From Network Inference Errors to Utility Suboptimality: How Much Is the Impact?

Shijing Li, Tian Lan
George Washington University
{shijing, tlan}@gwu.edu

***Abstract*—Accurate and timely network inference, known as network tomography, is a vital ingredient in efficient network operation, supporting more sophisticated and ambitious traffic optimization algorithms. In practice, however, precise knowledge of the dynamic internal network bottlenecks/states can be impossible to obtain under restricted network visibility on network edge where only end-to-end measurements are available without cooperation of internal network components. In this paper, we acknowledge that network tomography on the edge is often imperfect, leading to dynamic, time-varying inference errors that could change from iteration to iteration on the same time-scale as distributed network optimization algorithms. We quantify the impact of such imperfect bottleneck/state inference on algorithm convergence and optimality. In particular, we show that under arbitrary, bounded inference errors (belonging to three common classes including absent and incorrect bottlenecks, and inaccurate capacity), the solution of the distributed optimization algorithm still converges to a bounded neighborhood of the optimal solution. The resulted optimality gap is quantified in closed form and shown to be proportional to average inference errors. These results provide a theoretical support for understanding the impact of imperfect inference on distributed network optimization.**

I. Introduction

Edge computing improves service latency and processing efficiency by enabling computation and storage capacity at the edge network [1], [2]. The explosive growth of the Internet of Things, combined with rapid advances in emerging applications such as real-time data analytics and smart cities have led to the increasing popularity of edge computing [1], and yet at the same time, has rendered edge network optimization a demanding task in heterogeneous edge environment.

Such optimization problems are often addressed through the Network Utility Maximization (NUM) framework [3], [4], [5], which often calls for distributed algorithmic solutions based on the Lagrangian function[6]. The idea is to interpret network metrics as primal or dual variables, and design distributed optimization algorithms that often update the primal and dual variables via a gradient-based approach[7], [8]. In edge networks, in order for these gradients to be computed at the various edge nodes, it is essential to know (dynamically) the internal network bottlenecks/states, which, however, is often not directly available due to restricted network visibility on the edge, as shown in Fig. 1. To this end, techniques to infer internal network bottlenecks/states via end-to-end measurements on the edge, which is referred to as *network tomography*[9], are becoming a vital ingredient in enabling more sophisticated and ambitious traffic optimization algorithms in edge networks. Unfortunately, due to growing network complexity and yet limited visibility on the edge, perfect network knowledge is often impossible to obtain in practice even with state-of-the-art network tomography techniques [9], [10], [11].

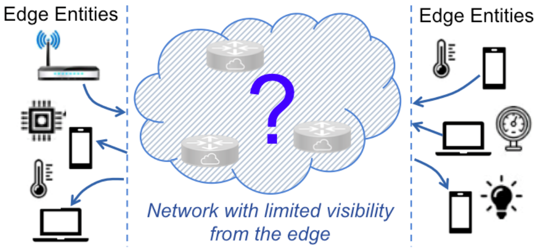

Fig. 1. Distributed traffic optimization must cope imperfect network inference due to limited network visibility on the network edge.

In this paper, we acknowledge that perfect network knowledge may not be available in edge NUM problems and address the question of how much network inference errors impact utility suboptimality. Interestingly, the answer is "not so much" when inference errors are dynamic and bounded. Traditionally, network monitoring relies on the cooperation of internal network components [12] or active probing [13], [14], which are often impractical due to high probing overhead, lack of cooperation, and growing privacy and proprietary concerns [15], [16], [17]. Alternatively, network tomography [18] infers internal network bottlenecks and states via edge-based measurements, e.g., [9], [10], [11], [19]. However, dynamically inferring precise network knowledge is difficult due to a number of reasons. First, network tomography based on end-to-end measurements often encounter the *lack of identifiability* problem [12], meaning that the measurements are insufficient to uniquely determine a network state and link metrics [20], [21], [22]. Second, network virtualization, widely used in cloud/fog systems, can introduce an opaqueness imposing serious obstacles on network inference [23]. Finally, edge networks are usually exposed to a high degree of dynamics induced by device mobility and application variability, which can degrade the quality of available edge-based measurements. Perfect network inference is hardly achievable in practice.

Under incomplete and imperfect network inference, our goal in this paper is to quantify the convergence and optimality of distributed NUM algorithms in edge networks. We consider *dynamic inference errors* that can vary during the process of network inference and on the same time-scale of distributed NUM algorithms. In particular, we focus on three common classes of inference errors: (i) absent bottlenecks that are not identified, (ii) incorrect bottlenecks in which flows are mistakenly inserted or erased, and (iii) inaccurate estimate of bottleneck link capacity. All of the three kinds of errors could vary in different network metric estimation iterations by using probe data, even adjacent iterations, which makes the optimization better adapt to fast changing network interference. In addition, for slowly changing network interference, three kinds of errors may remain the same for many iterations which make the optimization converge faster.

Focusing on a family of Projected Gradient Algorithm(PGA)[24] that are widely employed to solve distributed NUM problems, we prove that under arbitrary, bounded inference errors, such distributed traffic optimization algorithms can still converge using running averages of the primal and dual variables. Further, by analyzing the Lagrangian and the non-expansive property of the projections, the converged solution is shown to be within a small neighborhood of the global optimal solution, with an optimality gap that is proportional to the average inference error and diminishes linearly to zero as the inference errors decrease.

We quantify the optimality gap in closed-form and validate our theoretic results using extensive NS2 simulations, with randomly generated topologies and routing decisions. In simulation, we use file size from two real IoT data sets: (1) consisting of 200 hours of accelerometer information recorded over 25 days from 5 participants [32], with each data point in the size range of 80-187MB. (2) a series of continuous frames extracted from a traffic video sequence recorded by stationary cameras[33][34]. For the transmission rate of nodes, we set an uniform combination of 250Kbps and 54Mbps, due to real IoT nodes communicating communicating on these common rates through IEEE 802.15.4, IEEE 802.11, NB-IoT, or ZigBee. We simulated our algorithm's performance under imperfect network inference in different setting of network scale, amount of flows, network inference, to validate the theoretical analysis. The results provide a solid theoretical support and illuminate important design space for distributed network optimization algorithms under imperfect network tomography inference.

The main contributions of the paper can be summarized as follow:

- We show that distributed traffic optimization algorithms based on the Lagrangian still converge under imperfect bottleneck inference with dynamic, bounded errors, including absent and incorrect bottlenecks, and inaccurate bottleneck capacities.
- The optimality gap due to network inference errors is quantified in closed form. The gap is shown to be proportional to average inference error and diminishes to zero as inference error decreases.
- Our theoretical analysis are evaluated using extensive NS2 simulations. The numerical results validate our theory under different inference error rates and with randomly generated networks and flows.

II. Related Works

Network bottleneck/state inference is known to be a challenging problem due to growing network size, increasing heterogeneity, and dynamic network changes [25], [26]. The active approach is often undesirable in practice because it normally requires the cooperation of internal network components [12] or active probing [13], [14]. A mechanism for estimating network bottlenecks and their link capacities is proposed in [9], [19] by minimizing the entropy of the inter-packet spacing via flow clustering. The work has since been extended to dynamic networks using Lypunov theory [10], [11], robust clustering based on Renyi entropy [19], and maximum-likelihood-based topology inference [16]. Network tomography has also been used for clustering online TCP flow clustering [17], estimating multicast trees [27], and locating network bottlenecks [14]. Given current network topology, there are also papers targeting to optimize network resource by predicting traffic by machine learning[28] or data analysis[29]. However, it is often not possible to achieve perfect network inference due to network dynamics, virtualization, and limited visibility on network edge. In this paper, we address a crucial questions as follows: with imperfect bottleneck inference on network edge, how does it affect the convergence and optimality of existing distributed traffic optimization algorithms?

Edge/fog computing enables a large number of heterogeneous and decentralized devices on network edge to communicate and perform storage and computing tasks [30], [1]. Many network design problems can be formulated as utility and traffic optimization under bottleneck/link capacity constraints. Focusing on imperfect network inference, our work in this paper is different from those that employ robust optimization to deal with uncertainty in the problem parameters and constraints [31]. In contrast, we acknowledge that perfect inference may be impossible to achieve in practice and analyze the performance of existing distributed traffic optimization algorithms under dynamic inference errors, which varies on the same time scale as the algorithms and could change from iteration to iteration [8].

III. System and Problem Statement

We consider a network consisting of m distributed edge nodes/entities, such as smart devices, access points, computing/storage servers, mobile terminals, which form various edge-computing clusters and enclaves. Typically, NUM aims to jointly determines all flow rates $\boldsymbol{x} = (x_1, x_2, \ldots, x_m)$ (denoted as the bandwidth allocation vector) to maximize an aggregate utility $f(\boldsymbol{x}) = \sum_i f(x_i)$, subject to determined topology and link capacity constraints. We note that $f(x_i)$ is assumed to be continuous, increasing, and concave. It measures the utility of assigning bandwidth x_i to edge source i. Let $\boldsymbol{b}$ denote the link capacity vector containing the link capacity constraint b_j of each link j, and $\boldsymbol{A}$ be the combined routing matrix such that $A_{ij} = 1$ if flow i traverses link j, and $A_{ij} = 0$ otherwise. The NUM problem can be formulated as follows:

$$\max \sum_i f(\boldsymbol{x}) \quad s.t. \quad g(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} \leq 0 \tag{1}$$

where $\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}$ is the network capacity constraint. This optimization framework has been shown to capture many crucial design objectives including meeting application deadlines, mitigating network congestion, and providing QoS guarantees.

To obtain a distributed solution to the NUM problem, we often decouple the problem using the primal-dual method[8]. More precisely, we consider flow rates $\boldsymbol{x}$ as prime variables and introduce $\boldsymbol{\lambda}$ as the Lagrangian dual variables with respect to the capacity constraints. The Lagragian function is then obtained as

$$L(\boldsymbol{x},\boldsymbol{\lambda}) = f(\boldsymbol{x}) - \boldsymbol{\lambda^T}(\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}) \tag{2}$$

Functions $f(\mathbf{x})$ is convex and continuous on feasible set $\mathcal{X}$ and $\mathcal{D}$, respectively. Suppose that the set of optimal solutions $\{\boldsymbol{x}^*\}$ and the set of optimal dual solutions (i.e., optimal Lagrange multipliers) $\{\boldsymbol{\lambda}^*\}$ are both nonempty, closed and bounded. Solving the traffic optimization problem is equivalent to finding a saddle point of the associated Lagrangian function over $\mathcal{X}\times\mathcal{D}$, which is solved by a Projected Gradient Algorithm (PGA) [8]. Let $P_{\mathcal{X}}$ and $P_{\mathcal{D}}$ denote the projections of the primal and dual variables onto $\mathcal{X}$ and $\mathcal{D}$, respectively. The Projected Gradient Algorithm employs the following updates on the primal and dual variables:

$$\boldsymbol{x}(\tau+1) = P_{\mathcal{X}}\left[\boldsymbol{x}(\tau) + \alpha\nabla_{\boldsymbol{x}}L(\boldsymbol{x}(\tau),\boldsymbol{\lambda}(\tau))\right] \tag{3}$$

$$\boldsymbol{\lambda}(\tau+1) = P_{\mathcal{D}}\left[\boldsymbol{\lambda}(\tau) - \alpha\nabla_{\boldsymbol{\lambda}}L(\boldsymbol{x}(\tau),\boldsymbol{\lambda}(\tau))\right] \tag{4}$$

where $\tau = 0, 1, 2, \ldots$ is the number of iterations, and $\alpha > 0$ is a constant stepsize. The algorithm can be carried out distributedly at different edge entities, and it is guaranteed to converge to an optimal solution $(\boldsymbol{x}^*, \boldsymbol{\lambda}^*)$ to both the primal and dual problems.

In primal-dual algorithm, prime variables $\boldsymbol{x}$ and dual variables $\boldsymbol{\lambda}$ are updated on the same time scale in a discrete fashion. In particular, for each iteration $\tau+1$, each flow i's data rate $x_i(\tau+1)$ will be updated by its source node according to (3) with respect to previous rate $x_i(\tau)$. And similarly, each router or network controller will calculate $\lambda_j(\tau+1)$ according to Lagragian function (2).

However, in this paper we consider such distributed updates under imprefect network inference tomography. More precisely, we assume that network topology and link capacity constraints can only be inferred on network edge from end-to-end measurements, while the network is considered as a ”black-box”[9], [19]. For example, the authors in [9] showed how to estimate bottlenecks by minimizing the entropy of the inter-packet spacing and clustering flows into groups. Yet such network inferences in evitably introduce estimation error , e.g., omitting ”hidden” flows in routing matrix and/or adding estimate errors in link bandwidth, due to the nature of statistical network inference. In addition, applying optimized data flow may also lead to time-varying dynamics that further degrade the accuracy of network inference.

Traffic optimization with time-varying imperfect inference

We acknowledge that perfect network knowledge may not be available in edge NUM problems and analyze the convergence and optimality of projected-gradient-based NUM algorithms (jointly) under three common classes of dynamic inference errors that can vary on the same time-scale of the inference and NUM algorithms. First, we consider the possibility of absent bottlenecks that are missing in the network inference from time to time. For each iteration τ, we model missing bottlenecks using a set of binary variables $d_{jj}(\tau)\ \forall j$, i.e., $d_{jj}(\tau) = 1$ if bottleneck link j is identified at the time of the τth iteration, and $d_{jj}(\tau) = 0$ otherwise. Then, we define a diagonal indicator matrix $\mathbf{D_e} = \text{diag}\{d_{jj},\ \forall j\}$. When there is no missing bottleneck, $\mathbf{D_e}(\tau)$ becomes an identify matrix. Next, we introduce an error matrix $\mathbf{R_e}(\tau)$ to model the incorrectly inferred bottlenecks where flows are mistakenly inserted or erased. Mor precisely, we have $\mathbf{R_e}_{ij}(\tau) = 1$ if flow i is incorrectly inserted on bottleneck link j, $\mathbf{R_e}_{ij}(\tau) = -1$ if flow i is undetected, and $\mathbf{R_e}_{ij}(\tau) = 0$ if the inference about flow i is correct. Thus, the (imperfect) inferred routing matrix at the time of ith iteration becomes $\boldsymbol{A}+\mathbf{R_e}(\tau)$. Finally, to model erroneous bottleneck capacity, we assume that the inferred capacity values on bottleneck links are corrupted by a noise vector $\mathbf{b_e}(\tau)$, i.e., $\boldsymbol{b} + \mathbf{b_e}(\tau)$.

In this paper, we do not make any assumptions on the time-scale or dependence of such inference errors. More precisely, the three types of errors could vary (dependently or independently) on arbitrary time scales, e.g., from iteration to iteration if network inference is carried out frequently on each iteration of the NUM algorithm, or remaining the same for a number of iterations under a slowly-changing network environment. Network inferences affect the topology, bottlenecks and variables of the NUM algorithm. For example, in [9], a wrong clustering in routing matrix estimation will introduce extra or missing flows represented by $\boldsymbol{R_e}$. Omitting bottlenecks will be represented by $\boldsymbol{D_e}$. Time-varying network quality and previous optimization feedback will be represented by $\boldsymbol{b_e}$.

To illustrate our model of NUM with imperfect inference, we consider a simple network shown in Fig. 2 with 3 flows sharing 2 bottlenecks, both with capacity 2 Mbps and 3 Mbps. Blue flow 1 is from node 1 to node 5. Yellow flow 2 is from node 2 to node 4 but share the same link with green flow 3 from node 3 to node 4. It is easy to see that the routing matrix is thus $\boldsymbol{A} = (1\ 0\ 0;\ 0\ 1\ 1)$ and capacity vector $\boldsymbol{b} = (2, 2)^T$, corresponding to constraints $x_1 \leq$2Mbps and $x_2+x_3 \leq$3Mbps. Thus, the gradient-based update will consider the following constraint instead:

$$\hat{g}(\boldsymbol{x}) = \mathbf{D_e}(\tau)\left[(\mathbf{A}+\mathbf{R_e}(\tau))\boldsymbol{x} - (\mathbf{b}+\mathbf{b_e}(\tau))\right] \leq \mathbf{0}, \tag{5}$$

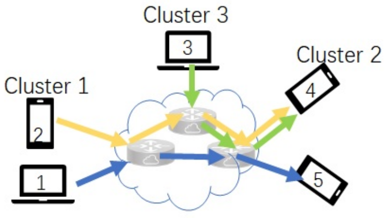

Fig. 2. A illustrative example of NUM with imperfect network inference.

Now consider the following network inference error during iteration τ of the optimization algorithm. Because node 1 and node 2 are far from node 3, they can not exchange information with node 3. So node 1 and node 2 estimate they share the same link, and thus cause link bottleneck $x_1 + x_2 \leq 2.2$Mbps. Then, there are 3 kinds ot estimation errors: (1)they omit one bottleneck including x_2 and x_3, which can be represented by $\mathbf{D_e}(\tau) = \mathrm{diag}(1,0)$; (2)they incorrectly includes x_2 in the first bottleneck which actually only includes x_1, which can be represented by $\boldsymbol{R_e}(\tau) = (0\ 1\ 0;\ 0\ 0\ 0)$; (3)they have a capacity estimate error of +0.2Mbps, which can be represented by $\boldsymbol{b_e}(\tau) = (0.2, 0)^T$. In this iteration, $\mathbf{D_e}(\tau)$ cause one missing constraint, $\mathcal{R}_e(\tau)$ cause one incorrect bottleneck link, and $\mathbf{b_e}(\tau)$ cause one larger maximum link capacity in constraint.

IV. Analyze Convergence and Optimality under Imperfect Inference

In this section, we prove that under the three types of network inference errors, the solution of the distributed traffic optimization algorithm still converges to a neighborhood of the optimal solution. Further, the optimality gap (i.e., the difference between the achieved utility and the maximum possible utility) can be quantified by an upper bound in closed-form.

In the τth iteration, consider inference errors modeled by $\mathbf{D_e}(\tau)$, $\mathbf{R_e}(\tau)$, and $\mathbf{b_e}(\tau)$. The distributed traffic optimization (**??**) now has a new constraint function $\hat{g}(\boldsymbol{x})$ (which takes into account the network inference errors) as follows:

$$\hat{g}(\boldsymbol{x}) = \mathbf{D_e}(\tau)\left[(\mathbf{A} + \mathbf{R_e}(\tau))\boldsymbol{x} - (\mathbf{b} + \mathbf{b_e}(\tau))\right] \leq \mathbf{0} \qquad (6)$$

To apply the projected gradient algorithm, we formulate the Lagrangian with respect to $\hat{g}(\boldsymbol{x})$. We have

$$\hat{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) - \boldsymbol{\lambda^T \hat{g}(x)} \qquad (7)$$

Because the Lagrangian function is different due to the inference errors, the gradient update will change accordingly:

$$\mathbf{x}(\tau+1) = P_{\mathcal{X}}\left[\mathbf{x}(\tau) + \alpha \nabla_{\mathbf{x}} \hat{L}(\mathbf{x}(\tau), \boldsymbol{\lambda}(\tau))\right] \qquad (8)$$

$$\boldsymbol{\lambda}(\tau+1) = P_{\mathcal{D}}\left[\boldsymbol{\lambda}(\tau) - \alpha \nabla_{\boldsymbol{\lambda}} \hat{L}(\mathbf{x}(\tau), \boldsymbol{\lambda}(\tau))\right] \qquad (9)$$

Suitable average can help deal with nonvanishing errors in gradient based approaches. To derive an upper bound on the optimality gap, we show that the following running average converges to a neighborhood of the optimal solution.

$$\bar{\boldsymbol{x}}(\tau) \triangleq \frac{1}{\tau}\sum_{i=0}^{\tau-1} \boldsymbol{x}(i) \text{ and } \bar{\boldsymbol{\lambda}}(\tau) \triangleq \frac{1}{\tau}\sum_{i=0}^{\tau-1} \boldsymbol{\lambda}(i) \qquad (10)$$

Let $||\cdot||$ denote the matrix L_2-norm. In the following, we first analyze the evolution of primal and dual variables $\boldsymbol{x}(\tau)$ and $\boldsymbol{\lambda}(\tau)$ in the projected gradient update in Lemma 1. In particular, it quantifies the distance between $\boldsymbol{x}(\tau)$ and an arbitrary point $\boldsymbol{x}$ as the distributed algorithm proceeds from iteration τ to $\tau+1$. Then, in Lemma 2, we utilize the result in Lemma 1 and add up the inequalities from iteration $i = 0$ to $i = \tau - 1$, which offer a bound for the difference between the optimal (error-free) utility and the sum of Lagrangian functions. Finally, we combine the lemmas to obtain the desired upper bound between the optimal solution and the sum of utilities of running averages. To simplify the notations, we will use $L = L(\boldsymbol{x}(\tau), \boldsymbol{\lambda}(\tau))$ and $\hat{L} = \hat{L}(\boldsymbol{x}(\tau), \boldsymbol{\lambda}(\tau))$. *The detailed proof procedures of lemma 1, 2 and 3 are shown in the Appendix.*

Lemma 1. *Let the sequences $\{\boldsymbol{x}(\tau)\}$ and $\{\boldsymbol{\lambda}(\tau)\}$ be generated by the projected gradient algorithm (9) with bottleneck inference errors, we then have:*

$$\begin{aligned}
&\|\boldsymbol{x}(\tau+1) - \boldsymbol{x}\|^2 - \|\boldsymbol{x}(\tau) - \boldsymbol{x}\|^2 \\
&\leq -2\alpha[L(\boldsymbol{x}, \boldsymbol{\lambda}(\tau)) - L(\boldsymbol{x}(\tau), \boldsymbol{\lambda}(\tau))] + \alpha^2\|\nabla_{\boldsymbol{x}}\hat{L}\|^2 \\
&\quad +2\alpha\|\Gamma_1(\tau)(\boldsymbol{x}(\tau) - \boldsymbol{x})\| \qquad (11) \\
&\|\boldsymbol{\lambda}(\tau+1) - \boldsymbol{\lambda}\|^2 - \|\boldsymbol{\lambda}(\tau) - \boldsymbol{\lambda}\|^2 \\
&\leq 2\alpha[L(\boldsymbol{x}(\tau), \boldsymbol{\lambda}(\tau)) - L(\boldsymbol{x}(\tau), \boldsymbol{\lambda}] + \alpha^2\|\nabla_{\boldsymbol{\lambda}}\hat{L}\|^2 \\
&\quad +2\alpha\|\Gamma_2(\tau)(\boldsymbol{\lambda}(\tau) - \boldsymbol{\lambda})\| \qquad (12)
\end{aligned}$$

where we defined $\Gamma_1(\tau) = \nabla_{\boldsymbol{x}}\hat{L} - \nabla_{\boldsymbol{x}}L$ as the introduced error in the $\boldsymbol{x}$-gradient, and $\Gamma_2(\tau) = \nabla_{\boldsymbol{\lambda}}\hat{L} - \nabla_{\boldsymbol{\lambda}}L$ as that of the $\boldsymbol{\lambda}$-gradient, or equivalently,

$$\begin{aligned}
\Gamma_1(\tau) &= \boldsymbol{\lambda}^T(\tau)[\boldsymbol{D_e}(\tau)\boldsymbol{A} + \boldsymbol{D_e}(\tau)\boldsymbol{R_e}(\tau) - \boldsymbol{A}] \qquad (13) \\
\Gamma_2(\tau) &= \boldsymbol{Ax} - \mathbf{D_e}(\tau)[\mathbf{A} + \mathbf{R_e}(\tau)]\boldsymbol{x} \\
&\quad +\mathbf{D_e}(\tau)[\mathbf{b} + \mathbf{b_e}(\tau)] - \boldsymbol{b} \qquad (14)
\end{aligned}$$

Next, we use the results in Lemma 1 to bound the difference between optimal solution and the average of Lagrangian functions over different iterations.

Lemma 2. *Let f^* be the optimal utility obtained without bottleneck inference error, and $x(0)$ and $\lambda(0)$ be the initial primal and dual variables at the beginning ($\tau = 0$) of the iterative optimization algorithms. We have*

$$\begin{aligned}
&f^* - \frac{1}{\tau}\sum_{i=0}^{\tau-1} L(\boldsymbol{x}(i), \boldsymbol{\lambda}(i)) \\
&\leq \frac{\|\boldsymbol{x}(0) - \boldsymbol{x}\|^2}{2\alpha\tau} + \frac{1}{\tau}\sum_{i=0}^{\tau-1}(\|\Gamma_0(i)\| + \frac{\alpha}{2}\|\nabla_{\boldsymbol{x}}\hat{L}\|^2), \qquad (15) \\
&\frac{1}{\tau}\sum_{i=0}^{\tau-1}[L(\boldsymbol{x}(\tau), \boldsymbol{\lambda}) - L(\boldsymbol{x}(\tau), \boldsymbol{\lambda}(\tau))] \leq \frac{\|\boldsymbol{\lambda}(0) - \boldsymbol{\lambda}\|^2}{2\alpha\tau} \\
&+\frac{1}{\tau}\sum_{i=0}^{\tau-1}(\|\Gamma_2(i)(\boldsymbol{\lambda}(i) - \boldsymbol{\lambda})\| + \frac{\alpha}{2}\|\nabla_{\boldsymbol{\lambda}}\hat{L}(i)\|^2) \qquad (16)
\end{aligned}$$

where we introduced the following auxiliary variable:

$$\Gamma_0(\tau) = \Gamma_1(\tau)[\boldsymbol{x}(\tau) - \boldsymbol{x}] \qquad (17)$$

Now, we combine Lemmas 1 and 2 to bound the difference between optimal utility f^* and the achieved utility $f(\bar{\boldsymbol{x}}(\tau))$. To this end, we define the following auxiliary variables:

$$K_0 = \|\boldsymbol{x}(0) - \boldsymbol{x^*}\|^2 + \|\boldsymbol{\lambda}(0)\|^2, \qquad (18)$$

$$K_{\hat{L}} = \frac{1}{\tau}\sum_{i=0}^{\tau-1}(\|\nabla_{\boldsymbol{x}}\hat{L}\|^2 + \|\nabla_{\boldsymbol{\lambda}}\hat{L}\|^2), \qquad (19)$$

$$K_{\boldsymbol{R_e}, \boldsymbol{b_e}, \boldsymbol{D_e}} = \frac{1}{\tau}\sum_{i=0}^{\tau-1}(\|\Gamma_0(i)\| + \|\Gamma_2(i)\boldsymbol{\lambda}(i)\|), \qquad (20)$$

which is used to state the following lemma.

Lemma 3. *The solution $\bar{\boldsymbol{x}}(\tau)$ generated by the distributed optimization algorithm under bottleneck inference errors converge to a neighborhood of the optimal solution, bounded by*

$$f^* - f(\bar{\boldsymbol{x}}(\tau)) \leq \frac{K_0}{2\alpha\tau} + \frac{\alpha K_{\hat{L}}}{2} + K_{\boldsymbol{R_e},\boldsymbol{b_e},\boldsymbol{D_e}}, \quad (21)$$

where K_0 in (18) is determined by the initial state of the algorithm, $K_{\hat{L}}$ in (19) depends on the average value of the Lagrangian, and $K_{\boldsymbol{R_e},\boldsymbol{b_e},\boldsymbol{D_e}}$ in (20) is determined by the three types of errors.

Remark 1: Lemma 3 shows that with sufficiently small α and sufficiently large $\alpha\tau$, distributed traffic optimization algorithms under inference errors converges to a neighborhood of the optimal solution. The gap $K_{\boldsymbol{R_e},\boldsymbol{b_e},\boldsymbol{D_e}}$ depends on the three types of inference errors. It provides us with a powerful tool for analyzing the optimality under inference errors.

We consider bounded network inference errors satisfying $\forall\tau$

$$\|\boldsymbol{D_e}(\tau) - \boldsymbol{I}\| \leq B_D, \ \|\boldsymbol{R_e}(\tau)\| \leq B_R, \ \|\boldsymbol{b_e}(\tau)\| \leq B_b, \quad (22)$$

for some constants $B_D, B_R, B_b \geq 0$. We assume that the sequence of primal and dual variables, $\{\boldsymbol{x}(\tau)\}_{\tau=0}^{\infty}$ and $\{\boldsymbol{\lambda}(\tau)\}_{\tau=0}^{\infty}$, are also bounded, i.e., there exists a constant $B_x \geq 0$, such that for any τ:

$$\|\boldsymbol{\lambda}(\tau)\| \leq B_x, \ \boldsymbol{x}(\tau) \leq B_x, \ \|\boldsymbol{x}(\tau) - \boldsymbol{x}^*\| \leq B_x, \quad (23)$$

which allow us to bound the Lagrangian and gradient in Lemma 3. We also use $B_{U'}$ to denote the bound on the gradient of utility, i.e., $||\nabla_{\boldsymbol{x}} f(\boldsymbol{x})|| \leq B_{U'}$, for bounded $\boldsymbol{x}(\tau)$.

Theorem 1. *When all three types of inference errors are bounded, the solution $\bar{\boldsymbol{x}}(\tau)$ generated by distributed optimization algorithm satisfy:*

$$f^* - f(\bar{\boldsymbol{x}}(\tau)) \leq \frac{B_x}{\alpha\tau} + \frac{\alpha B_1}{2} + \frac{1}{\tau}\sum_{i=0}^{\tau-1} Q_i, \quad (24)$$

where $B_1 = (B_{U'} + 2B_x||A|| + 2B_xB_R + B_b + ||b||)^2$ is a constant, and Q_i is the weighted sum of the three types of network inference errors in iteration i, given by

$$\begin{aligned} Q_i \quad &= 2B_x^2(\|(\boldsymbol{D_e}(\tau) - \boldsymbol{I})\| \cdot \|\boldsymbol{A}\| + \|\boldsymbol{R_e}(\tau)\|) \\ &+ B_x(\|\boldsymbol{D_e} - \boldsymbol{I}\|\|\mathbf{b}\| + \|\mathbf{b_e}\|) \end{aligned} \quad (25)$$

Proof. We just need to analyze and bound K_0, $K_{\hat{L}}$ and $K_{\boldsymbol{R_e},\boldsymbol{b_e},\boldsymbol{D_e}}$ in Lemma 3. The result is then directly from (21). First, it is easy to see that due to (22), we have

$$K_0 = \|\boldsymbol{x}(0) - \boldsymbol{x}^*\|^2 + \|\boldsymbol{\lambda}(0)\|^2 \leq 2B_x. \quad (26)$$

For $K_{\hat{L}}$ in (19), we need to bound $\|\nabla_{\boldsymbol{x}}\hat{L}\|^2$ and $\|\nabla_{\boldsymbol{\lambda}}\hat{L}\|^2$ separately. To this end, we derive for any iteration τ

$$\begin{aligned} \|\nabla_{\boldsymbol{x}}\hat{L}\| \quad &= \|\nabla_{\boldsymbol{x}} f(\boldsymbol{x}) - \boldsymbol{\lambda}^T\boldsymbol{D_e}(\tau)(A + \boldsymbol{R_e}(\tau))\| \\ &\leq B_{U'} + B_x\left(\|\boldsymbol{A}\| + B_R\right), \end{aligned} \quad (27)$$

where we used $\|\mathbf{D_e}(\tau)\| \leq 1$ because it is a diagonal matrix with only 0 or 1, and the bound of $\boldsymbol{R_e}(\tau)$ in (22). Similarly, we also have for any iteration τ

$$\begin{aligned} \|\nabla_{\boldsymbol{\lambda}}\hat{L}\| \quad &= \|\boldsymbol{D_e}(\tau)\left[(\boldsymbol{A} + \boldsymbol{R_e}(\tau))x - \boldsymbol{b} - \boldsymbol{b_e}(\tau)\right]\| \\ &\leq \|\boldsymbol{A}\|B_x + \|\boldsymbol{b}\| + B_xB_R + B_b, \end{aligned} \quad (28)$$

where we again used $\|\mathbf{D_e}(i)\| \leq 1$ and the bound for $\boldsymbol{b_e}(i)$ and $\boldsymbol{R_e}(i))$ in (22). Combining these results and using $c^2 + d^2 \leq (c+d)^2$ for $c = \|\nabla_{\boldsymbol{x}}\hat{L}\|$ and $d = \|\nabla_{\boldsymbol{\lambda}}\hat{L}\|$, we can easily derive the bound for $K_{\hat{L}}$.

Next, to bound $K_{\boldsymbol{R_e},\boldsymbol{b_e},\boldsymbol{D_e}}$ in (20), we need to bound $\|\Gamma_0\|$ and $\|\Gamma_2(\tau)\boldsymbol{\lambda}(\tau)\|$. It is easy to show that

$$\begin{aligned} \|\Gamma_0\| \quad &\leq B_x\|\boldsymbol{\lambda}^T(\tau)[\boldsymbol{D_e}(\tau)\boldsymbol{A} + \boldsymbol{D_e}(\tau)\boldsymbol{R_e}(\tau) - \boldsymbol{A}]\| \\ &\leq B_x^2(\|(\boldsymbol{D_e}(\tau) - \boldsymbol{I})\boldsymbol{A} + \boldsymbol{R_e}(\tau)\|) \\ &\leq B_x^2(\|(\boldsymbol{D_e}(\tau) - \boldsymbol{I})\| \cdot \|\boldsymbol{A}\| + \|\boldsymbol{R_e}(\tau)\|) \end{aligned} \quad (29)$$

where we used $\|\boldsymbol{\lambda}\| \leq B_x$. On the other hand, we can also show:

$$\begin{aligned} \|\Gamma_2(\tau)\boldsymbol{\lambda}(\tau)\| \quad &\leq B_x\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{D_e}(\tau)[\boldsymbol{A} + \boldsymbol{R_e}(\tau)]\boldsymbol{x} \\ &+ \mathbf{D_e}(\tau)[\mathbf{b} + \mathbf{b_e}(\tau)] - \boldsymbol{b}\| \\ &\leq B_x\|(\boldsymbol{D_e}(\tau) - \boldsymbol{I})(\mathbf{A}\boldsymbol{x} - \mathbf{b})\| \\ &+ B_x\|\boldsymbol{D_e}(\tau)(\boldsymbol{R_e}(\tau)\boldsymbol{x} - \mathbf{b_e}(\tau))\| \\ &\leq B_x^2(\|(\boldsymbol{D_e}(\tau) - \boldsymbol{I})\| \cdot \|\boldsymbol{A}\| + \|\boldsymbol{R_e}(\tau)\|) \\ &+ B_x(\|\boldsymbol{D_e} - \boldsymbol{I}\|\|\mathbf{b}\| + \|\mathbf{b_e}\|) \end{aligned} \quad (30)$$

where the first step used $\|\boldsymbol{\lambda}\| \leq B_x$, and the second step used $\|\boldsymbol{D_e}\| \leq 1$ and $\|\boldsymbol{x}\| \leq B_x$. Combining (29) and (30), we can bound $K_{\boldsymbol{R_e},\boldsymbol{b_e},\boldsymbol{D_e}}$. Finally, by plugging (26)-(30) into (21) in Lemma 3 and rearranging the terms, it is straight forward to prove the result in this theorem. □

Corollary 1. (**Convergence**) *For any $\epsilon_1 > 0$, there exist stepsize α and iteration number τ, such that*

$$f^* - f(\bar{\boldsymbol{x}}(\tau)) \leq \epsilon_1 + \frac{1}{\tau}\sum_{i=0}^{\tau-1} Q_i. \quad (31)$$

Proof. We choose $\alpha = 1/Z$ and $\tau = Z^2$. Since B_x and B_1 are constants, not depending on α or τ, we have

$$\frac{B_x}{\alpha\tau} + \frac{\alpha B_1}{2} = \frac{B_x + B_1/2}{Z} \to 0, \text{ as } Z \to \infty, \quad (32)$$

which implies that there exist α and τ, such that (32) is no greater than ϵ_1. According to Theorem 1, this is exactly the result in this corollary. □

Corollary 2. (**Optimality bound**) *Suppose the three types of network inference errors occur independently in each iteration with probabilities $\theta_D, \theta_R, \theta_b$, respectively. For any $\epsilon_2 > 0$, there exist stepsize α and iteration number τ, such that*

$$f^* - f(\bar{\boldsymbol{x}}(\tau)) \leq \epsilon_2 + B_2B_D\theta_D + 2B_x^2\theta_R + B_x\theta_b. \quad (33)$$

In other words, the optimality gap diminishes linearly as the probabilities of inference errors decrease.

Proof. We again choose $\alpha = 1/Z$ and $\tau = Z^2$. From Corollary 1, for $\epsilon_2/2$, there exists Z_1, such that $f^* - f(\bar{\boldsymbol{x}}(\tau)) \leq \epsilon_2/2 + \frac{1}{\tau}\sum_{i=0}^{\tau-1} Q_i$. Next, we recognize that $||\boldsymbol{D_e}(i) - \boldsymbol{I}||$ are a sequence of *i.i.d.* random variables bounded by B_D. Due to the law of large number, it is easy to see that there exists Z_2, such that $\tau = Z_2^2$ satisfy

$$\frac{1}{\tau}\sum_{i=0}^{\tau-1} ||\boldsymbol{D_e}(i) - \boldsymbol{I}|| \leq \frac{\epsilon_2}{6} + \mathbb{E}[||\boldsymbol{D_e}(i) - \boldsymbol{I}||] \leq \frac{\epsilon_2}{6} + \theta_D B_D,$$

where $\mathbb{E}[\cdot]$ is the expectation function, and we used the bound B_D in the last step. Using exactly the same method, we can prove that there exists Z_3 and Z_4, such that

$$\frac{1}{\tau}\sum_{i=0}^{\tau-1}||\boldsymbol{R}_{\boldsymbol{e}}(i)|| \le \frac{\epsilon_2}{6}+\theta_R B_R,\ \frac{1}{\tau}\sum_{i=0}^{\tau-1}||\boldsymbol{b}_{\boldsymbol{e}}(i)|| \le \frac{\epsilon_2}{6}+\theta_b B_b.$$

If we choose $Z=\max(Z_1,Z_2,Z_3,Z_4)$, then all the inequalities above are satisfied simultaneously. It directly leads to

$$\begin{aligned} & f^* - f(\bar{\boldsymbol{x}}(\tau)) \\ & \le \frac{\epsilon_2}{2}+\frac{1}{\tau}\sum_{i=0}^{\tau-1}Q_i \\ & = \frac{\epsilon_2}{2}+\frac{\epsilon_2}{6}+\theta_D B_D+\frac{\epsilon_2}{6}+\theta_R B_R+\frac{\epsilon_2}{6}+\theta_b B_b, \end{aligned}$$

which completes the proof of this corollary. □

Remark 2: Corollary 1 proves that for sufficiently small α and sufficiently large $\alpha\tau$, distributed optimization algorithms under imperfect network inference can converge to the optimal solution with an optimality gap $v = \frac{1}{\tau}\sum_{i=0}^{\tau-1}Q_i = B_2 B_D\theta_D + 2B_x^2\theta_R + B_x\theta_b$, determined by average inference error. Further, as the probabilities $\theta_D,\theta_R,\theta_b$ of inference errors decrease, the optimality gap diminishes linearly and becomes righter, as shown in Corollary 2. It also allows us to separately characterize the impact of each type of inference error.

Corollary 3. *(***Iteration number bound***) Suppose the three types of network inference errors occur independently in each iteration with probabilities $\theta_D,\theta_R,\theta_b$, respectively. For any $\epsilon_2 > 0$, there exist step size $\alpha = \sqrt{\frac{B_xB_1}{2\tau}}$, for any iteration number $\tau > \frac{16B_x+B_1^4B_x+8B_1^2B_x}{(f^*-v-f(\bar{\boldsymbol{x}}(\tau))-\epsilon_2)^2}$, the difference between $f(\bar{\boldsymbol{x}}(\tau))$ and theoretical optimal value f^* converges to optimality gap v:*

$$f^* - f(\bar{\boldsymbol{x}}(\tau)) \le \epsilon_2 + v \tag{34}$$

$$\tag{35}$$

Proof. From theorem 1 and $\alpha = \sqrt{\frac{B_xB_1}{2\tau}}$, we have

$$f^* - f(\bar{\boldsymbol{x}}(\tau)) \le \sqrt{\frac{2B_x}{B_1\tau}}+\sqrt{\frac{B_1^3B_x}{8\tau}}+\frac{1}{\tau}\sum_{i=0}^{\tau-1}Q_i \tag{36}$$

From corollary 2, given optimality gap $v = B_2B_D\theta_D + 2B_x^2\theta_R + B_x\theta_b$, and (36), we have

$$\begin{aligned} f^* - f(\bar{\boldsymbol{x}}(\tau)) \ & \le \ \epsilon_2+\sqrt{\frac{2B_x}{B_1\tau}}+\sqrt{\frac{B_1^3B_x}{8\tau}}+B_2B_D\theta_D \\ & \quad\ +2B_x^2\theta_R+B_x\theta_b \\ & \le \ \epsilon_2+\sqrt{\frac{2B_x}{B_1\tau}}+\sqrt{\frac{B_1^3B_x}{8\tau}}+v \end{aligned} \tag{37}$$

From (37), we have

$$\begin{aligned} f^*-v-f(\bar{\boldsymbol{x}}(\tau))-\epsilon_2 \ & \le \ \sqrt{\frac{2B_x}{B_1\tau}}+\sqrt{\frac{B_1^3B_x}{8\tau}} \\ (f^*-v-f(\bar{\boldsymbol{x}}(\tau))-\epsilon_2)^2 \ & \le \ \frac{2B_x}{B_1\tau}+\frac{B_1^3B_x}{8\tau}+\frac{B_1B_x}{\tau} \\ \tau \ & \le \ \frac{16B_x+B_1^4B_x+8B_1^2B_x}{(f^*-v-f(\bar{\boldsymbol{x}}(\tau))-\epsilon_2)^2} \end{aligned}$$

which completes the proof of this corollary. □

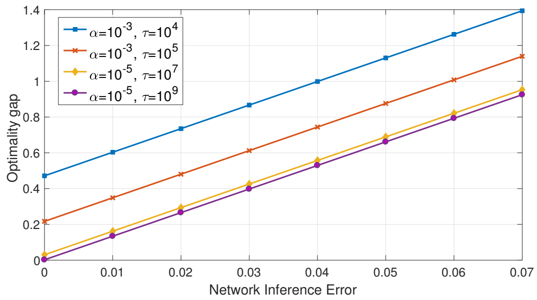

Fig. 3. An example of our optimality bound for different network inference errors and algorithm parameters.

To demonstrate application of our results, we consider the simple network example shown in Section III and compute its optimality bound without running the distributed algorithm. There are 3 flows and 2 bottlenecks $x_1+x_2 \le 2$ and $x_2+x_3 \le 2$. Thus, we have $\|\mathbf{A}\|=\sqrt{3}$ and $\|\mathbf{b}\|=2\sqrt{2}$. Since each flow is limited by the maximum link capacity, we also have $\|\boldsymbol{x}\|=2\sqrt{3}$. Consider the throughput optimization with objective $f=\sum_i x_i$. It is easy to see that $\|\nabla_{\boldsymbol{x}}f(\boldsymbol{x})\| \le B_{U'}=\sqrt{3}$. For each iteration τ, we consider all three types of (independent) inference errors with probability $\theta_D=\theta_B=\theta_b$. More precisely, with the same probability, one of the two bottlenecks will be be absent (thus $\|\mathbf{D}_{\mathbf{e}}(\tau)-\boldsymbol{I}\| \le B_D = 1$), one flow will be incorrectly added/inserted into the bottleneck (thus $B\|\mathbf{R}_{\mathbf{e}}(\tau)\| \le B_R = 1$), while the link capacity estimate error could swing in band ± 0.1 (thus $\|\mathbf{b}_{\mathbf{e}}(\tau)\| \le B_b = 0.1\cdot\sqrt{2}$). Leveraging our analytical bound, we can compute an upper bound of $f^*-f(\bar{\boldsymbol{x}}(\tau))$ in closed form, which is shown in Fig. 3 for different inference error probabilities and algorithm parameters. The bound improves with smaller stepsize α and larger number of iterations τ. It also diminishes linearly as the inference error decreases, as shown in Fig. 3.

V. Evaluation and Numerical Results

A. Methodology and NS2 Setup

We validate our theoretical results on algorithm convergence and optimality through extensive NS2 simulations with real world IoT nodes' transmission rate as well as file size with real IoT data set.

In simulation, we use file size from two real IoT data sets: (1) consisting of 200 hours of accelerometer information recorded over 25 days from 5 participants [32], with each data point in the size range of 80-187MB. (2) a series of continuous frames extracted from a traffic video sequence recorded by stationary cameras[33][34]. For the transmission rate of nodes, we set an uniform combination of 250Kbps and 54Mbps. This is because if IoT nodes communicating with its cluster header through IEEE 802.15.4, NB-IoT, or ZigBee, in which the transmission rates are 250Kbps; if the IoT nodes communicating through IEEE 802.11 a/g, in which the transmission rate is 54 Mbps.

We write a script to randomly generate a network with K clusters (as an input parameter in our evaluation) in an edge network. Each cluster contains a random number of edge entities, uniformly distributed in $[2, 5]$, one of which is selected as the cluster head. Then, we randomly select edge entities as traffic sources and destinations to generate three types of data flows (totally F flows): (i) peer flows, beginning with and ending in the same cluster, (ii) multiple-to-one flows, which transfer data between edge entities and cluster heads, and (iii) cross-cluster flows, from one cluster to another, traversing cluster head and intermediary nodes. We also vary assigned capacities to central links (connecting intermediaries and cluster heads) and local links (connecting edge entities).

We implement the distributed traffic optimization algorithm with the projected gradient method for stepsize α. Two different utility functions are considered, including total throughput $\sum_i x_i$ and proportional fair utility $\sum_i w_i \log(x_i)$, where weights w_i are uniformly distributed in $[1, 2]$. We run the algorithm under all three types of (coupled) inference errors - absent bottlenecks, incorrect bottlenecks, and inaccurate bottleneck capacity - with error rates $r\%$ ranging from 0 to 20%. Since most existing works on network tomography demonstrated estimation inference error below 20%, we set 20% as the maximum error rate of our simulation setting. The inference errors are generated randomly and vary from iteration to iteration.

Since NS2 records the status and time stamps of each packet, including the time it departs and arrives at each intermediary/source/destination, we analyze the package logs to obtain flow rates, throughput, utility values, and average packet delays. An example of our randomly generated networks is illustrated in Fig. 4, where the denser packets indicate the higher flow rates. All numerical results presented in our evaluation are the average of 2000-5000 runs. Table I summarizes key parameters.

B. Numerical Results

Fig. 8 shows the impact of bottleneck inference error on the speed of convergence for the distributed traffic optimization algorithms. We consider the α-fair utility function $u = \sum_i w_i log(x_i)$ with uniform weights in $[1, 2]$. While it is proven that the algorithms always converge under bounded inference errors, higher inference error leads to slower convergence, as the (average) required number of iterations increases from 292 to 458 (i.e., 58% increase) when bottleneck inference error grows to 20%. For the same setup, Fig 9 plots the distribution of normalized utility (i.e., the Cumulative Distribution Function, or CDF) achieved under different error rates. Higher bottleneck inference error not only causes lower achievable utility on average, it also increases the spread of achievable utility, implying higher variation in the outcome.

In Fig. 7, we optimize the total throughout $\sum_i x_i$ under 0%, 10%, and 20% bottleneck inference error (for all three types of errors), as the central link capacity (connecting intermediaries and cluster heads) increases from 1Gbps to 10Gbps, while local link are at 2Mbps. It can be seen that, as central link capacity increases, the relative difference between the optimal throughput achieved under different error rates becomes smaller. For example, the normalized throughput loss due to 20% inference error decreases from 52% to less than 10% as central link capacity grows from 1Gbps to 10Gbps. It implies tha accurate bottleneck inference is more crucial for optimizing resource constrained networks with high congestion.

For the same simulation setup, Fig. 5 illustrates the adverse effect of bottleneck inference on packet delay. Higher bottleneck inference error results in higher packet delay, for up to 49%. Also, we notice that under bottleneck inference error, packet delay increases quickly as central link capacity drops, highlighting the importance of accurate inference in networks that are already congested. Similarly, Fig. 6 shows the breakdown of data rates assigned to each flow in the network with central link capacity , when central link capacity is 2Gbps. While total throughput suffers from higher bottleneck inference error (as witnessed in Fig. 7), there exist flows that actually benefit from such inference error, since missing bottlenecks may unfairly assign high data rate to certain flows, at the expense of decreasing overall network performance.

Next, we study the impact of each type of inference errors, including (i) absent bottlenecks, (ii) incorrect bottlenecks, and (iii) inaccurate bottleneck capacity. In particular, we vary each type of inference error from 2% to 20% error rate, while keeping the other two types of errors fixed. The local link capacity is a combination of 0.25Mbps and 45Mbps, and the central link capacity is 10 Gbps. Fig. 10 shows the (achieved) normalized utility with respect to each type of inference error. Interestingly, absent bottlenecks result in the smallest utility loss. It is because dynamic inference errors change on the same time-scale as the distributed optimization algorithm, and thus the gradient-based update can still take different subsets of bottlenecks (i.e., active constraints) into account from iteration to iteration, leading to efficient computation of optimal utility. On the other hand, inaccurate bottleneck capacities have the highest impact on achievable utility, due to the accumulation and propagation of gradient error in the distributed algorithm.

Finally, we study the scalability of distributed traffic optimization under bottleneck inference errors, by increasing the network size from $N = 100$ nodes to $N = 500$ nodes, while the number of flows also increases proportionally at $F = 0.7 \cdot N$. It can be seen in Fig.11 that the network grows, the achieved utility (normalized by the optimal utility in the error-free case) drops quickly. This is because larger networks lead to longer flows (that are subject to more link capacity constraints) and more network bottlenecks (both on central and local links), and thus are more sensitive to inference error. Fig. 12 demonstrates similar results when the number of flows increases from 100 to 400, while the size of the network is fixed at 500 nodes. Again, inference errors cause larger utility loss in more heavily congested networks.

VI. Conclusions

Under imperfect bottleneck inference, we prove the convergence and optimality of distributed traffic optimization algorithms relying on projected gradient method. It is shown that the algorithms converge to a neighborhood of the optimal solution arbitrary, bounded inference errors. The optimality

Figures	# of nodes N	# of flows F	Error rate, type	Utility Function	Local Link	Central Link
7, 5, 6	500	20	0-20%, all	$\sum_i x_i$ (throughput)	0.25,54 Mbps	1-10 Gbps
8	500	$0.7 \cdot N$	0-20%, all	$\sum_i w_i log(x_i)$	0.25,54 Mbps	10 Gbps
10	500	$0.7 \cdot N$	0-20%, single	$\sum_i w_i log(x_i)$	0.25,54 Mbps	10 Gbps
9	500	$0.7 \cdot N$	0-20%, all	$\sum_i w_i log(x_i)$	0.25,54 Mbps	10 Gbps
11	100-500	$0.7 \cdot N$	0-20%, all	$\sum_i w_i log(x_i)$	0.25,54 Mbps	10 Gbps
12	500	$0.1 - 0.8 \cdot N$	0-20%, all	$\sum_i w_i log(x_i)$	0.25,54 Mbps	10 Mbps

TABLE I

KEY PARAMETERS IN OUR NS2 SIMULATION.

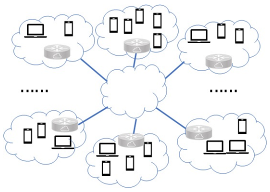

Fig. 4. Topology

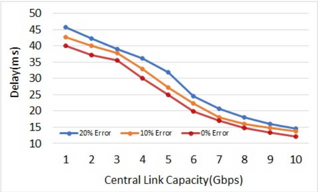

Fig. 5. Average packet delay

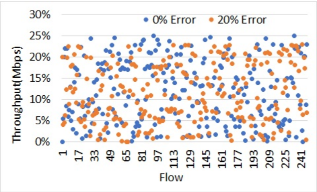

Fig. 6. Breakdown of different flows rates

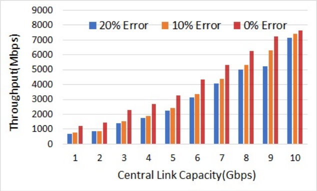

Fig. 7. Average packet delay

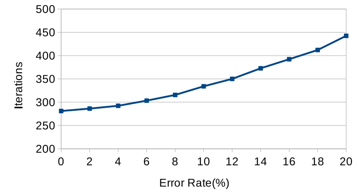

Fig. 8. Speed of convergence for different error rates

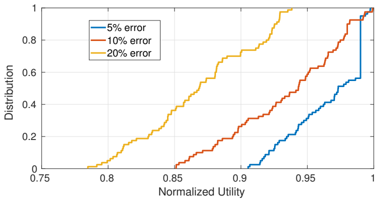

Fig. 9. Distribution of normalized utility

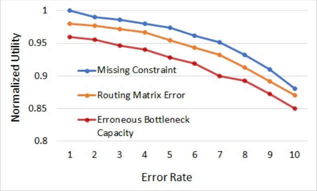

Fig. 10. The impact of different types of errors

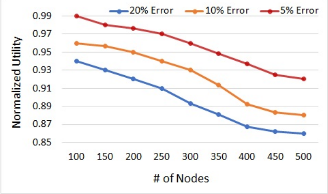

Fig. 11. Increasing # of nodes

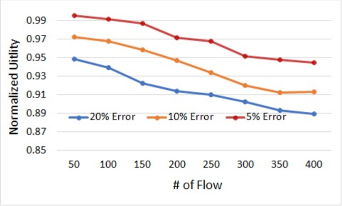

Fig. 12. Increasing # of flows

gap is quantified in closed form and is shown to be proportional to average inference error. The theoretical results are validated via extensive NS2 simulations and provide valuable insights on distributed network operations and algorithms under the impact of imperfect network inference.

REFERENCES

[1] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[2] R. Mahmud, R. Kotagiri, and R. Buyya, "Fog computing: A taxonomy, survey and future directions," in *Internet of everything*. Springer, 2018, pp. 103–130.

[3] J. Gao, L. Zhao, and X. Shen, "Network utility maximization based on an incentive mechanism for truthful reporting of local information," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 8, pp. 7523–7537, 2018.

[4] Y. Wang, W. Wang, Y. Cui, K. G. Shin, and Z. Zhang, "Distributed packet forwarding and caching based on stochastic network utility maximization," *IEEE/ACM Transactions on Networking*, vol. 26, no. 3, pp. 1264–1277, 2018.

[5] S. Singh, S.-p. Yeh, N. Himayat, and S. Talwar, "Optimal traffic aggregation in multi-rat heterogeneous wireless networks," in *Communications Workshops (ICC), 2016 IEEE International Conference on*. IEEE, 2016, pp. 626–631.

[6] D. P. Palomar and M. Chiang, "Alternative distributed algorithms for network utility maximization: Framework and applications," *IEEE Transactions on Automatic Control*, vol. 52, no. 12, pp. 2254–2269, 2007.

[7] A. Koppel, F. Y. Jakubiec, and A. Ribeiro, "A saddle point algorithm for networked online convex optimization," *IEEE Transactions on Signal Processing*, vol. 63, no. 19, pp. 5149–5164, 2015.

[8] N. Gatsis and G. B. Giannakis, "Power control with imperfect exchanges and applications to spectrum sharing," *IEEE Transactions on Signal Processing*, vol. 59, no. 7, pp. 3410–3423, 2011.

[9] D. Katabi, I. Bazzi, and X. Yang, "A passive approach for detecting shared bottlenecks," in *Computer Communications and Networks, 2001. Proceedings. Tenth International Conference on*. IEEE, 2001, pp. 174–181.

[10] M. Fazlyab and V. M. Preciado, "Robust topology identification and control of lti networks," in *Signal and Information Processing (GlobalSIP), 2014 IEEE Global Conference on*. IEEE, 2014, pp. 918–922.

[11] Y. Che, R. Li, C. Han, J. Wang, S. Cui, B. Deng, and X. Wei, "Adaptive lag synchronization based topology identification scheme of uncertain general complex dynamical networks," *The European Physical Journal B*, vol. 85, no. 8, pp. 1–8, 2012.

[12] T. He, A. Gkelias, L. Ma, K. K. Leung, A. Swami, and D. Towsley, "Robust and efficient monitor placement for network tomography in dynamic networks," *IEEE/ACM Transactions on Networking*, 2017.

[13] R. L. Carter and M. E. Crovella, "Measuring bottleneck link speed in packet-switched networks," *Performance evaluation*, vol. 27, pp. 297–318, 1996.

[14] N. Hu, L. E. Li, Z. M. Mao, P. Steenkiste, and J. Wang, "Locating internet bottlenecks: Algorithms, measurements, and implications," in *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 4.

ACM, 2004, pp. 41–54.

[15] D. Katabi, I. Bazzi, and X. Yang, “An information theoretic approach for shared bottleneck inference based on end-to-end measurements,” *Class Project, MIT Laboratory for Computer Science*, 1999.

[16] M. Coates, R. Castro, R. Nowak, M. Gadhiok, R. King, and Y. Tsang, “Maximum likelihood network topology identification from edge-based unicast measurements,” in *ACM SIGMETRICS Performance Evaluation Review*, vol. 30, no. 1. ACM, 2002, pp. 11–20.

[17] O. Younis and S. Fahmy, “Flowmate: Scalable on-line flow clustering,” *IEEE/ACM Transactions on Networking (TON)*, vol. 13, no. 2, pp. 288–301, 2005.

[18] Y. Vardi, “Network tomography: Estimating source-destination traffic intensities from link data,” *Journal of the American statistical association*, vol. 91, no. 433, pp. 365–377, 1996.

[19] D. Katabi and C. Blake, “Inferring congestion sharing and path characteristics from packet interarrival times,” *Mass. Inst. Technol., Cambridge, MA, MIT-LCS-TR-828*, 2001.

[20] O. Gurewitz and M. Sidi, “Estimating one-way delays from cyclic-path delay measurements,” in *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2. IEEE, 2001, pp. 1038–1044.

[21] Y. Chen, D. Bindel, H. Song, and R. H. Katz, “An algebraic approach to practical and scalable overlay network monitoring,” in *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 4. ACM, 2004, pp. 55–66.

[22] A. Chen, J. Cao, and T. Bu, “Network tomography: Identifiability and fourier domain estimation,” *IEEE Transactions on Signal Processing*, vol. 58, no. 12, pp. 6029–6039, 2010.

[23] D. Battré, N. Frejnik, S. Goel, O. Kao, and D. Warneke, “Evaluation of network topology inference in opaque compute clouds through end-to-end measurements,” in *Cloud Computing (CLOUD), 2011 IEEE International Conference on*. IEEE, 2011, pp. 17–24.

[24] Y. Hu, X. Yang, and C. Yu, “Subgradient methods for saddle point problems of quasiconvex optimization,” *Pure Appl Funct Anal*, vol. 2, no. 1, pp. 83–97, 2017.

[25] G. Wang and T. E. Ng, “The impact of virtualization on network performance of amazon ec2 data center,” in *INFOCOM, 2010 Proceedings IEEE*. IEEE, 2010, pp. 1–9.

[26] V. Chandra, M. E. McCorry, D. V. H. Quan, P. J. Schwaller, C. D. Selvaggi, and J. L. Wood, “Methods, systems and computer program products for scheduled network performance testing,” May 28 2002, uS Patent 6,397,359.

[27] S. Ratnasamy and S. McCanne, “Inference of multicast routing trees and bottleneck bandwidths using end-to-end measurements,” in *INFOCOM’99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 1. IEEE, 1999, pp. 353–360.

[28] F. Morales, M. Ruiz, and L. Velasco, “Virtual network topology reconfiguration based on big data analytics for traffic prediction,” in *Optical Fiber Communication Conference*. Optical Society of America, 2016, pp. Th3I–5.

[29] F. Morales, M. Ruiz, L. Gifre, L. M. Contreras, V. López, and L. Velasco, “Virtual network topology adaptability based on data analytics for traffic prediction,” *IEEE/OSA Journal of Optical Communications and Networking*, vol. 9, no. 1, pp. A35–A45, 2017.

[30] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, “Fog computing: A platform for internet of things and analytics,” in *Big Data and Internet of Things: A Roadmap for Smart Environments*. Springer, 2014, pp. 169–186.

[31] S. Sardellitti, G. Scutari, and S. Barbarossa, “Joint optimization of radio and computational resources for multicell mobile-edge computing,” *IEEE Transactions on Signal and Information Processing over Networks*, vol. 1, no. 2, pp. 89–103, 2015.

[32] M. Cong, K. Kim, M. Gorlatova, J. Sarik, J. Kymissis, and G. Zussman, “CRAWDAD dataset columbia/kinetic (v. 2014-05-13),” Downloaded from https://crawdad.org/columbia/kinetic/20140513/kinetic-energy, May 2014, traceset: kinetic-energy.

[33] H. Yan, X. Li, Y. Wang, and C. Jia, “Centralized duplicate removal video storage system with privacy preservation in iot,” *Sensors*, vol. 18, no. 6, p. 1814, 2018.

[34] M. Wang, W. Li, and X. Wang, “Transferring a generic pedestrian detector towards specific scenes,” in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 3274–3281.

Appendix

A. Proof of Lemma 1

Proof. First, according to the non-expansive property of the projection [8] and the definition of gradient (3), we have:

$$\begin{aligned}&\|\boldsymbol{x}(\tau+1)-\boldsymbol{x}\|^2 \\ &\leq \|\boldsymbol{x}(\tau)+\alpha\nabla_{\boldsymbol{x}}\hat{L}-\boldsymbol{x}\|^2 \\ &\leq \|\boldsymbol{x}(\tau)-\boldsymbol{x}\|^2+2\alpha(\boldsymbol{x}(\tau)-\boldsymbol{x})\nabla_{\boldsymbol{x}}\hat{L}+\alpha^2(\nabla_{\boldsymbol{x}}\hat{L})^2\end{aligned} \tag{38}$$

Consider the definition of erroneous Lagrangian functions (6) and (7). We rewrite the Lagrangian functions as:

$$\begin{aligned}\nabla_{\boldsymbol{x}}\hat{L} &= \nabla_{\boldsymbol{x}}L+\boldsymbol{\lambda}^T(\tau)(\boldsymbol{D}_{\mathbf{e}}(\tau)\boldsymbol{A}+\boldsymbol{D}_{\mathbf{e}}(\tau)\boldsymbol{R}_{\mathbf{e}}(\tau)-\boldsymbol{A}) \\ &= \nabla_{\boldsymbol{x}}L+\Gamma_1(\tau)\end{aligned} \tag{39}$$

Then, using the concavity of the Lagrangian functions with respect to primal variable $\boldsymbol{x}$, we have

$$\begin{aligned}&L(\boldsymbol{x},\boldsymbol{\lambda}(\tau))-L(\boldsymbol{x}(\tau),\boldsymbol{\lambda}(\tau)) \\ &\quad\leq \nabla_{\boldsymbol{x}}^T L(\boldsymbol{x}(\tau),\boldsymbol{\lambda}(\tau))(\boldsymbol{x}-\boldsymbol{x}(\tau))\end{aligned} \tag{40}$$

Pluging (40) and (39) into (38), we can directly obtain the result (11) in Lemma 1, i.e.,

$$\begin{aligned}&\|\boldsymbol{x}(\tau+1)-\boldsymbol{x}\|^2-\|\boldsymbol{x}(\tau)-\boldsymbol{x}\|^2 \\ &\leq 2\alpha(\boldsymbol{x}(\tau)-\boldsymbol{x})(\nabla_{\boldsymbol{x}}L+\boldsymbol{\lambda}^T\Gamma_1(\tau))+\alpha^2(\nabla_{\boldsymbol{x}}\hat{L})^2 \\ &\leq -2\alpha[L(\boldsymbol{x},\boldsymbol{\lambda}(\tau))-L(\boldsymbol{x}(\tau),\boldsymbol{\lambda}(\tau))] \\ &\quad +2\alpha(\boldsymbol{x}(\tau)-\boldsymbol{x})\Gamma_1(\tau)+\alpha^2(\nabla_{\boldsymbol{x}}\hat{L})^2\end{aligned} \tag{41}$$

where the first step is obtained by plugging (40) into (38), and the second step follows directly by applying (39).

Next, the proof of (12) is similar. Due to the non-expansive property of the projection and the definition of gradient (4), we have

$$\begin{aligned}&\|\boldsymbol{\lambda}(\tau+1)-\boldsymbol{\lambda}\|^2 \\ &\leq \|\boldsymbol{\lambda}(\tau)-\alpha\nabla_{\boldsymbol{\lambda}}\hat{L}-\boldsymbol{\lambda}\|^2 \\ &\leq \|\boldsymbol{\lambda}(\tau)-\boldsymbol{\lambda}\|^2+2\alpha(\boldsymbol{\lambda}(\tau)-\boldsymbol{\lambda})\nabla_{\boldsymbol{\lambda}}\hat{L}+\alpha^2(\nabla_{\boldsymbol{\lambda}}\hat{L})^2\end{aligned} \tag{42}$$

Using the erroneous Lagrangian function (6) and (7), we derive

$$\begin{aligned}\nabla_{\boldsymbol{\lambda}}\hat{L} &= \nabla_{\boldsymbol{\lambda}}L-\mathbf{D}_{\mathbf{e}}(\tau)[\mathbf{A}+\mathbf{R}_{\mathbf{e}}(\tau)]\boldsymbol{x} \\ &\quad +\mathbf{D}_{\mathbf{e}}(\tau)[\mathbf{b}+\mathbf{b}_{\mathbf{e}}(\tau)]+\boldsymbol{A}\boldsymbol{x}-b \\ &= \nabla_{\boldsymbol{\lambda}}L+\Gamma_2(\tau)\end{aligned} \tag{43}$$

Based on the convexity of the Lagrangian functions with respect to dual variable $\boldsymbol{\lambda}$, we have:

$$L(\boldsymbol{x}(\tau),\boldsymbol{\lambda}(\tau)-L(\boldsymbol{x}(\tau),\boldsymbol{\lambda})\geq\nabla_{\boldsymbol{\lambda}}^T L(\boldsymbol{x}(\tau),\boldsymbol{\lambda}(\tau))(\boldsymbol{\lambda}(\tau)-\boldsymbol{\lambda}) \tag{44}$$

Finally, plugging (43) and (44) into (42), we can obtain the result (12) in Lemma 1:

$$\begin{aligned}&\|\boldsymbol{\lambda}(\tau+1)-\boldsymbol{\lambda}\|^2-\|\boldsymbol{\lambda}(\tau)-\boldsymbol{\lambda}\|^2 \\ &\leq 2\alpha(\boldsymbol{\lambda}(\tau)-\boldsymbol{\lambda})(\nabla_{\boldsymbol{\lambda}}L+\Gamma_2(\tau))+\alpha^2(\nabla_{\boldsymbol{\lambda}}\hat{L})^2 \\ &\leq 2\alpha[L(\boldsymbol{x}(\tau),\boldsymbol{\lambda}(\tau)-L(\boldsymbol{x}(\tau),\boldsymbol{\lambda})] \\ &\quad +2\alpha\|(\boldsymbol{\lambda}(\tau)-\boldsymbol{\lambda})\Gamma_2(\tau))\|+\alpha^2(\nabla_{\boldsymbol{\lambda}}\hat{L})^2,\end{aligned} \tag{45}$$

where the first step uses (43), (42), the last step uses (44). □

B. Proof of Lemma 2

Proof. We first prove (15). Dividing both sides of (11) by 2α and applying (17), we can obtain:

$$\begin{aligned} & L(\boldsymbol{x},\boldsymbol{\lambda}(\tau)) - L(\boldsymbol{x}(\tau),\boldsymbol{\lambda}(\tau)) \\ & \le \frac{1}{2\alpha}(\|\boldsymbol{x}(\tau)-\boldsymbol{x}\|^2 - \|\boldsymbol{x}(\tau+1)-\boldsymbol{x}\|^2) \\ & \quad + \frac{\alpha}{2}\|\nabla_{\boldsymbol{x}}\hat{L}\|^2 + \|\Gamma_0(\tau)\| \end{aligned} \tag{46}$$

We consider the sum of a sequence of inequalities (46) from iteration 0 to $\tau-1$. It results in

$$\begin{aligned} & \frac{1}{\tau}\sum_{i=0}^{\tau-1}[L(\boldsymbol{x},\boldsymbol{\lambda}(\tau)) - L(\boldsymbol{x}(\tau),\boldsymbol{\lambda}(\tau))] \\ & \le \frac{1}{2\alpha\tau}(\|\boldsymbol{x}(0)-\boldsymbol{x}\|^2 - \|\boldsymbol{x}(\tau)-\boldsymbol{x}\|^2) \\ & \quad + \frac{1}{\tau}\sum_{i=0}^{\tau-1}(\|\Gamma_0(i)\| + \frac{\alpha}{2}\|\nabla_{\boldsymbol{x}}\hat{L}\|^2) \end{aligned} \tag{47}$$

Due to the convexity of $L(\boldsymbol{x},\boldsymbol{\lambda})$ with respect to $\boldsymbol{\lambda}$, we have

$$\begin{aligned} \tfrac{1}{\tau}\textstyle\sum_{i=0}^{\tau-1} L(\boldsymbol{x},\boldsymbol{\lambda}(i)) & \le L(\boldsymbol{x},\bar{\boldsymbol{\lambda}}(\tau)) \le L(\boldsymbol{x},\boldsymbol{\lambda}) \\ & \le L(\boldsymbol{x}^*,\boldsymbol{\lambda}^*) \le f^* \end{aligned} \tag{48}$$

Then, to obtain (15) in Lemma 2:

$$\begin{aligned} & f^* - L(\boldsymbol{x}(\tau),\boldsymbol{\lambda}(\tau))] \qquad (49) \\ & \le \frac{1}{\tau}\sum_{i=0}^{\tau-1}[L(\boldsymbol{x},\boldsymbol{\lambda}(\tau)) - L(\boldsymbol{x}(\tau),\boldsymbol{\lambda}(\tau))] \\ & \le \frac{\|\boldsymbol{x}(0)-\boldsymbol{x}\|^2}{2\alpha\tau} + \frac{1}{\tau}\sum_{i=0}^{\tau-1}(\|\Gamma_0(i)\| + \frac{\alpha}{2}\|\nabla_{\boldsymbol{x}}\hat{L}\|^2), \end{aligned}$$

where the first step is obtained by plugging (48) into (**??**), and the second step since (for $\forall\tau$) $\|\boldsymbol{x}(\tau)-\boldsymbol{x}\|^2/(2\alpha\tau) \ge 0$.

Next, we prove (16). Dividing both sides of (12) in Lemma 1 by 2α, we can obtain:

$$\begin{aligned} & L(\boldsymbol{x}(\tau),\boldsymbol{\lambda}) - L(\boldsymbol{x}(\tau),\boldsymbol{\lambda}(\tau)) \\ & \le \frac{\alpha\|\nabla_{\boldsymbol{\lambda}}\hat{L}\|^2}{2} + \|\Gamma_2(\tau)(\boldsymbol{\lambda}(\tau)-\boldsymbol{\lambda})\| \\ & \quad + \frac{1}{2\alpha}(\|\boldsymbol{\lambda}(\tau)-\boldsymbol{\lambda}\|^2 - \|\boldsymbol{\lambda}(\tau+1)-\boldsymbol{\lambda}\|^2) \end{aligned} \tag{50}$$

Then, adding up the inequalities (50) from iteration 0 to $\tau-1$ and dividing the result by τ, we find

$$\begin{aligned} & \frac{1}{\tau}\sum_{i=0}^{\tau-1}[L(\boldsymbol{x}(\tau),\boldsymbol{\lambda}) - L(\boldsymbol{x}(\tau),\boldsymbol{\lambda}(\tau))] \\ & \le \frac{\|\boldsymbol{\lambda}(0)-\boldsymbol{\lambda}\|^2}{2\alpha\tau} - \frac{\|\boldsymbol{\lambda}(\tau)-\boldsymbol{\lambda}\|^2}{2\alpha\tau} \\ & \quad + \frac{1}{\tau}\sum_{i=0}^{\tau-1}(\|\Gamma_2(i)(\boldsymbol{\lambda}(i)-\boldsymbol{\lambda})\| + \frac{\alpha}{2}\|\nabla_{\boldsymbol{\lambda}}\hat{L}\|^2) \end{aligned}$$

Since (for $\forall\tau$) $\|\boldsymbol{\lambda}(\tau)-\boldsymbol{\lambda}\|^2/(2\alpha\tau) \ge 0$, this directly leads to (16). □

C. Proof of Lemma 3

Proof. We use the concavity of the utility function, or $f(\boldsymbol{x})$:

$$f(\bar{\boldsymbol{x}}(\boldsymbol{\tau})) \ge \frac{1}{\tau}\sum_{i=0}^{\tau-1} f(\boldsymbol{x}(\tau)) \tag{51}$$

Applying the Lagrangian function (2) to (51), we can obtain:

$$\begin{aligned} & f^* - f(\bar{\boldsymbol{x}}(\tau)) \\ & \le -\frac{1}{\tau}\sum_{i=0}^{\tau-1} f(\boldsymbol{x}(\tau)) + f^* \\ & \le f^* - \frac{1}{\tau}\sum_{i=0}^{\tau-1}[L(\boldsymbol{x}(i),\boldsymbol{\lambda}(i)) - \boldsymbol{\lambda}^T(i)\boldsymbol{g}(\boldsymbol{x}(i))] \end{aligned} \tag{52}$$

Further, applying (**??**) to (52), make $\boldsymbol{x} = \boldsymbol{x}^*$, we can get:

$$\begin{aligned} & f^* - f(\bar{\boldsymbol{x}}(\tau)) \\ & \le f^* - \frac{1}{\tau}\sum_{i=0}^{\tau-1} L(\boldsymbol{x}(i),\boldsymbol{\lambda}(i)) + \frac{1}{\tau}\sum_{i=0}^{\tau-1}\boldsymbol{\lambda}^T(i)\boldsymbol{g}(\boldsymbol{x}(i)) \\ & \le \frac{\|\boldsymbol{x}(0)-\boldsymbol{x}^*\|^2}{2\alpha\tau} + \frac{1}{\tau}\sum_{i=0}^{\tau-1}[\frac{\alpha}{2}\|\nabla_{\boldsymbol{x}}\hat{L}\|^2 \\ & \quad + \|\Gamma_0(i)\| + \boldsymbol{\lambda}^T(i)\boldsymbol{g}(\boldsymbol{x}(i))] \end{aligned} \tag{53}$$

Since $L(\boldsymbol{x}(i),\boldsymbol{\lambda}(i)) - L(\boldsymbol{x}(i),\boldsymbol{0}) = -\boldsymbol{\lambda}^T(i)\boldsymbol{g}(\boldsymbol{x}(i))$, we make $\boldsymbol{\lambda} = \boldsymbol{0}$ in (16) in Lemma 2, which leads to

$$\begin{aligned} & \frac{1}{\tau}\sum_{i=0}^{\tau-1} -\boldsymbol{\lambda}^T(i)\boldsymbol{g}(\boldsymbol{x}(i)) \qquad (54) \\ & = \frac{1}{\tau}\sum_{i=0}^{\tau-1}[L(\boldsymbol{x}(i),\boldsymbol{0}) - L(\boldsymbol{x}(i),\boldsymbol{\lambda}(i))] \\ & \le \frac{\|\boldsymbol{\lambda}(0)\|^2}{2\alpha\tau} + \frac{1}{\tau}\sum_{i=0}^{\tau-1}(\|\Gamma_2(i)\boldsymbol{\lambda}(i)\| + \frac{\alpha}{2}\|\nabla_{\boldsymbol{\lambda}}\hat{L}\|^2) \end{aligned}$$

Then, plugging (54) into (53), we can obtain:

$$\begin{aligned} & f^* - f(\bar{\boldsymbol{x}}(\tau)) \\ & \le \frac{\|\boldsymbol{x}(0)-\boldsymbol{x}^*\|^2}{2\alpha\tau} + \frac{1}{\tau}\sum_{i=0}^{\tau-1}[\frac{\alpha}{2}\|\nabla_{\boldsymbol{x}}\hat{L}\|^2 + \|\Gamma_0(i)\|] \\ & + \frac{\|\boldsymbol{\lambda}(0)\|^2}{2\alpha\tau} + \frac{1}{\tau}\sum_{i=0}^{\tau-1}(\|\Gamma_2(i)\boldsymbol{\lambda}(i)\| + \frac{\alpha}{2}\|\nabla_{\boldsymbol{\lambda}}\hat{L}\|^2) \end{aligned}$$

which is exactly the result in Theorem 1 following (18)-(20). □