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
University of Southern California

***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 a parameter V in the algorithm, 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 $[O(1/V), O(V)]$ 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, the existing backpressure algorithms necessarily yield an arbitrarily large queue length. 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.**

I. Introduction

In multi-hop data networks, the problem of joint rate control and routing is to accept data into the network and to make routing decisions at each node such that certain utilities are maximized and all accepted data are delivered to intended destinations without overflowing any queue in intermediate nodes. The original backpressure algorithm proposed in the seminal work [1] 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 [1], 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 data network with both utility maximization and queue stability considerations [2], [3], [4]. Alternative extensions for both utility maximization and queue stabilization are developed in [5], [6], [7], [8]. 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 [9], [5], [7], [10] 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 [5] achieves an $O(1/V)$ utility optimality gap with an $O(V^2)$ queue length, where V is an algorithm parameter. By controlling parameter V, a small utility optimality gap is available only at the cost of a large queue length. The drift-plus-penalty backpressure algorithm [4], which has the best utility-delay tradeoff among all existing first order Lagrangian dual type methods for general networks, can only achieve an $O(1/V)$ utility optimality gap with an $O(V)$ queue length. Under certain restrictive assumptions over the network, a better $[O(1/V), O(\log(V))]$ tradeoff is achieved via an exponential Lyapunov function in [11], and an $[O(1/V), O(\log^2(V))]$ tradeoff is achieved via a LIFO-backpressure algorithm in [12]. The existing utility-delay tradeoff results seem to suggest that a larger queueing delay is unavoidable if a smaller utility optimality gap is demanded.

Recently, there have been many attempts in obtaining new variations of backpressure algorithms by applying Newton's method to the Lagrangian dual function. In the recent work [10], the authors develop a Newton's method for joint rate control and routing. However, the utility-delay tradeoff in [10] is still $[O(1/V), O(V^2)]$; and the algorithm requires a centralized projection step (although Newton directions can be approximated in a distributed manner). Work [13] 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 [14] 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 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 [4], [5], [7] but has a better utility-delay tradeoff.

Hao Yu and Michael J. Neely are with the Department of Electrical Engineering, University of Southern California, Los Angeles, USA.

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 $[O(1/V), O(V^2)]$ utility-delay tradeoff in [5] and the $[O(1/V), O(V)]$ utility-delay tradeoff of the drift-plus-penalty algorithm in [4], both of which yield an unbounded queue length to have a vanishing utility optimality gap. 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 [15].

Note that the benefit of introducing a quadratic term in network optimization has been observed in [16]. Work [16] considers network utility maximization with given routing paths that is a special case of the problem treated in this paper. The algorithm of [16] 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 [16] is not a backpressure type and hence is fundamentally different from ours. For example, the algorithm in [16] needs to update the primal variables (source session rates for each path) at least twice per iteration, while our algorithm only updates the primal variables (source session rates and link session rates) once per iteration. The prior work [16] shows that the utility optimality gap is asymptotically zero without analyzing the decay rate, while this paper shows the utility optimality gap decays like $O(1/t)$.

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 $|\mathcal{N}|$ and $|\mathcal{L}|$ be the respective cardinalities of $\mathcal{N}$ and $\mathcal{L}$. 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 $\text{Src}(f)$ 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 [5], [10] 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. It is assumed throughout that either $\text{dom}(U_f) = [0, \infty)$ or $\text{dom}(U_f) = (0, \infty)$, the latter being important for *proportionally fair utilities* [17] $U_f(x) = \log(x)$ that have singularities at $x = 0$.

Denote the capacity of link l as C_l and assume it is a fixed and positive constant.[1] 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 equalities with inequalities, 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 [18], 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 [16] where each flow is restricted to using links from a set of predefined paths can be treated by modifying the sets $\mathcal{S}_l$ accordingly.

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

[1] As stated in [10], this is a suitable model for wireline networks and wireless networks with fixed transmission power and orthogonal channels.

network topology and the solution is a centralized one, which is not practical for large data networks. As observed in [9], [5], [7], [10], 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

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

where $q(\boldsymbol{\lambda}) = \sup_{[x_f;\mu_l^{(f)}]\in\mathcal{C}} \big\{ \sum_{f\in\mathcal{F}} U_f(x_f) - \sum_{f\in\mathcal{F}}\sum_{n\in\mathcal{N}\setminus\{\text{Dst}(f)\}} \lambda_n^{(f)} \big[x_f \mathbf{1}_{\{n=\text{Src}(f)\}} + \sum_{l\in\mathcal{I}(n)} \mu_l^{(f)} - \sum_{l\in\mathcal{O}(n)} \mu_l^{(f)}\big]\big\}$ 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 ensures Assumption 2. Since constraints (2)-(6) are linear, Proposition 6.4.2 in [19] guarantees that Lagrange multipliers exist when constraints (2)-(6) are feasible and 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 $\widetilde{U}_f : (-\epsilon, \infty) \mapsto \mathbb{R}$ such that $\widetilde{U}_f(x) = U_f(x)$ whenever $x \geq 0$.[2]

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. ∎

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.

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

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

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 $\sum_{l\in\mathcal{I}(n)} \mu_l^{(f)}[t]$ when those other nodes do 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 $Z_n^{(f)}[t] \leq \sum_{l\in\mathcal{O}(n)} \mu_l^{(f)}[t]$, 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

[2] 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 [19].

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 $|Q_n^{(f)}[t]| \leq B, \forall t$, then

1) $Z_n^{(f)}[t] \leq 2B + \sum_{l\in\mathcal{O}(n)} C_l$ for all $t \in \{0, 1, 2, \ldots\}$.
2) $Y_n^{(f)}[t] \leq 2B + \sum_{l\in\mathcal{O}(n)} C_l$ 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 $Q_n^{(f)}[t] \geq -B, \forall t$ by assumption, we have $\widehat{Q}_n^{(f)}[t] \geq \sum_{l\in\mathcal{O}(n)} C_l \geq \sum_{l\in\mathcal{O}(n)} \mu_l^{(f)}[t], \forall t$. This implies that $\widehat{Q}_n^{(f)}[t]$ also satisfies:
$$\widehat{Q}_n^{(f)}[t+1] = \max\big\{\widehat{Q}_n^{(f)}[t] - \sum_{l\in\mathcal{O}(n)} \mu_l^{(f)}[t], 0\big\} + x_f[t]\mathbf{1}_{\{n=\text{Src}(f)\}} + \sum_{l\in\mathcal{I}(n)} \mu_l^{(f)}[t], \forall t \tag{10}$$
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, $Z_n^{(f)}[t] \leq \widehat{Q}_n^{(f)}[t], \forall t$.
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 $Z_n^{(f)}[t] \leq 2B + \sum_{l\in\mathcal{O}(n)} C_l, \forall t$.
2) The proof is similar and is omitted for brevity.

■

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
$$\frac{1}{t}\sum_{\tau=0}^{t-1}\sum_{f\in\mathcal{F}} U_f(x_f[t]) \geq \sum_{f\in\mathcal{F}} U_f(x_f^\ast) - O(1/t), \forall t,$$
where $x_f^\ast$ are from the optimal solution to (1)-(6). Note that Jensen's inequality further implies that
$$\sum_{f\in\mathcal{F}} U_f\big(\frac{1}{t}\sum_{\tau=0}^{t-1} x_f[\tau]\big) \geq \sum_{f\in\mathcal{F}} U_f(x_f^\ast) - O(1/t), \forall t.$$

The new backpressure algorithm[3] 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 [15].

Algorithm 1 The New Backpressure Algorithm

Let $\alpha > 0$ be a constant parameter. 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 $W_n^{(f)}[t]$:
$$W_n^{(f)}[t] = Q_n^{(f)}[t] + x_f[t-1]\mathbf{1}_{\{n=\text{Src}(f)\}} + \sum_{l\in\mathcal{I}(n)} \mu_l^{(f)}[t-1] - \sum_{l\in\mathcal{O}(n)} \mu_l^{(f)}[t-1], \tag{11}$$
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 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
$$\max_{x_f} \quad U_f(x_f) - W_n^{(f)}[t]x_f - \alpha\big(x_f - x_f[t-1]\big)^2 \tag{12}$$
$$\text{s.t.} \quad x_f \in \text{dom}(U_f) \tag{13}$$
- 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:
$$\max_{\mu_{(n,m)}^{(f)}} \sum_{f\in\mathcal{F}} \big(W_n^{(f)}(t) - W_m^{(f)}(t)\big)\mu_{(n,m)}^{(f)} - \alpha\sum_{f\in\mathcal{F}}\big(\mu_{(n,m)}^{(f)} - \mu_{(n,m)}^{(f)}[t-1]\big)^2 \tag{14}$$
$$\text{s.t.} \quad \sum_{f\in\mathcal{F}} \mu_{(n,m)}^{(f)} \leq C_{(n,m)} \tag{15}$$
$$\mu_{(n,m)}^{(f)} \geq 0, \forall f \in \mathcal{S}_{(n,m)} \tag{16}$$
$$\mu_{(n,m)}^{(f)} = 0, \forall f \notin \mathcal{S}_{(n,m)} \tag{17}$$
- 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).

---

[3]Note that Algorithm 1 involves a global parameter α that should be chosen according to the number of total links and sessions in the network (see Section IV). In the extended version [20], we propose another backpressure algorithm only with local parameters that can be chosen by each individual node.

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 $h(x_f) = U_f'(x_f) - 2\alpha x_f + 2\alpha x_f[t-1] - W_n^{(f)}[t]$. If $h(0) < 0$, then $\hat{x}_f = 0$; otherwise $\hat{x}_f$ is the root to the equation $h(x_f) = 0$ 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:

$$\hat{x}_f = (2\alpha x_f[t-1] - W_n^{(f)}[t])/(4\alpha) + \sqrt{(W_n^{(f)}[t] - 2\alpha x_f[t-1])^2 + 8\alpha w_f}/(4\alpha)$$

Proof: The proof is easy and is omitted for brevity. ■

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

$$\min \quad \frac{1}{2}\sum_{k=1}^{K}(z_k - a_k)^2 \tag{18}$$

$$\text{s.t.} \quad \sum_{k=1}^{K} z_k \leq b \tag{19}$$

$$z_k \geq 0, \forall k \in \{1, 2, \ldots, K\} \tag{20}$$

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 or by Algorithm 2 with complexity $O(K \log K)$.

Proof: A similar problem where (19) is replaced with an equality constraint in considered in [21]. 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. The proof is omitted for brevity. ■

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. Basic Facts from Convex Analysis

For any vector $\mathbf{z} \in \mathbb{R}^n$, $\|\mathbf{z}\|$ represents its Euclidean norm.

Definition 1 (Lipschitz Continuity): Let $\mathcal{Z} \subseteq \mathbb{R}^n$ be a convex set. Function $h : \mathcal{Z} \to \mathbb{R}^m$ is said to be Lipschitz continuous on $\mathcal{Z}$ with modulus β if there exists $\beta > 0$ such that $\|h(\mathbf{z}_1) - h(\mathbf{z}_2)\| \leq \beta\|\mathbf{z}_1 - \mathbf{z}_2\|$ for all $\mathbf{z}_1, \mathbf{z}_2 \in \mathcal{Z}$.

Definition 2 (Strongly Concave Functions): Let $\mathcal{Z} \subseteq \mathbb{R}^n$ be a convex set. Function h is said to be strongly concave on $\mathcal{Z}$ with modulus α if there exists a constant $\alpha > 0$ such that $h(\mathbf{z}) + \frac{1}{2}\alpha\|\mathbf{z}\|^2$ is concave on $\mathcal{Z}$.

By the definition of strongly concave functions, it is easy to show that if $h(\mathbf{z})$ is concave and $\alpha > 0$, then $h(\mathbf{z}) - \alpha\|\mathbf{z} - \mathbf{z}_0\|^2$ is strongly concave with modulus 2α for any constant $\mathbf{z}_0$.

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

B. Preliminaries

Define column vector $\boldsymbol{\mu} = [\mu_l^{(f)}]_{f \in \mathcal{F}, l \in \mathcal{L}}$ as the stacked vector of all link session rates. Define column vector $\mathbf{x} = [x_f]_{f \in \mathcal{F}}$ as the stacked vector of all source session rates. Note that $\boldsymbol{\mu}$ has length $|\mathcal{L}||\mathcal{F}|$ and $\mathbf{x}$ has length $|\mathcal{F}|$. Thus, constraints (2) can be vectorized as

$$\mathbf{g}(\mathbf{x}, \boldsymbol{\mu}) = \mathbf{A}\mathbf{x} + \mathbf{R}\boldsymbol{\mu} \leq \mathbf{0}, \tag{21}$$

where matrix $\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_{|\mathcal{F}|} \end{bmatrix}$ is a $|\mathcal{F}|(|\mathcal{N}| - 1) \times |\mathcal{F}|$ source-node incidence matrix such that each sub-matrix $\mathbf{A}_f$ is a $\{0, 1\}$ matrix of size $(|\mathcal{N}| - 1) \times |\mathcal{F}|$ whose (n, f)-th entry is equal to 1 if and only if node n is the source node of session f; and matrix $\mathbf{R} = \text{Diag}\{\mathbf{R}_1, \ldots, \mathbf{R}_{|\mathcal{F}|}\}$ is a block diagonal matrix with $\mathbf{R}_1, \ldots, \mathbf{R}_{|\mathcal{F}|}$ on its diagonal such that each sub-matrix $\mathbf{R}_f$ is a $\{\pm 1, 0\}$ node-arc incidence matrix of size $(|\mathcal{N}| - 1) \times |\mathcal{L}|$ whose (n, l)-th entry is equal to 1 if and only if link l flows into node n and is equal to -1 if and only if link l flows out of node n.

Define $\mathbf{y} = \begin{bmatrix} \mathbf{x} \\ \boldsymbol{\mu} \end{bmatrix}$ and $\mathbf{B} = [\mathbf{A}, \mathbf{R}]$. Then, constraints (2) can be further rewritten as

$$\mathbf{g}(\mathbf{y}) = \mathbf{B}\mathbf{y} \leq \mathbf{0}. \tag{22}$$

Lemma 5: The vector function $\mathbf{g}(\mathbf{y})$ is Lipschitz continuous with modulus

$$\beta = \sqrt{|\mathcal{F}|} + \sqrt{2|\mathcal{L}|}. \tag{23}$$

Proof: Note that linear function $\mathbf{g}(\mathbf{y}) = \mathbf{B}\mathbf{y}$ is Lipschitz continuous with modulus $\|\mathbf{B}\|_2$ where $\|\mathbf{B}\|_2$ is the induced matrix l_2 norm defined as $\|\mathbf{B}\|_2 = \sup_{\mathbf{x} \neq \mathbf{0}}\{\frac{\|\mathbf{B}\mathbf{x}\|}{\|\mathbf{x}\|}\}$. Applying matrix norm inequalities (for block matrices) $\|[\mathbf{H}_1, \mathbf{H}_2]\|_2 \leq \|\mathbf{H}_1\|_2 + \|\mathbf{H}_2\|_2$ and $\|\text{Diag}\{\mathbf{H}_1, \ldots, \mathbf{H}_K\}\|_2 \leq \max_{1 \leq k \leq K}\{\|\mathbf{H}_k\|_2\}$ yields $\|\mathbf{B}\|_2 \leq \|\mathbf{A}\|_2 + \|\mathbf{R}\|_2 \leq \|\mathbf{A}\|_2 + \max_{f \in \mathcal{F}}\{\|\mathbf{R}_f\|_2\}$. Note that exactly $|\mathcal{F}|$ entries in matrix $\mathbf{A}$ are 1 and all the other entries are 0; and each matrix $\mathbf{R}_f$ has at most $2|\mathcal{L}|$ non-zero entries whose absolute values are 1. By the fact $\|\mathbf{H}\|_2 \leq \sqrt{\sum_{i=1}^m \sum_{j=1}^n |H_{ij}|}$ for any matrix $\mathbf{H} \in \mathbb{R}^{m\times n}$, we know $\|\mathbf{A}\|_2 \leq \sqrt{|\mathcal{F}|}$ and $\|\mathbf{R}_f\|_2 \leq \sqrt{2|\mathcal{L}|}, \forall f \in \mathcal{F}$. It follows that $\|\mathbf{B}\|_2 \leq \sqrt{|\mathcal{F}|} + \sqrt{2|\mathcal{L}|}$. ∎

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). Define $L(t) = \frac{1}{2}\|\mathbf{Q}[t]\|^2$ and call it a *Lyapunov function*. Define the Lyapunov drift as $\Delta[t] = L[t+1] - L[t]$.

The update equation (9) can be written in vector form:

$$\mathbf{Q}[t+1] = \mathbf{Q}[t] + \mathbf{g}(\mathbf{y}[t]) \tag{24}$$

Lemma 6: At each iteration $t \in \{0, 1, \dots\}$ in Algorithm 1, the Lyapunov drift is given by

$$\Delta[t] = \mathbf{Q}^\mathsf{T}[t]\mathbf{g}(\mathbf{y}[t]) + \frac{1}{2}\|\mathbf{g}(\mathbf{y}[t])\|^2 \tag{25}$$

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

$$\begin{aligned}\Delta[t] &= \frac{1}{2}\mathbf{Q}^\mathsf{T}[t+1]\mathbf{Q}[t+1] - \frac{1}{2}\mathbf{Q}^\mathsf{T}[t]\mathbf{Q}[t]\\ &\overset{(a)}{=} \frac{1}{2}(\mathbf{Q}^\mathsf{T}[t] + \mathbf{g}^\mathsf{T}(\mathbf{y}[t]))(\mathbf{Q}[t] + \mathbf{g}(\mathbf{y}[t])) - \frac{1}{2}\mathbf{Q}^\mathsf{T}[t]\mathbf{Q}[t]\\ &= \mathbf{Q}^\mathsf{T}[t]\mathbf{g}(\mathbf{y}[t]) + \frac{1}{2}\|\mathbf{g}(\mathbf{y}[t])\|^2\end{aligned}$$

where (a) follows from (24). ∎

Define $f(\mathbf{y}) = \sum_{f\in\mathcal{F}} U_f(x_f)$. At each time t, consider choosing a decision vector $\mathbf{y}[t] = [\mathbf{x}[t]; \boldsymbol{\mu}[t]]$ to solve the following:

$$\max_{\mathbf{y}} \; f(\mathbf{y}) - \big(\mathbf{Q}[t] + \mathbf{g}(\mathbf{y}[t-1])\big)^\mathsf{T}\mathbf{g}(\mathbf{y}) - \alpha\|\mathbf{y} - \mathbf{y}[t-1]\|^2 \tag{26}$$

$$\text{s.t.} \quad (3)\text{-}(6) \tag{27}$$

The expression (26) is a *modified drift-plus-penalty expression*. Unlike the standard drift-plus-penalty expressions from [4], the above expression augments the virtual queues by the amount $\mathbf{g}(\mathbf{y}[t-1])$. 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 the above problem (26)-(27). This is formalized in the next lemma.

Lemma 7: At each iteration $t \in \{0, 1, \dots\}$, the action $\mathbf{y}[t] = [\mathbf{x}[t]; \boldsymbol{\mu}[t]]$ jointly chosen in Algorithm 1 is the solution to (26)-(27).

Proof: Note that if we define column vector $\mathbf{W}[t] = \big[W_n^{(f)}[t]\big]_{f\in\mathcal{F}, n\in\mathcal{N}\setminus\{\text{Dst}(f)\}}$ as the stacked vector of all weights $W_n^{(f)}[t]$ in (11), then (26) can be written as $f(\mathbf{y}) - \mathbf{W}^\mathsf{T}[t]\mathbf{g}(\mathbf{y}) - \alpha\|\mathbf{y} - \mathbf{y}[t-1]\|^2$. The proof involves collecting terms associated with the $x_f[t]$ and $\mu_l^{(f)}[t]$ decisions and is omitted for brevity. ∎

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

Lemma 8: Let $\mathbf{y}^*$ be an optimal solution to problem (1)-(6) given in Fact 1, i.e., $\mathbf{g}(\mathbf{y}^*) = \mathbf{0}$. If $\alpha \geq \frac{1}{2}\big(\sqrt{|\mathcal{F}|} + \sqrt{2|\mathcal{L}|}\big)^2$, then the action $\mathbf{y}[t] = [\mathbf{x}[t]; \boldsymbol{\mu}[t]]$ chosen in Algorithm 1 at each iteration $t \in \{0, 1, \dots\}$ satisfies

$$\begin{aligned}&-\Delta[t] + f(\mathbf{y}[t])\\ \geq& f(\mathbf{y}^*) + \alpha\big(\|\mathbf{y}^* - \mathbf{y}[t]\|^2 - \|\mathbf{y}^* - \mathbf{y}[t-1]\|^2\big)\end{aligned}$$

Proof: Note that $\mathbf{y}[t-1]$ and $\mathbf{Q}[t] + \mathbf{g}(\mathbf{y}[t-1])$ appear as known constants in (26)-(27). Since $f(\mathbf{y})$ is concave and $\mathbf{g}(\mathbf{y})$ is linear, it follows that $f(\mathbf{y}) - \big(\mathbf{Q}[t] + \mathbf{g}(\mathbf{y}[t-1])\big)^\mathsf{T}\mathbf{g}(\mathbf{y}) - \alpha\|\mathbf{y} - \mathbf{y}[t-1]\|^2$ is strongly concave with respect to $\mathbf{y}$ with modulus 2α. By Lemma 7, $\mathbf{y}[t]$ is chosen to solve problem (26)-(27). Note that $\mathbf{y}^*$ satisfies (3)-(6) by definition. Thus, by Lemma 4, we have

$$\begin{aligned}&f(\mathbf{y}[t]) - (\mathbf{Q}[t] + \mathbf{g}(\mathbf{y}[t-1]))^\mathsf{T}\mathbf{g}(\mathbf{y}[t]) - \alpha\|\mathbf{y}[t] - \mathbf{y}[t-1]\|^2\\ \geq& f(\mathbf{y}^*) - \big(\mathbf{Q}[t] + \mathbf{g}(\mathbf{y}[t-1])\big)^\mathsf{T}\mathbf{g}(\mathbf{y}^*) - \alpha\|\mathbf{y}^* - \mathbf{y}[t-1]\|^2\\ &+ \alpha\|\mathbf{y}^* - \mathbf{y}[t]\|^2\\ \overset{(a)}{=}& f(\mathbf{y}^*) - \alpha\|\mathbf{y}^* - \mathbf{y}[t-1]\|^2 + \alpha\|\mathbf{y}^* - \mathbf{y}[t]\|^2\end{aligned} \tag{28}$$

where (a) follows from the fact that $\mathbf{g}(\mathbf{y}^*) = \mathbf{0}$.

Recall that $\mathbf{u}_1^\mathsf{T}\mathbf{u}_2 = \frac{1}{2}\big(\|\mathbf{u}_1\|^2 + \|\mathbf{u}_2\|^2 - \|\mathbf{u}_1 - \mathbf{u}_2\|^2\big)$ for any $\mathbf{u}_1, \mathbf{u}_2 \in \mathbb{R}^m$. Thus, we have

$$\begin{aligned}&\mathbf{g}^\mathsf{T}(\mathbf{y}[t-1])\mathbf{g}(\mathbf{y}[t])\\ =&\frac{1}{2}\big(\|\mathbf{g}(\mathbf{y}[t-1])\|^2 + \|\mathbf{g}(\mathbf{y}[t])\|^2 - \|\mathbf{g}(\mathbf{y}[t-1]) - \mathbf{g}(\mathbf{y}[t])\|^2\big).\end{aligned} \tag{29}$$

Substituting (29) into (28) and rearranging terms yields

$$\begin{aligned}&f(\mathbf{y}[t]) - \mathbf{Q}^\mathsf{T}[t]\mathbf{g}(\mathbf{y}[t])\\ \geq& f(\mathbf{y}^*) - \alpha\|\mathbf{y}^* - \mathbf{y}[t-1]\|^2 + \alpha\|\mathbf{y}^* - \mathbf{y}[t]\|^2\\ &+ \alpha\|\mathbf{y}[t] - \mathbf{y}[t-1]\|^2 + \frac{1}{2}\|\mathbf{g}(\mathbf{y}[t-1])\|^2 + \frac{1}{2}\|\mathbf{g}(\mathbf{y}[t])\|^2\\ &- \frac{1}{2}\|\mathbf{g}(\mathbf{y}[t-1]) - \mathbf{g}(\mathbf{y}[t])\|^2\\ \overset{(a)}{\geq}& f(\mathbf{y}^*) - \alpha\|\mathbf{y}^* - \mathbf{y}[t-1]\|^2 + \alpha\|\mathbf{y}^* - \mathbf{y}[t]\|^2 + \frac{1}{2}\|\mathbf{g}(\mathbf{y}[t])\|^2\\ &+ \frac{1}{2}\|\mathbf{g}(\mathbf{y}[t-1])\|^2\\ &+ \big(\alpha - \frac{1}{2}\big(\sqrt{|\mathcal{F}|} + \sqrt{2|\mathcal{L}|}\big)^2\big)\|\mathbf{y}[t] - \mathbf{y}[t-1]\|^2\\ \overset{(b)}{\geq}& f(\mathbf{y}^*) - \alpha\|\mathbf{y}^* - \mathbf{y}[t-1]\|^2 + \alpha\|\mathbf{y}^* - \mathbf{y}[t]\|^2 + \frac{1}{2}\|\mathbf{g}(\mathbf{y}[t])\|^2\\ &+ \frac{1}{2}\|\mathbf{g}(\mathbf{y}[t-1])\|^2\end{aligned}$$

where (a) follows from the fact that $\|\mathbf{g}(\mathbf{y}[t]) - \mathbf{g}(\mathbf{y}[t-1])\| \leq \big(\sqrt{|\mathcal{F}|} + \sqrt{2|\mathcal{L}|}\big)\|\mathbf{y}[t] - \mathbf{y}[t-1]\|$, i.e., Lemma 5; and (b) follows from the fact that $\alpha \geq \frac{1}{2}\big(\sqrt{|\mathcal{F}|} + \sqrt{2|\mathcal{L}|}\big)^2$.

Subtracting (25) from the above inequality and cancelling the common terms on both sides yields

$$\begin{aligned}&-\Delta[t] + f(\mathbf{y}[t])\\ \geq& f(\mathbf{y}^*) - \alpha\|\mathbf{y}^* - \mathbf{y}[t-1]\|^2 + \alpha\|\mathbf{y}^* - \mathbf{y}[t]\|^2\\ &+ \frac{1}{2}\|\mathbf{g}(\mathbf{y}[t-1])\|^2\\ \geq& f(\mathbf{y}^*) - \alpha\|\mathbf{y}^* - \mathbf{y}[t-1]\|^2 + \alpha\|\mathbf{y}^* - \mathbf{y}[t]\|^2.\end{aligned}$$

∎

C. Utility Optimality Gap Analysis

Lemma 9: Let $\mathbf{y}^*$ be an optimal solution to problem (1)-(6) given in Fact 1, i.e., $\mathbf{g}(\mathbf{y}^*) = \mathbf{0}$. If $\alpha \geq \frac{1}{2}\big(\sqrt{|\mathcal{F}|} + \sqrt{2|\mathcal{L}|}\big)^2$ in Algorithm 1, then for all $t \geq 1$,

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

Proof: By Lemma 8, we have $-\Delta[\tau] + f(\mathbf{y}[\tau]) \geq f(\mathbf{y}^*) + \alpha\big(\|\mathbf{y}^* - \mathbf{y}[t]\|^2 - \|\mathbf{y}^* - \mathbf{y}[t-1]\|^2\big), \forall \tau \in \{0, 1, \ldots, t-1\}$. Summing over $\tau \in \{0, 1, \ldots, t-1\}$ yields

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

where (a) follows from $\mathbf{y}[-1] = \mathbf{0}$ in Algorithm 1.

Recall $\Delta[\tau] = L[\tau+1] - L[\tau] = \frac{1}{2}\|\mathbf{Q}[\tau+1]\|^2 - \frac{1}{2}\|\mathbf{Q}[\tau]\|^2$, simplifying summations and rearranging terms yields

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

where (a) follows from the fact that $\mathbf{Q}[0] = \mathbf{0}$. ■

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

Theorem 1: Let $\mathbf{y}^*$ be an optimal solution to problem (1)-(6) given in Fact 1, i.e., $\mathbf{g}(\mathbf{y}^*) = \mathbf{0}$. If $\alpha \geq \frac{1}{2}\big(\sqrt{|\mathcal{F}|} + \sqrt{2|\mathcal{L}|}\big)^2$ in Algorithm 1, then for all $t \geq 1$, we have

$$\frac{1}{t}\sum_{\tau=0}^{t-1}\sum_{f\in\mathcal{F}} U_f(x_f[\tau]) \geq \sum_{f\in\mathcal{F}} U_f(x_f^*) - \frac{\alpha}{t}\|\mathbf{y}^*\|^2.$$

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

$$\sum_{f\in\mathcal{F}} U_f(\overline{x}_f[t]) \geq \sum_{f\in\mathcal{F}} U_f(x_f^*) - \frac{\alpha}{t}\|\mathbf{y}^*\|^2.$$

Proof: Recall that $f(\mathbf{y}) = \sum_{f\in\mathcal{F}} U_f(x_f)$. By Lemma 9, we have

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

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. Finite Queue Bound Analysis

Lemma 10: Let $\mathbf{Q}[t], t \in \{0, 1, \ldots\}$ 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 $\mathbf{Q}[t+1] = \mathbf{Q}[t] + \mathbf{g}(\mathbf{y}[t-1])$. ■

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

Theorem 2: Let $\mathbf{y}^*$ be an optimal solution to problem (1)-(6) given in Fact 1, i.e., $\mathbf{g}(\mathbf{y}^*) = \mathbf{0}$, and $\boldsymbol{\lambda}^*$ be a Lagrange multiplier vector given in Assumption 2. If $\alpha \geq \frac{1}{2}\big(\sqrt{|\mathcal{F}|} + \sqrt{2|\mathcal{L}|}\big)^2$ in Algorithm 1, then for all $t \geq 1$,

$$|Q_n^{(f)}[t]| \leq 2\|\boldsymbol{\lambda}^*\| + \sqrt{2\alpha}\|\mathbf{y}^*\|, \forall f \in \mathcal{F}, \forall n \in \mathcal{N}\setminus\{\mathrm{Dst}(f)\}.$$

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

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

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

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

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

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

where (a) follows form Lemma 10 and (b) follows from Cauchy-Schwarz inequality. On the other hand, by Lemma 9, we have

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

Combining the last two inequalities and cancelling the common terms yields

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

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

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

■

This theorem shows that the absolute values of all virtual queues $Q_n^{(f)}[t]$ are bounded by the constant $B = 2\|\boldsymbol{\lambda}^*\| + \sqrt{2\alpha}\|\mathbf{y}^*\|$. By Lemma 1, 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 $\mathbf{y}^*$ be an optimal solution to problem (1)-(6) given in Fact 1, i.e., $\mathbf{g}(\mathbf{y}^*) = \mathbf{0}$, and $\boldsymbol{\lambda}^*$ be a Lagrange multiplier vector given in Assumption 2. If $\alpha \geq \frac{1}{2}\big(\sqrt{|\mathcal{F}|} + \sqrt{2|\mathcal{L}|}\big)^2$ in Algorithm 1, 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

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

E. Performance of Algorithm 1

Theorems 1 and 2 together imply that Algorithm 1 with $\alpha \geq \frac{1}{2}\big(\sqrt{|\mathcal{F}|} + \sqrt{2|\mathcal{L}|}\big)^2$ 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 are always bounded by a constant that is independent of the utility optimality gap.

This is superior to existing backpressure algorithms from [5], [10], [4] that achieve an $O(1/V)$ utility gap only at the cost of an $O(V^2)$ or $O(V)$ queue length, where V is an algorithm parameter. To obtain a vanishing utility gap, existing backpressure algorithms in [5], [10], [4] necessarily yield unbounded queues. We also comment that $O(V^2)$ queue bound in the primal-dual type backpressure algorithm [5] is actually of the order $V^2\|\boldsymbol{\lambda}^*\| + B_1$ 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 [22] shows that the $O(V)$ 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, both algorithms are claimed to have $O(V^2)$ or $O(V)$ queue bounds. By Corollary 1, Algorithm 1 guarantees physical queues at each node are bounded by $4\|\boldsymbol{\lambda}^*\| + B_3$, where $B_3 = 2\sqrt{2\alpha}\|\mathbf{y}^*\| + \sum_{l\in\mathcal{O}(n)} C_l$ is constant for a given problem. Thus, the constant queue bound guaranteed by Algorithm 1 is typically smaller than the $O(V^2)$ or $O(V)$ queue bounds from [5] and [22] even for a small V. (A small V can yield a poor utility performance for the backpressure algorithms in [5], [4].)

V. Numerical Experiment

In this section, we consider a simple network with 6 nodes and 8 links and 2 sessions 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 the network [17].) 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.

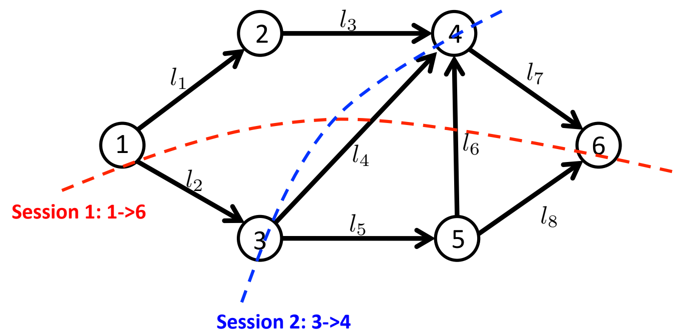

Fig. 1. A simple network with 6 nodes, 8 links and 2 sessions.

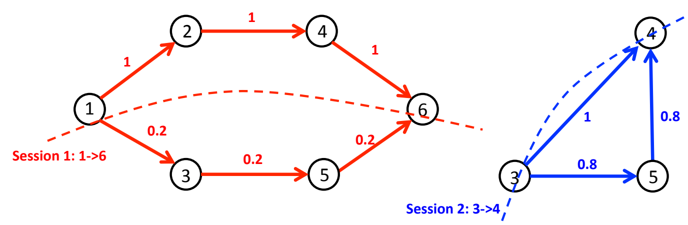

Fig. 2. The optimal routing for the network in Figure 1.

To compare the convergence performance of Algorithm 1 and the backpressure algorithm in [4] (with the best utility-delay tradeoff among all existing backpressure algorithms), we run both Algorithm 1 with $\alpha = \frac{1}{2}\big(\sqrt{|\mathcal{F}|} + \sqrt{2|\mathcal{L}|}\big)^2 = 14.7$ and the backpressure algorithm in [4] with $V = 500$ to plot Figure 3. It can be observed from Figure 3 that Algorithm 1 converges to the optimal source session rates faster than the backpressure algorithm in [4]. The backpressure algorithm in [4] with $V = 500$ takes around 2500 iterations to converges to source rates close to $(1.2, 1.8)$ while Algorithm 1 only takes around 800 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 [4] 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 the backpressure algorithm in [4]. 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 1 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 [4] with $V = 500$ oscillates around a point slightly larger than 1.2.

The analysis in Section IV-D shows that Algorithm 1 guarantees the actual queues in the network are bounded by constant $4\|\boldsymbol{\lambda}^*\| + 2\sqrt{2\alpha}\|\mathbf{y}^*\| + \sum_{l\in\mathcal{O}(n)} C_l$. Recall that the backpressure algorithm in [4] can guarantee the actual queues

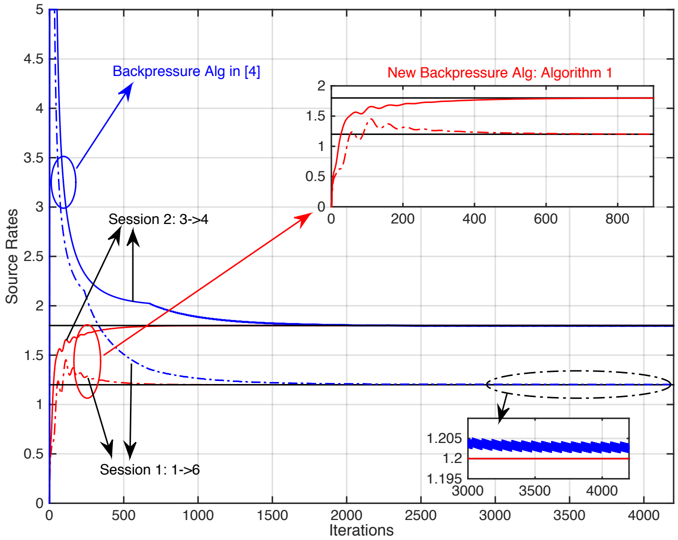

Fig. 3. Convergence performance comparison between Algorithm 1 and the backpressure algorithm in [4].

in the network are bounded by a constant of order $V\|\boldsymbol{\lambda}^*\|$. Figure 4 plots the sum of **actual queue lengths** at all nodes in the network for Algorithm 1 and the backpressure algorithm in [4] with $V = 10, 100$ and 500. (Recall a larger V in the backpressure algorithm in [4] 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 [4] scales linearly with respect to V.

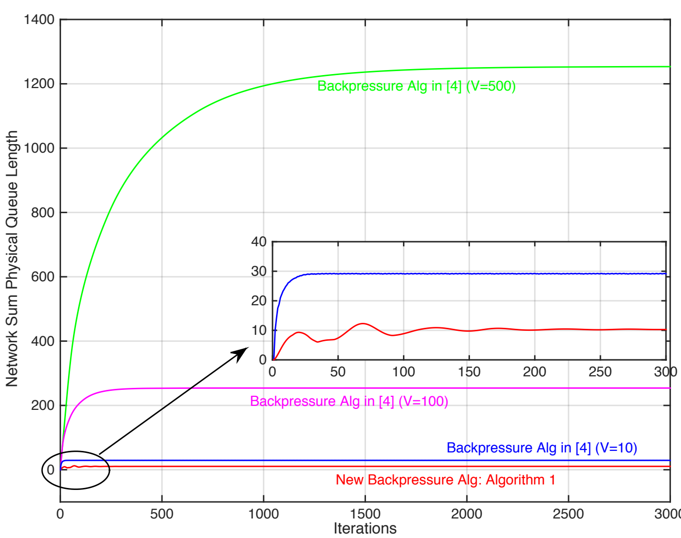

Fig. 4. Actual queue length comparison between Algorithm 1 and the backpressure algorithm in [4].

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 with finite queue lengths. This improves the state-of-art $[O(1/V), O(V^2)]$ or $[O(1/V), O(V)]$ utility-delay tradeoff attained by existing backpressure algorithms [5], [9], [7], [10].

References

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

[2] 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.

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

[4] M. J. Neely, *Stochastic network optimization with application to communication and queueing systems*. Morgan & Claypool Publishers, 2010.

[5] 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.

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

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

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

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

[16] 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.

[17] 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*, pp. 237–252, 1998.

[18] 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.

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

[20] 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," *arXiv:1701.04519*, 2017.

[21] 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.

[22] M. J. Neely, "A simple convergence time analysis of drift-plus-penalty for stochastic optimization and convex programs," *arXiv preprint arXiv:1412.079*, 2014.