1$ at time $2k+k$. Thus, the backlog in node 0 can be arbitrarily large when k is large. This example demonstrates that, even when there is only one destination, the deviation between virtual queues under dynamics (7) and actual queues under dynamics (8) can be arbitrarily large, even with an out-degree of at most 1 and an in-degree of at most 2.

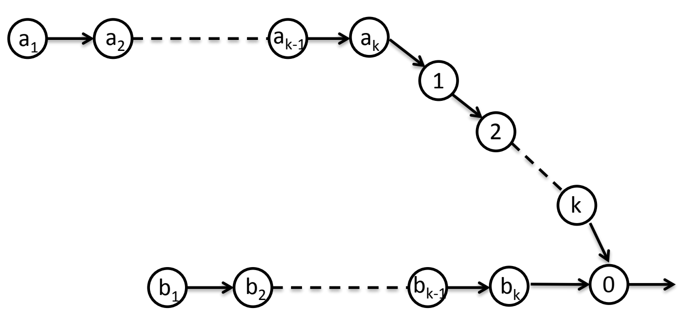

Fig. 5. An example illustrating the possibly large gap between queue model (7) and queue model (8)

Appendix C
Proof of Part (2) in Lemma 1

Fix $f \in \mathcal{F}, n \in \mathcal{N} \setminus \{\mathrm{Dst}(f)\}$. By (10),

$$\begin{aligned}
&\widehat{Q}_n^{(f)}[t+1] \\
&= \max\Big\{\widehat{Q}_n^{(f)}[t] - \sum_{l\in\mathcal{O}(n)} \mu_l^{(f)}[t], 0\Big\} + x_f[t]\mathbf{1}_{\{n=\mathrm{Src}(f)\}} \\
&\quad + \sum_{l\in\mathcal{I}(n)} \mu_l^{(f)}[t] \\
&= \max\Big\{\widehat{Q}_n^{(f)}[t] + x_f[t]\mathbf{1}_{\{n=\mathrm{Src}(f)\}} + \sum_{l\in\mathcal{I}(n)} \mu_l^{(f)}[t] \\
&\qquad - \sum_{l\in\mathcal{O}(n)} \mu_l^{(f)}[t], \quad x_f[t]\mathbf{1}_{\{n=\mathrm{Src}(f)\}} + \sum_{l\in\mathcal{I}(n)} \mu_l^{(f)}[t]\Big\} \\
&\overset{(a)}{\geq} \max\Big\{\widehat{Q}_n^{(f)}[t] + x_f[t]\mathbf{1}_{\{n=\mathrm{Src}(f)\}} + \sum_{l\in\mathcal{I}(n)} \mu_l^{(f)}[t] \\
&\qquad - \sum_{l\in\mathcal{O}(n)} \mu_l^{(f)}[t], \quad 0\Big\}
\end{aligned}$$

where (a) follows from the fact that $\mu_l^{(f)}[t], x_f[t], \forall f, l, t$ are non-negative. Note that the right side of the above equation is identical to the right side of (7) except that $Y_n^{(f)}(t)$ is rewritten as $\widehat{Q}_n^{(f)}(t)$. Since $Y_n^{(f)}(0) = 0 < \widehat{Q}_n^{(f)}(0)$, by induction, we have $Y_n^{(f)}[t] \leq \widehat{Q}_n^{(f)}[t], \forall t$. Since $\widehat{Q}_n^{(f)}[t] = Q_n^{(f)}[t] + B + \sum_{l\in\mathcal{O}(n)} C_l, \forall t$ and $Q_n^{(f)}[t] \leq B, \forall t$, we have $\widehat{Q}_n^{(f)}[t] \leq 2B + \sum_{l\in\mathcal{O}(n)} C_l, \forall t$. It follows that $Y_n^{(f)}[t] \leq 2B + \sum_{l\in\mathcal{O}(n)} C_l, \forall t$.

Appendix D
Proof of Lemma 3

Note that problem (18)-(20) satisfies Slater's condition. So the optimal solution to problem (18)-(20) is characterized by KKT conditions [34]. Introducing Lagrange multipliers $\theta \in \mathbb{R}_+$ for inequality constraint $\sum_{k=1}^K z_k \leq b$ and $\boldsymbol{\nu} = [\nu_1, \ldots, \nu_K]^\mathsf{T} \in \mathbb{R}_+^K$ for inequality constraints $z_k \geq 0, k \in \{1, 2, \ldots, K\}$. Let $\mathbf{z}^* = [z_1^*, \ldots, z_K^*]^\mathsf{T}$ and $(\theta^*, \boldsymbol{\nu}^*)$ be any primal and dual pair with the zero duality gap. By KKT conditions, we have $z_k^* - a_k + \theta^* - \nu_k^* = 0, \forall k \in \{1, 2, \ldots, K\}; \sum_{k=1}^K z_k^* \leq b; \theta^* \geq 0; \theta^*\big(\sum_{k=1}^K z_k^* - b\big) = 0; z_k^* \geq 0, \forall k \in \{1, 2, \ldots, K\}; \nu_k^* \geq 0, \forall k \in \{1, 2, \ldots, K\}; \nu_k^* z_k^* = 0, \forall k \in \{1, 2, \ldots, K\}$.

Eliminating $\nu_k^*, \forall k \in \{1, 2, \ldots, K\}$ in all equations yields $\theta^* \geq a_k - z_k^*, k \in \{1, 2, \ldots, K\}; \sum_{k=1}^K z_k^* \leq b; \theta^* \geq 0; \theta^*\big(\sum_{k=1}^K z_k^* - b\big) = 0; z_k^* \geq 0, \forall k \in \{1, 2, \ldots, K\}; (z_k^* - a_k + \theta^*) z_k^* = 0, \forall k \in \{1, 2, \ldots, K\}$.

For all $k \in \{1, 2, \ldots, K\}$, we consider $\theta^* < a_k$ and $\theta^* \geq a_k$ separately:

1) If $\theta^* < a_k$, then $\theta^* \geq a_k - z_k^*$ holds only when $z_k^* > 0$, which by $(z_k^* - a_k + \theta^*) z_k^* = 0$ implies that $z_k^* = a_k - \theta^*$.
2) If $\theta^* \geq a_k$, then $z_k^* > 0$ is impossible, because $z_k^* > 0$ implies that $z_k^* - a_k + \theta^* > 0$, which together with $z_k^* > 0$ contradicts the slackness condition $(z_k^* - a_k + \theta^*) z_k^* = 0$. Thus, if $\theta^* \geq a_k$, we must have $z_k^* = 0$.

Summarizing both cases, we have $z_k^* = \max\{0, a_k - \theta^*\}, \forall k \in \{1, 2, \ldots, K\}$, where θ^* is chosen such that $\sum_{k=1}^K z_k^* \leq b, \theta^* \geq 0$ and $\theta^*\big(\sum_{k=1}^K z_k^* - b\big) = 0$.

To find such θ^*, we first check if $\theta^* = 0$. If $\theta^* = 0$ is true, the slackness condition $\theta^*\big(\sum_{k=1}^K z_k^* - b\big)$ is guaranteed to hold and we need to further require $\sum_{k=1}^{K} z_k^* = \sum_{k=1}^{K} \max\{0, a_k\} \leq b$. Thus $\theta^* = 0$ if and only if $\sum_{k=1}^{K} \max\{0, a_k\} \leq b$. Thus, Algorithm 2 check if $\sum_{k=1}^{K} \max\{0, a_k\} \leq b$ holds at the first step and if this is true, then we conclude $\theta^* = 0$ and we are done!

Otherwise, we know $\theta^* > 0$. By the slackness condition $\theta^*\big(\sum_{k=1}^{K} z_k^* - b\big) = 0$, we must have $\sum_{k=1}^{K} z_k^* = \sum_{k=1}^{K} \max\{0, a_k - \theta^*\} = b$. To find $\theta^* > 0$ such that $\sum_{k=1}^{K} \max\{0, a_k - \theta^*\} = b$, we could apply a bisection search by noting that all z_k^* are decreasing with respect to θ^*.

Another algorithm of finding θ^* is inspired by the observation that if $a_j \geq a_i, \forall i, j \in \{1, 2, \ldots, K\}$, then $z_j^* \geq z_i^*$. Thus, we first sort all a_k in a decreasing order, say π is the permutation such that $a_{\pi(1)} \geq a_{\pi(2)} \geq \cdots \geq a_{\pi(K)}$; and then sequentially check if $k \in \{1, 2, \ldots, K\}$ is the index such that $a_{\pi(k)} - \theta^* \geq 0$ and $a_{\pi(k+1)} - \theta^* < 0$. To check this, we first assume k is indeed such an index and solve the equation $\sum_{j=1}^{k}(a_{\pi(j)} - \theta^*) = b$ to obtain θ^*; (Note that in Algorithm 2, to avoid recalculating the partial sum $\sum_{j=1}^{k} a_{\pi(j)}$ for each k, we introduce the parameter $S_k = \sum_{j=1}^{k} a_{\pi(j)}$ and update S_k incrementally. By doing this, the complexity of each iteration in the loop is only $O(1)$.) then verify the assumption by checking if $\theta^* \geq 0$, $a_{\pi(k)} - \theta^* \geq 0$ and $a_{\pi(k+1)} - \theta^* \leq 0$. The algorithm is described in Algorithm 2 and has complexity $O(K \log(K))$. The overall complexity is dominated by the step of sorting all a_k.

Appendix E
Proof of Lemma 6

The objective function (29) can be rewritten as

$$
\begin{aligned}
& h(\mathbf{y}) - \sum_{\substack{f \in \mathcal{F}, \\ n \in \mathcal{N} \setminus \mathrm{Dst}(f)}} \Big(W_n^{(f)}[t] g_n^{(f)}(\mathbf{y}_n^{(f)}) \\
& \qquad\qquad + \alpha_n \|\mathbf{y}_n^{(f)} - \mathbf{y}_n^{(f)}[t-1]\|^2 \Big) \\
& - \sum_{f \in \mathcal{F}, n = \mathrm{Dst}(f)} \alpha_n \sum_{l \in \mathcal{I}(n)} (\mu_l^{(f)} - \mu_l^{(f)}[t-1])^2 \\
\overset{(a)}{=} & \sum_{f \in \mathcal{F}} U_f(x_f) \\
& - \sum_{\substack{f \in \mathcal{F}, \\ n \in \mathcal{N} \setminus \mathrm{Dst}(f)}} W_n^{(f)}[t]\big(x_f \mathbf{1}_{\{n = \mathrm{Src}(f)\}} + \textstyle\sum_{l \in \mathcal{I}(n)} \mu_l^{(f)} - \sum_{l \in \mathcal{O}(n)} \mu_l^{(f)}\big) \\
& - \sum_{\substack{f \in \mathcal{F}, \\ n \in \mathcal{N} \setminus \mathrm{Dst}(f)}} \alpha_n (x_f - x_f[t-1])^2 \mathbf{1}_{\{n = \mathrm{Src}(f)\}} \\
& - \sum_{\substack{f \in \mathcal{F}, \\ n \in \mathcal{N} \setminus \mathrm{Dst}(f)}} \alpha_n \sum_{l \in \mathcal{I}(n)} (\mu_l^{(f)} - \mu_l^{(f)}[t-1])^2 \\
& - \sum_{\substack{f \in \mathcal{F}, \\ n \in \mathcal{N} \setminus \mathrm{Dst}(f)}} \alpha_n \sum_{l \in \mathcal{O}(n)} (\mu_l^{(f)} - \mu_l^{(f)}[t-1])^2 \\
& - \sum_{f \in \mathcal{F}, n = \mathrm{Dst}(f)} \alpha_n \sum_{l \in \mathcal{I}(n)} (\mu_l^{(f)} - \mu_l^{(f)}[t-1])^2 \\
\overset{(b)}{=} & \sum_{f \in \mathcal{F}} \big(U_f(x_f) - W_{\mathrm{Src}(f)}^{(f)}[t] x_f - \alpha_{\mathrm{Src}(f)} (x_f - x_f[t-1])^2 \big) \\
& + \sum_{(n,m) \in \mathcal{L}} \sum_{f \in \mathcal{F}} \big(W_n^{(f)}[t] - W_m^{(f)}[t] \big) \mu_{(n,m)}^{(f)} \\
& - \sum_{(n,m) \in \mathcal{L}} (\alpha_n + \alpha_m) \sum_{f \in \mathcal{F}} (\mu_{(n,m)}^{(f)} - \mu_{(n,m)}^{(f)}[t-1])^2
\end{aligned} \tag{32}
$$

where (a) follows from the fact that $\|\mathbf{y}_n^{(f)} - \mathbf{y}_n^{(f)}[t-1]\|^2 = (x_f - x_f[t-1])^2 \mathbf{1}_{\{n = \mathrm{Src}(f)\}} + \sum_{l \in \mathcal{I}(n)} (\mu_l^{(f)} - \mu_l^{(f)}[t-1])^2 + \sum_{l \in \mathcal{O}(n)} (\mu_l^{(f)} - \mu_l^{(f)}[t-1])^2$; and (b) follows by collecting each linear term $\mu_l^{(f)}$ and each quadratic term $(\mu_l^{(f)} - \mu_l^{(f)}[t-1])^2$. In the last step (b), the quadratic term $\sum_{(n,m) \in \mathcal{L}} (\alpha_n + \alpha_m) \sum_{f \in \mathcal{F}} (\mu_{(n,m)}^{(f)} - \mu_{(n,m)}^{(f)}[t-1])^2$ is obtained by simply collecting each quadratic term $(\mu_l^{(f)} - \mu_l^{(f)}[t-1])^2$ in the last 3 terms in step (a); and the linear term $\sum_{(n,m) \in \mathcal{L}} \sum_{f \in \mathcal{F}} \big(W_n^{(f)}[t] - W_m^{(f)}[t]\big) \mu_{(n,m)}^{(f)}$ is obtained by noting each link session rate $\mu_l^{(f)}$ appears twice with opposite signs in the summation term $\sum_{f \in \mathcal{F}, n \in \mathcal{N} \setminus \{\mathrm{Dst}(f)\}} W_n^{(f)}[t]\big(x_f \mathbf{1}_{\{n = \mathrm{Src}(f)\}} + \sum_{l \in \mathcal{I}(n)} \mu_l^{(f)} - \sum_{l \in \mathcal{O}(n)} \mu_l^{(f)}\big)$ unless link l flows into $\mathrm{Dst}(f)$ and recalling that $W_{\mathrm{Dst}(f)}^{(f)} = 0, \forall f \in \mathcal{F}$.

Note that equation (32) is now separable for each scalar x_f and vector $[\mu_{(n,m)}^{(f)}]_{f \in \mathcal{F}}$. Thus, problem (29)-(30) can be decomposed into independent smaller optimization problems in the form of problem (12)-(13) with respect to x_f, and in the form of problem (14)-(17) with respect to $\mu_{(n,m)}^{(f)}, \forall f \in \mathcal{F}$.

Appendix F
Proof of Lemma 7

Definition 2 (Lipschitz Continuity): Let $\mathcal{Z} \subseteq \mathbb{R}^n$ be a convex set. Function $f : \mathcal{Z} \to \mathbb{R}^m$ is said to be Lipschitz continuous on $\mathcal{Z}$ with modulus β if there exists $\beta > 0$ such that $\|f(\mathbf{z}_1) - f(\mathbf{z}_2)\| \leq \beta \|\mathbf{z}_1 - \mathbf{z}_2\|$ for all $\mathbf{z}_1, \mathbf{z}_2 \in \mathcal{Z}$.

The following fact summarizes the Lipschitz continuity of each function $g_n^{(f)}(\cdot)$.

Fact 2: Each function $g_n^{(f)}(\cdot)$ defined in (22) is Lipschitz continuous with respect to vector $\mathbf{y}_n^{(f)}$ with modulus

$$\beta_n \leq \sqrt{d_n + 1}.$$

where d_n is the degree of node n.

Proof: This fact can be easily shown by noting that each $g_n^{(f)}(\mathbf{y}_n^{(f)})$ is a linear function with respect to vector $\mathbf{y}_n^{(f)}$ and has at most $d_n + 1$ non-zero coefficients that are equal to ± 1. ■

Now, we are ready to present the main proof.

Note that $W_n^{(f)}[t]$ appears as a known constant in (12). Since $U_f(x_f)$ is concave and $W_n^{(f)}[t] x_f$ is linear, it follows that (12) is strongly concave with respect to x_f with modulus $2\alpha_n$. Since $x_f[t]$ is chosen to solve (12)-(13), by Lemma 4, $\forall f \in \mathcal{F}$, we have

$$
\begin{aligned}
& \underbrace{U_f(x_f[t]) - W_{\mathrm{Src}(f)}^{(f)}[t] x_f[t] - \alpha_n (x_f[t] - x_f[t-1])^2}_{\text{(33)-left}} \\
& \geq \underbrace{U_f(x_f^*) - W_{\mathrm{Src}(f)}^{(f)}[t] x_f^* - \alpha_n (x_f^* - x_f[t-1])^2 + \alpha_n (x_f^* - x_f[t])^2}_{\text{(33)-right}}.
\end{aligned} \tag{33}
$$

Similarly, we know (14) is strongly concave with respect to vector $[\mu^f_{(n,m)}]_{f\in\mathcal{F}}$ with modulus $2(\alpha_n+\alpha_m)$. By Lemma 4, $\forall(n,m)\in\mathcal{O}(n)$, we have

$$\underbrace{\begin{array}{l}\sum_{f\in\mathcal{F}}\left(W_n^{(f)}[t]-W_m^{(f)}[t]\right)\mu_{(n,m)}^{(f)}[t]\\ -\left(\alpha_n+\alpha_m\right)\sum_{f\in\mathcal{F}}\left(\mu_{(n,m)}^{(f)}[t]-\mu_{(n,m)}^{(f)}[t-1]\right)^2\end{array}}_{\text{(34)-left}}$$
$$\geq \underbrace{\begin{array}{l}\sum_{f\in\mathcal{F}}\left(W_n^{(f)}[t]-W_m^{(f)}[t]\right)\mu_{(n,m)}^{(f),*}\\ -\left(\alpha_n+\alpha_m\right)\sum_{f\in\mathcal{F}}\left(\mu_{(n,m)}^{(f),*}-\mu_{(n,m)}^{(f)}[t-1]\right)^2\\ +\left(\alpha_n+\alpha_m\right)\sum_{f\in\mathcal{F}}\left(\mu_{(n,m)}^{(f),*}-\mu_{(n,m)}^{(f)}[t]\right)^2\end{array}}_{\text{(34)-right}} \tag{34}$$

Recall that each column vector $\mathbf{y}_n^{(f)}$ defined in (21) is composed by control actions that appear in each constraint (2); column vector $\mathbf{y}=[x_f;\mu_l^{(f)}]_{f\in\mathcal{F},l\in\mathcal{L}}$ is the collection of all control actions; and $h(\mathbf{y})=\sum_{f\in\mathcal{F}}U_f(x_f)$. Summing term (33)-left over all $f\in\mathcal{F}$ and term (34)-left over all $(n,m)\in\mathcal{L}$ and using an argument similar to the proof of Lemma 6 (Recall that $\mathbf{y}[t]$ jointly chosen in Algorithm 1 is to minimize (29) by Lemma 6) yields

$$\begin{aligned}&\sum_{f\in\mathcal{F}}\text{(33)-left}+\sum_{(n,m)\in\mathcal{N}}\text{(34)-left}\\ =&h(\mathbf{y}[t])-\sum_{\substack{f\in\mathcal{F},\\ n\in\mathcal{N}\setminus\mathrm{Dst}(f)}}\Big(W_n^{(f)}[t]g_n^{(f)}(\mathbf{y}_n^{(f)}[t])\\ &\qquad+\alpha_n\|\mathbf{y}_n^{(f)}[t]-\mathbf{y}_n^{(f)}[t-1]\|^2\Big)\\ &-\sum_{\substack{f\in\mathcal{F},\\ n=\mathrm{Dst}(f)}}\alpha_n\sum_{l\in\mathcal{I}(n)}(\mu_l^{(f)}[t]-\mu_l^{(f)}[t-1])^2.\end{aligned} \tag{35}$$

Recall that $\Phi[t]=\sum_{f\in\mathcal{F},n\in\mathcal{N}}\left(\alpha_n\mathbf{1}_{\{n\neq\mathrm{Dst}(f)\}}\|\mathbf{y}_n^{(f),*}-\mathbf{y}_n^{(f)}[t]\|^2+\alpha_n\mathbf{1}_{\{n=\mathrm{Dst}(f)\}}\sum_{l\in\mathcal{I}(n)}(\mu_l^{(f),*}-\mu_l^{(f)}[t])^2\right)$. Summing term (33)-right over all $f\in\mathcal{F}$ and term (34)-right over all $(n,m)\in\mathcal{L}$ yields

$$\begin{aligned}&\sum_{f\in\mathcal{F}}\text{(33)-right}+\sum_{(n,m)\in\mathcal{N}}\text{(34)-right}\\ =&h(\mathbf{y}^*)+\Phi[t]-\Phi[t-1]\\ &-\sum_{f\in\mathcal{F}}\sum_{n\in\mathcal{N}\setminus\{\mathrm{Dst}(f)\}}W_n^{(f)}[t]g_n^{(f)}(\mathbf{y}_n^{(f),*}),\end{aligned} \tag{36}$$

Combining (33)-(36) and rearranging terms yields

$$\begin{aligned}&h(\mathbf{y}[t])\\ \geq&h(\mathbf{y}^*)+\Phi[t]-\Phi[t-1]-\sum_{\substack{f\in\mathcal{F},\\ n\in\mathcal{N}\setminus\mathrm{Dst}(f)}}W_n^{(f)}[t]g_n^{(f)}(\mathbf{y}_n^{(f),*})\\ &+\sum_{\substack{f\in\mathcal{F},\\ n\in\mathcal{N}\setminus\mathrm{Dst}(f)}}\Big(W_n^{(f)}[t]g_n^{(f)}(\mathbf{y}_n^{(f)}[t])\\ &\qquad+\alpha_n\|\mathbf{y}_n^{(f)}[t]-\mathbf{y}_n^{(f)}[t-1]\|^2\Big)\\ &+\sum_{\substack{f\in\mathcal{F},\\ n=\mathrm{Dst}(f)}}\alpha_n\sum_{l\in\mathcal{I}(n)}(\mu_l^{(f)}[t]-\mu_l^{(f)}[t-1])^2\\ \overset{(a)}{\geq}&h(\mathbf{y}^*)+\Phi[t]-\Phi[t-1]\\ &+\sum_{\substack{f\in\mathcal{F},\\ n\in\mathcal{N}\setminus\mathrm{Dst}(f)}}\Big(W_n^{(f)}[t]g_n^{(f)}(\mathbf{y}_n^{(f)}[t])\\ &\qquad+\alpha_n\|\mathbf{y}_n^{(f)}[t]-\mathbf{y}_n^{(f)}[t-1]\|^2\Big)\\ \overset{(b)}{=}&h(\mathbf{y}^*)+\Phi[t]-\Phi[t-1]\\ &+\sum_{\substack{f\in\mathcal{F},\\ n\in\mathcal{N}\setminus\mathrm{Dst}(f)}}\Big(Q_n^{(f)}[t]g_n^{(f)}(\mathbf{y}_n^{(f)}[t])\\ &\qquad+g_n^{(f)}(\mathbf{y}_n^{(f)}[t-1])g_n^{(f)}(\mathbf{y}_n^{(f)}[t])\\ &\qquad+\alpha_n\|\mathbf{y}_n^{(f)}[t]-\mathbf{y}_n^{(f)}[t-1]\|^2\Big)\end{aligned} \tag{37}$$

where (a) follows from the fact that $g_n^{(f)}(\mathbf{y}_n^{(f),*})=0$ and $\sum_{f\in\mathcal{F},n=\mathrm{Dst}(f)}\alpha_n\sum_{l\in\mathcal{I}(n)}(\mu_l^{(f)}[t]-\mu_l^{(f)}[t-1])^2\geq 0$; (b) follows from the fact that $W_n^{(f)}[t]=Q_n^{(f)}[t]+g_n^{(f)}(\mathbf{y}_n^{(f)}[t-1])$.

Recall that $u_1u_2=\frac{1}{2}u_1^2+\frac{1}{2}u_2^2-\frac{1}{2}(u_1-u_2)^2$ for any $u_1,u_2\in\mathbb{R}$. Thus, for all $f\in\mathcal{F},n\in\mathcal{N}\setminus\mathrm{Dst}(f)$, we have

$$\begin{aligned}&g_n^{(f)}(\mathbf{y}_n^{(f)}[t-1])g_n^{(f)}(\mathbf{y}_n^{(f)}[t])\\ =&\frac{1}{2}\big(g_n^{(f)}(\mathbf{y}_n^{(f)}[t-1])\big)^2+\frac{1}{2}\big(g_n^{(f)}(\mathbf{y}_n^{(f)}[t])\big)^2\\ &-\frac{1}{2}\big(g_n^{(f)}(\mathbf{y}_n^{(f)}[t-1])-g_n^{(f)}(\mathbf{y}_n^{(f)}[t])\big)^2.\end{aligned} \tag{38}$$

Substituting (38) into (37) yields

$$\begin{aligned}&h(\mathbf{y}[t])\\ \geq&h(\mathbf{y}^*)+\Phi[t]-\Phi[t-1]\\ &+\sum_{\substack{f\in\mathcal{F},\\ n\in\mathcal{N}\setminus\mathrm{Dst}(f)}}\Big(Q_n^{(f)}[t]g_n^{(f)}(\mathbf{y}_n^{(f)}[t])+\frac{1}{2}\big(g_n^{(f)}(\mathbf{y}_n^{(f)}[t-1])\big)^2\\ &\qquad+\frac{1}{2}\big(g_n^{(f)}(\mathbf{y}_n^{(f)}[t])\big)^2\\ &\qquad-\frac{1}{2}\big(g_n^{(f)}(\mathbf{y}_n^{(f)}[t-1])-g_n^{(f)}(\mathbf{y}_n^{(f)}[t])\big)^2\\ &\qquad+\alpha_n\|\mathbf{y}_n^{(f)}[t]-\mathbf{y}_n^{(f)}[t-1]\|^2\Big)\\ \overset{(a)}{\geq}&h(\mathbf{y}^*)+\Phi[t]-\Phi[t-1]\\ &+\sum_{\substack{f\in\mathcal{F},\\ n\in\mathcal{N}\setminus\mathrm{Dst}(f)}}\Big(Q_n^{(f)}[t]g_n^{(f)}(\mathbf{y}_n^{(f)}[t])+\frac{1}{2}\big(g_n^{(f)}(\mathbf{y}_n^{(f)}[t-1])\big)^2\\ &\qquad+\frac{1}{2}\big(g_n^{(f)}(\mathbf{y}_n^{(f)}[t])\big)^2\\ &\qquad+\big(\alpha_n-\frac{1}{2}\beta_n^2\big)\|\mathbf{y}_n^{(f)}[t]-\mathbf{y}_n^{(f)}[t-1]\|^2\Big)\\ \overset{(b)}{\geq}&h(\mathbf{y}^*)+\Phi[t]-\Phi[t-1]\\ &+\sum_{\substack{f\in\mathcal{F},\\ n\in\mathcal{N}\setminus\mathrm{Dst}(f)}}\Big(Q_n^{(f)}[t]g_n^{(f)}(\mathbf{y}_n^{(f)}[t])+\frac{1}{2}\big(g_n^{(f)}(\mathbf{y}_n^{(f)}[t])\big)^2\Big)\end{aligned} \tag{39}$$

where (a) follows from the Fact 2, i.e., each $g_n^{(f)}(\cdot)$ is Lipschitz with modulus β_n and (b) follows from the fact that $\alpha_n\geq\frac{1}{2}(d_n+1)$, $\beta_n\leq\sqrt{d_n+1}$ and $\frac{1}{2}\big(g_n^{(f)}(\mathbf{y}_n^{(f)}[t-1])\big)^2\geq 0$.

Substituting (26) into (39) yields

$$h(\mathbf{y}[t])\geq h(\mathbf{y}^*)+\Phi[t]-\Phi[t-1]+\Delta[t].$$

References

[1] H. Yu and M. J. Neely, "A new backpressure algorithm for joint rate control and routing with vanishing utility optimality gaps and finite queue lengths," in *Proceedings of IEEE International Conference on Computer Communications (INFOCOM)*, 2017.

[2] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Transactions on Automatic Control*, vol. 37, no. 12, pp. 1936–1948, 1992.

[3] M. J. Neely, "Dynamic power allocation and routing for satellite and wireless networks with time varying channels," Ph.D. dissertation, Massachusetts Institute of Technology, 2003.

[4] L. Georgiadis, M. J. Neely, and L. Tassiulas, "Resource allocation and cross-layer control in wireless networks," *Foundations and Trends in Networking*, 2006.

[5] M. J. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*. Morgan & Claypool Publishers, 2010.

[6] 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*, vol. 24, no. 8, pp. 1514–1524, 2006.

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

[8] X. Lin and N. B. Shroff, "Joint rate control and scheduling in multihop wireless networks," in *Proceedings of IEEE Conference on Decision and Control (CDC)*, 2004.

[9] J.-W. Lee, R. R. Mazumdar, and N. B. Shroff, "Opportunistic power scheduling for dynamic multi-server wireless systems," *IEEE Transactions on Wireless Communications*, vol. 5, no. 6, pp. 1506–1515, 2006.

[10] 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*, vol. 23, no. 1, pp. 89–103, 2005.

[11] J. Liu, N. B. Shroff, C. H. Xia, and H. D. Sherali, "Joint congestion control and routing optimization: An efficient second-order distributed approach," *IEEE/ACM Transactions on Networking*, vol. 24, no. 3, pp. 1404–1420, 2015.

[12] M. J. Neely, "Super-fast delay tradeoffs for utility optimal fair scheduling in wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, pp. 1489–1501, 2006.

[13] L. Huang, S. Moeller, M. J. Neely, and B. Krishnamachari, "LIFO-backpressure achieves near-optimal utility-delay tradeoff," *IEEE/ACM Transactions on Networking*, vol. 21, no. 3, pp. 831–844, 2013.

[14] R. A. Berry and R. G. Gallager, "Communication over fading channels with delay constraints," *IEEE Transactions on Information Theory*, vol. 48, no. 5, pp. 1135–1149, 2002.

[15] M. J. Neely, "Optimal energy and delay tradeoffs for multiuser wireless downlinks," *IEEE Transactions on Information Theory*, vol. 53, no. 9, pp. 3095–3113, 2007.

[16] A. Eryilmaz and R. Srikant, "Asymptotically tight steady-state queue length bounds implied by drift conditions," *Queueing Systems*, vol. 72, pp. 311–359, 2012.

[17] M. J. Neely, "Energy-aware wireless scheduling with near-optimal backlog and convergence time tradeoffs," *IEEE/ACM Transactions on Networking*, vol. 24, no. 4, pp. 2223–2236, 2016.

[18] E. Wei, A. Ozdaglar, and A. Jadbabaie, "A distributed Newton method for network utility maximization–I: algorithm," *IEEE Transactions on Automatic Control*, vol. 58, no. 9, pp. 2162–2175, 2013.

[19] M. Zargham, A. Ribeiro, and A. Jadbabaie, "Accelerated backpressure algorithm," in *Proceedings of IEEE Global Communications Conference (GLOBECOM)*, 2013.

[20] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 123–231, 2013.

[21] X. Lin and N. B. Shroff, "Utility maximization for communication networks with multipath routing," *IEEE Transactions on Automatic Control*, vol. 51, no. 5, pp. 766–781, 2006.

[22] X. Lin, "Simplification of network dynamics in large systems," Ph.D. dissertation, Purdue University, 2005.

[23] H. Yu and M. J. Neely, "A simple parallel algorithm with an $O(1/t)$ convergence rate for general convex programs," *SIAM Journal on Optimization*, vol. 27, no. 2, pp. 759–783, 2017.

[24] ——, "A new backpressure algorithm for joint rate control and routing with vanishing utility optimality gaps and finite queue lengths," *arXiv:1701.04519*, 2017.

[25] S. Wang and N. Shroff, "Towards fast-convergence, low-delay and low-complexity network optimization," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, Dec. 2017.

[26] B. He and X. Yuan, "On non-ergodic convergence rate of Douglas-Rachford alternating direction method of multipliers," *Numerische Mathematik*, vol. 130, no. 3, pp. 567–577, 2015.

[27] W. Deng and W. Yin, "On the global and linear convergence of the generalized alternating direction method of multipliers," *Journal of Scientific Computing*, vol. 66, no. 3, pp. 889–916, 2016.

[28] D. Han, D. Sun, and L. Zhang, "Linear rate convergence of the alternating direction method of multipliers for convex composite programming," *Mathematics of Operations Research*, 2017.

[29] F. P. Kelly, A. K. Maulloo, and D. K. Tan, "Rate control for communication networks: Shadow prices, proportional fairness and stability," *Journal of the Operational Research Society*, vol. 49, no. 3, pp. 237–252, 1998.

[30] S. H. Low and D. E. Lapsley, "Optimization flow control—I: basic algorithm and convergence," *IEEE/ACM Transactions on Networking*, vol. 7, no. 6, pp. 861–874, 1999.

[31] D. P. Bertsekas, A. Nedić, and A. E. Ozdaglar, *Convex Analysis and Optimization*. Athena Scientific, 2003.

[32] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra, "Efficient projections onto the l_1-ball for learning in high dimensions," in *Proceedings of International Conference on Machine Learning (ICML)*, 2008.

[33] M. J. Neely, "A simple convergence time analysis of drift-plus-penalty for stochastic optimization and convex programs," *arXiv:1412.0791*, 2014.

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