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

Sinong Wang
Department of ECE
The Ohio State University
Columbus, Ohio
wang.7691@osu.edu

Ness Shroff
Departments of ECE and CSE
The Ohio State University
Columbus, Ohio
shroff.11@osu.edu

ABSTRACT

Distributed network optimization has been studied several years. 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. A theoretical contribution of independent interest is a new pathway we provide to prove global and linear convergence rate of Uzawa-ADMM without requiring the full rank assumption of the constraint matrix.

ACM Reference Format:
Sinong Wang and Ness Shroff. 2018. Towards Fast-Convergence, Low-Delay and Low-Complexity Network Optimization. In *SIGMETRICS '18 Abstracts: ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems Abstracts, June 18–22, 2018, Irvine, CA, USA.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3219617.3219671

1 INTRODUCTION

Consider a fixed data network $\mathcal{G} = \{\mathcal{N}, \mathcal{L}\}$ shared by F end-to-end flows, 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. Each flow f is described by its source-destination node pair (s_f, d_f) and associated utility function $U_f(\cdot)$, without a priori established routes. The set of source nodes are defined as $\mathcal{F}$ and $|\mathcal{F}| = F$. The set of destination nodes are defined as $\mathcal{D} = \{d_f, f \in \mathcal{F}\}$, and let $|\mathcal{D}| = D$. 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*,

$$\max_{x_f, r_l^d} \quad \sum_{f\in\mathcal{F}} U_f(x_f) \tag{1}$$
$$s.t. \quad \sum_{f\in\mathcal{F}} x_f \mathbb{1}_{\{s_f=n, d_f=d\}} + \sum_{l\in\mathcal{I}(n)} r_l^d = \sum_{l\in\mathcal{O}(n)} r_l^d, \forall d, n \in \mathcal{N}\backslash d,$$
$$\left[\sum_{d\in\mathcal{D}} r_l^d\right] \in C, r_l^d \geq 0, \forall d \in \mathcal{D}, l \in \mathcal{L},$$
$$m_f \leq x_f \leq M_f, \forall f \in \mathcal{F}.$$

where r_l^d represent the amount of capacity on link l that is allocated for data towards destination d, C is the capacity region defined as $C \triangleq \text{Conv}(\Gamma)$ and $\Gamma = \{\mathbf{r}^{(1)}, \mathbf{r}^{(2)}, \ldots, \mathbf{r}^{(I)}\}$ is the set of feasible link rate vectors, and $\text{Conv}(\cdot)$ represents the convex hull operation. 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 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., [1, 2, 5–9], and [4] 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 methods: This class of algorithms apply the subgradient descent method to the dual function of this problem and leads to a beautiful queue-length-based control algorithmic (QCA) framework. 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 from optimality will yield a large queuing delay.

Second-order Newton method: To improve the convergence speed, there have been many attempts in obtaining new algorithms by applying the second-order method [5, 9]. This kind of algorithm has a faster convergence rate, i.e., $O(\log^2(1/\epsilon))$ iterations. 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

SIGMETRICS '18 Abstracts, June 18–22, 2018, Irvine, CA, USA

ACM ISBN 978-1-4503-5846-0/18/06.
https://doi.org/10.1145/3219617.3219671

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.

$[O(1/K), O(K^2)]$ in [5]; and (iii) it cannot efficiently handle the wireless interference channel.

Proximal method: The proximal method was first introduced in the work [3] to tackle the oscillation problem in a network optimization problem with given routing paths. The work [10] extends this method to the scenario of dynamic routing and prove that the proximal method not only exhibits the feature of low-latency, it also offers an improved convergence speed of $O^*(1/\epsilon)$.

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 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 existing schemes in an $L-$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 existing methods. The rationale behind our algorithm design is to utilize the Alternating Directional Method of Multiplier (ADMM). 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, 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 the congestion control and routing components are different, as is the coupling among the different functionalities. 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 [2], it is unclear how to solve the new scheduling problem derived from the ADMM-based decomposition.

2 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.

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

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

$$\max_{x_f\in[m_f,M_f]} U_f(x_f)-(z_{s_f}^{d_f}[t]+\rho\Delta r_f[t])x_f-\frac{\rho}{2}(x_f-x_f[t-1])^2. \tag{3}$$

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

$$\Delta r_f[t]=\sum_{l\in\mathcal{I}(s_f)}\left(r_l^{d_f}[t]-r_l^{d_f}[t-1]\right)-\sum_{l\in\mathcal{O}(s_f)}\left(r_l^{d_f}[t]-r_l^{d_f}[t-1]\right). \tag{4}$$

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

$$\lambda_n^d[t]=\lambda_n^d[t-1]-\rho\tau\sum_{l\in\mathcal{O}(n)}r_l^d[t]+\rho\tau\sum_{l\in\mathcal{I}(n)}r_l^d[t]+\rho\tau\sum_{f\in\mathcal{F}}x_f[t]\mathbb{1}_{\{s_f=n,d_f=d\}}. \tag{5}$$

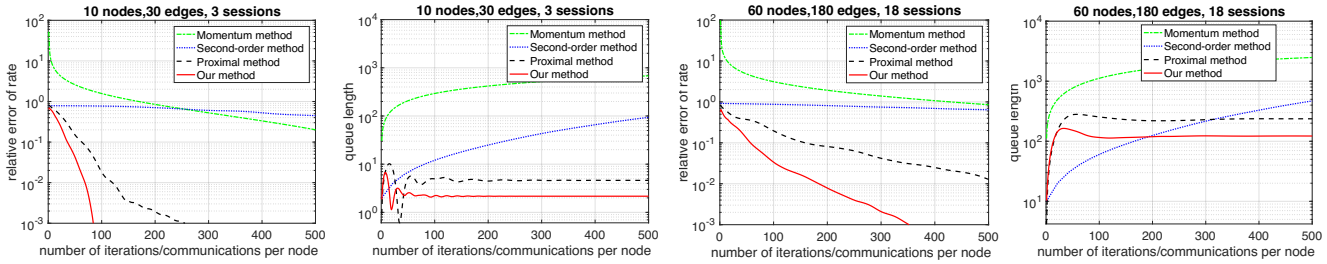

Figure 1: Comparison of Algorithm 1 and existing methods in a small-scale and a medium-scale wireline networks.

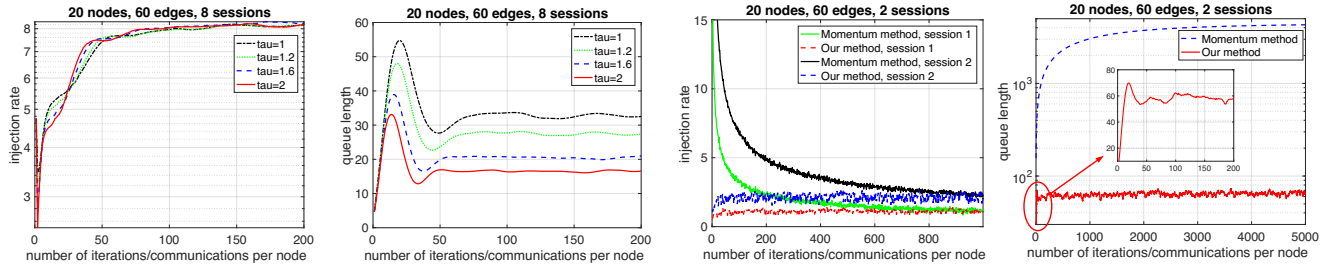

Figure 2: The two left figures shows the impact of parameter τ on convergence and queue length. The two right figures compare our algorithm and the momentum method for a wireless network with fading channel.

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

3 NUMERICAL ANALYSIS

We numerically compare our algorithm with existing algorithms. 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: $\sum_{d\in\mathcal{D}} \sum_{n\in\mathcal{N}\setminus d} Q_n^d[t]$.

4 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.

REFERENCES

[1] 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.
[2] 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.
[3] Xiaojun Lin and Ness B Shroff. 2006. Utility maximization for communication networks with multipath routing. *IEEE Trans. Automat. Control* 51, 5 (2006), 766–781.
[4] 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.
[5] 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.
[6] 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.
[7] Alexander L Stolyar. 2005. Maximizing queueing network utility subject to stability: Greedy primal-dual algorithm. *Queueing Systems* 50, 4 (2005), 401–457.
[8] 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.
[9] 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.
[10] 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).