Towards Fast-Convergence, Low-Delay and Low-Complexity Network Optimization

SINONG WANG, The Ohio State University, USA
NESS SHROFF, The Ohio State University, USA

Distributed network optimization has been studied for well over a decade. However, we still do not have a good idea of how to design schemes that can simultaneously provide good performance across the dimensions of utility optimality, convergence speed, and delay. To address these challenges, in this paper, we propose a new algorithmic framework with all these metrics approaching optimality. The salient features of our new algorithm are three-fold: (i) fast convergence: it converges with only $O(\log(1/\epsilon))$ iterations that is the fastest speed among all the existing algorithms; (ii) low delay: it guarantees optimal utility with finite queue length; (iii) simple implementation: the control variables of this algorithm are based on virtual queues that do not require maintaining per-flow information. The new technique builds on a kind of inexact Uzawa method in the Alternating Directional Method of Multiplier, and provides a new theoretical path to prove global and linear convergence rate of such a method without requiring the full rank assumption of the constraint matrix.

Additional Key Words and Phrases: Network Utility Maximization, cross-layer optimization, distributed optimization

ACM Reference Format:
Sinong Wang and Ness Shroff. 2017. Towards Fast-Convergence, Low-Delay and Low-Complexity Network Optimization. *Proc. ACM Meas. Anal. Comput. Syst.* 1, 2, Article 34 (December 2017), 32 pages. https://doi.org/10.1145/3154492

1 INTRODUCTION

Consider a fixed data network shared by F end-to-end flows. Each flow f is described by its source-destination node pair and associated utility function, without a priori established routes. The nodes within the network cooperate by forwarding each others' packets toward their destinations. The network optimization problem is *how does one jointly choose the end-to-end data rate x_f of each flow f, the schedule for each link and the link rate for each flow to maximize the network utilities defined as*

$$\max \sum_{f=1}^{F} U_f(x_f) \text{ s.t. } [x_f] \in \Lambda, \tag{1}$$

where Λ is the capacity region of data network, dependent on the limited power resources and interference among concurrent transmissions. The optimization problems of the above form plays a key role in resource control and optimization for both wireline and wireless networks.

In distributed network optimization, each iteration of the algorithm corresponds to one communication among different nodes, which could require a very large amount of information exchange

Authors' addresses: Sinong Wang, Department of ECE, The Ohio State University, Columbus, Ohio, USA, 43220, wang.7691@osu.edu; Ness Shroff, Departments of ECE and CSE, The Ohio State University, Columbus, Ohio, USA, 43220, shroff.11@osu.edu.

2476-1249/2017/12-ART34 $$15.00
https://doi.org/10.1145/3154492

Table 1. Comparison of Existing Algorithms in Network Optimization

	Optimality gap	Queue-length	Convergence speed	Routing complexity	Scheduling complexity[1]
Dual decomposition method	$O(1/K)$	$O(K)$	$O(1/\epsilon^2)$	$O(F)$	poly(L, F)
Proximal method	optimal	$O(1)$	$O^*(1/\epsilon)$	$O(F\log(F))$	unknown
Second-order method	$O(1/K)$	$O(K^2)$	$O(\log^2(1/\epsilon))$	$O(F^2+L^2)$	exp(L, F)
Momentum method	$O(1/K)$	$O(\sqrt{K})$	$O(1/\epsilon^2)$	$O(F)$	poly(L, F)
Our new method	optimal	$O(1)$	$O(\log(1/\epsilon))$	$O(F\log(F))$	poly(L, F)

[1] The scheduling complexity derives from the traditional node-exclusive interference model.
[2] Momentum method refers to heavy-ball method and Nesterov's accelerated method.

overhead. Therefore, one important metric to measure the performance of algorithm is the convergence speed, i.e., how many iterations are required to obtain an $\epsilon-$accurate solution. In addition, other important metrics are utility and the physical queue length in steady state, which measures the throughput and transmission delay that is achieved by the algorithm.

1.1 Existing Algorithms

The large body of work (see, e.g., [9, 20, 21, 23–27, 34–36], and [22] for a survey) in this area has given rise to several efficient and distributed control algorithmic frameworks. We first review the state-of-the-art of all the existing algorithms.

First-order dual decomposition method: This kind of algorithm applies the subgradient descent method to the dual function of problem (1) and leads to a beautiful queue-length-based control algorithmic (QCA) framework, based on which the components of congestion control, routing and scheduling are naturally coupled by queuing states [9, 20]. However, the classical QCA method achieves an $O(1/K)$ utility optimality gap at an expense of $O(K)$ steady-state queue-length, where $K > 0$ is a system parameter. Hence, a small utility gap will yield a large queuing delay. Significant efforts have been made to improve this tradeoff including the development of virtual queue techniques [1, 18], the threshold-based packeting-dropping scheme [15] and the $[O(1/K), O(\sqrt{K})]$ tradeoff produced by recent momentum-based methods [23, 24]. Due to the nonsmoothness of dual function and the subgradient nature, all the above methods suffer from a slow convergence that requires $O(1/\epsilon^2)$ iterations to obtain an $\epsilon-$accurate solution.

Second-order Newton method: To improve the convergence speed, there have been many attempts in obtaining new algorithms by applying the second-order method [25, 26, 36]. Compared with the first-order method, this kind of algorithm has a faster convergence rate, i.e., $O(\log^2(1/\epsilon))$ iterations (three-level convergence structure with interior point, Newton and matrix splitting method). However, it has several limitations: (i) the complexity of computing the Hessian inverse in the second-order method is quite high and does not scale well with the network size; (ii) a worse utility-delay tradeoff $[O(1/K), O(K^2)]$ in [25]; (iii) it cannot efficiently handle the wireless interference channel. For example, in the algorithm [26], even the number of variables (time sharing parameters) in the control plane is exponentially large.

Proximal method: The proximal method was first introduced in the work [21] to tackle the oscillation problem in a network optimization problem with given routing paths. Unlike the QCA method, it adds a quadratic regularizer in the routing component to stabilize the solution, and is proven to be the first algorithm to break the existing utility-delay tradeoff that offers both the zero utility optimality gap and finite queue length. Recently, the work [37] generalizes this idea to the scenario of dynamic routing and designs a new backpressure routing algorithm for wireline network. They prove that the proximal method not only exhibits the feature of low-latency, it also offers an improved convergence speed of $O^*(1/\epsilon)$[1].

[1]Here the $O^*(1/\epsilon)$ means that the convergence rate is in the ergodic sense. A sequence $\{a_n\}$ converges with ergodic rate $O^*(1/\epsilon)$ if $\frac{1}{N}\sum_{n=1}^{N} a_n = O(1/N)$, with rate $O(1/\epsilon)$ if $a_n = O(1/N)$.

It can be observed that all the existing algorithms sacrifice the performance of one or more metrics to improve the others. In particular, the slow convergence of all these algorithms will result in large information exchange overhead. The key question that we aim to answer in this paper is that: *is it possible to develop a joint congestion control, routing and scheduling algorithm with the fast convergence speed, routing complexity as low as the first-order method and delay as low as the proximal method?*

1.2 Our Results

In this paper, we positively answer this open question and propose a new algorithmic framework. The comparison of our algorithm and the existing schemes in a $L-$links and $F-$flows network are listed in TABLE 1. One can see that our algorithm offers the **fastest** convergence speed, **optimal** utility, **finite** queue length, and **low** routing and scheduling complexity compared with all the existing methods. The rationale behind our algorithm design is to utilize the Alternating Directional Method of Multiplier (ADMM), first appeared in [10]. Our key idea is to reformulate the joint scheduling-routing-congestion control problem as a $2-$block separable optimization problem, and apply the ADMM to the Augmented Lagrangian function of problem (1), which then allows us to obtain an optimization framework with a layered structure and only a limited degree of cross-layer coupling.

However, due to a number of technical challenges, developing an ADMM-based method is highly non-trivial. First, the ADMM's focus is on minimizing the Augmented Lagrangian function that is the summation of original utility function and a quadratic penalty function of the constraints. It will produce a routing-scheduling problem with a non-separable objective function regarding the rate vector among different links. Therefore, it is difficult to be solved in a low-complexity and distributed manner. Second, the structure of this method is substantially different from both the dual decomposition method and the proximal method. For example, the form of congestion control, routing component, and the coupling among the different layers are different. Hence, the analytical techniques used in existing methods for utility optimality and queue stability are not applicable. Third, in a wireless network with interference constraints, unlike the clear relationship between the linear program-based scheduling problem in the dual decomposition method and the combinatorial optimization problem, i.e., maximum weighted matching [20, 33], it is unclear how to solve the new scheduling problem derived from the ADMM-based decomposition.

The main contribution of this paper is that we develop a new algorithmic framework that addresses the aforementioned challenges. The detailed results and technical contributions of this paper are as follows:

- We utilize a kind of inexact Uzawa method of Alternating Directional Method of Multiplier [2, 38] to approximately solve a local second-order approximation of the Augmented Lagrangian function with respect to the link rates. This technique will yield a routing and scheduling problem with a separable quadratic objective function and a constraint set defined by a convex hull of feasible link rate vectors.
- We establish the utility optimality and finite queue length of our proposed framework. In particular, we show that, as the algorithm keeps running, the network utility gap will vanish, while the queue lengths in each node are bounded throughout by a finite constant. This result is much stronger than the best tradeoff $[O(1/K), O(\sqrt{K})]$ of the traditional QCA framework. Moreover, we prove that our new algorithmic framework converges at a global and linear rate that obtains an $\epsilon-$accurate solution with only $O(\log(1/\epsilon))$ number of iterations, which is faster than the existing second-order methods.

- We provide several algorithms to implement the new routing and scheduling problem in our proposed framework. More precisely, for the wireline network, we show that the new routing problem can be solved in a distributed manner and in $O(F\log(F))$ time within each link, which is much lower than $O(F^2+L^2)$ complexity of the second-order method in [25]. For the wireless networks with interference constraints, we show that the complexity of solving our new scheduling problem is equivalent to the classical MaxWeight scheduling. This result not only implies a deep connection between these two problems, but also paves a path to use the existing algorithms [17, 33] of MaxWeight scheduling to solve this new problem.

One technical contribution independent of interest is the global and linear convergence rate of our proposed algorithm. As mentioned earlier, this algorithm is indeed applying an inexact Uzawa method of ADMM to the optimization problem of the form $\min f(\mathbf{x})+g(\mathbf{y}),\ \text{s.t. } \mathbf{Ax}+\mathbf{By}=\mathbf{b}$. All the existing global and linear convergence results [3, 5, 19] of this generalized ADMM requires an assumption that one of the constraint matrices is of full rank. However, in our problem, both matrices $\mathbf{A}$ and $\mathbf{B}$ do not satisfy this condition. We provide a new technical path to overcome this challenge. The critical technical step is to estimate the distance from the primal and dual iterates of ADMM to the optimal solution set by the distance to an inscribed polyhedron of the optimal set. This enables us to utilize the isolated calmness of polyhedral mapping to upper bound such distance by certain amount of constraint violation.

The remainder of this paper is organized as follows. In Section 2, we introduce the network model and problem formulation. Section 3 presents our proposed algorithmic framework and the main results. In Section 4, we provide the detailed theoretical analysis of convergence speed and queuing stability. Section 5 develops the algorithms for the principal components of our framework. Section 6 presents numerical results. Section 7 provides some discussions and Section 8 concludes this paper. Due to the space limit, all the proofs are listed in Appendix.

2 PROBLEM STATEMENT

2.1 Network Model

We consider a slotted communication network system with time slot units being indexed by $t=1,2,\ldots$. As shown in Fig. 1, we represent the network by a *directed* graph $\mathcal{G}=\{\mathcal{N},\mathcal{L}\}$, where $\mathcal{N}$ is the set of nodes and $\mathcal{L}$ is the set of edges. Let $|\mathcal{N}|=N$ and $|\mathcal{L}|=L$. For each node n, denote the sets of its incoming links and outgoing links as $\mathcal{I}(n)$ and $\mathcal{O}(n)$, respectively. Let $\deg(n)$ be the number of adjacent links of node n. We define $\mathrm{Tx}(l)$ and $\mathrm{Rx}(l)$ as the transmitting and receiving node for each edge l. There are F end-to-end sessions in the network, indexed by $f\in\mathcal{F}\triangleq\{1,2,\ldots,F\}$. Each session f has a source node s_f and a destination node d_f in the node set $\mathcal{N}$. To avoid triviality, suppose that different sources are located at different nodes.

2.2 Congestion Control

Let scalar x_f be the injection rate of session f with which data is sent from s_f to d_f, possibly via multiple hops and multiple paths. We assume that injection rate x_f is bounded in $[m_f,M_f]$. Associated with each flow f is a utility function $U_f(x_f)$, which reflects the "utility" to session f when it can transmit at rate x_f. We assume that the utility function $U_f(\cdot)$ satisfy the following conditions.

Assumption 1. (Utility function) *For each session f, the utility function $U_f(\cdot)$ is a nondecreasing and concave function in the interval $[m_f,M_f]$.*

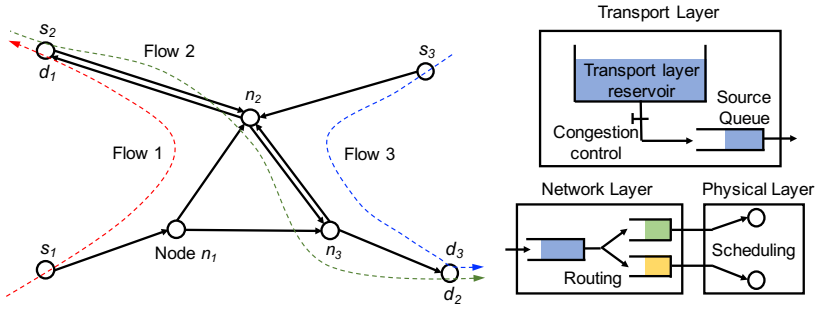

Fig. 1. Illustrative example of model.

The use of such utility functions is common in the congestion control literature to model fairness. For example, these conditions hold for the following two typically used utility functions: (i) weighted proportionally fair utilities $U_f(x_f) = w_f \log(x_f)$, where $w_f, f = 1, \ldots, F$ are the weights; (ii) general weighted proportionally fair utilities,

$$U_f(x_f) = w_f \frac{x^{1-\gamma}}{1-\gamma}, \gamma > 0. \tag{2}$$

Note that these two examples are also strictly concave functions.

2.3 Routing and Scheduling

For each edge l in the set $\mathcal{L}$, suppose that $l = (m, n)$ and the data is transmitted from node m to node n. Let r_l^d represent the amount of capacity on link l that is allocated for data towards destination d. In the sequel, we call it the link rate for simplicity. The set of destination nodes are defined as $\mathcal{D} = \{d_f, f \in \mathcal{F}\}$, and let $|\mathcal{D}| = D$. Then we can describe the capacity region of the data network.

Definition 2.1. (Capacity Region [27, 35]) The capacity region Λ of the network is the largest set of injection rate vector $[x_f]_{f\in\mathcal{F}}$ for which there exists a link rate vector $[r_l^d]_{l\in\mathcal{L}}^{d\in\mathcal{D}}$ that satisfies the following constraints.

(1) Flow conservation: for each destination d in $\mathcal{D}$, each node n in $\mathcal{N}\backslash\{d\}$,

$$\sum_{f\in\mathcal{F}} x_f \mathbf{1}_{\{s_f=n, d_f=d\}} + \sum_{l\in\mathcal{I}(n)} r_l^d = \sum_{l\in\mathcal{O}(n)} r_l^d, \tag{3}$$

where $\mathbf{1}_{\{\cdot\}}$ is an indicator function that takes the value 1 if $s_f = n, d_f = d$ and 0 otherwise.

(2) Capacity constraint: for each link $l \in \mathcal{L}$ and $d \in \mathcal{D}$,

$$\left[\sum_{d\in\mathcal{D}} r_l^d\right] \in C \triangleq \text{Conv}(\Gamma), r_l^d \geq 0, \tag{4}$$

where $\Gamma = \{\mathbf{r}^{(1)}, \mathbf{r}^{(2)}, \ldots, \mathbf{r}^{(I)}\}$ is the set of feasible link rate vectors, and $\text{Conv}(\cdot)$ represents the convex hull operation.

2.4 Queue Stability

We use $Q_n^d[t]$ to denote the length of the physical queue that are destined for node d, waiting for service at node n in time slot t. For each $d \in \mathcal{D}$ and $n \in \mathcal{N}\backslash d$, the evolution of physical queue

length is given by

$$Q_n^d[t] = \left[Q_n^d[t-1] - \sum_{l\in O(n)} r_l^d[t]\right]_+ + \sum_{l\in I(n)} \hat{r}_l^d[t] + \sum_{f\in\mathcal{F}} x_f[t]\mathbf{1}_{\{s_f=n, d_f=d\}}, \tag{5}$$

where $[\cdot]_+ \triangleq \max\{\cdot, 0\}$. The rate $r_l^d[t]$ is the capacity provided to d-destined packets over link l in time slot t and the rate $\hat{r}_l^d[t]$ is the actual used capacity over link l for d-destined packets in time slot t. We have $\hat{r}_l^d[t] \le r_l^d[t]$ since node n may have less than $r_l^d[t]$ amount of data to transmit for destination d. Note that the definition of $Q_n^d[t]$ is only used to measure the delay performance of our algorithm. The actual operation of our algorithm does not require this information (details in Section 3.1).

Definition 2.2. (Network Stability) Under a congestion control, routing and scheduling scheme, we say that the network is stable if the sum of queue lengths in steady state remains finite.

$$\limsup_{t\to\infty} \sum_{d\in\mathcal{D}} \sum_{n\in\mathcal{N}\setminus d} Q_n^d[t] < +\infty. \tag{6}$$

2.5 Problem Formulation

Our objective is to develop a joint congestion control, routing and scheduling algorithm to maximize the total utility $\sum_{f\in\mathcal{F}} U_f(x_f)$, subject to the network capacity constraints. Putting together the models presented earlier leads to the following general multi- commodity network flow formulation.

JCRS:

$$\begin{aligned} \max_{x_f, r_l^d} \quad & \sum_{f\in\mathcal{F}} U_f(x_f) \\ s.t. \quad & \sum_{f\in\mathcal{F}} x_f \mathbf{1}_{\{s_f=n, d_f=d\}} + \sum_{l\in I(n)} r_l^d = \sum_{l\in O(n)} r_l^d, \forall d, n \in \mathcal{N}\setminus d, \\ & \left[\sum_{d\in\mathcal{D}} r_l^d\right] \in C, r_l^d \ge 0, \forall d \in \mathcal{D}, l \in \mathcal{L}, \\ & m_f \le x_f \le M_f, \forall f \in \mathcal{F}. \end{aligned} \tag{7}$$

Problem (7) is a convex program with affine constraints. The objective is to develop a distributed algorithm to solve the above problem. We make the following standard assumption that is used in all the existing works.

Assumption 2. (Existence of optimal solutions) *There exists an optimal injection rate vector* $[x_f^*]_{f\in\mathcal{F}}$, *link rate vector* $[r_l^{d^*}]_{l\in\mathcal{L}}^{d\in\mathcal{D}}$ *and the Lagrangian multiplier vector* $[\lambda_n^{d^*}]_{n\in\mathcal{N}\setminus d}^{d\in\mathcal{D}}$ *in the problem (7).*

Note that the existence of optimal primal and dual solutions can be be guaranteed if a certain constraint qualification such as the Slater condition holds [32]. In what follows, we will investigate a new distributed joint congestion control, routing and scheduling algorithm.

3 JOINT CONGESTION CONTROL, ROUTING AND SCHEDULING FRAMEWORK

In Section 3.1, we first introduce our new algorithmic framework. Then, in Section 3.2, we present the main results on the utility optimality, queue stability and the convergence speed of the proposed algorithm.

3.1 Algorithmic Framework

The main procedure of our new joint congestion control, routing and scheduling method is described in Algorithm 1.

Algorithm 1: New Joint Congestion Control, Routing and Scheduling Framework

Initialization:

Choose parameters $\rho > 0$, $\tau \in [1, \frac{\sqrt{5}+1}{2})$ and $\beta_{m,n} > \deg(m) + \deg(n), \forall (m,n) \in \mathcal{L}$. Set $t = 0$. Let both physical and virtual queues be empty at the initial state $Q_n^d[0] = \lambda_n^d[0] = \lambda_n^d[-1] = 0, \forall d \in \mathcal{D}$ and $n \in \mathcal{N}\backslash\{d\}$. Let injection rates $x_f[0] = 0, \forall f \in \mathcal{F}$ and service rates $r_l^d[0] = 0, \forall d \in \mathcal{D}, l \in \mathcal{L}$.

Iteration: In each time slot $t \geq 1$, repeat the following three steps.

1: **Routing and Scheduling:** For each destination $d \in \mathcal{D}$ and node $n \in \mathcal{N}\backslash\{d\}$, calculate the new weight $z_n^d[t] = (1 + 1/\tau)\lambda_n^d[t-1] - \lambda_n^d[t-2]/\tau$. Let $z_d^d[t] = 0, \forall d \in \mathcal{D}$. Then choose the link rate $[r_l^d[t], l \in \mathcal{L}, d \in \mathcal{D}]$ as the solution to the following quadratic program.

$$\max_{r_{m,n}^d} \sum_{(m,n)\in\mathcal{L}} \sum_{d\in\mathcal{D}} (z_m^d[t] - z_n^d[t]) r_{m,n}^d - \frac{\rho\beta_{m,n}}{2}(r_{m,n}^d - r_{m,n}^d[t-1])^2$$
$$s.t. \quad \left[\sum_d r_{m,n}^d\right] \in C, r_{m,n}^d \geq 0, \forall (m,n) \in \mathcal{L}, d \in \mathcal{D}. \tag{8}$$

2: **Congestion Control:** For each node s_f, calculate the injection rate $x_f[t]$ as the solution to the following optimization problem.

$$\max_{x_f \in [m_f, M_f]} U_f(x_f) - (z_{s_f}^{d_f}[t] + \rho\Delta r_f[t])x_f - \frac{\rho}{2}(x_f - x_f[t-1])^2. \tag{9}$$

where the quantity $\Delta r_f[t]$ is given by

$$\Delta r_f[t] = \sum_{l\in\mathcal{I}(s_f)} \left(r_l^{d_f}[t] - r_l^{d_f}[t-1]\right) - \sum_{l\in\mathcal{O}(s_f)} \left(r_l^{d_f}[t] - r_l^{d_f}[t-1]\right). \tag{10}$$

3: **Virtual Queue Update:** For each destination $d \in \mathcal{D}$ and node $n \in \mathcal{N}\backslash\{d\}$, update the virtual queue length by

$$\lambda_n^d[t] = \lambda_n^d[t-1] - \rho\tau \sum_{l\in\mathcal{O}(n)} r_l^d[t] + \rho\tau \sum_{l\in\mathcal{I}(n)} r_l^d[t] + \rho\tau \sum_{f\in\mathcal{F}} x_f[t]\mathbf{1}_{\{s_f=n, d_f=d\}}. \tag{11}$$

Some important remarks on Algorithm 1 are in order:

Relation to QCA: In the QCA method [9, 20], the congestion control component has the form of

$$\max_{x_f \in [m_f, M_f]} U_f(x_f) - Q_{s_f}^{d_f}[t]x_f, \tag{12}$$

and the routing and scheduling component is given by

$$\max_{r_{m,n}^d} \sum_{(m,n)\in\mathcal{L}} \sum_{d\in\mathcal{D}} (Q_m^d[t] - Q_n^d[t]) r_{m,n}^d$$
$$s.t. \quad \left[\sum_d r_{m,n}^d\right] \in C, r_{m,n}^d \geq 0, \forall (m,n) \in \mathcal{L}, d \in \mathcal{D}. \tag{13}$$

Each component in this method is "loosely" connected by the physical queue length $Q_n^d[t]$. Similarly, our new algorithm also exhibits a layered structure, however, each component is "densely" connected by several quantities including the virtual queue length $\lambda_n^d[t]$, the injection rate $x_f[t]$ and the link rate $r_l^d[t]$. For example, the congestion control in the source node is dependent on both the virtual queue length and the change of link rate $\Delta r_f[t]$ in the adjacent links.

Quadratic congestion control and routing: Unlike the QCA method, Algorithm 1 contains a separable quadratic function in each component. In [21], it has been observed that such a l_2-regularization in the routing component can resolve the oscillation problem that occurs in traditional backpressure routing (13). Technically, we will see later that this technique also leads to significant delay reduction and convergence speed up, moreover, it can be derived from a kind of inexact Uzawa method in Alternating Directional Method of Multiplier [2, 38].

Virtual queue-based control: Existing methods such as the dual decomposition and the momentum-based methods require each node to maintain a separate physical queue for each flow, which is usually difficult to implement, especially in large networks. However, one can see that all the operations of congestion control, routing and scheduling in Algorithm 1 are based on the virtual queue length $\lambda_n^d[t]$. In practice, each node will maintain a separate virtual queue (i.e., a counter) for each flow going through it and a FIFO queue for storing packets of all the flows going through the corresponding link. This technique has been used in some existing works [1, 18]. However, they cannot guarantee the utility optimality and fast convergence rate.

3.2 Main Results

For notational convenience, we use vectors $\mathbf{x}[t], \mathbf{r}[t], \boldsymbol{\lambda}[t]$ to group all the injection rates, link rates and virtual queue lengths in time slot t, respectively. The first result in this paper is on the utility optimality and queue stability of Algorithm 1.

Theorem 3.1. (Utility optimality and queue stability) *Under the Assumptions 1 and 2, the network utility and physical queue length produced by Algorithm 1 satisfies*

$$\limsup_{t\to\infty}\left|\sum_{f\in\mathcal{F}} U_f(x_f[t]) - \sum_{f\in\mathcal{F}} U_f(x_f^*)\right| = 0, \tag{14}$$

$$\limsup_{t\to\infty}\sum_{d\in\mathcal{D}}\sum_{n\in\mathcal{N}\setminus d} Q_n^d[t] < +\infty, \tag{15}$$

where $[x_f^*, f\in\mathcal{F}]$ *is the optimal injection rate vector.*

Theorem 3.1 says that our proposed algorithm achieves optimal utility while guaranteeing that the physical queue length at each node is a finite constant. This result improves the utility-delay tradeoffs of prior works including $[O(1/K), O(K^2)]$ in [25], $[O(1/K), O(K)]$ in [20] and $[O(1/K), O(\sqrt{K})]$ in [23, 24]. All these methods will produce an unbounded queue length to obtain a vanishing utility optimality gap.

Theorem 3.2. (Global and linear convergence rate) *Under Assumptions 1 and 2 and the assumption that utility function is strictly concave, the Algorithm 1 converges at a global and linear rate. More specifically, there exists one of the optimal injection rate vector* $\mathbf{x}^*$*, link rate vector* $\mathbf{r}^*$ *and dual variable* $\boldsymbol{\lambda}^*$ *of the problem (7) such that* $\|\mathbf{x}[t]-\mathbf{x}^*\| \le O(c^t)$, $\|\mathbf{r}[t]-\mathbf{r}^*\| \le O(c^t)$, $\|\boldsymbol{\lambda}[t]-\boldsymbol{\lambda}^*\| \le O(c^t)$ *for all* $t \ge 1$*, where* c *is a constant satisfying* $0 < c < 1$.

As can be seen in Theorem 3.2, to obtain an $\epsilon-$accurate solution, our new algorithm only requires $O(\log(1/\epsilon))$ iterations, or equivalently, solving number of $O(\log(1/\epsilon))$ congestion control and routing components. This iteration complexity is much less than the traditional first-order method including dual decomposition method with $O(1/\epsilon^2)$ or the proximal method with $O(1/\epsilon)$. Moreover, it is even faster than the three-layered second-order Newton method [26].

Currently, several natural questions arise are: (i) how to design this new joint scheduling-routing-congestion control algorithm? (ii) how to prove the linear convergence rate, optimal utility and finite queue length of this new algorithm? (iii) how to efficiently solve the quadratic congestion

control, routing and scheduling component in our new algorithm? In the sequel, we focus on answering these questions.

4 THEORETICAL ANALYSIS

In this section, we first provide some necessary notations and basics in the variational analysis. Then, we will show how to apply the inexact Uzawa method in the Alternating Directional Method of Multiplier to obtain Algorithm 1. Finally, we will prove the technical results stated in Theorems 3.1 and 3.2.

4.1 Notations and Preliminaries

We use the bold letter $\mathbf{x}$ to represent the vector, and capital and bold letter $\mathbf{A}$ to denote the matrix. The element of a vector $\mathbf{x}$ is denoted by a scalar x_i, and the element of a matrix $\mathbf{A}$ is denoted by a scalar A_{ij}. We use $\mathbf{0}$ to represent a vector with each elements equal to zero. Let $\mathbf{x}^T$ and $\mathbf{A}^T$ to denote the transpose of a vector and a matrix, respectively. Let $\langle \cdot, \cdot \rangle$ represent the standard inner product, and let $\| \cdot \|$ denote the l_2 norm (the Euclidean norm of a vector or the spectral norm of a matrix). Let matrix norm $\|\mathbf{x}\|_{\mathbf{M}} = \mathbf{x}^T\mathbf{M}\mathbf{x}$, where $\mathbf{M}$ is a positive semidefinite matrix. We use $\lambda_{\min}(\mathbf{A})$ and $\lambda_{\max}(\mathbf{A})$ to represent the smallest and largest eigenvalues of a symmetric matrix $\mathbf{A}$. The spectral norm of a matrix $\mathbf{A}$ is then given by $\|\mathbf{A}\| = \lambda_{\max}(\mathbf{A}^T\mathbf{A})^{\frac{1}{2}}$. One basic inequality regarding the spectral norm is $\|\mathbf{A}\mathbf{x}\| \leq \|\mathbf{A}\|\|\mathbf{x}\|$.

Definition 4.1. (subdifferential) The subdifferential $\partial f(\mathbf{x})$ of a convex function $f : \mathbb{R}^n \to \mathbb{R}$ at $\mathbf{x}$ is the set of all subgradients.

$$\partial f(\mathbf{x}) = \{\mathbf{g} \in \mathbb{R}^n | \mathbf{g}^T(\mathbf{y} - \mathbf{x}) \leq f(\mathbf{y}) - f(\mathbf{x}), \forall \mathbf{y} \in \mathrm{dom}(f)\}.$$

The definition of subgradients is a generalization of the basic inequality from differentiable convex function to the non-differentiable function. For example, the indicator function over a convex set $I_C(\mathbf{x}) = 0, \mathbf{x} \in C$ and $I_C(\mathbf{x}) = \infty, \mathbf{x} \notin C$, is a convex and non-differentiable function. The subdifferential $\partial I_C(\mathbf{x})$ is the classical normal cone $N_C(\mathbf{x}) = \{\mathbf{g} | \mathbf{g}^T(\mathbf{y} - \mathbf{x}) \leq 0, \forall \mathbf{y} \in C\}$.

Definition 4.2. (Convex function) A function $f : \mathbb{R}^n \to \mathbb{R}$ is called convex with modulus $v \geq 0$ if for all $\mathbf{x}, \mathbf{y} \in \mathrm{dom}(f)$ and $\mathbf{g} \in \partial f(\mathbf{x})$, it satisfies

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \mathbf{g}^T(\mathbf{y} - \mathbf{x}) + \frac{v}{2}\|\mathbf{y} - \mathbf{x}\|^2.$$

As a consequence of the above definition, we have the following inequality, which will be used in our theoretical development. For arbitrary $\mathbf{x}, \mathbf{y} \in \mathrm{dom}(f)$,

$$\langle \mathbf{g}_x - \mathbf{g}_y, \mathbf{x} - \mathbf{y} \rangle \geq v\|\mathbf{x} - \mathbf{y}\|^2, \mathbf{g}_x \in \partial f(\mathbf{x}), \mathbf{g}_y \in \partial f(\mathbf{y}). \tag{16}$$

Note that the strictly convex function refers to that modulus $v > 0$.

Definition 4.3. (Moreau-Yosida proximal mapping) The proximal mapping of a closed and convex function $f : \mathbb{R}^n \to \mathbb{R}$ is defined as

$$\mathbf{Pr}_f(\mathbf{y}) = \arg\min_{\mathbf{x}} f(\mathbf{x}) + \frac{1}{2}\|\mathbf{x} - \mathbf{y}\|^2.$$

If the function f is the indicator function over a closed and convex set C, then $\mathbf{Pr}_f(\cdot) = \Pi_C(\cdot)$ is the metric projection operator over C. For simplicity, we use $[\cdot]_+$ to denote $\Pi_C(\cdot)$ when C is the positive orthant $[0, +\infty)^n$. One important property of Moreau-Yosida proximal mapping is non-expansiveness, which can be interpreted as the globally Lipschitz continuous with modulus one.

$$\|\mathbf{Pr}_f(\mathbf{x}) - \mathbf{Pr}_f(\mathbf{y})\| \leq \|\mathbf{x} - \mathbf{y}\|, \forall \mathbf{x}, \mathbf{y}.$$

4.2 Rationale behind the Algorithm Design

Algorithm 1 is inspired by an inexact Uzawa method in Alternating Directional Method of Multiplier (ADMM). For the sake of brevity, we will use the following vector notation in the rest of the paper. The node-arc incidence matrix $\mathbf{A}^d \in \mathbb{R}^{(N-1)\times L}$ is defined as

$$\mathbf{A}^d_{nl} = \begin{cases} 1, & \text{if } n = \mathrm{Tx}(l) \\ -1, & \text{if } n = \mathrm{Rx}(l) \\ 0, & \text{otherwise} \end{cases}, \forall n \in \mathcal{N}\backslash\{d\}, l \in \mathcal{L}.$$

The matrix $\mathbf{B}^d \in \mathbb{R}^{(N-1)\times F}$ is defined as

$$\mathbf{B}^d_{nf} = \begin{cases} -1, & \text{if } n = s_f, d = d_f \\ 0, & \text{otherwise} \end{cases}, \forall n \in \mathcal{N}\backslash\{d\}, f \in \mathcal{F}.$$

Define matrix $\mathbf{A} \in \mathbb{R}^{D(N-1)\times DL}$ and $\mathbf{B} \in \mathbb{R}^{D(N-1)\times F}$ as

$$\mathbf{A} = \mathrm{diag}\{\mathbf{A}^1, \mathbf{A}^2, \ldots, \mathbf{A}^D\} = \begin{bmatrix} \mathbf{A}^1 & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{A}^D \end{bmatrix}, \mathbf{B} = \begin{bmatrix} \mathbf{B}^1 \\ \vdots \\ \mathbf{B}^D \end{bmatrix}.$$

We denote the objective function $f(\mathbf{x}) = U(\mathbf{x}) + h(\mathbf{x})$, where the function $U(\mathbf{x}) = -\sum_{f\in\mathcal{F}} U_f(x_f)$ and the indicator function $h(\mathbf{x})$ is defined as

$$h(\mathbf{x}) = \begin{cases} 0, & \text{if } m_f \le x_f \le M_f, \forall f \in \mathcal{F} \\ +\infty, & \text{otherwise} \end{cases}.$$

Let the indicator function $g(\mathbf{r})$ represent the capacity constraints of link rate vector.

$$g(\mathbf{r}) = \begin{cases} 0, & \text{if } \left[\sum_d r^d_l\right] \in C, r^d_l \ge 0, \forall l, d \\ +\infty, & \text{otherwise} \end{cases}.$$

Based on the above notation, we can reformulate the JCRS problem (7) as the following equivalent form.

$$\begin{aligned} \min_{\mathbf{x},\mathbf{r}} \quad & f(\mathbf{x}) + g(\mathbf{r}) \\ s.t. \quad & \mathbf{B}\mathbf{x} + \mathbf{A}\mathbf{r} = \mathbf{0}. \end{aligned} \tag{17}$$

Note that optimization of this form contains a separable objective function and a separable constraint between injection rate vector $\mathbf{x}$ and the link rate vector $\mathbf{r}$. Therefore, it inspires us to adopt the Alternating Directional Method of Multiplier (ADMM) to split the decision variables $\mathbf{x}$ and $\mathbf{r}$, which results in a nice layered structure during the operation of the algorithm. Formally, the Augmented Lagrangian function of problem (17) is defined as

$$L(\mathbf{x}, \mathbf{r}, \boldsymbol{\lambda}) = f(\mathbf{x}) + g(\mathbf{r}) + \frac{\rho}{2}\|\mathbf{B}\mathbf{x} + \mathbf{A}\mathbf{r} - \boldsymbol{\lambda}/\rho\|^2, \tag{18}$$

where ρ is a pre-defined penalty parameter, $\boldsymbol{\lambda}$ is the Lagrangian multiplier. Then the ADMM optimizes the Augmented Lagrangian function $L(\mathbf{x}, \mathbf{r}, \boldsymbol{\lambda})$ in a Gauss-Seidel fashion. In each time slot t, go through the following three steps.

(1) Primal update: $\mathbf{r}[t] = \underset{\mathbf{r}}{\mathrm{argmin}}\, L(\mathbf{x}[t-1], \mathbf{r}, \boldsymbol{\lambda}[t-1])$.

(2) Primal update: $\mathbf{x}[t] = \underset{\mathbf{x}}{\mathrm{argmin}}\, L(\mathbf{x}, \mathbf{r}[t], \boldsymbol{\lambda}[t-1])$.

(3) Dual update: $\boldsymbol{\lambda}[t] = \boldsymbol{\lambda}[t-1] - \tau\rho(\mathbf{B}\mathbf{x}[t] + \mathbf{A}\mathbf{r}[t])$.

Based on the definition of the matrix $\mathbf{A}$ and $\mathbf{B}$, it is clear that the third step is the virtual queue update (11) in the Algorithm 1. We then show that the second step is indeed the congestion control component in Algorithm 1. We first omit the constant term $g(\mathbf{r}[t])$ and write it as

$$\mathbf{x}[t] = \underset{\mathbf{x}}{\operatorname{argmin}} f(\mathbf{x}) + \frac{\rho}{2}\|\mathbf{B}\mathbf{x} + \mathbf{A}\mathbf{r}[t] - \boldsymbol{\lambda}[t-1]/\rho\|^2.$$

Transforming the indicator function $h(\mathbf{x})$ in $f(\mathbf{x})$ into the box constraints, we have

$$\mathbf{x}[t] = \underset{\mathbf{m}\le\mathbf{x}\le\mathbf{M}}{\operatorname{argmax}} \sum_{f\in\mathcal{F}} U_f(x_f) - \frac{\rho}{2}\|\mathbf{B}\mathbf{x} + \mathbf{A}\mathbf{r}[t] - \boldsymbol{\lambda}[t-1]/\rho\|^2.$$

Based on the separability of both objective function and box constraints with respect to the variable x_f, we can decompose the original problem into F one-dimensional optimization problems.

$$x_f[t] = \underset{x_f\in[m_f,M_f]}{\operatorname{argmax}} U_f(x_f) - \frac{\rho}{2}\left(x_f + \sum_{l\in\mathcal{I}(s_f)} r_l^{d_f}[t] - \sum_{l\in\mathcal{O}(s_f)} r_l^{d_f}[t] + \lambda_{s_f}^{d_f}[t-1]/\rho\right)^2.$$

Rearranging the terms by utilizing the virtual queue length update in the time slot $t-1$, we can obtain the congestion control component in Algorithm 1.

The next step is to derive the routing component in Algorithm 1. As discussed before, the challenge in the first primal update step of ADMM is that the quadratic term $\|\mathbf{B}\mathbf{x}[t-1] + \mathbf{A}\mathbf{r} - \boldsymbol{\lambda}[t-1]/\rho\|^2$ in the objective function is non-separable with respect to the decision variable $\mathbf{r}$ due to the non-diagonal structure of the matrix $\mathbf{A}$.[2] The basic idea to overcome this difficulty is to *inexactly solve the* $\mathbf{r}$*−subproblem*, which is based on minimizing a second-order local approximation of the function $\|\mathbf{B}\mathbf{x}[t-1] + \mathbf{A}\mathbf{r} - \boldsymbol{\lambda}[t-1]/\rho\|^2$ instead of the original one. The approximation of the above function at the point $\mathbf{r}[t-1]$ is given by the Taylor expansion.

$$\begin{aligned}&\|\mathbf{A}\mathbf{r} + \mathbf{B}\mathbf{x}[t-1] - \boldsymbol{\lambda}[t-1]/\rho\|^2\\ &\approx \text{constant} + \langle \mathbf{g}[t-1], \mathbf{r} - \mathbf{r}[t-1]\rangle + \|\mathbf{r} - \mathbf{r}[t-1]\|_{\mathbf{M}}^2,\end{aligned}$$

where the gradient $\mathbf{g}[t-1] = 2\mathbf{A}^T(\mathbf{A}\mathbf{r}[t-1] + \mathbf{B}\mathbf{x}[t-1] - \boldsymbol{\lambda}[t-1]/\rho)$ and the matrix $\mathbf{M}$ is diagonal with $\mathbf{M} = \operatorname{diag}\{\ldots, \beta_l^d, \ldots\}$. Then, substituting this local approximation into the first step, we can write it as the following form.

$$\mathbf{r}[t] = \underset{\mathbf{r}}{\operatorname{argmin}}\, g(\mathbf{r}) + \rho\langle \mathbf{A}^T(\mathbf{A}\mathbf{r}[t-1] + \mathbf{B}\mathbf{x}[t-1] - \boldsymbol{\lambda}[t-1]/\rho), \mathbf{r} - \mathbf{r}[t-1]\rangle + \frac{\rho}{2}\|\mathbf{r} - \mathbf{r}[t-1]\|_{\mathbf{M}}^2. \tag{19}$$

Transforming the indicator function $g(\mathbf{r})$ into constraints, we are ready to obtain the routing component in the Algorithm 1.

The idea of approximately solving the subproblem in the ADMM has been widely applied in the existing literatures [13, 38]. The method is called the inexact Uzawa method and can be actually recovered by the following equivalent form.

$$\mathbf{r}[t] = \arg\min_{\mathbf{r}} L(\mathbf{x}[t-1], \mathbf{r}, \boldsymbol{\lambda}[t-1]) + \frac{1}{2}\|\mathbf{r} - \mathbf{r}[t-1]\|_{\mathbf{Q}}^2, \tag{20}$$

with matrix $\mathbf{Q} = \rho(\mathbf{M} - \mathbf{A}^T\mathbf{A})$. In the sequel, we will use this simplified form to prove all the theoretical results of Algorithm 1.

[2]One can actually utilize the randomized coordinate descent algorithm to solve this problem, which can lead to a asynchronous updates. However, the complexity is increased due to a two-layer iteration structure.

4.3 Convergence Analysis

In this subsection, we establish the global convergence of Algorithm 1. We first exploit the structure of matrix $\mathbf{B}$ and write the standard ADMM model (17) as the following form.

$$\begin{aligned} \min_{\mathbf{x},\mathbf{r}} \quad & f(\mathbf{x}) + g(\mathbf{r}) \\ s.t. \quad & \mathbf{A}_s\mathbf{r} = \mathbf{x}, \quad \mathbf{A}_r\mathbf{r} = \mathbf{0}, \end{aligned} \tag{21}$$

where $\mathbf{A}_s$ is a $F \times DL$ dimensional matrix formed by extracting the rows of matrix $\mathbf{A}$ whose index node is a source for one flow. The matrix $\mathbf{A}_r$ is formed by the rest of rows of the matrix $\mathbf{A}$. Therefore, the first equation $\mathbf{A}_s\mathbf{r} = \mathbf{x}$ in (21) denotes the flow conservation law in those source nodes and the second equation $\mathbf{A}_r\mathbf{r} = \mathbf{0}$ describes the flow conservation law in those intermediate nodes. Let the associated Lagrangian multiplier of constraints $\mathbf{A}_s\mathbf{r} = \mathbf{x}$, $\mathbf{A}_r\mathbf{r} = \mathbf{0}$ be $\boldsymbol{\lambda}_s$, $\boldsymbol{\lambda}_r$, respectively and let $\boldsymbol{\lambda} = [\boldsymbol{\lambda}_s; \boldsymbol{\lambda}_r]$. In the sequel, we write the Assumption 2 as the following equivalent form.

Assumption 3. (Existence of optimal solution) *There exists a saddle point $(\mathbf{x}^*, \mathbf{r}^*, \boldsymbol{\lambda}^*)$ of the problem (17), i.e., optimal primal variables $\mathbf{x}^*, \mathbf{r}^*$ and dual variables $\boldsymbol{\lambda}^*$, satisfying the KKT conditions:*

$$-\boldsymbol{\lambda}_r^* \in \partial f(\mathbf{x}^*), \tag{22}$$

$$\mathbf{A}_s^T\boldsymbol{\lambda}_s^* + \mathbf{A}_r^T\boldsymbol{\lambda}_r^* \in \partial g(\mathbf{r}^*), \tag{23}$$

$$\mathbf{A}_s\mathbf{r}^* = \mathbf{x}^*, \mathbf{A}_r\mathbf{r}^* = \mathbf{0}. \tag{24}$$

As discussed before, this assumption is a mild condition and can be guaranteed by various conditions. When this assumption fails to hold, Algorithm 1 has either unsolvable or unbounded subproblems or a diverging sequence of $\boldsymbol{\lambda}[t]$.

Lemma 4.4. (Sufficient descent of primal and dual variables) *Assume Assumption 1 and 2. If $\tau \in [1, (\sqrt{5}+1)/2)$, there exists an $\alpha, \eta > 0$ such that*

$$V(\mathbf{x}[t-1], \mathbf{r}[t-1], \boldsymbol{\lambda}[t-1]) - V(\mathbf{x}[t], \mathbf{r}[t], \boldsymbol{\lambda}[t]) \geq \alpha\left(\left\|\begin{bmatrix}\boldsymbol{\lambda}[t-1]-\boldsymbol{\lambda}[t] \\ \mathbf{x}[t-1]-\mathbf{x}[t]\end{bmatrix}\right\|^2 + \|\mathbf{r}[t-1]-\mathbf{r}[t]\|_{\mathbf{Q}}^2\right) + 2v\|\mathbf{x}[t]-\mathbf{x}^*\|^2 + 2v\|\mathbf{x}[t]-\mathbf{x}[t-1]\|^2. \tag{25}$$

The function $V(\mathbf{x}[t], \mathbf{r}[t], \boldsymbol{\lambda}[t])$ is defined as

$$V(\mathbf{x}[t], \mathbf{r}[t], \boldsymbol{\lambda}[t]) = \frac{1}{\rho\tau}\|\boldsymbol{\lambda}[t]-\boldsymbol{\lambda}^*\|^2 + \rho\|\mathbf{x}[t]-\mathbf{x}^*\|^2 + \|\mathbf{r}[t]-\mathbf{r}^*\|_{\mathbf{Q}}^2 + \frac{\rho}{\eta}\|\mathbf{A}_s\mathbf{r}[t]-\mathbf{x}[t]\|^2. \tag{26}$$

where matrix $\mathbf{Q} = \rho(\mathbf{M} - \mathbf{A}^T\mathbf{A})$, v is the convexity modulus of function $f(\mathbf{x})$ and $(\mathbf{x}^*, \mathbf{r}^*, \boldsymbol{\lambda}^*)$ is one of the saddle points of the problem (17).

In Lemma 4.4, the function $V(\mathbf{x}[t], \mathbf{r}[t], \boldsymbol{\lambda}[t])$ describes the distance between the current iterates and the optimal solution set. To guarantee that the function $V(\cdot)$ has sufficient descent, the matrix $\mathbf{M}$ should be chosen such that matrix $\mathbf{Q}$ is positive definite with $\|\mathbf{r}[t]-\mathbf{r}^*\|_{\mathbf{Q}}^2 > 0$. One simple choice is that each diagonal element of matrix $\mathbf{M}$ satisfies

$$\beta_{m,n}^d > \deg(m) + \deg(n), \forall (m,n) \in \mathcal{L}, \forall d \in \mathcal{D}. \tag{27}$$

Then one can see that $\mathbf{Q}$ is a diagonally dominant matrix, thus it is also positive definite by Gershgorin circle theorem. The proof of Lemma 4.4 is modified from [5].

Now we are ready to use the sufficient descent of the function $V(\cdot)$ to establish the global convergence of Algorithm 1.

Theorem 4.5. (Global convergence of Algorithm 1) *For any* $\tau \in [1, (\sqrt{5}+1)/2)$ *and any parameter* $\beta^d_{m,n} > \deg(m) + \deg(n)$ *for all* $(m,n) \in \mathcal{L}, d \in \mathcal{D}$, *the sequences* $(\mathbf{x}[t], \mathbf{r}[t], \boldsymbol{\lambda}[t])$ *converges to a saddle point of (17), namely,*

$$\begin{aligned} \limsup_{t\to\infty} \|\mathbf{x}[t] - \mathbf{x}^*\| &= 0, \\ \limsup_{t\to\infty} \|\mathbf{r}[t] - \mathbf{r}^*\| &= 0, \\ \limsup_{t\to\infty} \|\boldsymbol{\lambda}[t] - \boldsymbol{\lambda}^*\| &= 0. \end{aligned} \tag{28}$$

Note that the convergence of Algorithm 1 only requires the concavity of the utility function without the assumption of smoothness and strictly concavity (v could be zero). The existing theoretical analysis of two-block ADMM [13] has shown that the algorithm converges at a globally sub-linear rate, i.e., $O(1/\epsilon)$, when both function f and g are proper closed convex. Clearly, our definition of function f and g satisfy this condition and Algorithm 1 converges in $O(1/\epsilon)$ iterations. However, in the next subsection, we will present a surprising result that, when the utility function is strictly concave (v is positive), the Algorithm 1 actually converges globally and linearly, which requires only $O(\log(1/\epsilon))$ iterations to achieve an $\epsilon-$accurate solution.

4.4 Linear Convergence Rate Analysis

Based on the result in Lemma 4.4, we have an inequality of the form, for arbitrary $t \geq 1$,

$$V(\mathbf{x}[t-1], \mathbf{r}[t-1], \boldsymbol{\lambda}[t-1]) - V(\mathbf{x}[t], \mathbf{r}[t], \boldsymbol{\lambda}[t]) \geq C.$$

To establish the global and linear convergence rate of Algorithm 1, it is sufficient to show that, there exists constant $\gamma > 0$ such that

$$C \geq \gamma V(\mathbf{x}[t], \mathbf{r}[t], \boldsymbol{\lambda}[t]), \forall t \geq 1. \tag{29}$$

The function $V(\mathbf{x}[t], \mathbf{r}[t], \boldsymbol{\lambda}[t])$ contains the terms including $\|\mathbf{r}[t] - \mathbf{r}^*\|^2_{\mathbf{Q}}$ and $\|\boldsymbol{\lambda}[t] - \boldsymbol{\lambda}^*\|^2$, but the lower bound C only contains the terms like $\|\mathbf{r}[t] - \mathbf{r}[t-1]\|^2_{\mathbf{Q}}$. Therefore, the challenge is how to bound the terms $\|\boldsymbol{\lambda}[t] - \boldsymbol{\lambda}^*\|^2$ and $\|\mathbf{r}[t] - \mathbf{r}^*\|^2_{\mathbf{Q}}$ using the existing terms in the lower bound C. In the existing works of theoretical ADMM [5], they assume that the matrix $\mathbf{B}$ is of full row rank and matrix $\mathbf{A}$ is of full column rank, and utilize this assumption to upper bound $\|\boldsymbol{\lambda}[t] - \boldsymbol{\lambda}^*\|^2$ and $\|\mathbf{r}[t] - \mathbf{r}^*\|^2_{\mathbf{Q}}$ by existing terms in C. However, in our problem, both matrix $\mathbf{B}$ and $\mathbf{A}$ do not satisfy this assumption (matrix $\mathbf{B}$ has several all-zero rows, i.e., those nodes do not contain sources; the number of rows of matrix $\mathbf{A}$ is less than the number of columns). In the sequel, we provide a completely new theoretical path to overcome this technical challenge. We first introduce some basics in the variational analysis.

Definition 4.6. (Calmness [29]) Define the multi-valued mapping $F : \mathbb{R}^n \to \mathbb{R}^m$. We say that F is calm at $\mathbf{x}_0$ if there exists a neighborhood U of $\mathbf{x}_0$ and a constant $\kappa_0 > 0$ such that

$$F(\mathbf{x}) \subseteq F(\mathbf{x}_0) + \kappa_0 \|\mathbf{x} - \mathbf{x}_0\| \mathbb{B}_y, \forall \mathbf{x} \in U. \tag{30}$$

where unit ball $\mathbb{B}_y \triangleq \{\mathbf{y} \in \mathbb{R}^m | \|\mathbf{y}\| \leq 1\}$.

The calmness property can be regarded as a generalization of Lipschitz continuous property from single-valued function to set-valued mapping. Recall that the set-valued mapping F is piecewise polyhedral if the graph of F is the union of finitely many polyhedral sets. The following Lemma in [30] establishes the calmness of the piecewise polyhedral mapping.

Lemma 4.7. (Calmness of piecewise polyhedral mapping) *If the set-valued mapping* $F : \mathbb{R}^n \to \mathbb{R}^m$ *is piecewise polyhedral, then* F *is calm at any* $\mathbf{x}_0$ *with modulus* κ *independent of choice of* $\mathbf{x}_0$.

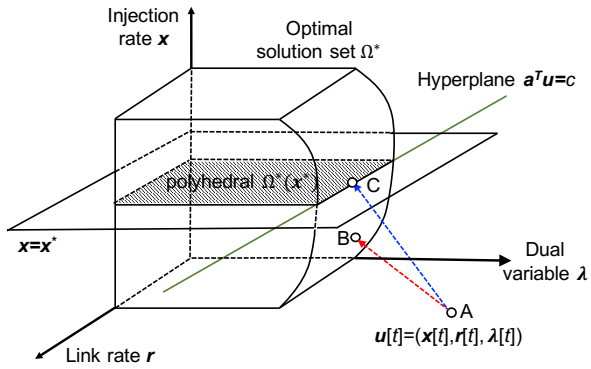

Fig. 2. The distance AB between current iterates and the optimal solution set Ω^* (non-polyhedral set) is less than the distance AC between current iterates and the set $\Omega^*(\mathbf{x}^*)$ (polyhedral set). The upper bound AC is the distance between a point A and a hyperplane $\mathbf{a}^T\mathbf{u} = c$ that can be implicitly given by $|\mathbf{a}^T\mathbf{u}[t] - b|/\|\mathbf{a}\| = O(|\mathbf{a}^T\mathbf{u}[t] - b|)$ (error bound in the simplest case).

The key technical path to obtain the inequality (29) is to utilize the calmness of piecewise polyhedral mapping to establish a global error bound. Then one can apply this error bound to estimate the distance to the optimal solution set, i.e., the terms in the function $V(\mathbf{x}[t], \mathbf{r}[t], \boldsymbol{\lambda}[t])$, by certain constraint violations, which can be further upper bounded by the existing terms in C. Denote the solution set of KKT system (22)-(24) by Ω^*. The main difficulty is that the set Ω^* is non-polyhedron, and one cannot use existing error bound such as Hoffman bound [14] or calmness to estimate the distance to the optimal solution set. However, one important observation is that, the intersection of the optimal solution set Ω^* and the hyperplane $\mathbf{x} = \mathbf{c}$, given by

$$\Omega^*(\mathbf{c}) = \Omega^* \cap \{(\mathbf{x}, \mathbf{r}, \boldsymbol{\lambda}) | \mathbf{x} = \mathbf{c}\}, \tag{31}$$

is actually the inverse image of a piecewise polyhedral mapping at origin. This result enables us to first upper bound the distance between current iterates $(\mathbf{x}[t], \mathbf{r}[t], \boldsymbol{\lambda}[t])$ and the optimal solution set Ω^* by the distance to the set $\Omega^*(\mathbf{c})$, then utilize the calmness property to further upper bound above distance by certain constraint violation.

Lemma 4.8. *For arbitrary optimal injection rate vector* $\mathbf{x}^*$*, define the set-valued mapping* $\mathbf{R}_{\mathbf{x}^*}(\mathbf{x}, \mathbf{r}, \boldsymbol{\lambda})$ *as*

$$\mathbf{R}_{\mathbf{x}^*}(\mathbf{x}, \mathbf{r}, \boldsymbol{\lambda}) = \begin{pmatrix} \mathbf{x} - \mathbf{Pr}_h(\mathbf{x} - \boldsymbol{\lambda}_s - \nabla U(\mathbf{x}^*)) \\ \mathbf{r} - \mathbf{Pr}_g(\mathbf{r} + (\mathbf{A}_s^T\boldsymbol{\lambda}_s + \mathbf{A}_r^T\boldsymbol{\lambda}_r)) \\ \mathbf{Ar} + \mathbf{Bx} \\ \mathbf{x} - \mathbf{x}^* \end{pmatrix}, \tag{32}$$

Then, for arbitrary $(\mathbf{x}, \mathbf{r}, \boldsymbol{\lambda})$*, we have* $(\mathbf{x}, \mathbf{r}, \boldsymbol{\lambda}) \in \Omega^*(\mathbf{x}^*)$ *if and only if* $\mathbf{R}_{\mathbf{x}^*}(\mathbf{x}, \mathbf{r}, \boldsymbol{\lambda}) = \mathbf{0}$.

Since functions $h(\cdot)$ and $g(\cdot)$ are the indicator functions of the closed and convex sets, the Moreau-Yosida proximal mappings $\mathbf{Pr}_h(\cdot)$ and $\mathbf{Pr}_g(\cdot)$ are projection mappings onto a convex set and therefore piecewise polyhedral by Proposition 12.30 in [7]. Considering the fact that mappings $\boldsymbol{\lambda}_s + \nabla U(\mathbf{x}^*)$, $\mathbf{A}_s^T\boldsymbol{\lambda}_s + \mathbf{A}_r^T\boldsymbol{\lambda}_r$, $\mathbf{Ar} + \mathbf{Bx}$ and $\mathbf{x} - \mathbf{x}^*$ are affine, the set-valued mapping $\mathbf{R}_{\mathbf{x}^*}(\cdot)$ is therefore piecewise polyhedral, and so is $\mathbf{R}_{\mathbf{x}^*}^{-1}(\cdot)$. Then, from the result of Lemma 4.8, we can regard the subset $\Omega^*(\mathbf{x}^*)$ as $\mathbf{R}_{\mathbf{x}^*}^{-1}(\mathbf{0})$ and utilize the calmness result in Lemma 4.7 to upper bound the distance between the current iterates and the set $\Omega^*(\mathbf{x}^*)$ by the constraint violation $\|\mathbf{R}_{\mathbf{x}^*}(\mathbf{x}[t], \mathbf{r}[t], \boldsymbol{\lambda}[t])\|$. Formally, we have the following global error bound.

Lemma 4.9. (Global error bound) *Assume Assumptions 1 and 2. If* $\tau \in [1, (\sqrt{5}+1)/2)$ *and parameter* $\beta^d_{m,n} > \deg(m) + \deg(n)$, *then there exists a constant* $\kappa > 0$ *such that the sequence* $(\mathbf{x}[t], \mathbf{r}[t], \boldsymbol{\lambda}[t])$ *generated by Algorithm 1 satisfies*

$$\text{dist}^2((\mathbf{x}[t], \mathbf{r}[t], \boldsymbol{\lambda}[t]), \Omega^*) \leq \kappa \|\mathbf{R}_{\mathbf{x}^*}(\mathbf{x}[t], \mathbf{r}[t], \boldsymbol{\lambda}[t])\|^2, t \geq 1, \tag{33}$$

where $\mathbf{x}^*$ *is an arbitrary optimal injection rate vector and the distance function is defined as*

$$\text{dist}^2((\mathbf{x}[t], \mathbf{r}[t], \boldsymbol{\lambda}[t]), \Omega^*) \triangleq \inf_{(\mathbf{x},\mathbf{r},\boldsymbol{\lambda})\in\Omega^*} \left\| \begin{bmatrix} \mathbf{x} \\ \mathbf{r} \\ \boldsymbol{\lambda} \end{bmatrix} - \begin{bmatrix} \mathbf{x}[t] \\ \mathbf{r}[t] \\ \boldsymbol{\lambda}[t] \end{bmatrix} \right\|^2, \tag{34}$$

We finally upper bound the residual $\|\mathbf{R}_{\mathbf{x}^*}(\mathbf{x}[t], \mathbf{r}[t], \boldsymbol{\lambda}[t])\|$ by the existing terms in lower bound C and combine the results in Lemma 4.4 and Lemma 4.9 to establish the global and linear convergence rate in Theorem 3.2. The detailed proof can be seen in Appendix E. An example of the key proof idea and the global error bound are illustrated in Fig 2.

Remark 1. *The existing work [6] also utilizes the isolated calmness to show that the ADMM converges at a linear rate when applying to the quadratic programs. In the quadratic programs, the optimal solution set can be characterized by a piecewise polyhedral mapping and one can straightforwardly apply the isolated calmness property. However, in our problem, due to lack of such a curvature information, the optimal solution set* Ω^* *is not piecewise polyhedral. The key in our proof is to find a subset of* Ω^* *that is also piecewise polyhedral.*

4.5 Queue Stability Analysis

Based on the evolution of physical queue length (5), we have the following inequality for each queue.

$$Q^d_n[t] \leq \left[Q^d_n[t-1] - \sum_{l\in O(n)} r^d_l[t] \right]_+ + \sum_{l\in I(n)} r^d_l[t] + \sum_{f\in\mathcal{F}} x_f[t]\mathbf{1}_{\{s_f=n, d_f=d\}}. \tag{35}$$

In the proof of Theorem 3.2, we have shown that the quantity $\mathbf{Bx}[t] + \mathbf{Ar}[t] \leq O(c^t)$, which implies that the change of physical queue length vanishes exponentially. This observation provides a simple path to establish the boundedness of physical queue length. However, it requires the assumption that utility function is strictly concave. In the sequel, we provide a different path, which only assumes the weakly concavity of the utility function. The following technical lemma connects the boundedness of the physical queue length $Q^d_n[t]$ and the virtual queue length $\lambda^d_n[t]$.

Lemma 4.10. *For each destination* $d \in \mathcal{D}$ *and node* $n \in \mathcal{N}\backslash d$, *suppose that* $\lambda^d_n[t]$ *and* $Q^d_n[t]$ *evolves by (11) and (35) with initializations* $\lambda^d_n[t] = Q^d_n[t] = 0$. *If there exists a constant* $M > 0$ *such that* $|\lambda^d_n[t]| < M, \forall t, d \in \mathcal{D}, n \in \mathcal{N}\backslash d$, *then*

$$Q^d_n[t] \leq \frac{2M}{\rho\tau} + B, \forall t, d \in \mathcal{D}, n \in \mathcal{N}\backslash d. \tag{36}$$

where B *is the constant dependent on the largest link capacity.*

From Theorem 4.5, we know that the virtual queue length $\boldsymbol{\lambda}[t]$ converges to an optimal dual variable $\boldsymbol{\lambda}^*$ and we can obtain that

$$|\lambda^d_n[t]| \leq \|\boldsymbol{\lambda}[t]\| = \|\boldsymbol{\lambda}[t] - \boldsymbol{\lambda}^* + \boldsymbol{\lambda}^*\| \leq \|\boldsymbol{\lambda}[t] - \boldsymbol{\lambda}^*\| + \|\boldsymbol{\lambda}^*\|.$$

Based on the result in Lemma 4.4, the function $V(\mathbf{x}[t], \mathbf{r}[t], \boldsymbol{\lambda}[t])$ is monotonically decreasing with respect to t. Then we have

$$|\lambda^d_n[t]| \leq \rho\tau V(\mathbf{x}[t], \mathbf{r}[t], \boldsymbol{\lambda}[t]) + \|\boldsymbol{\lambda}^*\|$$

$$\leq \rho\tau V(\mathbf{x}[0], \mathbf{r}[0], \boldsymbol{\lambda}[0]) + \|\boldsymbol{\lambda}^*\| \triangleq M, \forall t \geq 1.$$

which is a finite constant dependent on the initial distance to the optimal solution set Ω^*. Therefore, combining the result in Lemma 4.10, one can conclude that the physical queue length for each node and destination is finite.

5 EFFICIENT SUBPROBLEM SOLVER

In this section, we develop several efficient algorithms to solve the congestion control, routing and scheduling components in Algorithm 1.

5.1 Congestion Control

The congestion control component is an one-dimensional optimization problem, which can be efficiently solved by Newton method or Fibonacci search. Moreover, if the utility function takes a specific form such as the weighted proportional fair utilities, $U_f(x_f) = w_f \log(x_f), x_f > 0$, the solution can be obtained in a close-form expression,

$$x_f[t] = \frac{x_f[t-1]}{2} - \frac{z_{s_f}^{d_f}[t] + \Delta r_f[t]}{2\rho} + \sqrt{\frac{w_f}{\rho} + \frac{(z_{s_f}^{d_f}[t] + \Delta r_f[t] - \rho x_f[t-1])^2}{4\rho^2}}. \tag{37}$$

5.2 New Backpressure Routing in Wireline Network

In the wireline network, there exist no interference among different links and the achievable rate region C is given by the following form [25, 37].

$$C = \left\{ [r_l^d] \middle| \sum_{d=1}^{D} r_l^d \leq C_l, \forall l \right\}, \tag{38}$$

where C_l is the capacity of link l. Then both the objective function and the constraints of problem (8) are separable among the rate vectors in different links. Therefore, the link rate $r_l^d[t]$ can be determined in a distributed fashion: for each link $l = (m, n)$, solving the following quadratic program to obtain $[r_{m,n}^d[t], d \in \mathcal{D}]$.

$$\begin{aligned} \max_{r_{m,n}^d} \quad & \sum_{d \in \mathcal{D}} (z_m^d[t] - z_n^d[t]) r_{m,n}^d - \frac{\rho\beta_{m,n}}{2} (r_{m,n}^d - r_{m,n}^d[t-1])^2 \\ s.t. \quad & \textstyle\sum_d r_{m,n}^d \leq C_{m,n}, r_{m,n}^d \geq 0, \forall d. \end{aligned} \tag{39}$$

We define this problem as the **new backpressure routing** problem. After rearrangement of the terms, it can be formulated as a problem that projects the point $(r_{m,n}^d[t-1] + (z_m^d[t] - z_n^d[t])/\rho\beta_{m,n})$ onto a simplex defined in (39), which has already been investigated in [8].

Lemma 5.1. (solution of routing component) *For each link $l = (m, n) \in \mathcal{L}$, the solution of new backpressure routing has the form of $r_{m,n}^d[t] = [r_{m,n}^d[t-1] + (z_m^d[t] - z_n^d[t])/\rho\beta_{m,n} - \theta^*]_+$, where θ^* can be determined in $O(F \log(F))$ time.*

The main procedure to solve problem (39) are listed in Algorithm 2. Note that the step 1-4 and step 6-8 have $O(F)$ complexity and hence the overall complexity of Algorithm 2 is dominated by the sorting step 5 with complexity $O(F \log(F))$.

5.3 New Scheduling in Wireless Network

In the wireless network, different links cannot be simultaneously activated due to the existence of interference. Therefore, in addition to the rate assignment at each link, we need to schedule the link itself. The basic challenge to solve the scheduling component is that the the number of feasible

Algorithm 2: New backpressure routing algorithm

Let $x_d = [r^d_{m,n}[t-1] + (z^d_m[t] - z^d_n[t])/\rho\beta_{m,n}]_+, \forall d \in \mathcal{D}$.
if $\sum_{d=1}^{D} x_d \le C_{m,n}$ **then**
    Let $\theta^* = 0$ and $r^d_{m,n}[t] = x^d, \forall d \in \mathcal{D}$ and terminate algorithm.
end if
Sort $\{x_d, d \in \mathcal{D}\}$ in an decreasing order π such that $x_{\pi(1)} \ge x_{\pi(2)} \ge \cdots \ge x_{\pi(D)}$.
Find $p = \max\left\{k \in [D] \middle| x_{\pi(k)} - \frac{1}{k}\left(\sum_{d=1}^{k} x_{\pi(d)} - C_{m,n}\right) > 0\right\}$.
Let $\theta^* = \frac{1}{p}\left(\sum_{d=1}^{p} x_{\pi(d)} - C_{m,n}\right)$.
Output $r^d_{m,n}[t] = [r^d_{m,n}[t-1] + (z^d_m[t] - z^d_n[t])/\rho\beta_{m,n} - \theta^*]_+, \forall d \in \mathcal{D}$.

link rate vectors $|\Gamma|$ is possibly exponentially large. For example, in the one-hop node-exclusive model [20], all the feasible link rate vectors correspond to all the matchings in the graph $\mathcal{G}$, which could be $O(2^L)$ even in the bipartite graph. In the QCA method, the scheduling component is the classical MaxWeight scheduling, and the objective function is linear and such a problem can be reduced to some classical combinatorial problems such as maximum weighted matching. Instead, in our scheduling component (8), the objective function is quadratic, and the optimal solution may not belong to the vertex set Γ of the convex hull C, which poses a significant challenge in solving this problem. However, utilizing the idea of ellipsoid method, we will show a surprising result that the complexity of solving our new scheduling component (8) is equivalent to the complexity of solving the traditional MaxWeight scheduling problem.

Before presenting our main result, we first briefly introduce several concepts and technical tools in geometric algorithms [12] that will be used in the sequel.

Definition 5.2. (Separation oracle) Let H be a non-empty convex polyhedron in $\mathbb{R}^n$. A separation oracle for H is that, given any $\mathbf{x} \in \mathbb{R}^n$, it either outputs $\mathbf{x} \in H$, and if not, find a hyperplane such that $\mathbf{c}^T\mathbf{x} > \mathbf{c}^T\mathbf{y}, \forall \mathbf{y} \in H$.

Lemma 5.3. (Separation and optimization) *Let H be a non-empty convex polyhedron in $\mathbb{R}^n$ and $f(\cdot)$ be a convex function in $\mathbb{R}^n$. **If** the separation oracle for H can be solved in* $\text{poly}(n)$ *time, then we can compute an $\mathbf{x}$ with $B(\mathbf{x}, \delta) \in H$ and $\max_{\mathbf{y}\in H} f(\mathbf{y}) - f(\mathbf{x}) \ge \delta$ in* $\text{poly}(n, \log(\delta^{-1}))$ *time.*

In this lemma, $B(\mathbf{x}, \delta)$ is the ball centering at $\mathbf{x}$ with radius δ, where the δ is the finite truncation error from irrational number to rational number.

Theorem 5.4. *Assume that the feasible link rate vector $\mathbf{r}^{(i)} \in \mathbb{N}^L, \forall \mathbf{r}^{(i)} \in \Gamma$. There is a* $\text{poly}(L, F)$ *time algorithm to compute the new scheduling component (8) **if and only if** there is a* $\text{poly}(L, F)$ *time algorithm to compute the MaxWeight scheduling problem (13).*

In practice, the link rate always refers to the number of transmitted packets, hence the integer assumption on the feasible link rate vector is reasonable. Theorem 5.4 shows that *our quadratic scheduling component is not much "harder" than the traditional MaxWeight scheduling problem.* Therefore, we can establish the hardness of our new scheduling problem based on all existing complexity results of MaxWeight scheduling. For example, under the node-exclusive interference model, the MaxWeight scheduling is actually a maximum weighted matching problem that can be solved in polynomial time [20]. This result implies that problem (8) can also be solved in polynomial time. Another example is the Maximum Weighted K-Valid Matching problem introduced in [33] to characterize the multi-hop interference. They show that this problem is NP-hard when we have at least 2−hop interference, which implies that the problem (8) is also NP-hard.

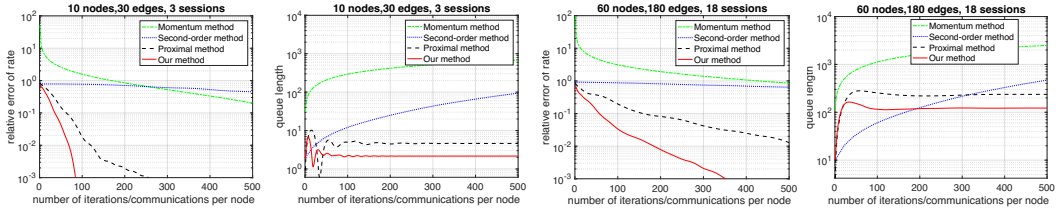

Fig. 3. Comparison of Algorithm 1 and existing methods in a small-scale and a medium-scale wireline networks.

Table 2. Comparison of Convergence Speed and Queue Length per Link

Problem size	Momentum method		Second-order method		Proximal method		Our method	
	# Iterations	Queue len	# Iterations	Queue len	# Iterations	Queue len	# Iterations	Queue len
(50, 150, 10)	4658	22.5	9600	35.1	369	1.10	207	0.66
(100, 300, 20)	9594	82.6	38900	145.2	512	1.94	298	0.83
(500, 1500, 100)	$> 10^5$	$> 10^3$	$> 10^5$	$> 10^3$	853	8.15	371	3.92
(1000, 3000, 200)	$> 10^5$	$> 10^4$	$> 10^5$	$> 10^4$	1921	15.30	639	6.61
benchmark	1044	31.2	1510	29.5	102	1.08	82	0.58

Remark 2. *The Theorem 5.4 has exhibited the complexity equivalence between the new scheduling component and the traditional MaxWeight scheduling. However, such a poly(L, F) time reduction is based on the ellipsoid method and requires high complexity. In practice, one can solve the new scheduling component (8) by several efficient optimization techniques. For example, both the subgradient method with a dual averaging [28] and Frank Wolfe [16] algorithm can reduce problem (8) to $O(1/\epsilon)$ MaxWeight problem, where ϵ is the accuracy.*

The rest of challenge is the implementation issue incurred by the non-integer solution of (8), because the optimal point may not lie in the set of feasible link rate vectors. We next show that this problem can be tackled by connecting the practical time sharing technique and the convex decomposition technique in the combinatorial optimization.

Lemma 5.5. *If there is a* poly(L, F) *time algorithm to compute the MaxWeight scheduling problem (13), then there is a* poly(L, F) *time algorithm that, given any optimal solution $\mathbf{r}^*$ of (8), yields $(L+1)$ feasible rate vectors $\mathbf{r}^{(i)} \in \Gamma$ such that $[\sum_d r_l^{d*}] = \sum_{i=1}^{L+1} \tau_i \mathbf{r}^{(i)}$ and $\sum_{i=1}^{L+1} \tau_i = 1, \tau_i \geq 0$.*

The proof of Lemma 5.5 is a straightforward application of Theorem 5.4 and the polynomial time reduction from the convex decomposition of point within a polyhedron H to the separation oracle problem [11]. Based on this result, we can first divide the time slots into mini slots and then operate the link rate vector $\mathbf{r}^{(i)}$ in τ_i fraction of time. For each link l, given the link rate vector $r_l^{(i)}$, the specific rate assignment for each $d-$destined packets can be determined by solving a problem same as (39) (only change C_l to $r_l^{(i)}$).

6 NUMERICAL ANALYSIS

In this section, we conduct some numerical studies to verify the theoretical improvements of our proposed method compared with the state-of-arts.

6.1 Simulation Setup

We adopt the well-known weighted proportional fair utilities $U_f(x_f) = w_f \log(x_f)$, where the weight w_f of each flow f is randomly generated from a uniform distribution $U(0, 1)$. The network typology $\mathcal{G} = (\mathcal{N}, \mathcal{L})$ is generated by the classic Erdős-Rényi (ER) random graph model $G(n, p)$, where n is the number of nodes and p is the connected probability between two nodes (we only

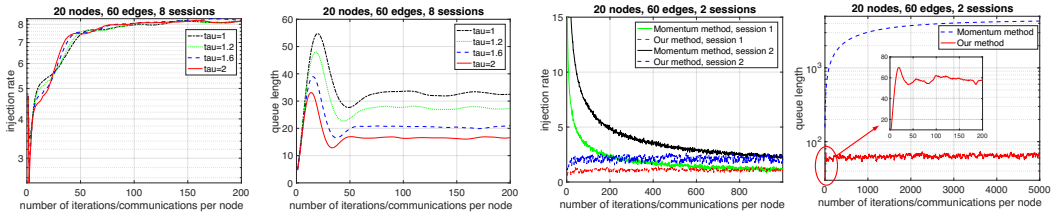

Fig. 4. The two left figures shows the impact of parameter τ on convergence and queue length. The two right figures compare our algorithm and the momentum method for a wireless network with fading channel.

consider the connected graph). We compare our algorithm with the following three benchmark algorithms.

Momentum method: Here the Momentum method refers to the Heavy-ball algorithm proposed in [24]. The existing works [23, 24] have shown that this method produces significantly faster convergence speed and lower queuing delay compared to the traditional QCA method[3].

Second-order method: There exists several versions of second-order algorithms [25, 26, 36] in solving this problem. We use the one with the fastest convergence speed proposed in [25]. This algorithm has a two-layered iteration structure: (i) each outer iteration corresponds to one Newton step; (ii) a Sherman-Morrison-Woodbury (SMW) based inner iteration to determine the Newton direction.

Proximal method: We use the one proposed in [37]. They have shown superior performance in the queue length reduction and improvement of convergence speed than the QCA method in the wireline network.

We adopt the following two comparison metrics: (i) the relative error of injection rate: $\|\mathbf{x}[t] - \mathbf{x}^*\|/\|\mathbf{x}^*\|$, where the $\mathbf{x}^*$ is obtained approximately by running our method with a strict stopping condition; (ii) total physical queue length of all nodes and all flows: $\sum_{d\in\mathcal{D}}\sum_{n\in\mathcal{N}\setminus d} Q_n^d[t]$. In the simulation, each iteration refers to one communication per node. For our method, momentum method and proximal method, each iteration refers to solving one congestion control and routing component. For the second-order method, each iteration refers to one SMW-based iteration.

6.2 Wireline Network

We first compare our algorithm with above three algorithms in a wireline network with link capacity C_l randomly generated from a uniform distribution $U(0, 1)$. As shown in Figure 3, we plot the relative error of rate and the total physical queue length versus the number of iterations under a small-scale network (10 nodes, 30 edges, 3 sessions) and a medium-scale network (60 nodes, 180 edges, 18 sessions). For the momentum method and second-order method, we choose parameter K and μ large enough to guarantee the utility optimality gap is less than 0.1%. For proximal method, we choose parameter $\alpha_n = (d_n + 1)/2$. It can be observed that our proposed algorithm converges at a global and linear rate with bounded physical queue length, which matches our theoretical results. Moreover, it produces the fastest convergence speed and lowest physical queue length among all the existing methods. Although the second-order method has only $40 - 80$ outer iterations (newton step), it still converges quite slowly due to the large number of inner iterations in computing the Newton direction. Another observation is that our method, proximal method and second-order method gradually increases the injection rates to the optimal point, instead, the momentum method

[3]Hence, we don't compare our algorithm with QCA.

first produces an extremely high injection rate, then gradually decrease it, which leads to a large physical queue length.

As we will discuss in the next section, the proximal algorithm can be recovered by a kind of inexact Uzawa ADMM with Jacobi updates, instead, our method is the inexact Uzawa ADMM with Gauss-Seidel updates. Therefore, our method is much efficient compared with the proximal method since the Gauss-Seidel updates is usually faster than the Jacobi updates in the 2−block ADMM [5]. Besides, the existing analysis shows that the ADMM with Jacobi update converges sub-linearly. Interestingly, in our simulations one observation is that this algorithm exhibits a linear convergence rate.

We then investigate the impact of the network size and compare our algorithm with the existing methods in number of iterations and physical queue lengths to obtain solution with a given accuracy. The stopping criterion is that both the relative error of rate $\|\mathbf{x}[t] - \mathbf{x}^*\|/\|\mathbf{x}^*\|$ and the constraint violation $\|\mathbf{B}\mathbf{x}[t] + \mathbf{A}\mathbf{r}[t]\|$ is less than 1%. Similarly, we set parameters K and μ of momentum method and second-order method large enough to guarantee the desired utility optimality gap. To avoid the random noises, we randomly generate 1000 instances in each problem size and take the average. Besides, we also use a benchmark network in [37], whose optimal solution is known. The results are listed in TABLE 2. Note that the queue length is normalized by the number of links. It can be observed that our proposed algorithm exhibits a $10 - 10^3$ order of improvement of both convergence speed and queue length compared with the momentum method and the second-order method. It also converges $2 - 3$ times faster and produces $40\% - 60\%$ less physical queue length than the proximal method. Moreover, our algorithm has an effect of relieving the curse of dimensionality in the traditional algorithms. For example, when the problem size increases 20 times (from the first instance to fourth instance), the number of iterations only increases 3 times.

6.3 Impact of Parameter

We next investigate the impact of parameter τ on the convergence speed and the queue length of our algorithm. Theorem 4.5 shows that the convergence of our algorithm is guaranteed when $\tau \in [1, (\sqrt{5}+1)/2)$. We test our algorithm in a 20−nodes 60−links and 8−sessions network with $\tau = \{1, 1.2, 1.6, 2.0\}$ and plot the sum of injection rate and queue length versus the number of iterations in Fig. 4. The basic observation is that when the parameter τ increases, the convergence speed of our algorithm will slightly increase and the queue length will decrease at an inversely proportional manner, which roughly matches the upper bound of physical queue length provided in Lemma 4.10 that $Q_n^d[t] \leq \frac{2M}{\rho\tau}$+constant. For example, when τ increases from 1 to 2, the queue length is reduced roughly 40%. However, from the simulation, we observe that when $\tau \geq (\sqrt{5}+1)/2$, the algorithm sometimes diverges. Therefore, we suggest a safe value $\tau = 1.618$ when using our algorithm.

6.4 Wireless Network

From the methods compared above, only the momentum method and our algorithm can be applied to the wireless networks with interference constraints. We compare our algorithm with it in a 20−nodes 60−links and 2−sessions wireless network with quasi-static block fading (channel states vary from one slot to the next but remain constant in each slot). We plot the injection rate of each session and sum of queue length versus the number of iterations in Fig. 4. It can be observed that our algorithm converges to the steady state in less than 50 iterations and the momentum method requires at least 5000 iterations. Moreover, our algorithm produces only 1% queue length compared to the momentum method.

7 DISCUSSIONS

We now discuss the connection of our algorithm to the existing proximal method and list some follow-up works as well as directions for future research.

7.1 Connection to Proximal Method

In the scenario of wireline networks, the existing proximal methods [21, 37] also contain a quadratic regularizer in the congestion control and routing component. Interestingly, we find that this method can be recovered by a kind of proximal linear ADMM with following Jacobi (parallel) updates.

(1) $\mathbf{x}[t] = \arg\min_{\mathbf{x}} L(\mathbf{x}, \mathbf{r}[t-1], \boldsymbol{\lambda}[t-1])$.
(2) $\mathbf{r}[t] = \arg\min_{\mathbf{r}} L(\mathbf{x}[t-1], \mathbf{r}, \boldsymbol{\lambda}[t-1]) + \frac{1}{2}\|\mathbf{r} - \mathbf{r}[t]\|_{\mathbf{Q}}^2$.
(3) $\boldsymbol{\lambda}[t] = \boldsymbol{\lambda}[t-1] - \rho(\mathbf{B}\mathbf{x}[t] + \mathbf{A}\mathbf{r}[t])$

The function $L(\cdot)$ is the Augmented Lagrangian function defined in (18). The matrix $\mathbf{Q} = \rho(\mathbf{M} - \mathbf{A}^T\mathbf{A})$ and matrix $\mathbf{M} = \text{diag}\{\ldots, \beta_l^d, \ldots\}$. Therefore, we can use the existing analysis in [4] to establish a stronger theoretical result that the proximal algorithm in [37] actually converges in a non-ergodic sublinear rate $o(1/\epsilon)$.

7.2 Stochastic Network Optimization

In the reality, the channel conditions will fluctuate due to the environmental changes (e.g., fading). To accommodate this situation, we assume that there exists a finite set $\mathcal{J}$ of states that channel conditions can be in. Let Γ_j denote the set of feasible link rates in state j and π_j be the stationary probability of jth channel state. We define the following average capacity region.

$$C = \sum_{j \in \mathcal{J}} \pi_j \text{Conv}(\Gamma_j).$$

Then, the problem becomes an optimization problem over this new capacity region. Accordingly, the routing and scheduling components in each time slot t can be modified to an optimization problem over instantaneous region $C[t]$. Although the numerical results have already exhibited improved performance over existing algorithms, the theoretical performance under this setting is unknown. One possible approach is to utilize some stochastic Alternating Directional Method of Multipliers. However, the challenge is that all existing stochastic ADMMs can only be applied to the smooth stochastic objective function, which is not the case in this problem.

8 CONCLUSION

In this paper, we have proposed a new joint congestion control, routing and scheduling algorithmic framework for distributed network optimization based on an inexact Uzawa method of the Alternating Directional Method of Multiplier. This algorithm offers zero utility optimality gap with finite queue length, the fastest convergence speed to date, i.e., $O(\log(1/\epsilon))$ iterations, among all the existing algorithms. Moreover, the virtual queue-based control provides an extremely low-complexity implementation of this algorithm. These results build a deep connection between the cross-layer decomposition of network optimization and the variable splitting in the multi-block Alternating Directional Method of Multiplier. One important theoretical contribution is that we prove that the ADMM with an inexact Uzawa method converges globally and linearly without requiring the full rank assumption of constraint matrix.

ACKNOWLEDGMENTS

The work is supported by the Office of Naval Research under Grant No.: N00014-17-1-2417 and N00014-15-1-2166, the Defense Thrust Reduction Agency under Grant No.: HDTRA1-14-1-0058 and the National Science Foundation under Grant No.: CNS-1719371 and CNS-1518829.

A APPENDIX

A.1 Proof of Lemma 4.4

The first step of the Algorithm 1 is

$$\begin{aligned}
&\mathbf{r}[t] = \arg\min_{\mathbf{r}} L(\mathbf{x}[t-1], \mathbf{r}, \boldsymbol{\lambda}[t-1]) + \frac{1}{2}\|\mathbf{r}-\mathbf{r}[t-1]\|_{\mathbf{Q}}^2 \\
\overset{(a)}{\Longleftrightarrow}\ &\mathbf{r}[t] = \arg\min_{\mathbf{r}} g(\mathbf{r}) + \frac{\rho}{2}\left\|\begin{bmatrix}\mathbf{A}_s\mathbf{r}-\mathbf{x}[t-1]-\boldsymbol{\lambda}_s[t-1]/\rho \\ \mathbf{A}_r\mathbf{r}-\boldsymbol{\lambda}_r[t-1]/\rho\end{bmatrix}\right\|^2 + \frac{1}{2}\|\mathbf{r}-\mathbf{r}[t-1]\|_{\mathbf{Q}}^2 \\
\overset{(b)}{\Longleftrightarrow}\ &\mathbf{A}_s^T\left[\boldsymbol{\lambda}_s[t-1]-\rho(\mathbf{A}_s\mathbf{r}[t]-\mathbf{x}[t-1])\right] + \mathbf{A}_r^T\left(\boldsymbol{\lambda}_r[t-1]-\rho\mathbf{A}_r\mathbf{r}[t]\right) + \mathbf{Q}(\mathbf{r}[t-1]-\mathbf{r}[t]) \in \partial g(\mathbf{r}[t]) \\
\overset{(c)}{\Longleftrightarrow}\ &\mathbf{A}_s^T\left[\bar{\boldsymbol{\lambda}}_s[t]-\rho(\mathbf{x}[t]-\mathbf{x}[t-1])\right] + \mathbf{A}_r^T\bar{\boldsymbol{\lambda}}_r[t] + \mathbf{Q}(\mathbf{r}[t-1]-\mathbf{r}[t]) \in \partial g(\mathbf{r}[t]). \qquad (40)
\end{aligned}$$

The above, step (a) utilizes definition of the Augmented Lagrangian function (18), step (b) is based on the first-order optimality condition, step (c) is based on the following definition of variables $\bar{\boldsymbol{\lambda}}_s[t]$ and $\bar{\boldsymbol{\lambda}}_r[t]$.

$$\bar{\boldsymbol{\lambda}}_s[t] = \boldsymbol{\lambda}_s[t-1] - \rho(\mathbf{A}_s\mathbf{r}[t]-\mathbf{x}[t]), \qquad (41)$$

$$\bar{\boldsymbol{\lambda}}_r[t] = \boldsymbol{\lambda}_r[t-1] - \rho\mathbf{A}_r\mathbf{r}[t]. \qquad (42)$$

Similarly, based on the the first-order optimality condition and the definition of the variable $\bar{\boldsymbol{\lambda}}_s[t]$ in (41), the second step of the Algorithm 1 is

$$\begin{aligned}
&\mathbf{x}[t] = \arg\min_{\mathbf{x}} f(\mathbf{x}) + \frac{\rho}{2}\|\mathbf{A}_s\mathbf{r}[t]-\mathbf{x}-\boldsymbol{\lambda}_s[t-1]/\rho\|^2 \\
\Longleftrightarrow\ &-\bar{\boldsymbol{\lambda}}_s[t] \in \partial f(\mathbf{x}[t]). \qquad (43)
\end{aligned}$$

Combining the KKT condition (23) that $\mathbf{A}_r^T\boldsymbol{\lambda}_r^* + \mathbf{A}_s^T\boldsymbol{\lambda}_s^* \in \partial g(\mathbf{r}^*)$, the optimality condition (40) in the second step of ADMM, we have

$$\langle \mathbf{r}[t]-\mathbf{r}^*, \mathbf{A}_s^T\left[\bar{\boldsymbol{\lambda}}_s[t]-\boldsymbol{\lambda}_s^*-\rho(\mathbf{x}[t]-\mathbf{x}[t-1])\right] + \mathbf{A}_r^T(\bar{\boldsymbol{\lambda}}_r[t]-\boldsymbol{\lambda}_r^*) + \mathbf{Q}(\mathbf{r}[t-1]-\mathbf{r}[t])\rangle \overset{(a)}{\geq} \mathbf{0}. \qquad (44)$$

The above, (a) utilizes the fact that function $h(\mathbf{x})$ is convex and the subdifferential of a convex function is a monotone operator, i.e., the inequality (16) holds with $v = 0$.

Similarly, combining the result of KKT condition (22) that $-\boldsymbol{\lambda}_r^* \in \partial f(\mathbf{x}^*)$ and the optimality condition (43) in the second step of ADMM, we have

$$\langle \mathbf{x}[t]-\mathbf{x}^*, \bar{\boldsymbol{\lambda}}_s[t]-\boldsymbol{\lambda}_s^*[t]\rangle \overset{(a)}{\leq} -v\|\mathbf{x}[t]-\mathbf{x}^*\|^2, v > 0. \qquad (45)$$

The above, step (a) utilizes the fact that function $f(\mathbf{x})$ is convex with modulus v and inequality (16).

Then, change the direction of inequality (45) and sum it with inequality (44), we have

$$\begin{aligned}
&\langle \mathbf{x}[t]-\mathbf{x}^*, \boldsymbol{\lambda}_s^*[t]-\bar{\boldsymbol{\lambda}}_s[t]\rangle + \langle \mathbf{r}[t]-\mathbf{r}^*, \mathbf{Q}(\mathbf{r}[t-1]-\mathbf{r}[t])\rangle + \langle \mathbf{r}[t]-\mathbf{r}^*, \mathbf{A}_s^T\left[\bar{\boldsymbol{\lambda}}_s[t]-\boldsymbol{\lambda}_s^*-\rho(\mathbf{x}[t]\right. \\
&\left.-\mathbf{x}[t-1])\right]\rangle + \langle \mathbf{r}[t]-\mathbf{r}^*, \mathbf{A}_r^T(\bar{\boldsymbol{\lambda}}_r[t]-\boldsymbol{\lambda}_r^*)\rangle \geq v\|\mathbf{x}[t]-\mathbf{x}^*\|^2 \\
\overset{(a)}{\Longleftrightarrow}\ &\frac{1}{\rho}\langle \boldsymbol{\lambda}_s[t-1]-\bar{\boldsymbol{\lambda}}_s[t], \bar{\boldsymbol{\lambda}}_s[t]-\boldsymbol{\lambda}_s^*\rangle + \rho\langle \mathbf{x}^*-\mathbf{x}[t], \mathbf{x}[t]-\mathbf{x}[t-1]\rangle + \langle \mathbf{r}[t]-\mathbf{r}^*, \mathbf{Q}(\mathbf{r}[t-1]-\mathbf{r}[t])\rangle
\end{aligned}$$

$$+\frac{1}{\rho}\langle\boldsymbol{\lambda}_r[t-1]-\bar{\boldsymbol{\lambda}}_r[t],\bar{\boldsymbol{\lambda}}_r[t]-\boldsymbol{\lambda}_r^*[t]\rangle\geq v\|\mathbf{x}[t]-\mathbf{x}^*\|^2+\langle\boldsymbol{\lambda}_s[t-1]-\bar{\boldsymbol{\lambda}}_s[t],\mathbf{x}[t]-\mathbf{x}[t-1]\rangle$$

$$\overset{(b)}{\Longleftrightarrow}\frac{1}{\rho}\langle\boldsymbol{\lambda}_s[t-1]-\bar{\boldsymbol{\lambda}}_s[t],\boldsymbol{\lambda}_s[t-1]-\boldsymbol{\lambda}_s^*\rangle+\rho\mathbf{x}[t-1]-\langle\mathbf{x}^*,\mathbf{x}[t-1]-\mathbf{x}[t]\rangle+\langle\mathbf{r}[t-1]-\mathbf{r}^*,$$

$$\mathbf{Q}(\mathbf{r}[t-1]-\mathbf{r}[t])\rangle+\frac{1}{\rho}\langle\boldsymbol{\lambda}_r[t-1]-\bar{\boldsymbol{\lambda}}_r[t],\boldsymbol{\lambda}_r[t-1]-\boldsymbol{\lambda}_r^*[t]\rangle\geq\frac{1}{\rho}\left\|\boldsymbol{\lambda}[t-1]-\bar{\boldsymbol{\lambda}}[t]\right\|^2+\rho\|\mathbf{x}[t-1]$$

$$-\mathbf{x}[t]\|^2+\|\mathbf{r}[t-1]-\mathbf{r}[t]\|_{\mathbf{Q}}^2+v\|\mathbf{x}[t]-\mathbf{x}^*\|^2+\langle\boldsymbol{\lambda}_s[t-1]-\bar{\boldsymbol{\lambda}}_s[t],\mathbf{x}[t]-\mathbf{x}[t-1]\rangle$$

$$\overset{(c)}{\Longleftrightarrow}\frac{1}{\rho\tau}\left(\|\boldsymbol{\lambda}[t-1]-\boldsymbol{\lambda}^*\|^2-\|\boldsymbol{\lambda}[t]-\boldsymbol{\lambda}^*\|^2\right)+\rho(\|\mathbf{x}[t-1]-\mathbf{x}^*\|^2-\|\mathbf{x}[t]-\mathbf{x}^*\|^2)+\|\mathbf{r}[t-1]-\mathbf{r}^*\|_{\mathbf{Q}}^2-$$

$$\|\mathbf{r}[t]-\mathbf{r}^*\|_{\mathbf{Q}}^2\geq\frac{2-\tau}{\rho}\|\boldsymbol{\lambda}[t-1]-\bar{\boldsymbol{\lambda}}[t]\|^2+\rho\|\mathbf{x}[t-1]-\mathbf{x}[t]\|^2+\|\mathbf{r}[t-1]-\mathbf{r}[t]\|_{\mathbf{Q}}^2+$$

$$2v\|\mathbf{x}[t]-\mathbf{x}^*\|^2+2\langle\boldsymbol{\lambda}_s[t-1]-\bar{\boldsymbol{\lambda}}_s[t],\mathbf{x}[t]-\mathbf{x}[t-1]\rangle. \tag{46}$$

The above, step (a) rearranges the terms in the original inequality and utilizes the definition of the variable $\bar{\boldsymbol{\lambda}}_s[t],\bar{\boldsymbol{\lambda}}_r[t]$ in (41), (42) and the KKT condition (24). The step (b) rearranges terms by writing $\mathbf{x}^*-\mathbf{x}[t]=\mathbf{x}^*-\mathbf{x}[t-1]+\mathbf{x}[t-1]-\mathbf{x}[t]$ (similarly for variables $\mathbf{r}[t]$ and $\bar{\boldsymbol{\lambda}}[t]$). The step (c) applies the three-point equality of Euclidean norms $\|\mathbf{x}-\mathbf{z}\|_{\mathbf{M}}^2-\|\mathbf{y}-\mathbf{z}\|_{\mathbf{M}}^2=2(\mathbf{x}-\mathbf{z})^T\mathbf{M}(\mathbf{x}-\mathbf{y})-\|\mathbf{x}-\mathbf{y}\|_{\mathbf{M}}^2$ to the left hand side of the inequality, and utilizes the equation $\boldsymbol{\lambda}[t-1]-\bar{\boldsymbol{\lambda}}[t]=(\boldsymbol{\lambda}[t-1]-\boldsymbol{\lambda}[t])/\tau$.

The rest is to show that the term $\langle\boldsymbol{\lambda}_s[t-1]-\bar{\boldsymbol{\lambda}}_s[t],\mathbf{x}[t]-\mathbf{x}[t-1]\rangle$ is lower bounded by certain form of other existing terms in (46) and the primal residual. Applying the equation (43) to the time slot $t-1$, we have

$$-\boldsymbol{\lambda}_s[t-2]+\rho(\mathbf{A}_s\mathbf{r}[t-1]-\mathbf{x}[t-1])\in\partial f(\mathbf{x}[t-1]).$$

Combing this result, equation (43) and inequality (16), we have

$$\langle\mathbf{x}[t-1]-\mathbf{x}[t],-\boldsymbol{\lambda}_s[t-2]+\rho(\mathbf{A}_s\mathbf{r}[t-1]-\mathbf{x}[t-1])+\bar{\boldsymbol{\lambda}}_s[t]\rangle\geq v\|\mathbf{x}[t-1]-\mathbf{x}[t]\|^2$$

$$\overset{(a)}{\Longleftrightarrow}\langle\mathbf{x}[t]-\mathbf{x}[t-1],\boldsymbol{\lambda}_s[t-1]-\bar{\boldsymbol{\lambda}}_s[t]\rangle\geq\langle\mathbf{x}[t]-\mathbf{x}[t-1],(1-\tau)\rho(\mathbf{A}_s\mathbf{r}[t-1]-\mathbf{x}[t-1])\rangle+$$

$$v\|\mathbf{x}[t-1]-\mathbf{x}[t]\|^2$$

$$\overset{(b)}{\Longleftrightarrow}\langle\mathbf{x}[t]-\mathbf{x}[t-1],\boldsymbol{\lambda}_s[t-1]-\bar{\boldsymbol{\lambda}}_s[t]\rangle\geq-\frac{\rho}{2\eta}\|\mathbf{A}_s\mathbf{r}[t-1]-\mathbf{x}[t-1]\|^2+$$

$$\left[v-\frac{(1-\tau)^2\rho\eta}{2}\right]\|\mathbf{x}[t]-\mathbf{x}[t-1]\|^2. \tag{47}$$

The above, step (a) is based on the virtual queue update $\boldsymbol{\lambda}_s[t-1]=\boldsymbol{\lambda}_s[t-2]-\rho\tau(\mathbf{A}_s\mathbf{r}[t-1]-\mathbf{x}[t-1])$, step (b) utilizes the following inequality.

$$\langle\sqrt{\rho\eta}(1-\tau)(\mathbf{x}[t-1]-\mathbf{x}[t]),\sqrt{\frac{\rho}{\eta}}(\mathbf{A}_s\mathbf{r}[t-1]-\mathbf{x}[t-1])\rangle\leq$$

$$\frac{\rho\eta(1-\tau)^2}{2}\|\mathbf{x}[t-1]-\mathbf{x}[t]\|^2+\frac{\rho}{2\eta}\|\mathbf{A}_s\mathbf{r}[t-1]-\mathbf{x}[t-1]\|^2,$$

where $\eta>1$ is an arbitrary constant. Substituting the above inequality into (46), we can finally obtain

$$V(\mathbf{x}[t-1],\mathbf{r}[t-1],\boldsymbol{\lambda}[t-1])-V(\mathbf{x}[t],\mathbf{r}[t],\boldsymbol{\lambda}[t])\geq\rho\left(2-\tau-\frac{1}{\eta}\right)\|\mathbf{A}_s\mathbf{r}[t]-\mathbf{x}[t]\|^2+$$

$$\frac{2-\tau}{\rho}\|\boldsymbol{\lambda}_r[t-1]-\bar{\boldsymbol{\lambda}}_r[t]\|^2+\rho\left[1-\eta(1-\tau)^2\right]\|\mathbf{x}[t]-\mathbf{x}[t-1]\|^2+\|\mathbf{r}[t-1]-\mathbf{r}[t]\|_{\mathbf{Q}}^2+$$

$$2v\|\mathbf{x}[t]-\mathbf{x}^*\|^2+2v\|\mathbf{x}[t]-\mathbf{x}[t-1]\|^2. \tag{48}$$

The existence of $\alpha>0$ can be guaranteed by $2-\tau-\frac{1}{\eta}>0$ and $1-\eta(1-\tau)^2>0$, or, equivalently, $\tau\in[1,(1+\sqrt{5})/2)$. Therefore, the lemma follows.

B PROOF OF THEOREM 4.5

By Lemma 4.4, if the parameter τ satisfies $\tau\in[1,(\sqrt{5}+1)/2)$, the function $V(\mathbf{x}[t],\mathbf{r}[t],\boldsymbol{\lambda}[t])$ is bounded. Then we have that $\|\boldsymbol{\lambda}[t]-\boldsymbol{\lambda}^*\|$, $\|\mathbf{x}[t]-\mathbf{x}^*\|$ and $\|\mathbf{r}[t]-\mathbf{r}^*\|_{\mathbf{Q}}^2$ are bounded, which implies that sequence $\boldsymbol{\lambda}[t]$ and $\mathbf{x}[t]$ are bounded. Based on the choice of parameter $\beta_{m,n}^d>\deg(m)+\deg(n)$, the matrix $\mathbf{Q}$ is positive definite, thus the sequence $\mathbf{r}[t]$ is also bounded. Being bounded, these sequences have the converging subsequences such that

$$\lim_{i\to\infty}\mathbf{x}[t_i]=\hat{\mathbf{x}},\ \lim_{i\to\infty}\mathbf{r}[t_i]=\hat{\mathbf{r}},\ \lim_{i\to\infty}\boldsymbol{\lambda}[t_i]=\hat{\boldsymbol{\lambda}}.$$

The function $V(\mathbf{x}[t],\mathbf{r}[t],\boldsymbol{\lambda}[t])$ is monotonically nonincreasing and thus converging. Due to the fact that $\alpha>0$, we have $\limsup\|\boldsymbol{\lambda}[t-1]-\boldsymbol{\lambda}[t]\|=0$, and then we have

$$\limsup\|\mathbf{A}_s\mathbf{r}[t]-\mathbf{x}[t]\|=\limsup\|\mathbf{A}_r\mathbf{r}[t]\|=0. \tag{49}$$

By passing the limit on (49) over subsequences, we have

$$\mathbf{A}_s\hat{\mathbf{r}}=\hat{\mathbf{x}},\mathbf{A}_r\hat{\mathbf{r}}=\mathbf{0}. \tag{50}$$

Similarly, we have $\limsup\|\mathbf{x}[t-1]-\mathbf{x}[t]\|=\limsup\|\mathbf{r}[t-1]-\mathbf{r}[t]\|=0$. Recall the optimality condition (40) and (43) of first and second step of ADMM, taking limit over the subsequence and applying Theorem 24.4 of [31], we obtain

$$-\hat{\boldsymbol{\lambda}}_s\in\partial f(\hat{\mathbf{x}}),\ \text{and}\ \mathbf{A}_s^T\hat{\boldsymbol{\lambda}}_s+\mathbf{A}_r^T\hat{\boldsymbol{\lambda}}_r\in\partial g(\hat{\mathbf{r}}). \tag{51}$$

Together with (50), $\hat{\mathbf{x}},\hat{\mathbf{r}},\hat{\boldsymbol{\lambda}}$ satisfy the KKT conditions of problem (17). Therefore, the theorem follows.

C PROOF OF LEMMA 4.8

Based on the fact that $f(\mathbf{x}^*)=U(\mathbf{x}^*)$, we have

$$\begin{aligned}
&\mathbf{R}_{\mathbf{x}^*}(\mathbf{x},\mathbf{r},\boldsymbol{\lambda})=\mathbf{0}\\
\Longleftrightarrow\ &\begin{cases}\mathbf{x}-\mathbf{Pr}_h(\mathbf{x}-\boldsymbol{\lambda}_s-\nabla U(\mathbf{x}^*))=\mathbf{0}\\ \mathbf{r}-\mathbf{Pr}_g(\mathbf{r}+\mathbf{A}_s^T\boldsymbol{\lambda}_s+\mathbf{A}_r^T\boldsymbol{\lambda}_r)=\mathbf{0}\\ \mathbf{A}_s\mathbf{r}-\mathbf{x}=\mathbf{0},\mathbf{A}_r\mathbf{r}=\mathbf{0}\\ \mathbf{x}=\mathbf{x}^*\end{cases}\\
\overset{(a)}{\Longleftrightarrow}\ &\begin{cases}-\boldsymbol{\lambda}_s\in\partial h(\mathbf{x})+\nabla U(\mathbf{x}^*)\\ \mathbf{A}_s^T\boldsymbol{\lambda}_s+\mathbf{A}_r^T\boldsymbol{\lambda}_r\in\partial g(\mathbf{r})\\ \mathbf{A}_s\mathbf{r}=\mathbf{x},\mathbf{A}_r\mathbf{r}=\mathbf{0}\\ \mathbf{x}=\mathbf{x}^*\end{cases}\\
\overset{(b)}{\Longleftrightarrow}\ &\begin{cases}-\boldsymbol{\lambda}_s\in\partial f(\mathbf{x})\\ \mathbf{A}_s^T\boldsymbol{\lambda}_s+\mathbf{A}_r^T\boldsymbol{\lambda}_r\in\partial g(\mathbf{r})\\ \mathbf{A}_s\mathbf{r}=\mathbf{x},\mathbf{A}_r\mathbf{r}=\mathbf{0}\\ \mathbf{x}=\mathbf{x}^*\end{cases}\\
\Longleftrightarrow\ &(\mathbf{x},\mathbf{r},\boldsymbol{\lambda})\in\Omega^*(\mathbf{x}^*).
\end{aligned} \tag{52}$$

The above, step (a) utilizes the definition of proximal mapping and the first-order optimality condition that

$$\begin{aligned}&\mathbf{x} = \arg\min_{\mathbf{u}} h(\mathbf{u}) + \frac{1}{2}\|\mathbf{u} - [\mathbf{x} - \boldsymbol{\lambda}_s - \nabla U(\mathbf{x}^*)]\|^2\\ \Longleftrightarrow\ &\mathbf{0} \in \partial h(\mathbf{x}) + \mathbf{x} - [\mathbf{x} - \boldsymbol{\lambda}_s - \nabla U(\mathbf{x}^*)],\end{aligned}$$

and

$$\begin{aligned}&\mathbf{r} = \arg\min_{\mathbf{u}} g(\mathbf{u}) + \frac{1}{2}\|\mathbf{u} - [\mathbf{r} + (\mathbf{A}_s^T\boldsymbol{\lambda}_s + \mathbf{A}_r^T\boldsymbol{\lambda}_r)]\|^2\\ \Longleftrightarrow\ &\mathbf{0} \in \partial g(\mathbf{r}) + \mathbf{r} - [\mathbf{r} + (\mathbf{A}_s^T\boldsymbol{\lambda}_s + \mathbf{A}_r^T\boldsymbol{\lambda}_r)].\end{aligned}$$

The step (b) is based on the following fact.

$$\begin{cases}-\boldsymbol{\lambda}_s \in \partial h(\mathbf{x}) + \nabla U(\mathbf{x}^*)\\ \mathbf{x} = \mathbf{x}^*\end{cases} \Longleftrightarrow \begin{cases}-\boldsymbol{\lambda}_s \in \partial f(\mathbf{x})\\ \mathbf{x} = \mathbf{x}^*\end{cases}.$$

Therefore, the lemma follows.

D PROOF OF LEMMA 4.9

For notational simplicity, let $\mathbf{u}[t] = (\mathbf{x}[t], \mathbf{r}[t], \boldsymbol{\lambda}[t])$. Based on the result of Lemma 4.7, there exists two constants κ_0 and η_0 such that, for all $\mathbf{u}[t] \in \{\mathbf{u}[t]|\mathbf{R}_{\mathbf{x}^*}(\mathbf{u}[t]) \leq \eta_0\}$,

$$\text{dist}^2\left(\mathbf{u}[t], \mathbf{R}_{\mathbf{x}^*}^{-1}(\mathbf{0})\right) \leq \kappa_0\|\mathbf{R}_{\mathbf{x}^*}(\mathbf{u}[t])\|^2. \tag{53}$$

From Theorem 4.5, we know that the sequence $\mathbf{u}[t]$ converges to a KKT point $\mathbf{u}^*$ with $\|\mathbf{u}[t] - \mathbf{u}^*\| \leq B_0$ for all $t \geq 1$, where B_0 is a finite constant. Then, for $\mathbf{u}[t]$ with $\|\mathbf{R}_{\mathbf{x}^*}(\mathbf{u}[t])\| > \eta_0$, it holds that

$$\begin{aligned}\text{dist}^2\left(\mathbf{u}[t], \mathbf{R}_{\mathbf{x}^*}^{-1}(\mathbf{0})\right) &\leq \|\mathbf{u}[t] - \mathbf{u}^*\|^2\\ &\leq B_0^2\\ &\leq \frac{B_0^2}{\eta_0^2}\|\mathbf{R}_{\mathbf{x}^*}(\mathbf{u}[t])\|^2\end{aligned}$$

Then, let $\kappa = \max\{\kappa_0, B_0^2/\eta_0^2\}$, we have

$$\text{dist}^2\left(\mathbf{u}[t], \mathbf{R}_{\mathbf{x}^*}^{-1}(\mathbf{0})\right) \leq \kappa\|\mathbf{R}_{\mathbf{x}^*}(\mathbf{u}[t])\|^2, \forall t \geq 1. \tag{54}$$

Based on the result of Lemma 4.8, the set $\mathbf{R}_{\mathbf{x}^*}^{-1}(\mathbf{0})$ is equivalent to the set $\Omega^*(\mathbf{x}^*)$. Therefore, we have for all $t \geq 1$,

$$\begin{aligned}\text{dist}^2\left(\mathbf{u}[t], \Omega^*\right) &= \inf_{\mathbf{u}\in\Omega^*}\|\mathbf{u} - \mathbf{u}[t]\|^2\\ &\overset{(a)}{\leq} \inf_{\mathbf{u}\in\Omega^*(\mathbf{x}^*)}\|\mathbf{u} - \mathbf{u}[t]\|^2\\ &= \text{dist}^2\left(\mathbf{u}[t], \mathbf{R}_{\mathbf{x}^*}^{-1}(\mathbf{0})\right)\\ &\leq \kappa\|\mathbf{R}_{\mathbf{x}^*}(\mathbf{u}[t])\|^2.\end{aligned} \tag{55}$$

The above, step (a) is based on the definition $\Omega^*(\mathbf{x}^*) = \Omega^* \cap \{(\mathbf{x}, \mathbf{r}, \boldsymbol{\lambda})|\mathbf{x} = \mathbf{x}^*\}$. Therefore, the lemma follows.

E PROOF OF THEOREM 3.2

For notational simplicity, let $\mathbf{u}[t] = (\mathbf{x}[t], \mathbf{r}[t], \boldsymbol{\lambda}[t])$. Based on the result of Lemma 4.9, we have

$$\begin{aligned}\text{dist}^2\left(\mathbf{u}[t], \Omega^*\right) \leq \kappa \|\mathbf{R}_{\mathbf{x}^*}(\mathbf{u}[t])\|^2 = \kappa \Big(&\|\mathbf{x}[t] - \mathbf{x}^*\|^2 + \|\mathbf{x}[t] - \mathbf{Pr}_h(\mathbf{x}[t] - \boldsymbol{\lambda}_s[t] - \nabla U(\mathbf{x}^*))\|^2 \\ &+ \|\mathbf{r}[t] - \mathbf{Pr}_g(\mathbf{r}[t] + \mathbf{A}_s^T\boldsymbol{\lambda}_s[t] + \mathbf{A}_r^T\boldsymbol{\lambda}_r[t])\|^2 + \|\mathbf{A}\mathbf{r}[t] + \mathbf{B}\mathbf{x}[t]\|^2\Big). \end{aligned} \tag{56}$$

Firstly, the term $\|\mathbf{A}\mathbf{r}[t] + \mathbf{B}\mathbf{x}[t]\| = \|\boldsymbol{\lambda}[t-1] - \boldsymbol{\lambda}[t]\|/\rho\tau$. Secondly, from the Proof of Lemma 4.4, we have shown that the optimality condition of the first step in Algorithm 1 is equivalent to the condition (40), which can be further written as

$$\mathbf{r}[t] = \mathbf{Pr}_g\left(\mathbf{r}[t] + \mathbf{A}_s^T\left[\bar{\boldsymbol{\lambda}}_s[t] - \rho(\mathbf{x}[t] - \mathbf{x}[t-1])\right] + \mathbf{A}_r^T\bar{\boldsymbol{\lambda}}_r[t] + \mathbf{Q}(\mathbf{r}[t-1] - \mathbf{r}[t])\right).$$

Then, we have

$$\begin{aligned}&\|\mathbf{r}[t] - \mathbf{Pr}_g(\mathbf{r}[t] + \mathbf{A}_s^T\boldsymbol{\lambda}_s[t] + \mathbf{A}_r^T\boldsymbol{\lambda}_r[t])\| \\ =&\Big\|\mathbf{Pr}_g(\mathbf{r}[t] + \mathbf{A}_s^T\left[\bar{\boldsymbol{\lambda}}_s[t] - \rho(\mathbf{x}[t] - \mathbf{x}[t-1])\right] + \mathbf{A}_r^T\bar{\boldsymbol{\lambda}}_r[t] + \mathbf{Q}(\mathbf{r}[t-1] - \mathbf{r}[t])) - \\ &\mathbf{Pr}_g(\mathbf{r}[t] + \mathbf{A}_s^T\boldsymbol{\lambda}_s[t] + \mathbf{A}_r^T\boldsymbol{\lambda}_r[t])\Big\| \\ \overset{(a)}{\leq}&\|\mathbf{A}_s^T(\bar{\boldsymbol{\lambda}}_s[t] - \boldsymbol{\lambda}_s[t]) + \mathbf{A}_r^T(\bar{\boldsymbol{\lambda}}_r[t] - \boldsymbol{\lambda}_r[t]) - \rho\mathbf{A}_s^T(\mathbf{x}[t] - \mathbf{x}[t-1]) + \mathbf{Q}(\mathbf{r}[t-1] - \mathbf{r}[t])\| \\ \overset{(b)}{\leq}&\|\mathbf{A}_s^T\|\|\bar{\boldsymbol{\lambda}}_s[t] - \boldsymbol{\lambda}_s[t]\| + \|\mathbf{A}_r^T\|\|\bar{\boldsymbol{\lambda}}_r[t] - \boldsymbol{\lambda}_r[t]\| + \rho\|\mathbf{A}_s^T\|\|\mathbf{x}[t] - \mathbf{x}[t-1]\| + \|\mathbf{Q}\|\|\mathbf{r}[t-1] - \mathbf{r}[t]\| \\ \overset{(c)}{\leq}&(1 - \frac{1}{\tau})\|\mathbf{A}_s^T\|\|\boldsymbol{\lambda}_s[t-1] - \boldsymbol{\lambda}_s[t]\| + \rho\|\mathbf{A}_s^T\|\|\mathbf{x}[t] - \mathbf{x}[t-1]\| + (1 - \frac{1}{\tau})\|\mathbf{A}_r^T\|\|\boldsymbol{\lambda}_r[t-1] - \boldsymbol{\lambda}_r[t]\| + \\ &\|\mathbf{Q}\|\|\mathbf{r}[t-1] - \mathbf{r}[t]\|.\end{aligned}$$

The above, step (a) is based on the non-expansiveness of the proximal mapping that $\|\mathbf{Pr}_f(\mathbf{x}) - \mathbf{Pr}_f(\mathbf{y})\| \leq \|\mathbf{x} - \mathbf{y}\|$, step (b) utilizes the triangle inequality and the matrix norm inequality that $\|\mathbf{A}\mathbf{x}\| \leq \|\mathbf{A}\|\|\mathbf{x}\|$, step (c) is based on the definition of $\bar{\boldsymbol{\lambda}}[t]$ in (41) and (42) such that $\bar{\boldsymbol{\lambda}}[t] - \boldsymbol{\lambda}[t] = (\tau - 1)\rho(\mathbf{A}\mathbf{r}[t] + \mathbf{B}\mathbf{x}[t]) = (1 - 1/\tau)(\boldsymbol{\lambda}[t-1] - \boldsymbol{\lambda}[t])$. Similarly, we have

$$\mathbf{x}[t] = \mathbf{Pr}_h(\mathbf{x}[t] - \bar{\boldsymbol{\lambda}}_s[t] - \nabla U(\mathbf{x}[t])), \tag{57}$$

and then

$$\begin{aligned}&\|\mathbf{x}[t] - \mathbf{Pr}_h(\mathbf{x}[t] - \boldsymbol{\lambda}_s[t] - \nabla U(\mathbf{x}^*))\| \\ =&\|\mathbf{Pr}_h(\mathbf{x}[t] - \bar{\boldsymbol{\lambda}}_s[t] - \nabla U(\mathbf{x}[t])) - \mathbf{Pr}_h(\mathbf{x}[t] - \boldsymbol{\lambda}_s[t] - \nabla U(\mathbf{x}^*))\| \\ \leq&\|\bar{\boldsymbol{\lambda}}_s[t] - \boldsymbol{\lambda}_s[t]\| + \|\nabla U(\mathbf{x}[t]) - \nabla U(\mathbf{x}^*)\| \\ \overset{(d)}{\leq}&(1 - \frac{1}{\tau})\|\boldsymbol{\lambda}_s[t-1] - \boldsymbol{\lambda}_s[t]\| + L_u\|\mathbf{x}[t] - \mathbf{x}^*\|.\end{aligned} \tag{58}$$

The above, step (d) is based on the assumption that utility function $U(\cdot)$ has Lipschitz continuous gradient with constant L_u. Then, substitute the above inequalities into upper bound (56) and rearrange the terms, we have

$$\text{dist}^2\left(\mathbf{u}[t], \Omega^*\right) \leq c_1\|\mathbf{x}^* - \mathbf{x}[t]\|^2 + c_2\|\boldsymbol{\lambda}[t-1] - \boldsymbol{\lambda}[t]\|^2 + c_3\|\mathbf{x}[t] - \mathbf{x}[t-1]\|^2 + c_4\|\mathbf{r}[t] - \mathbf{r}[t-1]\|^2, \tag{59}$$

where the constant c_1, c_2, c_3 and c_4 are given by

$$\begin{aligned}c_1 &= \kappa(1 + 2L_u^2), \\ c_2 &= (1 - \frac{1}{\tau})^2(4\max\{\|\mathbf{A}_s^T\|^2, \|\mathbf{A}_r^T\|^2\} + 2) + \frac{1}{\rho^2\tau^2},\end{aligned}$$

$$c_3 = 4\rho^2\|\mathbf{A}_s^T\|^2,$$
$$c_4 = 4\|\mathbf{Q}\|^2.$$

Note that the constants 2 and 4 in coefficients c_i derive from the Cauchy-Schwartz inequality. For all $t \geq 1$, define

$$(\overline{\mathbf{x}}_t, \overline{\mathbf{r}}_t, \overline{\boldsymbol{\lambda}}_t) = \arg \min_{(\mathbf{x},\mathbf{r},\boldsymbol{\lambda})\in\Omega^*} \|\mathbf{x} - \mathbf{x}[t]\|^2 + \|\mathbf{r} - \mathbf{r}[t]\|^2 + \|\boldsymbol{\lambda} - \boldsymbol{\lambda}[t]\|^2.$$

Then we have

$$\text{dist}^2(\mathbf{u}[t], \Omega^*) = \|\mathbf{x}[t] - \overline{\mathbf{x}}_t\|^2 + \|\mathbf{r}[t] - \overline{\mathbf{r}}_t\|^2 + \|\boldsymbol{\lambda}[t] - \overline{\boldsymbol{\lambda}}_t\|^2. \tag{60}$$

Further, define

$$\mathbf{x}_t^* = \arg \min_{(\mathbf{x},\mathbf{r},\boldsymbol{\lambda})\in\Omega^*} \|\mathbf{x} - \mathbf{x}[t]\|,$$
$$\mathbf{r}_t^* = \arg \min_{(\mathbf{x},\mathbf{r},\boldsymbol{\lambda})\in\Omega^*} \|\mathbf{r} - \mathbf{r}[t]\|,$$
$$\boldsymbol{\lambda}_t^* = \arg \min_{(\mathbf{x},\mathbf{r},\boldsymbol{\lambda})\in\Omega^*} \|\boldsymbol{\lambda} - \boldsymbol{\lambda}[t]\|.$$

Based on the fact that matrix $\mathbf{Q}$ is positive definite, we have $\lambda_{\min}(\mathbf{Q}) > 0$ and $\|\mathbf{r}[t] - \mathbf{r}^*\|_{\mathbf{Q}}^2 \geq \lambda_{\min}(\mathbf{Q})\|\mathbf{r}[t] - \mathbf{r}^*\|^2$. Then, we can write the inequality (25) in Lemma 4.4 as the following form.

$$V(\mathbf{x}[t-1], \mathbf{r}[t-1], \boldsymbol{\lambda}[t-1]) - V(\mathbf{x}[t], \mathbf{r}[t], \boldsymbol{\lambda}[t]) \geq c_5\|\boldsymbol{\lambda}[t-1] - \boldsymbol{\lambda}[t]\| + c_5\|\mathbf{x}[t-1] - \mathbf{x}[t]\|^2 + c_6\|\mathbf{r}[t] - \mathbf{r}[t-1]\|^2 + c_7\|\mathbf{x}[t] - \mathbf{x}^*\|^2 + c_7\|\mathbf{x}[t] - \mathbf{x}[t-1]\|^2, \tag{61}$$

where the coefficients c_5, c_6 and c_7 are positive constants. Combining the above inequality with the error bound (59), we conclude that there exists a positive constant $\gamma > 0$ such that

$$V(\mathbf{x}[t-1], \mathbf{r}[t-1], \boldsymbol{\lambda}[t-1]) - V(\mathbf{x}[t], \mathbf{r}[t], \boldsymbol{\lambda}[t]) \geq \gamma \left(\frac{1}{\rho\tau}\|\boldsymbol{\lambda}[t] - \overline{\boldsymbol{\lambda}}_t\|^2 + \rho\|\mathbf{x}[t] - \overline{\mathbf{x}}_t\|^2 + \|\mathbf{r}[t] - \overline{\mathbf{r}}_t\|_{\mathbf{Q}}^2 + \frac{\rho}{\eta}\|\mathbf{A}_s\mathbf{r}[t] - \mathbf{x}[t]\|^2 \right).$$

Let $\mathbf{x}^* = \mathbf{x}_{t-1}^*$, $\mathbf{r}^* = \mathbf{r}_{t-1}^*$ and $\boldsymbol{\lambda}^* = \boldsymbol{\lambda}_{t-1}^*$ in the function $V(\cdot)$ of the above inequality, then we have

$$\frac{1}{\rho\tau}\|\boldsymbol{\lambda}[t-1] - \boldsymbol{\lambda}_{t-1}^*\|^2 + \rho\|\mathbf{x}[t-1] - \mathbf{x}_{t-1}^*\|^2 + \frac{\rho}{\eta}\|\mathbf{A}_s\mathbf{r}[t-1] - \mathbf{x}[t-1]\|^2 + \|\mathbf{r}[t-1] - \mathbf{r}_{t-1}^*\|_{\mathbf{Q}}^2 \geq \left(\frac{1}{\rho\tau}\|\boldsymbol{\lambda}[t] - \boldsymbol{\lambda}_{t-1}^*\|^2 + \rho\|\mathbf{x}[t] - \mathbf{x}_{t-1}^*\|^2 + \|\mathbf{r}[t] - \mathbf{r}_{t-1}^*\|_{\mathbf{Q}}^2 + \frac{\rho}{\eta}\|\mathbf{A}_s\mathbf{r}[t] - \mathbf{x}[t]\|^2 \right) + \gamma \left(\frac{1}{\rho\tau}\|\boldsymbol{\lambda}[t] - \overline{\boldsymbol{\lambda}}_t\|^2 + \rho\|\mathbf{x}[t] - \overline{\mathbf{x}}\|^2 + \|\mathbf{r}[t] - \overline{\mathbf{r}}_t\|_{\mathbf{Q}}^2 + \frac{\rho}{\eta}\|\mathbf{A}_s\mathbf{r}[t] - \mathbf{x}[t]\|^2 \right). \tag{62}$$

Based on the definition of sequences $(\overline{\mathbf{x}}_t, \overline{\mathbf{r}}_t, \overline{\boldsymbol{\lambda}}_t)$ and $(\mathbf{x}_t^*, \mathbf{r}_t^*, \boldsymbol{\lambda}_t^*)$, we have

$$\begin{aligned} &\|\mathbf{x}[t] - \mathbf{x}_{t-1}^*\| \geq \|\mathbf{x}[t] - \mathbf{x}_t^*\|, \|\mathbf{x}[t] - \overline{\mathbf{x}}\| \geq \|\mathbf{x}[t] - \mathbf{x}_t^*\|, \\ &\|\mathbf{r}[t] - \mathbf{r}_{t-1}^*\| \geq \|\mathbf{r}[t] - \mathbf{r}_t^*\|, \|\mathbf{r}[t] - \overline{\mathbf{x}}\| \geq \|\mathbf{r}[t] - \mathbf{r}_t^*\|, \\ &\|\boldsymbol{\lambda}[t] - \boldsymbol{\lambda}_{t-1}^*\| \geq \|\boldsymbol{\lambda}[t] - \boldsymbol{\lambda}_t^*\|, \|\boldsymbol{\lambda}[t] - \overline{\boldsymbol{\lambda}}_t\| \geq \|\boldsymbol{\lambda}[t] - \boldsymbol{\lambda}_t^*\|. \end{aligned} \tag{63}$$

Combining inequality (62) and (63) together, we can get the following contraction.

$$G[t] \leq \frac{1}{1+\gamma} G[t-1], t \geq 1.$$

where $G[t]$ is defined as

$$G[t] = \frac{1}{\rho\tau}\|\boldsymbol{\lambda}[t] - \boldsymbol{\lambda}_t^*\|^2 + \rho\|\mathbf{x}[t] - \mathbf{x}_t^*\|^2 + \|\mathbf{r}[t] - \mathbf{r}_t^*\|_{\mathbf{Q}}^2 + \frac{\rho}{\eta}\|\mathbf{A}_s\mathbf{r}[t] - \mathbf{x}[t]\|^2. \tag{64}$$

Telescoping the above inequality for all iterations t, we arrive that

$$G[t] \leq \left(\frac{1}{1+\gamma}\right)^t D_0, \tag{65}$$

where D_0 is the initial distance to the optimal solution set,

$$D_0 = \frac{1}{\rho\tau}\|\boldsymbol{\lambda}[0] - \boldsymbol{\lambda}_0^*\|^2 + \rho\|\mathbf{x}[0] - \mathbf{x}_0^*\|^2 + \|\mathbf{r}[0] - \mathbf{r}_0^*\|_{\mathbf{Q}}^2 + \frac{\rho}{\eta}\|\mathbf{A}_s\mathbf{r}[0] - \mathbf{x}[0]\|^2.$$

Therefore, the theorem follows.

F PROOF OF LEMMA 4.10

Define an auxiliary queue $\hat{\lambda}_n^d[t]$ that evolves according to (11). Initializing the auxiliary queue with $\hat{\lambda}_n^d[0] = M + \rho\tau\sum_{l\in O(n)}\eta_l$, where η_l is the upper bound of the capacity of link l. Then we can prove by induction that

$$\hat{\lambda}_n^d[t] = \lambda_n^d[t] + M + \rho\tau\sum_{l\in O(n)}\eta_l, \forall t, d \in \mathcal{D}, n \in \mathcal{N}\backslash d.$$

Since $\lambda_n^d[t] \geq -M, \forall t, n, d$ by assumption, we have that

$$\hat{\lambda}_n^d[t] \geq \rho\tau\sum_{l\in O(n)}\eta_l, \forall t, d \in \mathcal{D}, n \in \mathcal{N}\backslash d.$$

Then the auxiliary queue $\hat{\lambda}_n^d[t]$ satisfies

$$\hat{\lambda}_n^d[t] = \left[\hat{\lambda}_n^d[t-1] - \rho\tau\sum_{l\in O(n)} r_l^d[t]\right]_+ + \rho\tau\sum_{l\in I(n)} r_l^d[t] + \rho\tau\sum_{f\in\mathcal{F}} x_f[t]\mathbf{1}_{\{s_f=n, d_f=d\}}, \forall t, d \in \mathcal{D}, n \in \mathcal{N}\backslash d.$$

Based on the fact that $\rho\tau > 0$, we can rewrite the above updating formula as

$$\frac{\hat{\lambda}_n^d[t]}{\rho\tau} = \left[\frac{\hat{\lambda}_n^d[t-1]}{\rho\tau} - \sum_{l\in O(n)} r_l^d[t]\right]_+ + \sum_{l\in I(n)} r_l^d[t] + \sum_{f\in\mathcal{F}} x_f[t]\mathbf{1}_{\{s_f=n, d_f=d\}}.$$

We next prove that $Q_n^d[t] \leq \hat{\lambda}_n^d[t]/\rho\tau, \forall t \geq 1$ by induction. For $t = 0$, we have $Q_n^d[0] = 0 \leq \hat{\lambda}_n^d[0]/\rho\tau$. Suppose that it holds for $k = t-1$, then for $k = t$, we have

$$\begin{aligned} Q_n^d[t] &\leq \left[Q_n^d[t-1] - \sum_{l\in O(n)} r_l^d[t]\right]_+ + \sum_{l\in I(n)} \hat{r}_l^d[t] + \sum_{f\in\mathcal{F}} x_f[t]\mathbf{1}_{\{s_f=n, d_f=d\}} \\ &\leq \left[\frac{\hat{\lambda}_n^d[t-1]}{\rho\tau} - \sum_{l\in O(n)} r_l^d[t]\right]_+ + \sum_{l\in I(n)} \hat{r}_l^d[t] + \sum_{f\in\mathcal{F}} x_f[t]\mathbf{1}_{\{s_f=n, d_f=d\}} \\ &= \frac{\hat{\lambda}_n^d[t]}{\rho\tau}. \end{aligned} \tag{66}$$

Finally, since $\hat{\lambda}_n^d[t] = \lambda_n^d[t] + M + \rho\tau\sum_{l\in O(n)}\eta_l$ and $\lambda_n^d[t] \leq M$, we have

$$Q_n^d[t] \leq \frac{2M}{\rho\tau} + \sum_{l\in O(n)}\eta_l.$$

Let constant $B = \max_{n\in\mathcal{N}} \sum_{l\in O(n)} \eta_l$. Therefore, the lemma follows.

G PROOF OF THEOREM 5.4

Let $\mathcal{P} \in \mathbb{R}^{L(D+1)}$ be a convex polyhedron, defined as

$$\mathcal{P} = \left\{ (\mathbf{y}, \mathbf{r}) \middle| \mathbf{y} \in C, y_l = \sum_{d=1}^{D} r_l^d, \text{ and } r_l^d \geq 0, \forall l \in \mathcal{L}, d \in \mathcal{D} \right\}.$$

Formally, we define following two problems. The first one is the scheduling component in Algorithm 1.

Definition G.1. (New scheduling problem) Given arbitrary weights $\mathbf{a} \in \mathbb{R}^{DL}$, $\mathbf{b} \in \mathbb{R}^{DL}$ and $\mathbf{c} \in \mathbb{R}^{L}$, output an $(\mathbf{r}^*, \mathbf{y}^*)$ such that, for arbitrary $(\mathbf{r}, \mathbf{y}) \in \mathcal{P}$,

$$\sum_{l=1}^{L}\sum_{d=1}^{D} a_l^d r_l^{d*} - c_l(r_l^{d*} - b_l^d)^2 \geq -\delta + \sum_{i=1}^{L}\sum_{j=1}^{D} a_l^d r_l^d - c_i(r_l^d - b_l^d)^2, \tag{67}$$

and $B((\mathbf{r}^*, \mathbf{y}^*), \delta) \in \mathcal{P}$.

Definition G.2. (MaxWeight scheduling) Given arbitrary weights $\mathbf{w} \in \mathbb{Z}^D$, output an $\mathbf{r}^* \in C$ such that

$$\mathbf{w}^T\mathbf{r}^* \geq \mathbf{w}^T\mathbf{r}, \forall \mathbf{r} \in C \text{ and } r_l \geq 0. \tag{68}$$

We can observe that the above defined problem is actually equivalent to the original MaxWeight scheduling problem (13) based on the fact that

$$\max_{r_l^d} \sum_{l=1}^{L}\sum_{d\in\mathcal{D}} (Q_m^d[t] - Q_n^d[t]) r_l^d, \text{ s.t. } \left[\sum_d r_l^d\right] \in C, r_l^d \geq 0.$$

$$\iff \max_{\mathbf{r}} \sum_{l=1}^{L} (Q_m^{d_l}[t] - Q_n^{d_l}[t]) r_l^{d_l}, \text{ s.t. } \mathbf{r} \in C, r_l^{d_l} \geq 0,$$

where d_l is defined as $d_l = \arg\max_{d\in\mathcal{D}}(Q_m^d[t] - Q_n^d[t])$, and the fact that the physical queue length in the QCA method is an integer (number of packets). According to the above definitions, to prove the Theorem 5.4, we need to construct a poly(L, F) time reduction between the above two problems.

We first prove the "if" direction.

Based on the result in Lemma 5.3, we know that solving the new scheduling problem in poly$(L, F, \log(\delta^{-1}))$ time if the separation oracle problem for polyhedron $\mathcal{P}$ can be solved in poly(L, F) time. Since the constraints $y_l = \sum_{d=1}^{D} r_l^d$ and $r_l^d \geq 0$ in $\mathcal{P}$ can be explicitly checked in $O(LF)$ time, then the separation oracle problem for polyhedron $\mathcal{P}$ can be reduced to the separation oracle problem for polyhedron C by the following procedure: given a separation hyperplane $\mathbf{c}^T\mathbf{y} \geq \mathbf{c}^T\mathbf{y}', \forall \mathbf{y}' \in C$, construct the hyperplane $\mathbf{c}^T\mathbf{y} + \mathbf{c}'^T\mathbf{r}$ with $c_l^{d\prime} = c_l, \forall l, d$. Then, we have

$$\begin{aligned}
\mathbf{c}^T\mathbf{y} + \mathbf{c}'^T\mathbf{r} &= \mathbf{c}^T\mathbf{y} + \sum_{l=1}^{L} c_l \sum_{D=1}^{D} r_l^d = \mathbf{c}^T\mathbf{y} + \mathbf{c}^T\mathbf{y} \\
&\geq \mathbf{c}^T\mathbf{y}' + \mathbf{c}^T\mathbf{y}' \\
&= \mathbf{c}^T\mathbf{y}' + \mathbf{c}'^T\mathbf{r}', \forall (\mathbf{y}', \mathbf{r}') \in \mathcal{P},
\end{aligned}$$

which implies that $\mathbf{c}^T\mathbf{y} + \mathbf{c}'^T\mathbf{r}$ is also a separating hyperplane of polyhedron $\mathcal{P}$. A classic result in the combinatorial optimization due to Grötschel and Lovász [11] establishes the equivalence between the linear optimization problem and the separation oracle problem for the same polyhedron.

Therefore, the new scheduling problem (67) can be reduced to the original MaxWeight scheduling problem (68) in poly(L, F) time.

We next prove the "only if" direction.

For any input instance $\mathbf{w} \in \mathbb{Z}^D$ in the MaxWeight scheduling problem, construct the input instance $\mathbf{a} \in \mathbb{R}^{DL}$, $\mathbf{b} \in \mathbb{R}^{DL}$ and $\mathbf{c} \in \mathbb{R}^L$ as following.

$$a_l^d = (LDB^2 + 1)w_l, \forall l, d,$$
$$b_l^d = 0, \forall l, d, \quad c_l = 1, \forall l.$$

The above, constant B is the upper bound of the all the link rate r_l^d. Suppose that we solve the new scheduling problem in poly$(L, F, \log(\delta^{-1}))$ time under the above input instance. Then we have an $(\mathbf{r}^*, \mathbf{y}^*)$ such that $B((\mathbf{r}^*, \mathbf{y}^*), \delta) \in \mathcal{P}$ and for arbitrary $(\mathbf{r}, \mathbf{y}) \in \mathcal{P}$,

$$(LDB^2 + 1)\sum_{l=1}^{L} w_l y_l^* - \sum_{l=1}^{L}\sum_{d=1}^{D} r_l^{d*2} \geq -\delta + (LDB^2 + 1)\sum_{l=1}^{L} w_l y_l - \sum_{l=1}^{L}\sum_{d=1}^{D} r_l^{d2}.$$

The quantity y_l^* and y_l derives from $y_l^* = \sum_{d=1}^{D} r_l^{d*}$ and $y_l = \sum_{d=1}^{D} r_l^d$. We prove the following argument by contradiction.

$$\sum_{l=1}^{L} w_l y_l^* \geq \sum_{l=1}^{L} w_l y_l, \forall \mathbf{y} \in \Gamma \text{ and } y_l \geq 0, \forall l.$$

Assume that there exists $\mathbf{y} \in \Gamma$ and $y_l \geq 0, \forall l$ such that $\sum_{l=1}^{L} w_l y_l^* < \sum_{l=1}^{L} w_l y_l$. Then, we have

$$\begin{aligned}
&(LDB^2 + 1)\sum_{l=1}^{L} w_l y_l^* < (LDB^2 + 1)\sum_{l=1}^{L} w_l y_l \\
\overset{(a)}{\Rightarrow}&(LDB^2 + 1)\left[1 + \sum_{l=1}^{L} w_l y_l^*\right] \leq (LDB^2 + 1)\sum_{l=1}^{L} w_l y_l - \delta \\
\overset{(b)}{\Rightarrow}&(LDB^2 + 1)\sum_{l=1}^{L} w_l y_l^* < \sum_{l=1}^{L}\sum_{d=1}^{D} r_l^{d*2} - \sum_{l=1}^{L}\sum_{d=1}^{D} r_l^{d2} - \delta + (LDB^2 + 1)\sum_{l=1}^{L} w_l y_l \\
\Rightarrow&(LDB^2 + 1)\sum_{l=1}^{L} w_l y_l^* - \sum_{l=1}^{L}\sum_{d=1}^{D} r_l^{d*2} < -\sum_{l=1}^{L}\sum_{d=1}^{D} r_l^{d2} + (LDB^2 + 1)\sum_{l=1}^{L} w_l y_l - \delta,
\end{aligned}$$

which is a contradiction. The above, step (a) is based on the assumption that the weight w_l, feasible link rate y_l, y_l^* are the integers, and that δ is sufficiently small, step (b) utilizes the definition that $r_l^d \leq B, \forall l, d$. Utilizing the fact that the optimal point of linear optimization lies in the vertex set of the feasible region, the y_l^* is also the optimal solution of the following optimization problem.

$$\max_{\mathbf{r}} \mathbf{w}^T \mathbf{r}, \text{ s.t. } \mathbf{r} \in C, r_l \geq 0, \forall l.$$

which is clearly the solution of the MaxWeight scheduling problem (68). Therefore, the theorem follows.

REFERENCES

[1] Eleftheria Athanasopoulou, Loc X Bui, Tianxiong Ji, R Srikant, and Alexander Stolyar. 2013. Back-pressure-based packet-by-packet adaptive routing in communication networks. *IEEE/ACM Transactions on Networking (TON)* 21, 1 (2013), 244–257.

[2] James H Bramble, Joseph E Pasciak, and Apostol T Vassilev. 1997. Analysis of the inexact Uzawa algorithm for saddle point problems. *SIAM J. Numer. Anal.* 34, 3 (1997), 1072–1092.

[3] Damek Davis and Wotao Yin. 2017. Faster convergence rates of relaxed Peaceman-Rachford and ADMM under regularity assumptions. *Mathematics of Operations Research* (2017).

[4] Wei Deng, Ming-Jun Lai, Zhimin Peng, and Wotao Yin. 2017. Parallel multi-block ADMM with O (1/k) convergence. *Journal of Scientific Computing* 71, 2 (2017), 712–736.

[5] Wei Deng and Wotao Yin. 2016. On the global and linear convergence of the generalized alternating direction method of multipliers. *Journal of Scientific Computing* 66, 3 (2016), 889–916.

[6] Defeng Sun Deren Han and Liwei Zhang. 2017. Linear Rate Convergence of the Alternating Direction Method of Multipliers for Convex Composite Programming. In *Mathematics of Operation Research*.

[7] Asen L Dontchev and R Tyrrell Rockafellar. 2009. Implicit functions and solution mappings. *Springer Monogr. Math.* (2009).

[8] John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. 2008. Efficient projections onto the l 1-ball for learning in high dimensions. In *Proceedings of the 25th international conference on Machine learning*. ACM, 272–279.

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

[10] Daniel Gabay and Bertrand Mercier. 1976. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications* 2, 1 (1976), 17–40.

[11] Martin Grötschel, László Lovász, and Alexander Schrijver. 1981. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica* 1, 2 (1981), 169–197.

[12] Martin Grötschel, László Lovász, and Lex Schrijver. 1993. Geometric algorithms and combinatorial optimization. *Algorithms and Combinatorics* 2 (1993), 1–362.

[13] Bingsheng He and Xiaoming Yuan. 2015. On non-ergodic convergence rate of Douglas–Rachford alternating direction method of multipliers. *Numer. Math.* 130, 3 (2015), 567–577.

[14] Alan J Hoffman. 2003. On approximate solutions of systems of linear inequalities. *Selected Papers Of Alan J Hoffman: With Commentary* (2003), 174–176.

[15] Longbo Huang and Michael J Neely. 2011. Delay reduction via Lagrange multipliers in stochastic network optimization. *IEEE Trans. Automat. Control* 56, 4 (2011), 842–857.

[16] Martin Jaggi. 2013. Revisiting Frank-Wolfe: Projection-Free Sparse Convex Optimization.. In *ICML (1)*. 427–435.

[17] Libin Jiang and Jean Walrand. 2010. A distributed CSMA algorithm for throughput and utility maximization in wireless networks. *IEEE/ACM Transactions on Networking (ToN)* 18, 3 (2010), 960–972.

[18] Srisankar Kunniyur and Rayadurgam Srikant. 2001. Analysis and design of an adaptive virtual queue (AVQ) algorithm for active queue management. In *ACM SIGCOMM Computer Communication Review*, Vol. 31. ACM, 123–134.

[19] Tianyi Lin, Shiqian Ma, and Shuzhong Zhang. 2015. On the global linear convergence of the admm with multiblock variables. *SIAM Journal on Optimization* 25, 3 (2015), 1478–1497.

[20] Xiaojun Lin and Ness B Shroff. 2004. Joint rate control and scheduling in multihop wireless networks. In *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, Vol. 2. IEEE, 1484–1489.

[21] Xiaojun Lin and Ness B Shroff. 2006. Utility maximization for communication networks with multipath routing. *IEEE Trans. Automat. Control* 51, 5 (2006), 766–781.

[22] Xiaojun Lin, Ness B Shroff, and Rayadurgam Srikant. 2006. A tutorial on cross-layer optimization in wireless networks. *IEEE Journal on Selected areas in Communications* 24, 8 (2006), 1452–1463.

[23] Jia Liu. 2016. Achieving low-delay and fast-convergence in stochastic network optimization: A nesterovian approach. In *Proceedings of the 2016 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Science*. ACM, 221–234.

[24] Jia Liu, Atilla Eryilmaz, Ness B Shroff, and Elizabeth S Bentley. 2016. Heavy-ball: A new approach to tame delay and convergence in wireless network optimization. In *Computer Communications, IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on*. IEEE, 1–9.

[25] Jia Liu, Ness B Shroff, Cathy H Xia, and Hanif D Sherali. 2016. Joint congestion control and routing optimization: An efficient second-order distributed approach. *IEEE/ACM Transactions on Networking* 24, 3 (2016), 1404–1420.

[26] Jia Liu, Cathy H Xia, Ness B Shroff, and Hanif D Sherali. 2013. Distributed cross-layer optimization in wireless networks: A second-order approach. In *INFOCOM, 2013 Proceedings IEEE*. IEEE, 2103–2111.

[27] Michael J Neely, Eytan Modiano, and Charles E Rohrs. 2003. Power allocation and routing in multibeam satellites with time-varying channels. *IEEE/ACM Transactions on Networking (TON)* 11, 1 (2003), 138–152.

[28] Yurii Nesterov. 2009. Primal-dual subgradient methods for convex problems. *Mathematical programming* 120, 1 (2009), 221–259.

[29] R. J.-B. Wets R. T. Rockafellar. 1998. Variational analysis. (1998).

[30] Stephen M Robinson. 1981. Some continuity properties of polyhedral multifunctions. *Mathematical Programming at Oberwolfach* (1981), 206–214.
[31] R Tyrrell Rockafellar. 1997. Convex Analysis. (1997).
[32] Ralph Tyrell Rockafellar. 2015. *Convex analysis*. Princeton university press.
[33] Gaurav Sharma, Ravi R Mazumdar, and Ness B Shroff. 2006. On the complexity of scheduling in wireless networks. In *Proceedings of the 12th annual international conference on Mobile computing and networking*. ACM, 227–238.
[34] Alexander L Stolyar. 2005. Maximizing queueing network utility subject to stability: Greedy primal-dual algorithm. *Queueing Systems* 50, 4 (2005), 401–457.
[35] Leandros Tassiulas and Anthony Ephremides. 1992. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE transactions on automatic control* 37, 12 (1992), 1936–1948.
[36] Ermin Wei, Asuman Ozdaglar, and Ali Jadbabaie. 2013. A distributed Newton method for network utility maximization–I: Algorithm. *IEEE Trans. Automat. Control* 58, 9 (2013), 2162–2175.
[37] Hao Yu and Michael J Neely. 2017. A New Backpressure Algorithm for Joint Rate Control and Routing with Vanishing Utility Optimality Gaps and Finite Queue Lengths. *arXiv preprint arXiv:1701.04519* (2017).
[38] Xiaoqun Zhang, Martin Burger, and Stanley Osher. 2011. A unified primal-dual algorithm framework based on Bregman iteration. *Journal of Scientific Computing* 46, 1 (2011), 20–46.

Received August 2017; revised October 2017; accepted December 2017