The Impact of Stochastic Noisy Feedback on Distributed Network Utility Maximization

Junshan Zhang, *Senior Member, IEEE*, Dong Zheng, and Mung Chiang, *Member, IEEE*

***Abstract*—The implementation of distributed network utility maximization (NUM) algorithms hinges heavily on information feedback through message passing among network elements. In practical systems the feedback is often obtained using error-prone measurement mechanisms and suffers from random errors. In this paper, we investigate the impact of noisy feedback on distributed NUM. We first study the distributed NUM algorithms based on the Lagrangian dual method, and focus on the primal-dual (P-D) algorithm, which is a single time-scale algorithm in the sense that the primal and dual parameters are updated simultaneously. Assuming strong duality, we study both cases when the stochastic gradients are unbiased or biased, and develop a general theory on the stochastic stability of the P-D algorithms in the presence of noisy feedback. When the gradient estimators are unbiased, we establish, via a combination of tools in Martingale theory and convex analysis, that the iterates generated by distributed P-D algorithms converge with probability one to the optimal point, under standard technical conditions. In contrast, when the gradient estimators are biased, we show that the iterates converge to a contraction region around the optimal point, provided that the biased terms are asymptotically bounded by a scaled version of the true gradients. We also investigate the rate of convergence for the unbiased case, and find that, in general, the limit process of the interpolated process corresponding to the normalized iterate sequence is a stationary reflected linear diffusion process, not necessarily a Gaussian diffusion process. We apply the above general theory to investigate stability of cross-layer rate control for joint congestion control and random access. Next, we study the impact of noisy feedback on distributed two time-scale NUM algorithms based on primal decomposition. We establish, via the mean ODE method, the convergence of the stochastic two time-scale algorithm under mild conditions, for the cases where the gradient estimators in both time scales are unbiased. Numerical examples are used to illustrate the finding that compared to the single time-scale counterpart, the two time-scale algorithm, although having lower complexity, is less robust to noisy feedback.**

***Index Terms*—Multihop wireless networks, network utility maximization, noisy feedback, primal-dual algorithm, stochastic approximation, stochastic networks, two time-scale algorithm.**

Manuscript received February 8, 2007; revised July 18, 2007. This research is supported in part by Office of Naval Research through the Grant N00014-05-1-0636 and National Science Foundation through the Grants ANI-0238550 and CNS-0721820.

J. Zhang is with the Department of Electrical Engineering, Ira A. Fulton School of Engineering, Arizona State University, Tempe, AZ 85287 USA (e-mail: junshan@asu.edu).

D. Zheng is with NextWave Broadband Inc., San Deigo, CA 92130 USA (e-mail: dzheng@nextwave.com).

M. Chiang is with the Department of Electrical Engineering, Princeton University, Princeton, NJ 08544 USA (e-mail: chiangm@princeton.edu).

Communicated by E. Modiano, Associate Editor for Communication Networks.

Color versions of Figures 2–11 in this paper are available at http:\\ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIT.2007.913572

I. Introduction

A. Motivation

THE past decade has witnessed an increasing intensity of interest in the utility maximization approach for Internet congestion control, and, more generally, for network resource allocation. Roughly speaking, network control is treated as a distributed solution to maximizing the utility functions of the users or the network, subject to resource constraints such as capacity constraints or power/energy constraints. The utility functions serve as a measure of efficiency or fairness, or model user satisfaction or traffic elasticity. More recently, this utility approach has been substantially generalized from an analytic tool of reverse-engineering TCP congestion control to a network utility maximization (NUM) framework that enables a better understanding of interactions across protocol layers and network architecture choices. This framework of "layering as optimization decomposition" offers a rigorous integration of the various protocol layers into a single coherent theory, by viewing them as carrying out an asynchronous distributed computation over the network to implicitly solve a global NUM problem. Different layers iterate on different subsets of the decision variables at different timescales using local information to achieve individual optimality.

Distributed solutions play a critical role in making the NUM framework attractive in practical networks where a centralized solution is often infeasible, nonscalable, too costly, or too fragile. The implementation of such distributed algorithms hinges heavily on the communications (message passing) among network elements (e.g., nodes and links), which take place in the form of information exchange and feedback. Unfortunately, in practical systems the feedback is often obtained using error-prone measurement mechanisms and also suffers from other random errors in its transmissions. That is to say, the feedback signals are stochastically noisy in nature. This gives rise to the following important questions.

- Would it still be possible for the distributed NUM solution, in the presence of stochastic noisy feedback, to converge to the optimal point obtained by the corresponding deterministic algorithms; and if yes, under what conditions and how quickly does it converge?
- In the case that it does not converge to the optimum point, would the distributed NUM solution be stable at all?
- Which decomposition method would yield more robust solutions, and what is the corresponding complexity?

A systematic study of the impacts of stochastic noisy feedback on distributed network utility maximization is lacking in the existing literature that almost always assumes ideal feedback. A

main objective of this study is to fill this void and to obtain some overriding principles for network resource allocation in the presence of stochastic noisy feedback.

In this paper, we consider distributed NUM under stochastic noisy feedback in wireline Internet and multihop wireless networks. Noisy feedback is ubiquitous in both wireline networks and wireless networks, although its impact in wireless systems is likely to be more significant. Indeed, we observe that the randomness in the noisy feedback signals has its root in a number of error-prone measurement mechanisms inherent in multihop networks, as illustrated by two examples we will explore in detail.

As the first example, in the Internet congestion control scheme, the congestion price from the routing nodes needs to be fed back to the source node for rate adaptation. A popular approach for obtaining the price information is the packet marking technique. Simply put, packets are marked with certain probabilities that reflect the congestion level at routing nodes. The overall marking probability depends on the marking events along the routing path that the packets traverse, and is estimated by observing the relative frequency of marked packets during a pre-specified time window at the source node. Clearly, such a feedback signal is probabilistic in nature.

As another important case, consider a wireless network based on random access, where packet transmissions take place over collision channels, and the link feedback information is (0, 1, e). Therefore, the measurement of the flow rate is sample-path-based and depends highly on the specifics of the random access mechanism and the delay can fluctuate significantly. Furthermore, the feedback signal is transmitted over the wireless channel, which is error-prone due to the channel variations of the link quality.

Researchers have recently started to appreciate that there are many decomposition methods that can solve the NUM problem in different but all distributed manners. We first investigate the impact of noise feedback on the distributed NUM algorithms based on the Lagrangian dual method. These algorithms, often regarded gradient or subgradient based algorithms, can be roughly classified into the following three categories: primal-dual (P-D) algorithms, primal algorithms and dual algorithms. We shall focus on the impact of noise feedback on the primal-dual algorithm, which is a single time-scale algorithm in the sense that the primal variables and dual variables are updated at the same time. Next, we study distributed two time-scale algorithms based on primal decomposition, and explore the impact of noisy feedback on stochastic stability therein. This study enables us to compare alternative decompositions by the important metric of robustness to noisy feedback.

B. Overview of Results

Assuming that strong duality holds, we study stochastic stability of primal-dual algorithms in the presence of noisy feedback. To this end, we examine the structure of the stochastic gradients (subgradients) that are the key elements in distributed NUM algorithms. Since in practical systems, the stochastic gradients can be either biased (e.g., when random exponential marking is used) or unbiased (e.g., when self-normalized additive marking is used), we investigate the stability for both cases: 1) for the unbiased case, we establish, via a combination of the stochastic Lyapunov Stability Theorem and local analysis, that the iterates generated by distributed P-D algorithms converge with probability one to the optimal solution corresponding to the deterministic algorithm, under standard technical conditions; and 2) in contrast, when the gradient estimator is biased, we cannot hope to obtain almost sure convergence of the iterates to the optimal points. Nevertheless, we are able to show that the iterates still move along the direction toward the optimal point and converge to a contraction region around the optimal point, provided that outside this contraction region the biased terms are asymptotically bounded by a scaled version $(0 \leq \eta < 1)$ of the true gradients. Our findings confirm that for a fixed network configuration the distributed NUM algorithms using constant step sizes are less robust to the random perturbation, in the sense that the fluctuation of the iterates remains significant after many iterations; in contrast, the fluctuation using diminishing step sizes vanishes eventually, even under stochastic noisy feedback. However, we caution that in scenarios where time-scale decomposition does not hold well, adaptive step size (not necessarily diminishing) is needed to track the network dynamics [19], [22] since diminishing step sizes may slow down convergence significantly.

We also investigate the rate of convergence for the unbiased case, and our results reveal that due to the link constraints, the limit process for the interpolated process corresponding to the normalized iterate sequence, is in general a stationary reflected linear diffusion process, not necessarily the standard Gaussian diffusion process (cf. [21]). Indeed, the reflection terms incurred by the link constraints help to increase the rate of convergence, and the spread (fluctuation) around the equilibrium point is typically smaller than the unconstrained case. Our result on the rate of convergence can be viewed as a generalization of that in the seminal work by Kelly, Maulloo, and Tan [21]. Analysis of the rate of convergence also serves as a basis for determining time scale separability in network protocol design.

To get a more concrete sense of the above results, we then apply the general theory developed above to investigate stability of cross-layer rate control for joint congestion control and MAC design. Specifically, we consider rate control in multihop wireless networks based on random access, and take into account the rate constraints imposed by the MAC/PHY layer. Appealing to the general theory above, we examine the convergence behavior of the stochastic algorithm for cross-layer rate control for both unbiased and biased cases. Our numerical examples corroborate the theoretic findings well.

Finally, we generalize the study to explore the stability behavior of distributed two time-scale NUM algorithms based on primal decomposition; In such algorithms, gradient estimators in both faster and slower time scales can be unbiased or biased. When the gradient estimators in both time scales are unbiased, we establish, via the mean ODE method, the convergence of the stochastic two time-scale algorithm under mild conditions. For the case where the gradient estimator at the faster time scale is unbiased and that at the slower time scale is biased, it can be shown that algorithm converges to a contraction region. We

should point out that stability is not well defined when the gradient estimator at the faster time scale is biased, because the cumulative effect of the biases can vary significantly. We use numerical examples to illustrate the finding that compared to the single time-scale counterpart, the two time-scale algorithm, although with lower complexity, is less robust to noisy feedback, partially due to the sensitivity of the faster time-scale loop to perturbation.

The proofs of our main results make use of a combination of tools in stochastic approximation, Martingale theory and convex analysis. We note that both rate of convergence and two time-scale algorithms for constrained optimization problems are current research topics in stochastic approximation, and indeed our proofs are built on some recent results.

C. Related Work

There has recently been a large number of work on the utility maximization approach for network resource allocation. Most relevant to Section IV of this paper are those on joint congestion control and MAC (either scheduling-based or random-access-based): e.g., [2], [4], [13], [17], [14], [24], [34], [30], [25], [33], [18], [28], [35], [36].

We now briefly summarize the differences between our work and the most closely related work, on the following two topics that form the core of this paper:

Stochastic NUM. There are four levels of stochastic dynamics that need to be brought back into the basic NUM formulations based on deterministic fluid models: session level (sessions come and go with finite workload), packet level (packets arrive in bursts and also go through stochastic marking and dropping), channel level (channel conditions vary), and topology level (network topology varies). Compared to session level stochastic issues such as stability, there has been much less work on packet level stochastic dynamics. Deb, Srikant, and Shakkottai [16] validated fluid approximation for packet level models in the many-user asymptote, and Chang and Liu [12] established the connection between HTTP layer utility and TCP layer utility. This paper investigates packet level stochastic dynamics from the viewpoint of noisy feedback where the "noise" is in part induced by packet level marking and dropping.

We note that, when dealing with the session level randomness where the number of flows changes, stochastic stability means that the number of users and the queue lengths on all the links remain finite [9], [25]. In contrast, in the presence of stochastic perturbations of network parameters, stochastic stability means that the proposed algorithms converge to the optimal solutions in some sense (e.g., almost sure), and this is the subject of our study here.

It is also worth noting that in Kelly's seminal work [21], [20], stochastic stability is examined using linear stochastic perturbation around the equilibrium point, which implicitly assumes the convergence of the distributed algorithms in the first place. Our findings on the rate of convergence provide another step toward understanding how quickly the large time-scale network parameters (e.g., the number of traffic flows) are allowed to fluctuate while the algorithm can still track the optimal point.

NUM with errors. This is an under-explored area with only limited treatment in very few recent publications. The case of deterministic errors in feedback has recently been studied by Mehyar, Spanos, and Low [28], using the methodologies on the gradient method with errors (e.g., [5]). The case of stochastic errors in distributed algorithms for single path congestion control has been investigated in [21], where convergence is assumed and the focus is on the rate of convergence. The multipath case has recently been investigated by Lin and Shroff [26], with focus on the models with unbiased errors on link load measurements for wired network congestion control. In [39], we have taken some preliminary steps to study the impact of stochastic feedback signals on a specific rate control algorithm, assuming that the gradient estimator is unbiased. The impact of noisy feedback on the primal-dual algorithm can be found in the conference version of this work [38].

D. Organization

The rest of the paper is organized as follows. Section II presents the general problem formulation. In Section III, we focus on the distributed NUM algorithm based on Lagrange dual decomposition, and develop a general theory on stochastic stability for distributed P-D algorithms in the presence of noisy feedback, for both the unbiased and biased cases. We also study the rate of convergence for the unbiased case. In Section IV, we apply the general theory summarized above to investigate stability of cross-layer rate control for joint congestion control and MAC design when strong duality holds. Our numerical examples corroborate the theoretic results well. Section V contains the study on stochastic two time-scale algorithms based on primal decomposition. Finally, Section VI concludes the paper.

II. Problem Formulation

A. Rate Control via Network Utility Utilization (Num)

Consider a communication network with L links, each with a (possibly time-varying) capacity of c_l, and S source nodes, each transmitting at a source rate of x_s. We assume that each source s emits one flow, using a fixed set of links $\mathcal{L}(s)$ in its path, and has a utility function $U_s(x_s)$. Let $\mathcal{S}$ denote the set of flows in the network, and $\mathcal{S}(l)$ the set of flows using link l. As is standard, we assume that the network is modeled as a directed graph $G=(N,E)$, where N is the set of the nodes and E is the set of the directed edges.

The general NUM problem is targeted at maximizing the total utility $\sum_s U_s(x_s)$, over the source rates $\mathbf{x}$, subject to flow rate constraints for all links l:

$$\boldsymbol{\Xi_1}: \quad \begin{array}{ll} \max_{\{0\le x_s\le M_s\}} & \sum_s U_s(x_s) \\ \text{subject to} & f_l(\mathbf{x}) \le c_l, \quad \forall l \end{array} \tag{1}$$

where the utilities U_s are twice-differentiable, increasing, and strictly concave functions, $\{f_l\}$ are twice-differentiable convex functions, M_s is an upper bound on the flow rate, and c_l is the ergodic capacity of link l, which depends on the specific MAC protocols and channel conditions.

The above basic NUM has recently been extended to much richer constraint sets involving variables other than just the

source rates. In particular, in some network models (e.g., MAC with random access), link capacities c_l can be adapted and the constraints may not be convex.

Built on the above NUM framework, a large number of decentralized algorithms have been devised, with the following general form on primal variables $\mathbf{x}$ and dual variables $\boldsymbol{\lambda}$ (discrete-time version with time index t):

$$x_s(t+1) = F_s(x_s(t), q_s(t)); \quad \forall s \in \mathcal{S} \tag{2}$$
$$\lambda_l(t+1) = G_l(y_l(t), \lambda_l(t)), \quad \forall l \in \mathcal{L} \tag{3}$$

where F_s and G_l are some nonnegative functions, q_s is the overall end-to-end price along the path of flow s, and y_l is the aggregated rate at link l. As surveyed in, e.g., [27], [15], from a reverse engineering perspective, in the case of $\mathbf{x}$ being just the source rates, these algorithms can be viewed as distributed solutions to some optimization problems, and the general algorithm in (2) and (3) is often implemented through gradient or subgradient algorithms.

Roughly speaking, the existing distributed algorithms can be classified into the following three categories (cf. [27]): 1) **Primal-dual algorithm:** In the primal-dual algorithm, the source rates x_s and the shadow prices λ_l are updated on the same time scale. 2) **Primal algorithm:** The primal algorithm often refers to a first-order gradient-descent algorithm for source rate updating and a *static* link price algorithm. 3) **Dual Algorithm:** The dual algorithm is based on dual decomposition, and consists of a gradient-type algorithm for shadow price updating and a *static* source rate algorithm. In general, the primal-dual algorithm has a faster convergence rate.

B. Distributed Num Under Noisy Feedback Based on Lagrangian Dual Decomposition

It is clear that to implement the distributed algorithms (2) and (3), some communications among network elements (e.g., nodes and links), in the form of information feedback (message passing), are critical for computing the gradients or subgradients. Unfortunately, in practical systems the feedback is often obtained using error-prone measurement mechanisms and contains random errors. In light of this, we seek to obtain a clear understanding of the conditions under which the general algorithm (2) and (3), in the presence of noisy feedback, converges to the equilibrium point obtained by deterministic algorithm. In the following, we focus on the distributed algorithms based on Lagrangian dual decomposition. Similar studies can be carried out for algorithms devised using other decomposition methods, and we will elaborate further on this in Section V.

We now turn to the algorithms based on the Lagrangian dual decomposition. To this end, we first form the Lagrangian of (1):

$$L(\mathbf{x}, \boldsymbol{\lambda}) = \sum_s U_s(x_s) + \sum_l \lambda_l (c_l - f_l(\mathbf{x})) \tag{4}$$

where $\lambda_l \geq 0$ is the Lagrange multiplier (link price) associated with the linear flow constraint on link l. Then, the Lagrange dual function is

$$Q(\boldsymbol{\lambda}) = \max_{\mathbf{x} \geq 0} L(\mathbf{x}, \boldsymbol{\lambda}) \tag{5}$$

and the dual problem is given by

$$\mathbf{D}: \min_{\boldsymbol{\lambda} \geq \mathbf{0}} Q(\boldsymbol{\lambda}). \tag{6}$$

When strong duality holds, the primal problem can be equivalently solved by solving the dual problem. Let Φ be the set of $\boldsymbol{\lambda}$ that minimizes $Q(\boldsymbol{\lambda})$. We assume that strong duality holds and Φ is nonempty and compact (which holds under mild conditions [37]). In what follows, we shall focus on the impact of noise feedback on the primal-dual algorithms based on the Lagrangian dual method, which are summarized as follows.

- **Deterministic primal-dual algorithm:** In the primal-dual algorithm, the updates of source rates x_s and the link prices λ_l are executed at the same time scale:

$$x_s(n+1) = [x_s(n) + \epsilon_n L_{x_s}(\mathbf{x}(n), \boldsymbol{\lambda}(n))]_{\mathcal{D}}, \quad \forall s \tag{7}$$
$$\lambda_l(n+1) = [\lambda_l(n) - \epsilon_n L_{\lambda_l}(\mathbf{x}(n), \boldsymbol{\lambda}(n))]_0^{\infty}, \quad \forall l \tag{8}$$

  where $L_{x_s}(\cdot)$ is the gradient of L with respect to x_s and $L_{\lambda_l}(\cdot)$ is the gradient of L with respect to λ_l, and $[\cdot]_{\mathcal{D}}$ stands for the projection onto the feasible set $\mathcal{D}$.
- **Stochastic primal-dual algorithm:** In the presence of noisy feedback, the gradients $L_{x_s}(\cdot)$ and $L_{\lambda_l}(\cdot)$ are stochastic. Let $\hat{L}_{x_s}(\cdot)$ and $\hat{L}_{\lambda_l}(\cdot)$ be the corresponding estimators, and the stochastic version of the primal-dual algorithm can be written as follows:

$$x_s(n+1) = [x_s(n) + \epsilon_n \hat{L}_{x_s}(\mathbf{x}(n), \boldsymbol{\lambda}(n))]_{\mathcal{D}}, \quad \forall s \tag{9}$$
$$\lambda_l(n+1) = [\lambda_l(n) - \epsilon_n \hat{L}_{\lambda_l}(\mathbf{x}(n), \boldsymbol{\lambda}(n))]_0^{\infty}, \quad \forall l \tag{10}$$

For the sake of convenience, we have used the same step size ϵ_n for both source rate and link price updating. We emphasize that this assumption does not incur any loss of generality. If the step sizes are different, the shadow prices and the utility functions can be rescaled with no effect on the stability.

III. Stochastic Stability of Primal-Dual Algorithm Under Noisy Feedback

As noted above, a main objective of this study is to examine the impact of noisy feedback on the convergence behavior of the distributed algorithms, which boils down to the stochastic stability of network dynamics. There are many notions of stochastic stability. In the presence of stochastic perturbations of network parameters, stochastic stability is used to indicate that the algorithms converge to the optimal solutions in some sense; and this is the subject of our study here. In what follows, we focus on the stability behavior of stochastic P-D algorithm, which is a single time-scale algorithm.

A. Structure of Stochastic Gradients

The structure of the stochastic gradients plays a critical role in the stability behavior of the distributed NUM algorithm. For convenience, let $\{\mathcal{F}_n\}$ be a sequence of σ-algebras generated by $\{(\mathbf{x}(i), \boldsymbol{\lambda}(i)), \forall\, i \leq n\}$, $E_n[\cdot]$ denote the conditional expectation $E[\cdot \mid \mathcal{F}_n]$. We have the following observations on $\hat{L}_{x_s}(\cdot, \cdot)$ and $\hat{L}_{\lambda_l}(\cdot, \cdot)$.

1) ***Stochastic gradient*** $\hat{L}_{x_s}(\cdot,\cdot)$: Observe that

$$\hat{L}_{x_s}(\mathbf{x}(n),\boldsymbol{\lambda}(n)) = L_{x_s}(\mathbf{x}(n),\boldsymbol{\lambda}(n)) + \alpha_s(n) + \zeta_s(n)$$

where $\alpha_s(n)$ is the biased estimation error of $L_{x_s}(\mathbf{x}(n),\boldsymbol{\lambda}(n))$, given by

$$\alpha_s(n) \triangleq E_n[\hat{L}_{x_s}(\mathbf{x}(n),\boldsymbol{\lambda}(n))] - L_{x_s}(\mathbf{x}(n),\boldsymbol{\lambda}(n)) \quad (11)$$

and $\zeta_s(n)$ is a martingale difference noise, given by

$$\zeta_s(n) \triangleq \hat{L}_{x_s}(\mathbf{x}(n),\boldsymbol{\lambda}(n)) - E_n\left[\hat{L}_{x_s}(\mathbf{x}(n),\boldsymbol{\lambda}(n))\right]. \quad (12)$$

2) ***Stochastic gradient*** $\hat{L}_{\lambda_l}(\cdot,\cdot)$: Observe that

$$\hat{L}_{\lambda_l}(\mathbf{x}(n),\boldsymbol{\lambda}(n)) = L_{\lambda_l}(\mathbf{x}(n),\boldsymbol{\lambda}(n))) + \beta_l(n) + \xi_l(n)$$

where $\beta_l(n)$ is the biased estimation error of $L_{\lambda_l}(\mathbf{x}(n),\boldsymbol{\lambda}(n))$, given by

$$\beta_l(n) \triangleq E_n[\hat{L}_{\lambda_l}(\mathbf{x}(n),\boldsymbol{\lambda}(n))] - L_{\lambda_l}(\mathbf{x}(n),\boldsymbol{\lambda}(n)) \quad (13)$$

and $\xi_l(n)$ is a martingale difference noise, given by

$$\xi_l(n) \triangleq \hat{L}_{\lambda_l}(\mathbf{x}(n),\boldsymbol{\lambda}(n)) - E_n[\hat{L}_{\lambda_l}(\mathbf{x}(n),\boldsymbol{\lambda}(n))]. \quad (14)$$

In practical systems, a popular mechanism for conveying the price information to the source is by packet marking. In particular, there are two standard marking schemes, random exponential marking (REM) [3] and self-normalized additive marking (SAM) [1]; REM is a biased estimator of the overall price along a route, whereas SAM is an unbiased one. Accordingly, $\hat{L}_{x_s}(\cdot,\cdot)$ can be either biased or unbiased. On the other hand, $\hat{L}_{\lambda_l}(\cdot,\cdot)$ can be either unbiased or biased when the aggregated source rate is estimated (e.g., in wireless networks with random access). In Section IV, we provide more examples to illustrate the structure of the stochastic gradients.

B. The Unbiased Case

In this section, we show that the proposed stochastic primal-dual algorithm converges with probability one to the optimal points, provided that the gradient estimator is asymptotically unbiased. To this end, we impose the following technical conditions:

A1. We assume that the noise terms in the gradient estimators are independent across iterations.

A2. Condition on the step size: $\epsilon_n > 0$, $\epsilon_n \to 0$, $\sum_n \epsilon_n \to \infty$ and $\sum_n \epsilon_n^2 < \infty$.

A3. Condition on the biased error: $\sum_n \epsilon_n|\alpha_s(n)| < \infty\ w.p.1, \forall\ s$ and $\sum_n \epsilon_n|\beta_l(n)| < \infty\ w.p.1, \forall\ l$.

A4. Condition on the martingale difference noise: $\sup_n E_n[\zeta_s(n)^2] < \infty\ w.p.1, \forall\ s$, and $\sup_n E_n[\xi_l(n)^2] < \infty\ w.p.1, \forall\ l$.

We now present out first main result on the convergence of the stochastic primal-dual algorithm in the presence of unbiased noisy feedback information. The complete proof can be found in Appendix VI-A.

Theorem 3.1: Under Conditions $\mathbf{A1}-\mathbf{A4}$, the iterates $\{(\mathbf{x}(n),\boldsymbol{\lambda}(n)), n = 1,2,\ldots\}$, generated by stochastic algorithms (9) and (10), converge with probability one to the optimal solution of the primal Problem $\boldsymbol{\Xi_1}$.

Sketch of the proof: The proof consists of two steps. First, define a stochastic Lyapunov function as the square of the distance between the iterates $(\mathbf{x},\boldsymbol{\lambda})$ and the set of the saddle points. Via a combination of tools in the Martingale theory and convex analysis, we establish that the iterates generated by (9) and (10) return to a neighborhood (denoted as A_μ) of the optimal points infinitely often. Then, we show that the recurrent iterates eventually reside in an arbitrary small neighborhood of the optimal points, and the proof involves some tedious local analysis.

Remarks: 1) Clearly, the first key step in the stability analysis is to construct the Lyapunov function. Different from the techniques in [21], [22, Ch. 5], the Lagrangian function in the primal-dual algorithm is neither convex nor concave in $(\mathbf{x},\boldsymbol{\lambda})$ because the primal problem is a maximization over $\mathbf{x}$ but the dual problem is a minimization over $\boldsymbol{\lambda}$. As a result, the P-D algorithm is much more involved than the standard gradient algorithm [22]. Observing that the ultimate goal is to show that the iterates approach the optimal solutions eventually, we set the distance between the iterates and the set of the optimal saddle points as the Lyapunov function. We elaborate further on this in Appendix A.

2) Condition $\mathbf{A2}$ is a standard technical condition in stochastic approximation for proving convergence with probability one (w.p. 1), and Condition $\mathbf{A3}$ essentially requires that the biased term is asymptomatically diminishing. However, if the step size ϵ_n does *not* go to zero (which occurs often in on-line applications) or the biased terms $\{\alpha_s(n),\beta_l(n)\}$ do not diminish (which may be the case in some practical systems), we cannot expect to get probability one convergence to the optimal point. To be quantified in the next subsection, the hope is that the iterates would converge to some neighborhood of the optimal points. It should be cautioned that even the expectation of the limiting process would not be the optimal point if the biased terms do not go to zero.

C. The Biased Case

As noted above, when the gradient estimators are biased, we cannot hope to obtain almost sure convergence of the stochastic primal-dual algorithm. Nevertheless, if the biased terms are asymptotically bounded, outside the contraction region (to be defined below), by a scaled version $(0 \le \eta < 1)$ of the true gradients, we show that the iterates still move toward the optimal point. For technical reasons, we impose the following condition on the biased terms:

A5. Condition on the biased error: There exist nonnegative constants $\{\alpha_s^u\}$ and $\{\beta_l^u\}$ such that $\limsup_n |\alpha_s(n)| \le \alpha_s^u,\ \forall\ s$, and $\limsup_n |\beta_l(n)| \le \beta_l^u,\ \forall\ l$, with probability one.

Define the "contraction region" A_η as follows:

$$A_\eta \triangleq \{(\mathbf{x},\boldsymbol{\lambda}) : \alpha_s^u \ge \eta|L_{x_s}(\mathbf{x},\boldsymbol{\lambda})|, \text{for some } s \text{ or } \beta_l^u \ge \eta|L_{\lambda_l}(\mathbf{x},\boldsymbol{\lambda})|, \text{for some } l, 0 \le \eta < 1\}. \quad (15)$$

It is not difficult to see that A_η defines a closed and bounded neighborhood around the equilibrium point, simply because at the equilibrium point $L_{x_s}(\mathbf{x}^*,\boldsymbol{\lambda}^*) = 0$ for all $s \in \mathcal{S}$ and $L_{\lambda_l}(\mathbf{x}^*,\boldsymbol{\lambda}^*) = 0$ for all $l \in \mathcal{L}$, and both $L_{x_s}(\mathbf{x},\boldsymbol{\lambda})$ and $L_{\lambda_l}(\mathbf{x},\boldsymbol{\lambda})$ are continuous. The size of A_η depends on the values of η and $\{\alpha_s^u, \beta_l^u\}$. The larger the value of η is, the smaller the size of A_η would be. In contrast, the larger the values of $\{\alpha_s^u, \beta_l^u\}$ are, the larger the size of A_η would be.

Next, we present our second main result on the stochastic stability of the primal-dual algorithm in the presence of biased noisy feedback information.

Theorem 3.2: Under Conditions **A1**, **A2**, **A4**, and **A5**, the iterates $\{(\mathbf{x}(n),\boldsymbol{\lambda}(n), n = 1,2,\ldots\}$, generated by stochastic algorithms (9) and (10), return to the set A_η infinitely often with probability one.

The proof is relegated to Appendix B. Note that condition **A5** is weaker than **A3**, and in this sense, Theorem 3.2 is a generalization of Theorem 3.1.

To get a more concrete sense of the regularity conditions, we first observe that outside the set A_η, i.e., in A_η^c,

$$\alpha_s^u \le \eta |L_{x_s}(\mathbf{x},\boldsymbol{\lambda})|, \quad \forall\, s$$

and

$$\beta_l^u \le \eta |L_{\lambda_l}(\mathbf{x},\boldsymbol{\lambda})|, \quad \forall\, l.$$

These, combined with Condition A5, indicate that the biased terms are asymptotically uniformly bounded by a scaled version of the true gradient outside A_η. As a result, the errors are relatively small and would not negate the true gradient. Therefore, the inexact gradient can still drive the iterate to move toward the optimal points until it enters A_η.

D. Rate of Convergence for the Unbiased Case

We now examine the rate of convergence when the stochastic gradients are unbiased. Roughly speaking, the rate of convergence is concerned with the asymptotic behavior of normalized distance of the iterates from the optimal points. Recall that the primal-dual algorithm takes the following general form:

$$\begin{bmatrix} x_s(n+1) \\ \lambda_l(n+1) \end{bmatrix} = \begin{bmatrix} x_s(n) \\ \lambda_l(n) \end{bmatrix} + \epsilon_n \begin{bmatrix} L_{x_s}(\mathbf{x}(n),\boldsymbol{\lambda}(n)) \\ -L_{\lambda_l}(\mathbf{x}(n),\boldsymbol{\lambda}(n)) \end{bmatrix} + \epsilon_n \begin{bmatrix} \alpha_s(n)+\zeta_s(n) \\ \beta_l(n)+\xi_l(n) \end{bmatrix} + \epsilon_n \begin{bmatrix} Z_n^{x_s} \\ Z_n^{\lambda_l} \end{bmatrix} \quad (16)$$

where $\epsilon_n Z_n^{x_s}$ and $\epsilon_n Z_n^{\lambda_l}$ are the reflection terms which "force" x_s and λ_l to reside inside the constraint sets. As is standard in the study on rate of convergence, we focus on local analysis, assuming the iterates generated by the stochastic primal-dual algorithm have entered a small neighborhood of $(\mathbf{x}^*,\boldsymbol{\lambda}^*)$.

To characterize the asymptotic properties, we define $U_{\mathbf{x}}(n) \triangleq (\mathbf{x}(n) - \mathbf{x}^*)/\sqrt{\epsilon_n}$ and $U_{\boldsymbol{\lambda}}(n) \triangleq (\boldsymbol{\lambda}(n) - \boldsymbol{\lambda}^*)/\sqrt{\epsilon_n}$, and we construct $U^n(t)$ to be the piecewise constant interpolation of $U(n) = \{U_{\mathbf{x}}(n), U_{\boldsymbol{\lambda}}(n)\}$, i.e., $U^n(t) = U(n+i)$, for $t \in [t_{n+i} - t_n, t_{n+i+1} - t_n)$, where $t_n \triangleq \sum_{i=0}^{n-1} \epsilon_n$. **A6**. Let $\boldsymbol{\theta}(n) \triangleq (\mathbf{x}(n),\boldsymbol{\lambda}(n))$ and $\boldsymbol{\phi}_n \triangleq (\boldsymbol{\zeta}(n),\boldsymbol{\xi}(n))$. Suppose for any given small $\rho > 0$, there exists a positive definite symmetric matrix $\boldsymbol{\Sigma} = \boldsymbol{\sigma}\boldsymbol{\sigma}'$ such that, as $n \to \infty$,

$$E_n\left[\boldsymbol{\phi}_n\boldsymbol{\phi}_n^T - \boldsymbol{\Sigma}\right] I_{\{|\boldsymbol{\theta}(n)-\boldsymbol{\theta}^*|\le\rho\}} \to 0.$$

A7. Let $\epsilon_n = 1/n$; and assume that $\mathbf{H} + \mathbf{I}/2$ is a Hurwitz matrix, where $\mathbf{H}$ is the Hessian matrix of the Lagrangian function at $(\mathbf{x}^*,\boldsymbol{\lambda}^*)$.[1]

We have the following proposition on the rate of convergence for the unbiased case.

Proposition 3.1: a) Under Conditions **A1**, **A3**, **A4**, **A6** and **A7**, $U^n(\cdot)$ converges in distribution to the solution (denoted as U) to the Skorohod problem

$$\begin{pmatrix} dU_{\mathbf{x}} \\ dU_{\boldsymbol{\lambda}} \end{pmatrix} = \left(\mathbf{H} + \frac{\mathbf{I}}{2}\right) \begin{pmatrix} U_{\mathbf{x}} \\ U_{\boldsymbol{\lambda}} \end{pmatrix} dt + \boldsymbol{\sigma} dw(t) + \begin{pmatrix} dZ_{\mathbf{x}} \\ dZ_{\boldsymbol{\lambda}} \end{pmatrix} \quad (17)$$

where $w(t)$ is a standard Wiener process and $Z(\cdot)$ is the reflection term.

b) If $(\mathbf{x}^*,\boldsymbol{\lambda}^*)$ is an interior point of the constraint set, then the limiting process U is a stationary Gaussian diffusion process, and $U(n)$ converges in distribution to a normally distributed random variable with mean zero and covariance $\boldsymbol{\Sigma}$.

c) If $(\mathbf{x}^*,\boldsymbol{\lambda}^*)$ is on the boundary of the constraint set, then the limiting process U is a stationary reflected linear diffusion process.

Sketch of the proof: Proposition 3.1 can be proved by combining [11, Proof of Th. 5.1] and that in [22, Ch. 6, Th. 2.1]. Roughly, we can expand, via a truncated Taylor series, the interpolated process $U^n(t)$ around the chosen saddle point $(\mathbf{x}^*,\boldsymbol{\lambda}^*)$. Then, a key step needed is to show the tightness of $U^n(t)$. To this end, we can follow [22, Ch. 6, Proof of Th. 2.1, Part 3] to establish that the biased term in the interpolated process diminishes asymptotically and does not affect the tightness of $U^n(t)$. More specifically, after showing the tightness of $U^n(t)$, one can work with a truncated version of $U^n(t)$ to establish the weak convergence of the truncated processes, and then use the properties of these weak sense limits to show that the truncation does not impact the conclusion. The rest follows from [11, Proof of Theorems 4.3, 4.4, and 5.1].

Remarks: Proposition 3.1 reveals that for the constrained case, the limiting process of the interpolated process for the normalized iterates depends on the specific structure of the Skorohod problem [22]. In general, the limit process is a stationary reflected linear diffusion process, not necessarily a standard Gaussian diffusion process. The limit process would be Gaussian only if there is no reflection term.

Based on Proposition 3.1, the rate of convergence depends heavily on the smallest eigenvalue of $(\mathbf{H} + \frac{\mathbf{I}}{2})$. The more negative the smallest eigenvalue is, the faster the rate of convergence is. Intuitively speaking, the reflection terms help increase the speed of convergence, which unfortunately cannot be characterized exactly. As noted in [11], one cannot readily compute the stationary covariance matrix for reflected diffusion processes,

[1]It can be shown that the real parts of the eigenvalues of $\mathbf{H}$ are all nonpositive (cf. S[5, p. 449]).

and we have to resort to simulations to get some understanding of the effect of the constraints on the asymptotic variances. Furthermore, the covariance matrix of the limit process gives a measure of the spread at the equilibrium point, which is typically smaller than that of the unconstrained case [11].

Our findings on the rate of convergence provide another step toward understanding how rapidly the slower timescale network parameters (e.g., the number of traffic flows) are allowed to change before the algorithm converges to the optimal point.

IV. Application to Joint Congestion Control and Random Access

In this section, we apply the general theory developed above to investigate the stability of cross-layer rate control for joint congestion control and MAC design, in the presence of noisy feedback. Specifically, we consider rate control in multihop wireless networks with random access, and take into account the rate constraints imposed by the MAC/PHY layer. Since random access in wireless networks is node-based, we abuse the notation and denote a link as a pair of nodes (i,j), where i is the transmitter of the link and j is the receiver. For convenience, let $N_{to}^I((i,j))$ denote the set of nodes whose transmissions interfere with link (i,j)'s, excluding node i. Let $N_{in}(i)$ denote the set of the nodes from which node i receives traffic, $N_{\text{out}}(i)$ the set of nodes to which node i is sending packets and $L_{\text{from}}^I(i)$ to denote the set of links whose transmissions are interfered by node i's transmission, excluding the outgoing links from node i.

We assume that the persistence transmission mechanism is used at the MAC layer, i.e., node i of link (i,j) contends the channel with a persistence probability $p_{(i,j)}$. Define $P_i = \sum_{j\in N_{\text{out}}(i)} p_{(i,j)}$. It is known that successful transmission probability of link (i,j) is given by $p_{(i,j)}\prod_{k\in N_{to}^I((i,j))}(1-P_k)$ [6]. In light of this, we impose the constraint that the total flow rate over link (i,j) should be no more than $r_{(i,j)}p_{(i,j)}\prod_{k\in N_{to}^I((i,j))}(1-P_k)$, where $r_{(i,j)}$ is the average link rate. For simplicity, in the following, we assume that $\{r_{(i,j)}\}$ is normalized to one.

The cross-layer rate optimization across the transport layer and the MAC layer can be put together as shown in (18) at the bottom of the page, where M_s is the maximum for flow data rate of s, and the utility function $U_s(\cdot)$ is assumed to be [29]:

$$U_\kappa(x_s) = \begin{cases} w_s \log x_s, & \text{if } \kappa = 1 \\ w_s(1-\kappa)^{-1}x_s^{1-\kappa}, & \kappa \geq 0, \kappa \neq 1. \end{cases} \tag{19}$$

The above problem is nonconvex since the link constraints involves the product term of $\{p_{i,j}\}$. By using a change of variables $\tilde{x}_s = \log(x_s)$ (cf. [35], [24]), however, it can be transformed into the following convex programming problem, provided that $\kappa \geq 1$ (cf. (19)), please see (20) shown at the bottom of the page, where $U_s'(\tilde{x}_s) = U_s(\exp(\tilde{x}_s))$ and $\tilde{M}_s = \log(M_s)$. More specifically, it is easy to show that $U_s'(\tilde{x}_s)$ is strictly concave in $\tilde{x}_s$ for utility functions in (19) with $\kappa \geq 1$ [24]. Furthermore, since both terms $\log(\sum_{s\in\mathcal{S}((i,j))}\exp(\tilde{x}_s))$ and $-\log(p_{(i,j)}) - \sum_{k\in N_{to}^I((i,j))}\log(1-P_k)$ are convex functions of $\tilde{x}_s$ and $p_{(i,j)}$, it follows that the above problem is strictly convex with a unique optimal point $\mathbf{x}^*$.

Worth noting is that when $0 \leq \kappa < 1$, $U_s'(\tilde{x}_s)$ becomes a convex function. Then the problem boils down to finding the maxima of a convex function over a convex constraint set (no longer a convex program), indicating that the maxima lies on the boundary of the constraint set.

A. Cross-Layer Rate Control: The $\kappa \geq 1$ Case

In this section, we assume that $\kappa \geq 1$. The Lagrangian function with the Lagrange multipliers $\{\lambda_{(i,j)}\}$ is given as follows:

$$\begin{aligned} &L(\tilde{\mathbf{x}}, p, \boldsymbol{\lambda}) \\ &\quad = \left\{ \sum_s U_s'(\tilde{x}_s) - \sum_{(i,j)} \lambda_{(i,j)} \log \right. \\ &\qquad \left. \times \left(\sum_{s\in\mathcal{S}((i,j))} \exp(\tilde{x}_s) \right) \right\} \\ &\qquad + \sum_{(i,j)} \lambda_{(i,j)} \log \left(p_{(i,j)} \prod_{k\in N_{to}^I((i,j))} (1-P_k) \right). \end{aligned} \tag{21}$$

$$\begin{aligned} \mathbf{P}: \quad \max_{\{x_s, p_{(i,j)}\}} \quad & \textstyle\sum_s U_s(x_s) \\ \text{subject to} \quad & \textstyle\sum_{s\in\mathcal{S}((i,j))} x_s \leq [p_{(i,j)} \prod_{k\in N_{to}^I((i,j))}(1-P_k)], \quad \forall\,(i,j) \\ & \textstyle\sum_{j\in N_{\text{out}}(i)} p_{(i,j)} = P_i, \quad \forall\, i \\ & 0 \leq x_s \leq M_s, \quad \forall\, s \\ & 0 \leq P_i \leq 1, \quad \forall\, i \end{aligned} \tag{18}$$

$$\begin{aligned} \mathbf{P_1}: \quad \max_{\{\tilde{x}_s, p_{(i,j)}\}} \quad & \textstyle\sum_{s\in\mathcal{S}} U_s'(\tilde{x}_s) \\ \text{subject to} \quad & \textstyle\log(\sum_{s\in\mathcal{S}((i,j))}\exp(\tilde{x}_s)) - \log(p_{(i,j)}) \\ & \textstyle -\sum_{k\in N_{to}^I((i,j))}\log(1-P_k) \leq 0, \quad \forall\,(i,j) \\ & \textstyle\sum_{j\in N_{\text{out}}(i)} p_{(i,j)} = P_i, \quad \forall\, i \\ & -\infty \leq \tilde{x}_s \leq \tilde{M}_s, \quad \forall\, s \\ & 0 \leq P_i \leq 1, \quad \forall\, i \end{aligned} \tag{20}$$

Then, the Lagrange dual function is

$$Q(\boldsymbol{\lambda}) = \max_{\substack{\sum_{j\in N_{\text{out}}(i)} p_{(i,j)}=P_i \\ \mathbf{0}\le\mathbf{P}\le\mathbf{1} \\ -\boldsymbol{\infty}\le\tilde{\mathbf{x}}\le\tilde{\mathbf{M}}}} L(\tilde{\mathbf{x}}, p, \boldsymbol{\lambda}) \tag{22}$$

where $\tilde{\mathbf{M}}$ is a vector of $\tilde{M}_s, \forall\, s \in \mathcal{S}$. Thus, the dual problem is given by

$$\mathbf{D}: \min_{\boldsymbol{\lambda}\ge\mathbf{0}} Q(\boldsymbol{\lambda}). \tag{23}$$

It is not difficult to see that the Slater condition is satisfied, indicating that there is no duality gap and the strong duality holds. Furthermore, it can be shown that Φ is nonempty and compact [37]. Summarizing, the primal problem can be equivalently solved by solving the dual problem. To this end, we rewrite $Q(\boldsymbol{\lambda})$ as shown in the equation at the bottom of the page which essentially "decomposes" the original utility optimization into two subproblems, i.e., maximizing $O_{\mathbf{x}}(\boldsymbol{\lambda})$ by flow control and maximizing $O_{\mathbf{p}}(\boldsymbol{\lambda})$ via MAC layer random access, which are loosely coupled by the shadow price $\boldsymbol{\lambda}$. The MAC layer problem can then be solved by [23]

$$p_{(i,j)} = \frac{\lambda_{(i,j)}}{\sum_{k\in N_{\text{out}}(i)} \lambda_{(i,k)} + \sum_{l:I\in L^I_{\text{from}}(i)} \lambda_l}. \tag{24}$$

The flow control subproblem can be readily solved by the following gradient method:

$$\tilde{x}_s(n+1) = [\tilde{x}_s(n) + \epsilon_n L_{\tilde{x}_s}(\tilde{\mathbf{x}}(n), \mathbf{p}(n), \boldsymbol{\lambda}(n))]^{\tilde{M}_s}_{-\infty} \tag{25}$$

where ϵ_n is the step size, $[x]^a_b$ stands for $\max(b, \min(a, x))$, and

$$L_{\tilde{x}_s}(\cdot,\cdot,\cdot) \triangleq \dot{U}'_s(\tilde{x}_s(n)) - \exp(\tilde{x}_s(n)) \times \sum_{(i,j)\in\mathcal{L}(s)} \frac{\lambda_{(i,j)(n)}}{\sum_{k\in\mathcal{S}((i,j))} \exp(\tilde{x}_k(n))} \tag{26}$$

where $\dot{U}'_s(\tilde{x}_s(n))$ is the first order derivative of $U'_s(\tilde{x}_s(n))$.

Next, by using the subgradient method, the shadow price updates are given by

$$\lambda_{(i,j)}(n+1) = [\lambda_{(i,j)}(n) - \epsilon_n L_{\lambda_{(i,j)}}(\tilde{\mathbf{x}}(n), \mathbf{p}(n), \boldsymbol{\lambda}(n))]^{\infty}_0 \tag{27}$$

where

$$L_{\lambda_{(i,j)}}(\cdot,\cdot,\cdot) \triangleq \log(p_{(i,j)}(n)) + \sum_{k\in N^I_{to}((i,j))} \log(1 - P_k(n)) - \log\left(\sum_{s\in\mathcal{S}((i,j))} \exp(\tilde{x}_s(n))\right). \tag{28}$$

As noted in Section III, message passing is required to feed back the price information for distributed implementation. In particular, the parameters $\frac{\lambda_{(i,j)}}{\sum_{s\in\mathcal{S}((i,j))} x_s}$ need to be generated at each link in $\mathcal{L}(s)$, and be fed back to the source node along the routing path. It can be seen that the calculation of the ratio $\frac{\lambda_{(i,j)}}{\sum_{s\in\mathcal{S}((i,j))} x_s}$ at node i requires the local information of the shadow price $\lambda_{(i,j)}$ and the total incoming traffic $\sum_{s\in\mathcal{S}((i,j))} x_s$. Similarly, in (27), in order to update the shadow price $\lambda_{(i,j)}$, it suffices to have the local information of the total incoming traffic and the two-hop information of $\{P_k, k \in N^I_{to}(j)\}$. Nevertheless, the feedback is noisy and contains possibly random errors. Summarizing, the stochastic primal-dual algorithm for cross-layer rate control, in the presence of noisy feedback, is given as follows:

Stochastic primal-dual algorithm for cross-layer rate control (in what follows, SA stands for stochastic approximation):

- *SA algorithm for source rate updating*:

$$\tilde{x}_s(n+1) = [\tilde{x}_s(n) + \epsilon_n \hat{L}_{\tilde{x}_s}(\tilde{\mathbf{x}}(n), \mathbf{p}(n), \boldsymbol{\lambda}(n))]^{\tilde{M}_s} \tag{29}$$

  where $\hat{L}_{\tilde{x}_s}(\cdot)$ is an estimator of $L_{\tilde{x}_s}(\cdot)$, i.e., the gradient of L with respect to $\tilde{x}_s$.
- *SA algorithm for shadow price updating*:

$$\lambda_{(i,j)}(n+1) = \left[\lambda_{(i,j)}(n) - \epsilon_n \hat{L}_{\lambda_{(i,j)}}(\tilde{\mathbf{x}}(n), \mathbf{p}(n), \boldsymbol{\lambda}(n))\right]^+ \tag{30}$$

  where $\hat{L}_{\lambda_{(i,j)}}(\cdot)$ is an estimator of $L_{\lambda_{(i,j)}}(\cdot)$, i.e., the gradient of L with respect to $\lambda_{(i,j)}$.
- *The persistence probabilities are updated by using (24).*

Based on Theorems 3.1 and 3.2, we have the following result on the stochastic stability of the above algorithm.

Corollary 4.1: (a)**(The unbiased case)** Under Conditions $\mathbf{A1}-\mathbf{A4}$, the iterates $\{(\tilde{\mathbf{x}}(n), \mathbf{P}(n), \boldsymbol{\lambda}(n)), n = 1, 2, \ldots\}$, generated by stochastic approximation algorithms (29), (30)

$$Q(\boldsymbol{\lambda}) = \max_{\tilde{\mathbf{m}}\le\tilde{\mathbf{x}}\le\tilde{\mathbf{M}}} \underbrace{\left\{\sum_s U'_s(\tilde{x}_s) - \sum_{(i,j)} \lambda_{(i,j)} \log\left(\sum_{s\in\mathcal{S}((i,j))} \exp(\tilde{x}_s)\right)\right\}}_{\triangleq O_{\mathbf{x}}(\boldsymbol{\lambda})} + \max_{\substack{\sum_{j\in N_{\text{out}}(i)} p_{(i,j)}=P_i \\ 0\le P_i\le 1,\ \forall\, i}} \underbrace{\sum_{(i,j)} \lambda_{(i,j)} \log(p_{(i,j)} \prod_{k\in N^I_{to}((i,j))} (1-P_k))}_{\triangleq O_{\mathbf{p}}(\boldsymbol{\lambda})}$$

and (24), converge with probability one to the equilibrium point.

(b) (**The biased case**) Under Conditions $\mathbf{A1}-\mathbf{A2}$, $\mathbf{A4}$ and $\mathbf{A5}$, the iterates $\{(\tilde{\mathbf{x}}(n), \mathbf{P}(n), \boldsymbol{\lambda}(n), n = 1, 2, \ldots\}$, generated by stochastic approximation algorithms (29), (30) and (24), return to the set A_η infinitely often with probability one.

The proof of part (a) follows essentially along the same line as that of Theorem 3.1, except that $G(\cdot)$ depends on $\tilde{\mathbf{x}}(n), \mathbf{p}(n)$ and $\boldsymbol{\lambda}(n)$. More specifically, to show that $G(\cdot) < 0$ for $(\tilde{\mathbf{x}}(n), \boldsymbol{\lambda}(n)) \in A_\mu^c$, we have that

$$\begin{aligned} & G(\tilde{\mathbf{x}}(n), \boldsymbol{\lambda}(n)) \\ & \quad \le L(\tilde{\mathbf{x}}(n), \mathbf{p}(n), \boldsymbol{\lambda}^*_{\min}) - L(\tilde{\mathbf{x}}^*, \mathbf{p}(n), \boldsymbol{\lambda}(n)) \\ & \quad = L(\tilde{\mathbf{x}}(n), \mathbf{p}(n), \boldsymbol{\lambda}^*_{\min}) - L(\tilde{\mathbf{x}}^*, \mathbf{p}^*, \boldsymbol{\lambda}^*_{\min}) \qquad (31) \\ & \qquad + L(\tilde{\mathbf{x}}^*, \mathbf{p}^*, \boldsymbol{\lambda}^*_{\min}) - L(\tilde{\mathbf{x}}^*, \mathbf{p}^*, \boldsymbol{\lambda}(n)) \qquad (32) \\ & \qquad + L(\tilde{\mathbf{x}}^*, \mathbf{p}^*, \boldsymbol{\lambda}(n)) - L(\tilde{\mathbf{x}}^*, \mathbf{p}(n), \boldsymbol{\lambda}(n)). \qquad (33) \end{aligned}$$

Observing that

1) (31) and (32) are negative due to the saddle point property of $(\tilde{\mathbf{x}}^*, \mathbf{p}^*, \boldsymbol{\lambda}^*)$, i.e., $L(\tilde{\mathbf{x}}(n), \mathbf{p}(n), \boldsymbol{\lambda}^*) \le L(\tilde{\mathbf{x}}^*, \mathbf{p}^*, \boldsymbol{\lambda}^*) \le L(\tilde{\mathbf{x}}^*, \mathbf{p}^*, \boldsymbol{\lambda}(n))$;
2) (33) is negative because when $(\tilde{\mathbf{x}}^*, \boldsymbol{\lambda}(n))$ is fixed, $\mathbf{p}(n)$ is the unique maximum for $L(\cdot, \cdot, \cdot)$;

we conclude that there exists $\delta_\mu > 0$ such that $G(\{\tilde{\mathbf{x}}(n), \boldsymbol{\lambda}(n)\}) < -\delta_\mu$ when $(\tilde{\mathbf{x}}(n), \boldsymbol{\lambda}(n)) \in A_\mu^c$.

The proof of part (b) is a direct application of Theorem 3.2.

Example 1: **Structure of** $\hat{L}_{\tilde{x}_s}(\tilde{\mathbf{x}}(n), p(n), \lambda(n))$ under random exponential marking (REM).

To get a more concrete sense of Condition $\mathbf{A3}$ on the biased terms and Condition $\mathbf{A4}$ on the martingale noise terms, we use the following example to elaborate further on this when random exponential marking is used.

Suppose that the exponential marking technique is used to feedback the price information to the source nodes. More specifically, every link (i, j) marks a packet independently with probability $1 - \exp(\frac{-\lambda_{(i,j)}}{\sum_{s\in\mathcal{S}(i,j)} \exp(\tilde{x}_s)})$. Therefore, the end-to-end nonmarking probability for flow s is given as follows:

$$q = \exp\left(\sum_{(i,j)\in\mathcal{L}(s)} \frac{-\lambda_{(i,j)}}{\sum_{s\in\mathcal{S}(i,j)} \exp(\tilde{x}_s)}\right)$$

To obtain the estimate of the overall price, source s sends N_n packets during round n and counts the nonmarked packets. For example, if K packets have been counted, then the estimation of the overall price can be $\log(\hat{q})$ where $\hat{q} = K/N_n$. Therefore,

$$\hat{L}_{\tilde{x}_s}(\tilde{\mathbf{x}}(n), \mathbf{p}(n), \boldsymbol{\lambda}(n)) = \dot{U}'_s(\tilde{x}_s(n)) + \exp(\tilde{x}_s(n)) \log(\hat{q}). \qquad (34)$$

By the definition of (11), we have that

$$\begin{aligned} \alpha_n &= E_n[\hat{L}_{\tilde{x}_s}(\tilde{\mathbf{x}}(n), \mathbf{p}(n), \boldsymbol{\lambda}(n))] \\ &\quad - L_{\tilde{x}_s}(\tilde{\mathbf{x}}(n), \mathbf{p}(n), \boldsymbol{\lambda}(n)), \\ &= \exp(\tilde{x}_s(n))(E_n[\log(\hat{q})] - \log(q)) \qquad (35) \end{aligned}$$

Using Jensen's inequality, it is clear that $\alpha_s(n) \ge 0$ for any s and n. Next, we find an upper bound for $\alpha_s(n)$. Note that K is a Binomial random variable with distribution $B(N_n, q)$. When

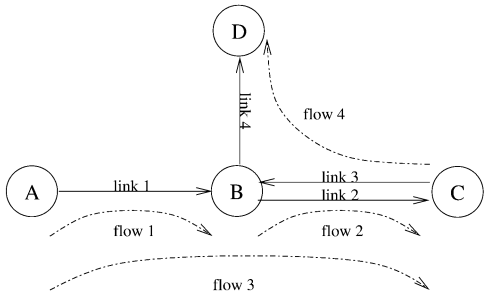

Fig. 1. A simple network model.

N_n is sufficiently large, it follows that $\hat{q} \sim \mathcal{N}(q, q(1-q)/N_n)$ and $\hat{q} \in [q - c/\sqrt{N_n}, q + c/\sqrt{N_n}]$ with high probability, where c is some positive constant (great than 3). That is, the estimation bias of the overall shadow price is upper bounded by

$$|\alpha_s(n)| \le \tilde{M}_s |E[\log(\hat{q})] - \log(q)| \le \frac{c'_s}{\sqrt{N_n}} \qquad (36)$$

for large N_n, where $\tilde{M}_s = \log(M_s)$ and c'_s is some positive constant.

Clearly, if a fixed number N_n of packets for round n is used, the biased term would exist. In contrast, to meet Condition $\mathbf{A3}$ such that the biased term should diminish asymptotically, it suffices to have that

$$\sum_n \frac{\epsilon_n}{\sqrt{N_n}} < \infty. \qquad (37)$$

For example, when $\epsilon_n = 1/n, N_n \sim O(\log^4(n))$ would satisfy (37), indicating that it suffices to have the measurement window size grows at the rate of $\log^4(n)$.

Next, we examine the variance of $\zeta_s(n)$ to check if Condition $\mathbf{A4}$ is satisfied. By (12) and (34), it follows that

$$\begin{aligned} E_n[\zeta_s(n)^2] &= E_n\Big[\hat{L}_{\tilde{x}_s}(\tilde{\mathbf{x}}(n), \mathbf{p}(n), \boldsymbol{\lambda}(n)) \\ &\qquad - E_n\left[\hat{L}_{\tilde{x}_s}(\tilde{\mathbf{x}}(n), \mathbf{p}(n), \boldsymbol{\lambda}(n))\right]\Big]^2 \\ &= E_n[\exp(\tilde{x}_s(n))(\log(\hat{q}) - E_n[\log(\hat{q})])]^2 \\ &\le \tilde{M}_s^2 E_n[\log(\hat{q})]^2 \\ &\le \tilde{M}_s^2 E_n[\log(q + c)]^2. \end{aligned}$$

Example 2: **Structure of** $\hat{L}_{\lambda_{(i,j)}}(\tilde{\mathbf{x}}(n), \mathbf{p}(n), \boldsymbol{\lambda}(n))$ **under random access:**

To update the shadow prices, each link needs to estimate the incoming flow rate $t = \sum_{s\in\mathcal{S}((i,j))} \exp(\tilde{x}_s(n))$ (or equivalently $\sum_{s\in\mathcal{S}((i,j))} x_s(n)$). Since the packets are transmitted using random access, there can be at most one successful transmission at each time slot. Therefore, the packet arrival rate at link (i, j) at each time slot follow a Bernoulli distribution with a successful probability $\sum_{s\in\mathcal{S}((i,j))} x_s$. The number of arrived packets K during a time window of M_n can be used to estimate the rate, and the estimator $\hat{t} = K/M_n$ follows a normal distribution $\mathcal{N}(t, t(1-t)/M_n)$ as M_n is reasonably large. To get an upper bound for $\beta_{(i,j)}(n)$, it is easy to see from (13) and (27) that

$$|\beta_{(i,j)}| \le |E[\log(\hat{t})] - log(t)| \le \frac{e_{(i,j)}}{\sqrt{M_n}} \qquad (38)$$

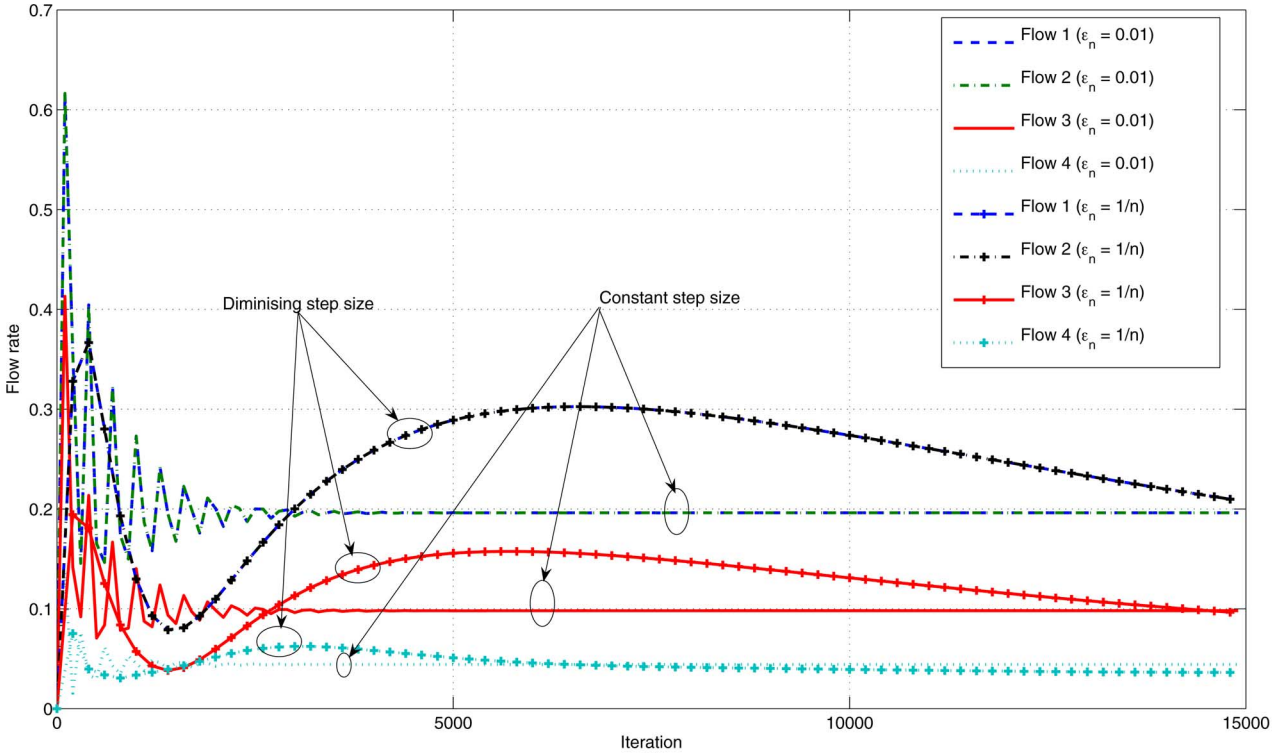

Fig. 2. Convergence of deterministic algorithm under different step sizes.

for large M_n, where $e_{(i,j)}$ is a positive constant. Thus, if M_n satisfies the same condition as N_n in (37), then Condition **A3** is met. Similar studies can be carried out to $\xi_{(i,j)}(n)$ in (14).

B. Numerical Examples

In this section, we illustrate, via numerical examples, the convergence performance of the single time-scale algorithm. Specifically, we consider a basic network scenario, as depicted in Fig. 1. There are four nodes (A, B, C, and D), four links and four flows, where flow 1 is from node A to node B, flow 2 from B and to C, flow 3 from A to C via B, and flow 4 from C to D via B. The utility function in Problem Ξ is taken to be the logarithm utility function where $\kappa = 1, w_s = 1$, and $M_s = 1, \forall\, s$.

1) Convergence of the Deterministic Algorithm: We first assume that there is no estimation error for the gradients. The results obtained by the primal-dual algorithm, together with the theoretical optimal solution, are summarized in Table I. It can be seen that the result obtained from the primal-dual algorithm corroborates the optimal solutions. Moreover, Fig. 2 illustrates the convergence rate of the primal-dual algorithm under different choices of the step size ϵ_n: Case I with a fixed value $\epsilon_n = 0.01$ and Case II with diminishing step sizes $\{\epsilon_n = 1/n\}$. Clearly, the iterates using step size $\epsilon_n = 0.01$ converge much faster than that using $\epsilon_n = 1/n$. We should caution that this does not necessarily indicate that using constant step sizes is better than diminishing step sizes, particularly in the presence of noisy feedback, and this is the subject of the next section.

2) Stability of the Stochastic Algorithm: The Unbiased Case: Next, we examine the convergence of the primal-dual algorithm under noisy feedback. Note that since the observation time window size has many samples, the random noise can be well approximated by the Gaussian random variable. Therefore, for ease of exposition, we let $\alpha_s(n) = \beta_{(i,j)}(n) = 1/n, \zeta_s(n) \sim \mathcal{N}(0,1)$ and $\xi_{(i,j)}(n) \sim \mathcal{N}(0,1)$, for all s and (i,j). Fig. 3 depicts the sequence of the iterates produced by the primal-dual algorithm under different ϵ_n. Comparing Figs. 3 and 2, we observe that the iterates using constant step sizes are less robust to the random perturbation, in the sense that the fluctuation of the iterates remains significant after many iterations; in contrast, the fluctuation using diminishing step sizes "vanishes" eventually. This corroborates Theorems 3.1 and 3.2 which reveal that with constant step sizes, the convergence to a neighborhood is the best we can hope; whereas by using diminishing step sizes, convergence with probability one to the optimal points is made possible.

3) Stability of the Stochastic Algorithm: The Biased Case: Recall that when the gradient estimation error is biased, we cannot hope to obtain almost sure convergence to the optimal solutions. Instead, we have shown that provided that the biased error is asymptotically uniformly bounded, the iterates return to

TABLE I
COMPARISON BETWEEN THE RESULT OF THE PROPOSED PRIMAL-DUAL ALGORITHM AND THE OPTIMAL SOLUTION

Link probabilities	$p_{(A,B)}$	$p_{(B,C)}$	$p_{(C,B)}$	$p_{(B,D)}$
Primal-dual algorithm	0.6655	0.3840	0.2009	0.0391
Optimal solution	0.6457	0.3750	0.2152	0.0443
Flow rate	x_1	x_2	x_3	x_4
Primal-dual algorithm	0.2065	0.2065	0.0974	0.0385
Optimal solution	0.1962	0.1962	0.0981	0.0443

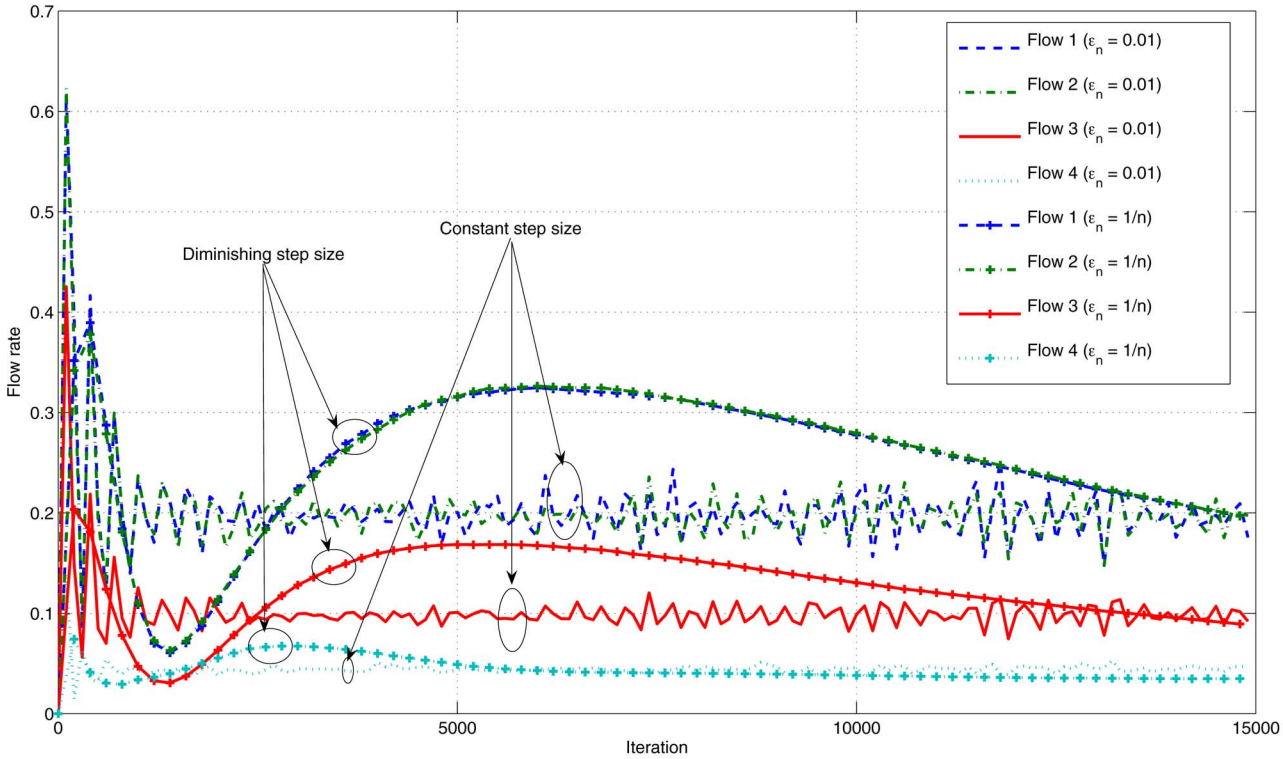

Fig. 3. Convergence under noisy feedback (the unbiased case).

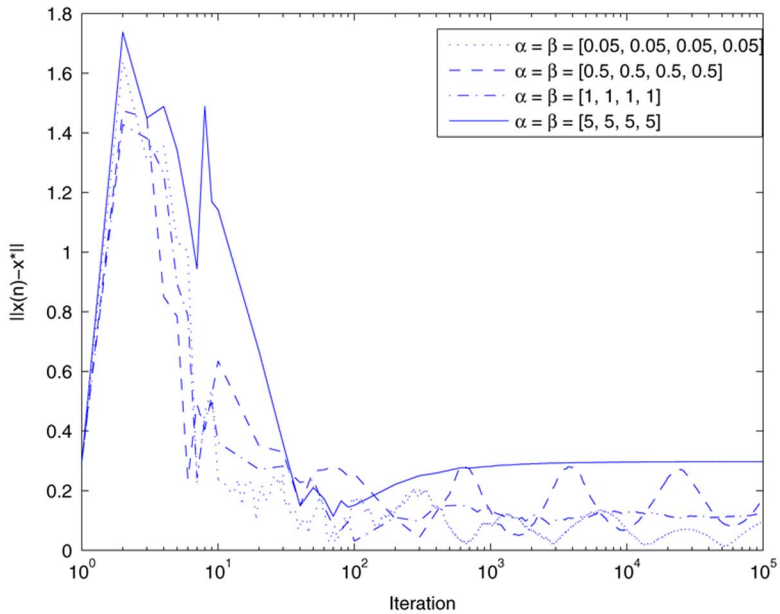

Fig. 4. Convergence under noisy feedback (the biased case).

a "contraction region" infinitely often. In this example, we assume that $\alpha_s(n) = \beta_{(i,j)}(n)$ and are uniformly bounded by a specified positive value. We also assume that $\zeta_s(n) \sim \mathcal{N}(0,1)$ and $\xi_{(i,j)}(n) \sim \mathcal{N}(0,1)$, for all s and (i,j).

We plot the iterates (using the relative distance to the optimal points) in Fig. 4, which is further "zoomed in" in Fig. 5. It can be observed from Fig. 4 that when the upper-bounds on the $\{\alpha_s, \beta_{(i,j)}\}$ are small, the iterates return to a neighborhood of the optimal solution. However, when the estimation errors are large, the recurrent behavior of the iterates may not occur, and the iterates may diverge. This corroborates the theoretical analysis. We can further observe from Fig. 5 that the smaller the upper-bound is, the smaller the "contraction region" A_η becomes, indicating that the iterates converge "closer" to the optimal points.

V. Stochastic Stability of Two Time-Scale Algorithm Under Noisy Feedback

In the previous sections, we have applied the dual decomposition method to Problem 1 and devised the primal-dual algorithm, which is a single time-scale algorithm. As noted in

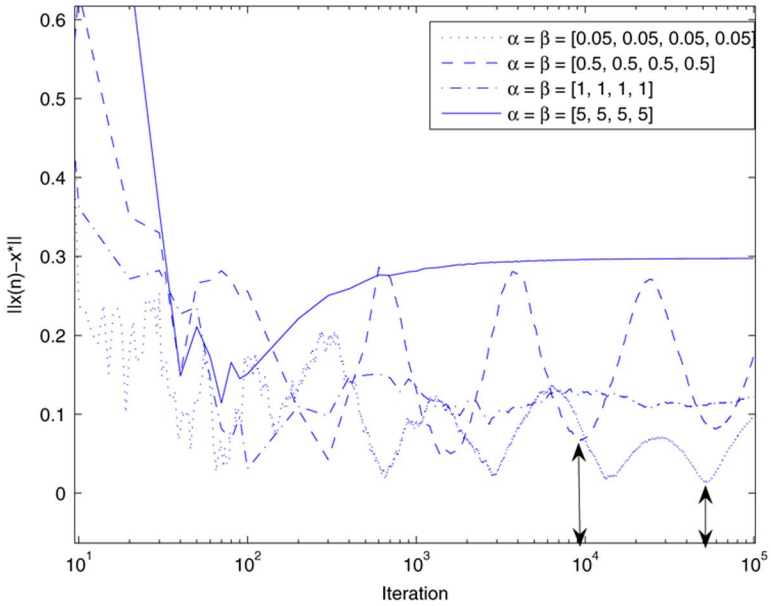

Fig. 5. "Zoomed-in" convergence behavior of the iterates in Fig. 4.

Section I, there are many other decomposition methods. In particular, the primal decomposition method is a useful machinery for problem with coupled variables [31]; and when some of the variables are fixed, the rest of the problem may decouple into several subproblems. This naturally yields multiple time-scale algorithms. It is also of great interest to examine the stability of the multiple time-scale algorithms in the presence of noisy feedback, and compare with the single time-scale algorithms, in terms of complexity and robustness.

To get a more concrete sense of the two time-scale algorithms based on primal decomposition, we consider the following NUM problem:

$$\begin{aligned} &\boldsymbol{\Xi_2}: \text{maximize}_{\{m_s \le x_s \le M_s,\ \mathbf{p}\}} \sum_s U_s(x_s) \\ &\text{subject to} \quad \sum_{s: l \in \mathcal{L}(s)} x_s \le c_l, \quad \forall l \\ &c_l = h_l(\mathbf{p}), \quad \forall l \\ &\mathbf{p} \in \mathcal{H} \end{aligned} \tag{39}$$

where the link capacities $\{c_l\}$ are functions of specific MAC parameters $\mathbf{p}$ (for instance, $\mathbf{p}$ can be transmission probabilities or powers), i.e., $c_l = h_l(\mathbf{p})$ with $h_l : \mathcal{R}^L \to \mathcal{R}, \{m_s\}$ and $\{M_s\}$ are positive lower and upper bounds on the flow rates, and $\mathcal{H}$ is the constraint set of the MAC parameters $\mathbf{p}$. Since the focus of this paper is on the impact of noisy feedback rather than that of nonconvexity, we assume that the above problem $\boldsymbol{\Xi_2}$ is a convex optimization (possibly after some transformation).

The deterministic two time-scale algorithm based on primal decomposition for solving (39) is given as follows:

- **Faster (smaller) time scale:**
  — *The source rates are updated by*

$$x_s(n) = \operatorname*{argmax}_{m_s \le x_s \le M_s} \left(U_s(x_s) - x_s \sum_{l \in \mathcal{L}(s)} \lambda_l(n) \right), \quad \forall s. \tag{40}$$

  — *The shadow prices are updated by*

$$\lambda_l(n+1) = \left[\lambda_l(n) - a_n \left(c_l(\mathbf{p}(n)) - \sum_{s \in \mathcal{S}(l)} x_s(n) \right) \right]_0^{\infty}, \quad \forall l \tag{41}$$

  where a_n is the positive step size.
- **Slower (Larger) time scale:**
  - *The MAC parameters are updated by*

$$ap_l(n+1) = \left[p_l(n) + b_n \sum_{k \in E} \lambda_k^* \frac{\partial h_k(\mathbf{p}(n))}{\partial p_l} \right]_{\mathcal{H}}, \ \forall l \tag{42}$$

  where b_n is the positive step size and λ^* is the optimal shadow price generated by the faster time-scale iteration when it converges.

Remarks: 1) Note that in the P-D algorithm based on the dual decomposition, local message passing is needed to update the persistence probabilities (cf. (24)) every iteration. However, in the two time-scale algorithm, the updating of the persistence probabilities is performed less frequently than that of the source rates and shadow prices. Therefore, message passing is significantly reduced in the two time-scale algorithm based on primal decomposition. Accordingly, the two time-scale algorithm has a lower communication complexity.

2) There are many ways to implement the above two time-scale algorithm. One method is to run the two loops sequentially, i.e., let the faster time-scale iteration (40) and (41) converge first, and then execute the slower time-scale loop. Alternatively, one can set $b_n = o(a_n)$ and run the algorithms at both time scales. A plausible way to implement this is to execute one update on the slower time-scale loop for every few iterations on the faster time-scale loop, i.e., the slower time-scale loop is run less frequently. Needless to say, different implementation

methods have different complexity and robustness issues. In the following, we focus on the second method for implementing the two time-scale algorithm, and examine its stability performance under noisy feedback.

A. Stability of the Stochastic Two Time-Scale Algorithm

Next, we examine the impact of the noisy feedback on the stability of the following stochastic two time-scale algorithm based on the primal decomposition method:

- **Faster time scale:**
  — *SA algorithm for source rate updating:*

$$x_s(n) = \underset{m_s \le x_s \le M_s}{\operatorname{argmax}} \left(U_s(x_s) - x_s \left[\sum_{l \in \mathcal{L}(s)} \lambda_l(n) + M_s^x(n) \right] \right). \tag{43}$$

where $\mathbf{M}^x(n)$ is the estimation error of the end-to-end price (possibly consisting of a bias term).

— *SA algorithm for shadow price updating:*

$$\lambda_l(n+1) = \left[\lambda_l(n) - a_n \left\{ \left(c_l(\mathbf{p}(n)) - \sum_{s \in \mathcal{S}(l)} x_s(n) \right) + M_l^{\lambda}(n) \right\} \right]_0^{\infty} \tag{44}$$

where $\mathbf{M}^{\lambda}(n)$ is the estimation error of the flow rate (possibly consisting of a bias term).

- **Slower time scale:**
  — *SA algorithm for MAC parameter updating*:

$$p_l(n+1) = \left[p_l(n) + b_n \times \left(\sum_{k \in E} \lambda_k(n+1) \frac{\partial h_k(\mathbf{p}(n))}{\partial p_l} + N_l(n) \right) \right]_{\mathcal{H}} \tag{45}$$

where $\mathbf{N}(n)$ is the estimation error of the subgradients (possibly consisting of a bias term).

Note that $\mathbf{M}^x(n), \mathbf{M}^{\lambda}(n)$ and $\mathbf{N}(n)$ are $\mathcal{F}(n)$-measurable, where $\mathcal{F}(n)$ is the σ-algebra generated by the observations made up to time n.

B. Stability for the Unbiased Case

Define $t_n^a \triangleq \sum_{i=0}^{n-1} a_i$, and $m^a(t) \triangleq m$ such that $t_m^a \le t < t_{m+1}^a$. Define t_n^b and $m^b(t)$ analogously with b_i in lieu of a_i. Define

$$\mathbf{M}^{a,x}(t) \triangleq \sum_{i=0}^{m^a(t)-1} a_i \mathbf{M}^x(i) \tag{46}$$

$$\mathbf{M}^{a,\lambda}(t) \triangleq \sum_{i=0}^{m^a(t)-1} a_i \mathbf{M}^{\lambda}(i) \tag{47}$$

$$\mathbf{N}^b(t) \triangleq \sum_{i=0}^{m^b(t)-1} b_i \mathbf{N}(i). \tag{48}$$

In order to establish the convergence of the above stochastic two time-scale algorithm in the presence of unbiased error, we impose the following assumptions: $\mathbf{B1}$. *Condition on the step sizes*:

$$a_n > 0,\ b_n < 0,$$
$$\sum_n a_n = \sum_n b_n = \infty,$$
$$a_n \longrightarrow 0, b_n \longrightarrow 0, b_n/a_n \longrightarrow 0.$$

$\mathbf{B2}$. *Condition on the noises: For some positive* T

$$\lim_n \sup_{j \ge n} \max_{0 \le t \le T} |\mathbf{M}^{a,x}(jT+t) - \mathbf{M}^{a,x}(jT)| = 0 \quad \text{w.p.1},$$
$$\lim_n \sup_{j \ge n} \max_{0 \le t \le T} |\mathbf{M}^{a,\lambda}(jT+t) - \mathbf{M}^{a,x}(jT)| = 0 \quad \text{w.p.1},$$
$$\lim_n \sup_{j \ge n} \max_{0 \le t \le T} |\mathbf{N}^{b}(jT+t) - \mathbf{N}^{b}(jT)| = 0 \quad \text{w.p.1}.$$

$\mathbf{B3}$. *Condition on the utility functions*: The curvatures of U_s are bounded: $0 < 1/\bar{\nu}_s \le -U_s''(x_s) \le 1/\bar{\mu}_s < \infty$ for all $x_s \in [m_s, M_s]$. Recall that Problem $\mathbf{\Xi_2}$ in (39) is convex. We have the following result.

Theorem 5.3: Under Conditions $\mathbf{B1} - \mathbf{B3}$, the iterates $\{(\mathbf{x}(n), \boldsymbol{\lambda}(n), \mathbf{p}(n)), n = 1, 2, \ldots\}$ generated by stochastic approximation algorithms (43), (44) and (45) converge with probability one to the optimal solutions of Problem $\mathbf{\Xi_2}$.

Sketch of the proof: A key step in this proof of is to establish the passage from the stochastic two time-scale algorithm to the mean ODE's at two time scales. Then the rest of the proof follows the convergence of the mean ODE's [10], [22]. The details of the proof are presented in Appendix C.

C. Stability for the Biased Case

The estimation error can also be biased in the two time-scale algorithm, i.e., there is no guarantee that $|\mathbf{M}^x(n)| \to 0$, $|\mathbf{M}^{\lambda}(n)| \to 0$ and/or $|\mathbf{N}(n)| \to 0$ with probability one. The convergence of the two time-scale algorithm is highly nontrivial in these cases.

We caution that besides the possible bias in gradient estimation, there may exist another cause of the biased error in the two time-scale algorithm. Recall that one popular approach is to let the faster time-scale iteration "converge" first, and then execute the slower time-scale iteration. Since the "convergence" of the faster time-scale iteration is determined by comparing the distance of the iterates with a pre-set threshold, an artificial biased error may be induced for the slower time-scale iteration.

For the case when $\mathbf{M}^x(n)$ and $\mathbf{M}^{\lambda}(n)$ are asymptotically unbiased but $\mathbf{N}(n)$ is biased, one can construct a contraction region, using the same method as in the stochastic P-D algorithm. However, when $\mathbf{M}^x(n)$ and $\mathbf{M}^{\lambda}(n)$ are biased, it is very challenging to define/examine the contraction region for the two time-scale algorithm, because the impact of the cumulative effect of the biases on slower time scale iteration can vary significantly.

D. Numerical Examples

We consider the same network model in Section IV.B. The utility functions are the logarithm function with $w_s = 1$, $m_s =$

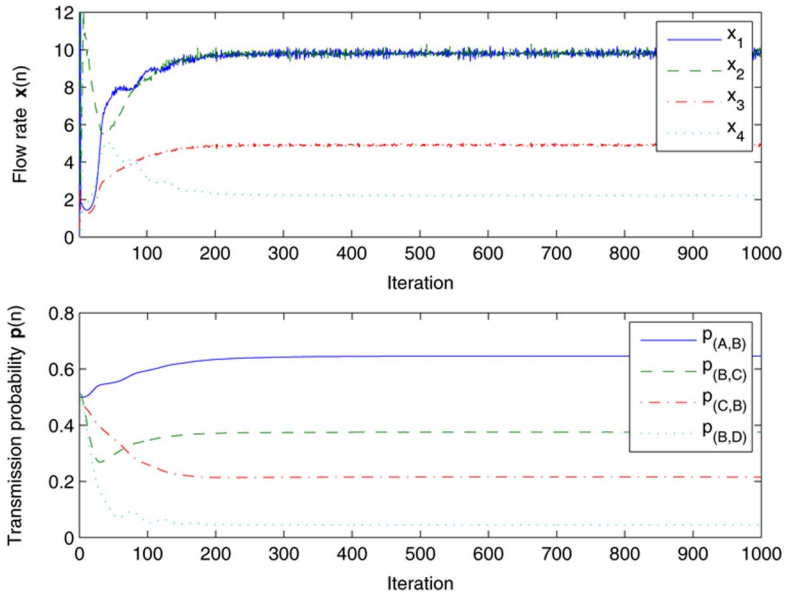

Fig. 6. Convergence of the two time-scale algorithm (standard deviation of $M_s^x(n)$ is 0.001).

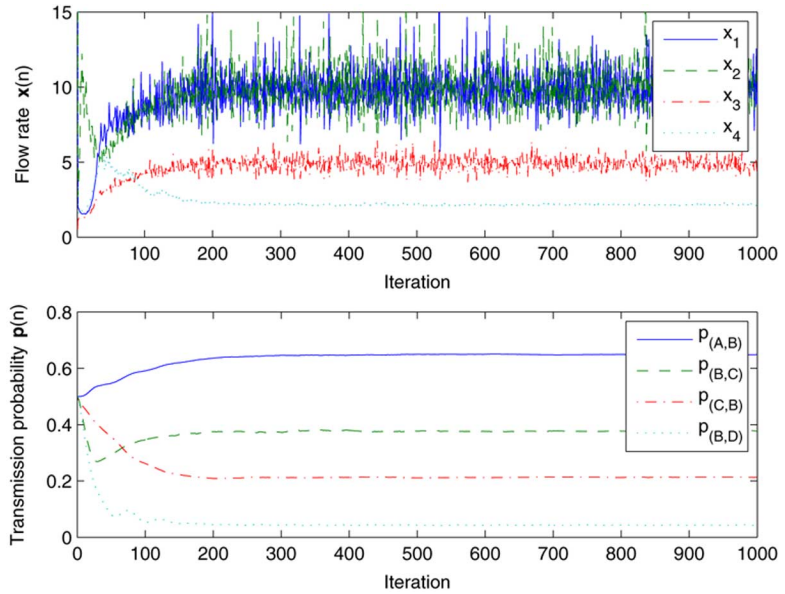

Fig. 7. Convergence of the two time-scale algorithm (standard deviation of $M_s^x(n)$ is 0.01).

1 and $M_s = 50$. The link capacity $r_{i,j}$ is scaled to be 50 for all i, j for numerical purposes. Therefore, the corresponding optimal flow rate is $\mathbf{x}^* = [9.810, 9.810, 4.900, 2.215]$ while the optimal link persistence probabilities remain the same as before. The step sizes are $a_n = \frac{0.01}{\lceil n^{0.75} * 500 \rceil}$ and $b_n = \frac{0.001}{\lceil n * 500 \rceil}$.

We first examine the convergence performance of the two time-scale algorithm (43), (44), and (45) when the noisy feedback is unbiased. The estimation noises $\{M_s^x(n)\}$, $\{M_l^\lambda(n)\}$ and $\{N_l(n)\}$ are set to be zero-mean Gaussian random variables, and the standard deviations of $\{M_l^\lambda(n)\}$ and $\{N_l(n)\}$ are fixed at 0.001 while the standard deviation of $\{M_s^x(n)\}$ varies from 0.001 to 0.01. We should point out that the variance of the gradient estimation error in (44) is significantly larger than that of $\{M^x(n)\}$, $\{M_l^\lambda(n)\}$ and $\{N_l(n)\}$, due to the nonlinear transformation of $\{M^x(n)\}$ in (44).

Figs. 6 and 7 depict the simulation results under different noise levels. Comparing Figs. 6 and 7, we observe that when the standard deviation of $\{M_s^x(n)\}$ increases, the faster time-scale iteration is more susceptible to the random perturbation than the slower time-scale one. Our intuition is as follows.

- The step size of the slower time-scale iteration is smaller than that of the faster time-scale one, and a smaller step size can help to reduce the noise effect;
- Due to nonlinear transformation, the noise $\{M_s^x(n)\}$ in (43) is amplified many folds in (44), and thus the induced noise in (44) is more significant. More specifically, the "amplified" noise due to $\{M_s^x(n)\}$ can be calculated as

$$\sum_{s \in \mathcal{S}(l)} \frac{1}{\sum_{l \in \mathcal{L}(s)} \lambda_l + M_s^x} - \sum_{s \in \mathcal{S}(l)} \frac{1}{\sum_{l \in \mathcal{L}(s)} \lambda_l}$$

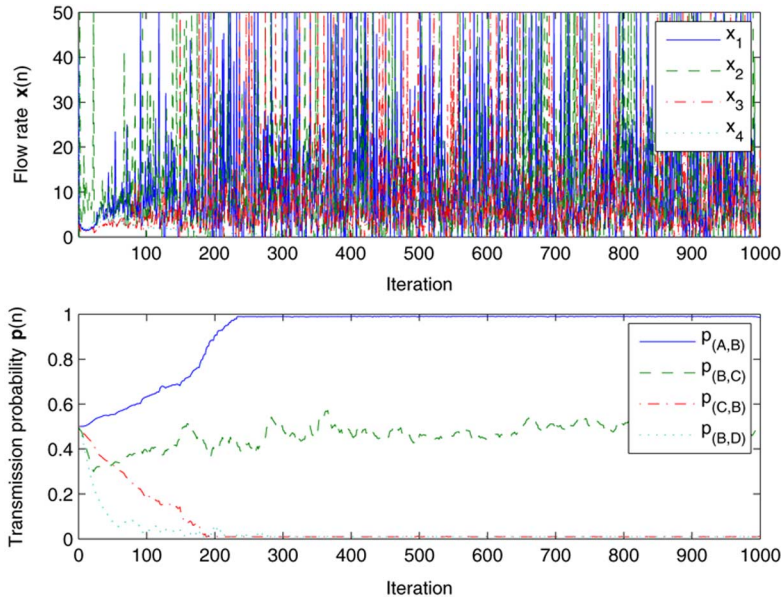

Fig. 8. Convergence of the two time-scale algorithm with standard deviation of $M_s^x(n)$ set to be 0.05.

$$\approx \sum_{s\in\mathcal{S}l} \frac{1}{\left(\sum_{l\in\mathcal{L}(s)}\lambda_l\right)^2} M_s^x$$
$$\approx \sum_{s\in\mathcal{S}(l)} x_s^2 M_s^x. \tag{49}$$

It follows that, around the equilibrium points, the "amplified" noises in (44) due to $M_s^x(n)$ are x_s^2 times larger, which amounts to 100 times for flow 1 and flow 2, 25 times for flow 3 and 5 times for flow 4. This also indicates that the impact of random perturbation on flows with higher rates are higher than that on flows with lower rates, which can be observed in Fig. 7.

We note that when the standard deviation of $\{M_s^x(n)\}$ are further increased to 0.05, the fluctuation of the faster time-scale iteration is so significant that it may drive the two time-scale algorithm to diverge (since $-U_s''(x_s) = 1/x_s^2$ is large when x_s is very small, and Condition **B3** is violated). In contrast, increasing the standard deviations of $\{M_l^\lambda\}$ or $\{N_l\}$ does not impact the convergence of the two time-scale algorithm much (see Fig. 9). This reveals that the noise term $\{M_s^x(n)\}$ plays an important role in the overall stability of the two time-scale algorithm.

Summarizing, our findings reveal that the faster time-scale iteration is more "vulnerable" to noisy perturbation than the slower time-scale one, and its behavior is closely tied to in the stability of the two time-scale algorithm. Consequently, we conclude that the two time-scale algorithm is less robust to the noisy feedback compared with the single time-scale algorithm.

Finally, we examine the performance of the two time-scale algorithm under biased noises. Particularly, we set the noises as follows: $M_s^x(n) = \mathcal{N}(0.1, 10^{-6}), \forall s, n, M_l^\lambda(n) = \mathcal{N}(0.1, 10^{-2}), \forall l, n$ and $N_l(n) = \mathcal{N}(1,1), \forall l, n$. It can be seen from Figs. 10 and 11 that the iterates converge to some point in the neighborhood of the optimal point. We speculate that there may exist a contraction region for the two time-scale algorithm under appropriate conditions on the biased noise terms, and more work in this avenue is needed to obtain a thorough understanding.

VI. Conclusion and Future Work

Stochastic noisy feedback is a practically inevitable phenomenon that has not been systematically investigated in the vast recent literature on distributed NUM. Considering the general distributed algorithms based on different decomposition methods, we make use of a combination of tools in stochastic approximation, Martingale theory and convex analysis to investigate the impact of stochastic noisy feedback.

- For the unbiased gradient case, we have established that the iterates, generated by the distributed P-D algorithm based on Lagrange dual method, converge with probability one to the optimal solution of the centralized algorithm.
- In contrast, when the gradient estimator is biased, we have showed that the P-D algorithm converges to a contraction region around the optimal point, provided that the biased terms are asymptotically (uniformly) bounded by a scaled version $(0 \le \eta < 1)$ of the true gradients.
- We have also studied the rate of convergence of the P-D algorithm for the unbiased case, and our results reveal that in general, the limit process of the interpolated process for the normalized iterate sequence is a stationary reflected linear diffusion process, not necessarily a Gaussian diffusion process.
- Finally, we have investigated stochastic NUM solutions based on alternative decomposition methods. Specifically, we have studied the impact of noisy feedback on the stability of the two time-scale algorithm based on primal decomposition. Using the mean ODE method, we show that the convergence with probability one can be assured under appropriate conditions. Compared with the single time-scale dual algorithm, the two time-scale algorithm has low complexity, but is less robust to noisy feedback,

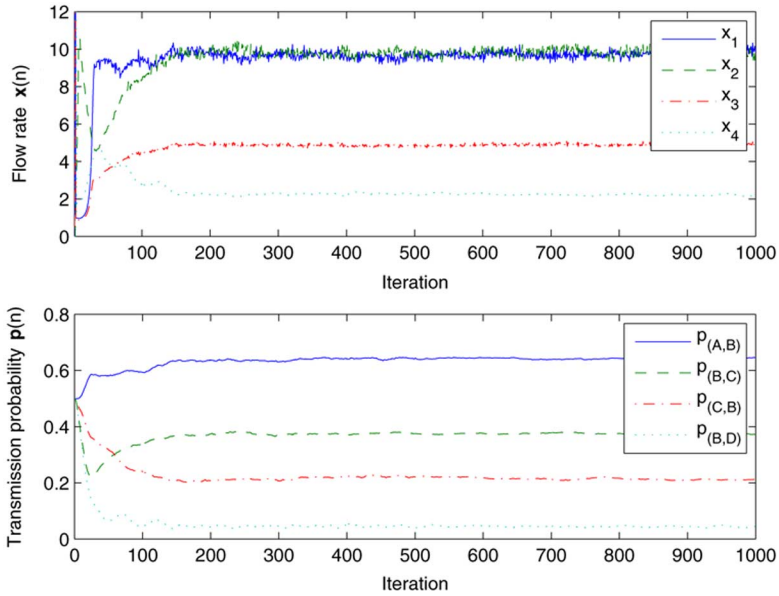

Fig. 9. Convergence of the two time-scale algorithm with standard deviations of $M_s^x(n)$, $M_l^\lambda(n)$ and $N_l(n)$ set to be 0.001, 0.1, and 1, respectively.

partially due to the sensitivity of the faster time-scale loop to perturbation.

The impact of stochastic noisy feedback on distributed algorithms is an important yet underexplored area. Besides the issues in this study, there are also many other challenges toward a complete theory under stochastic noisy feedback, for example, when we have nonconvex problem formulations or when there exists asynchronism. An even more fundamental issue is when the network utility maximization approach would be robust and fast-converging in multihop wireless networks, given the noisy and lossy nature of wireless communications, and which decomposition and distributed algorithm among the alternatives has the best robustness property.

APPENDIX

A. Proof of Theorem 3.1

Proof: Let $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$ be a saddle point where $\mathbf{x}^*$ is the unique optimum solution to the primal problem and $\boldsymbol{\lambda}^* \in \Phi$. Define the Lyapunov function $V(\cdot,\cdot)$ as follows:[2]

$$V(\mathbf{x},\lambda)\|\mathbf{x}-\mathbf{x}^*\|^2 + \min_{\lambda^*\in\Phi}\|\lambda-\lambda^*\|^2 \tag{50}$$

and define the set

$$A_\mu \triangleq \{\{\mathbf{x},\lambda\} : V(\mathbf{x},\lambda) \le \mu\} \tag{51}$$

for any given $\mu > 0$. In general, there can be multiple optimal shadow prices, meaning that

$$A_\mu = \bigcup_{\lambda^*\in\Phi} \{\{\mathbf{x},\lambda\} : \|\mathbf{x}-\mathbf{x}^*\|^2 + \|\lambda-\lambda^*\|^2 \le \mu\}.$$

A pictorial illustration of A_μ is provided in Fig. 12.

Step I: In what follows, we show that $\forall \mu > 0, A_\mu$ is recurrent, i.e., the iterates return to A_μ infinitely often with probability one.

Since $V(\cdot,\cdot)$ is continuous and Φ is compact, it follows that A_μ is compact and that the optimal shadow price is bounded above. Let $\lambda_0 > 0$ be an upper bound for the shadow price. For convenience, we define a "correction term" Z_n^λ to stand for the reflection effect that keeps the iterate $\lambda_l(n+1)$ in the constraint set. We rewrite (30) as

$$\lambda_l(n+1) = \lambda_l(n) - \epsilon_n[L_{\lambda_l}(\mathbf{x}(n),\boldsymbol{\lambda}(n)) + \beta_l(n) + \xi_l(n)] + \epsilon_n Z_n^{\lambda_l}. \tag{52}$$

Similarly, define Z_n^x as the correction term for x_s, and accordingly

$$x_s(n+1) = x_s(n) + \epsilon_n[L_{x_s}(\mathbf{x}(n),\boldsymbol{\lambda}(n)) + \alpha_s(n) + \zeta_s(n)] + \epsilon_n Z_n^{x_s}. \tag{53}$$

Next, observe that

$$\begin{aligned}&\|\mathbf{x}(n+1)-\mathbf{x}^*\|^2 \\ &\quad\le \|\mathbf{x}(n)-\mathbf{x}^*\|^2 + \epsilon_n^2\|L_{\mathbf{x}}(\mathbf{x}(n),\boldsymbol{\lambda}(n)) \\ &\qquad + \alpha(n)+\zeta(n)\|^2, +2\epsilon_n(\mathbf{x}(n)-\mathbf{x}^*)^T[L_{\mathbf{x}}(\mathbf{x}(n),\boldsymbol{\lambda}(n)) \\ &\qquad + \alpha(n)+\zeta(n)]\end{aligned}$$

where the inequality follows from the fact that the projection term $\epsilon_n Z_n^{x_s}$ is nonexpansive [5]. Using the Cauchy–Schwartz Inequality, we have that

$$(\mathbf{x}(n)-\mathbf{x}^*)^T\alpha(n) \le \|\mathbf{x}(n)-\mathbf{x}^*\|\cdot\|\alpha(n)\|.$$

Recall that both $\{U_s\}$ and $\{f_l\}$ are twice-differentiable functions. It follows that the Lagrangian function $L(\mathbf{x},\boldsymbol{\lambda})$ is also twice-differentiable and that both $L_x(\mathbf{x}(n),\boldsymbol{\lambda}(n))$ and $L_\lambda(\mathbf{x}(n),\boldsymbol{\lambda}(n))$ are bounded. Since $\|\mathbf{x}(n)-\mathbf{x}^*\|$ is bounded and $E_n(\zeta(n)) = 0$, combining Condition **A3** and the boundness of $\alpha(n)$ yields that

$$\begin{aligned}E_n(\|\mathbf{x}(n+1)-\mathbf{x}^*\|^2) &\le \|\mathbf{x}(n)-\mathbf{x}^*\|^2 \\ &\quad + 2\epsilon_n(\mathbf{x}(n)-\mathbf{x}^*)^T L_{\mathbf{x}}(\mathbf{x}(n),\boldsymbol{\lambda}(n)) \\ &\quad + O(\epsilon_n\|\alpha(n)\|) + O(\epsilon_n^2).\end{aligned} \tag{54}$$

[2]Throughout $\|\cdot\|$ refers to the *Euclidean* norm.

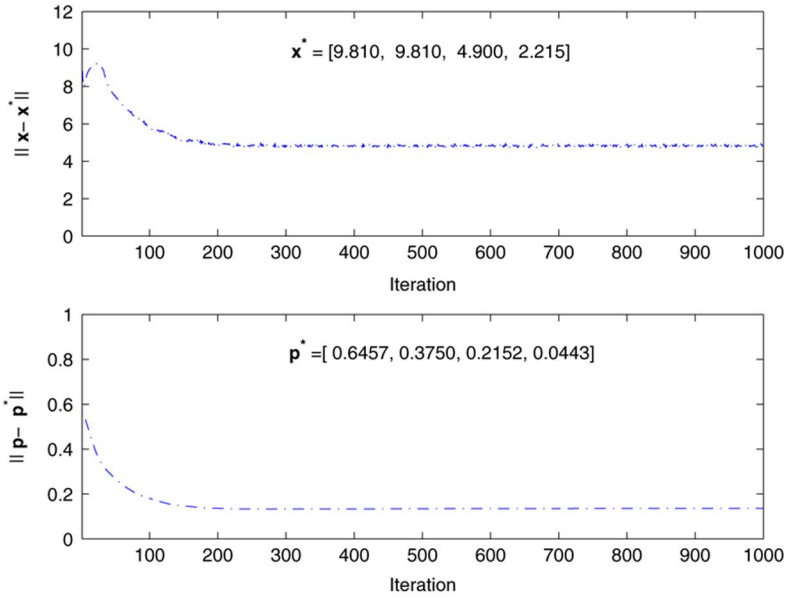

Fig. 10. Distance of the iterates to the optimal solution under biased feedback.

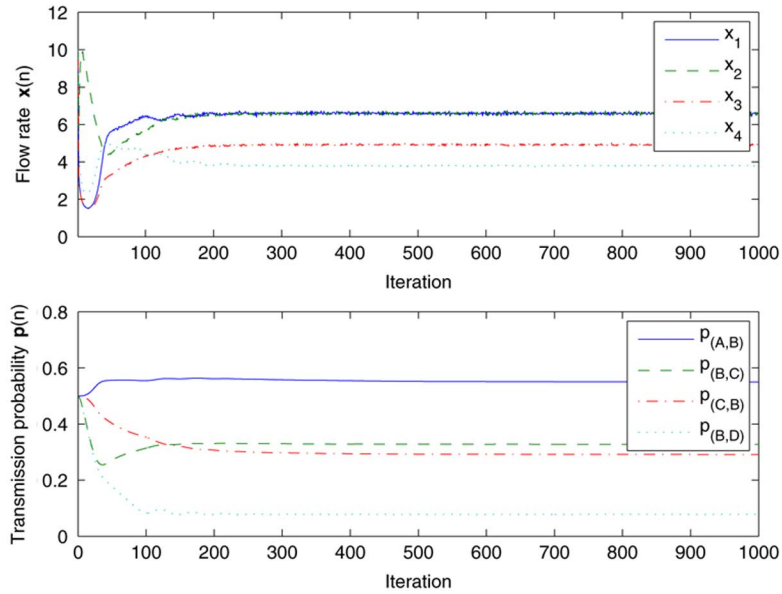

Fig. 11. Convergence under biased feedback.

Along the same lines, we have that

$$E_n(\|\boldsymbol{\lambda}(n+1)-\boldsymbol{\lambda}^*\|^2) \le \|\boldsymbol{\lambda}(n)-\boldsymbol{\lambda}^*\|^2 - 2\epsilon_n(\boldsymbol{\lambda}(n)-\boldsymbol{\lambda}^*)^T L_\lambda(\mathbf{x}(n),\boldsymbol{\lambda}(n)) + O(\epsilon_n\|\beta(n)\|) + O(\epsilon_n^2). \quad (55)$$

Combining (54) and (55) yields (56) shown at the bottom of the page.

$$E_n[\|\mathbf{x}(n+1)-\mathbf{x}^*\|^2 + \|\boldsymbol{\lambda}(n+1)-\boldsymbol{\lambda}^*\|^2] - (\|\mathbf{x}(n)-\mathbf{x}^*\|^2 + \|\boldsymbol{\lambda}(n)-\boldsymbol{\lambda}^*\|^2) \le 2\epsilon_n \underbrace{[(\mathbf{x}(n)-\mathbf{x}^*)^T L_{\mathbf{x}}(\mathbf{x}(n),\boldsymbol{\lambda}(n)) - (\boldsymbol{\lambda}(n)-\boldsymbol{\lambda}^*)^T L_\lambda(\mathbf{x}(n),\boldsymbol{\lambda}(n))]}_{\triangleq G(\mathbf{x}(n),\boldsymbol{\lambda}(n))} + O(\epsilon_n(\|\alpha(n)\| + \|\beta(n)\|)) + O(\epsilon_n^2). \quad (56)$$

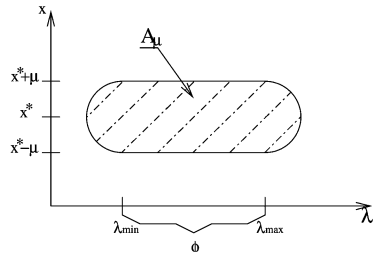

Fig. 12. A pictorial sketch of the set A_μ.

For convenience, define $\boldsymbol{\lambda}^*_{\min}(n) \triangleq \arg\min_{\boldsymbol{\lambda}\in\Phi} \|\boldsymbol{\lambda}(n) - \boldsymbol{\lambda}\|^2$. Since Inequality (56) holds for every $\boldsymbol{\lambda}^* \in \Phi$, substituting $\boldsymbol{\lambda}^* = \boldsymbol{\lambda}^*_{\min}(n)$ yields that

$$\begin{aligned} &E_n[V(\mathbf{x}(n+1),\boldsymbol{\lambda}(n+1))] \\ &\quad\le \quad E_n[\|\mathbf{x}(n+1)-\mathbf{x}^*\|^2 \\ &\qquad + \|\boldsymbol{\lambda}(n+1)-\boldsymbol{\lambda}^*_{\min}(n)\|^2] \\ &\quad\le \quad V(\mathbf{x}(n),\boldsymbol{\lambda}(n)) + 2\epsilon_n G(\mathbf{x}(n),\boldsymbol{\lambda}(n)) \\ &\qquad + O(\epsilon_n(\|\alpha(n)\| + \|\beta(n)\|)) + O(\epsilon_n^2) \end{aligned} \tag{57}$$

with the understanding that

$$\begin{aligned} &G(\mathbf{x}(n),\boldsymbol{\lambda}(n)) \\ &\quad = (\mathbf{x}(n)-\mathbf{x}^*)^T L_{\mathbf{x}}(\mathbf{x}(n),\boldsymbol{\lambda}(n)) \\ &\qquad - (\boldsymbol{\lambda}(n)-\boldsymbol{\lambda}^*_{\min}(n))^T L_\lambda(\mathbf{x}(n),\boldsymbol{\lambda}(n)). \end{aligned}$$

Based on the structure of (57), we need the following lemma from [7] to establish the recurrence of A_μ.

*Lemma A.1 (***A Super-Martingale Lemma***):* Let $\{X_n\}$ be an $\mathcal{R}^r$-valued stochastic process, and $V(\cdot)$ be a real-valued nonnegative function in $\mathcal{R}^r$. Suppose that $\{Y_n\}$ is a sequence of random variables satisfying that $\sum_n |Y_n| < \infty$ with probability one. Let $\{\mathcal{F}_n\}$ be a sequence of $\sigma-$algebras generated by $\{X_i, Y_i, i \le n\}$. Suppose that there exists a compact set $A \subset \mathcal{R}^r$ such that for all n

$$E_n[V(X_{n+1})] - V(X_n) \le -\epsilon_n\delta + Y_n, \text{for} X_n \notin A \tag{58}$$

where ϵ_n satisfies **A1** and δ is a positive constant. Then the set A is recurrent for $\{X_n\}$, i.e., $X_n \in A$ for infinitely many n with probability one.

Appealing to Lemma A.1, it suffices to show that $G(\mathbf{x}(n),\boldsymbol{\lambda}(n)) < 0$ for $(\mathbf{x}(n),\boldsymbol{\lambda}(n)) \in A^c_\mu$. Since the Lagrangian function $L(\mathbf{x}(n),\boldsymbol{\lambda}(n))$ is concave in $\mathbf{x}$, it follows that

$$\begin{aligned} &L(\mathbf{x}(n),\boldsymbol{\lambda}(n)) - L(\mathbf{x}^*,\boldsymbol{\lambda}(n)) \\ &\qquad \ge (\mathbf{x}(n)-\mathbf{x}^*)^T L_{\mathbf{x}}(\mathbf{x}(n),\boldsymbol{\lambda}(n)). \end{aligned} \tag{59}$$

Since $L(\mathbf{x},\boldsymbol{\lambda})$ is linear in $\boldsymbol{\lambda}$, we have that

$$\begin{aligned} &L(\mathbf{x}(n),\boldsymbol{\lambda}^*_{\min}) - L(\mathbf{x}(n),\boldsymbol{\lambda}(n)) \\ &\qquad = (\boldsymbol{\lambda}^*_{\min}-\boldsymbol{\lambda}(n))^T L_\lambda(\mathbf{x}(n),\boldsymbol{\lambda}(n)). \end{aligned} \tag{60}$$

Combining (59) and (60) yields that

$$\begin{aligned} &G(\mathbf{x}(n),\boldsymbol{\lambda}(n)) \\ &\quad\le \quad L(\mathbf{x}(n),\boldsymbol{\lambda}^*_{\min}) - L(\mathbf{x}^*,\boldsymbol{\lambda}(n)) \\ &\quad = L(\mathbf{x}(n),\boldsymbol{\lambda}^*_{\min}) - L(\mathbf{x}^*,\boldsymbol{\lambda}^*_{\min}) \qquad (61) \\ &\qquad + L(\mathbf{x}^*,\boldsymbol{\lambda}^*_{\min}) - L(\mathbf{x}^*,\boldsymbol{\lambda}(n)) \qquad (62) \end{aligned}$$

Since $(\mathbf{x}^*,\boldsymbol{\lambda}^*)$ is a saddle point, it follows that

$$L(\mathbf{x}(n),\boldsymbol{\lambda}^*) \le L(\mathbf{x}^*,\boldsymbol{\lambda}^*) \le L(\mathbf{x}^*,\boldsymbol{\lambda}(n)),$$

indicating that both (61) and (62) are nonpositive. Furthermore, there exists $\delta_\mu > 0$ such that $G(\{\mathbf{x}(n),\boldsymbol{\lambda}(n)\}) < -\delta_\mu$ when $(\mathbf{x}(n),\boldsymbol{\lambda}(n)) \in A^c_\mu$, where A^c_μ is the complement set of A_μ.

Summarizing, we conclude that $\{\mathbf{x}(n),\boldsymbol{\lambda}(n)\}$ return to A_μ infinitely often with probability one.

Step II: Next, we use local analysis to show that $\{(\mathbf{x}(n),\boldsymbol{\lambda}(n)\}), n = 1,2,\ldots$ leaves $A_{3\mu}$ only finitely often with probability one. Let $\{n_k, k = 1,2\ldots\}$ denote the recurrent times with $(\mathbf{x}(n_k),\boldsymbol{\lambda}(n_k)) \in A_\mu$. It suffices to show that there exists $n_{k_0} \in \{n_k, k = 1,2\ldots\}$, such that for all $n \ge n_{k_0}$, the original iterates $\{(\mathbf{x}(n),\boldsymbol{\lambda}(n)), n = 1,2,\ldots\}$ reside in $A_{3\mu}$ almost surely. The basic idea of Step II is depicted in Fig. 13. Specifically, rewrite the algorithm as

$$\begin{aligned} &\begin{bmatrix} x_s(n+1) \\ \lambda_l(n+1) \end{bmatrix} \\ &\quad = \begin{bmatrix} x_s(n) \\ \lambda_l(n) \end{bmatrix} + \epsilon_n \begin{bmatrix} L_{x_s}(\mathbf{x}(n),\boldsymbol{\lambda}(n)) \\ -L_{\lambda_l}(\mathbf{x}(n),\boldsymbol{\lambda}(n)) \end{bmatrix} \\ &\qquad + \epsilon_n \begin{bmatrix} \alpha_s(n)+\zeta_s(n) \\ \beta_l(n)+\xi_l(n) \end{bmatrix} + \epsilon_n \begin{bmatrix} Z_n^{x_s} \\ Z_n^{\lambda_l} \end{bmatrix}. \end{aligned} \tag{63}$$

Observe the following.

1) Combining Condition **A1** and the boundedness of $L_x(\cdot,\cdot)$ and $L_\lambda(\cdot,\cdot)$, yields that $\epsilon_n L_x(\mathbf{x}(n_k),\boldsymbol{\lambda}(n_k)) \to 0$ and $\epsilon_n L_\lambda(\mathbf{x}(n_k),\boldsymbol{\lambda}(n_k)) \to 0$.
2) Condition **A2** implies that $\epsilon_n\alpha_s(n) \to 0$ and $\epsilon_n\beta_l(n) \to 0$.
3) Using Chebyshev's Inequality, for any small positive ρ, we have that
$$P(|\epsilon_n\zeta_s(n)| > \rho) \le \epsilon_n^2 E[\zeta_s(n)^2]/\rho^2. \tag{64}$$
Based on Conditions **A1** and **A4**, we use the Borel–Cantelli Lemma to conclude that $\epsilon_n\zeta_s(n) > \rho$ only finitely often with probability one [8]. The same result can be obtained for $\epsilon_n\xi_l(n)$.

Combining the above observations with the properties of the reflection terms, we conclude that there exists n_{k_1} such that for all $n_k \ge n_{k_1}$, the overall change in one single step is no greater than μ and therefore the (n_k+1)th iterate $(\mathbf{x}(n_k+1),\boldsymbol{\lambda}(n_k+1))$ resides in $A_{2\mu}$ with probability one.

Next, we show that there exists $n_{k_0} \ge n_{k_1}$ such that for all $n \ge n_{k_0}, \{\mathbf{x}(n),\boldsymbol{\lambda}(n)\} \in A_{3\mu}$ almost surely. Using (63), it suffices to show that the combined accumulative effect of $\epsilon_n \begin{bmatrix} L_{x_s}(\mathbf{x}(n),\boldsymbol{\lambda}(n)) \\ -L_{\lambda_l}(\mathbf{x}(n),\boldsymbol{\lambda}(n)) \end{bmatrix}$ and $\epsilon_n \begin{bmatrix} \alpha_s(n)+\zeta_s(n) \\ \beta_l(n)+\xi_l(n) \end{bmatrix}$ cannot "drive" $\{\mathbf{x}(n),\boldsymbol{\lambda}(n)\}$ out of $A_{3\mu}$ for sufficiently large n_{k_0}.

From Step I, it is clear that $\begin{bmatrix} L_{x_s}(\mathbf{x}(n),\boldsymbol{\lambda}(n)) \\ -L_{\lambda_l}(\mathbf{x}(n),\boldsymbol{\lambda}(n)) \end{bmatrix}$ is the gradient and would drive $(\mathbf{x}(n),\boldsymbol{\lambda}(n))$ toward A_μ because of the gradient descent properties. Furthermore, Condition **A2** reveals that $\lim_{m\to\infty}\sum_{n=m}^{\infty}\epsilon_n\alpha_s(n) \to 0$ and $\lim_{m\to\infty}\sum_{n=m}^{\infty}\epsilon_n\beta_l(n) \to 0$. Hence, the only possible terms

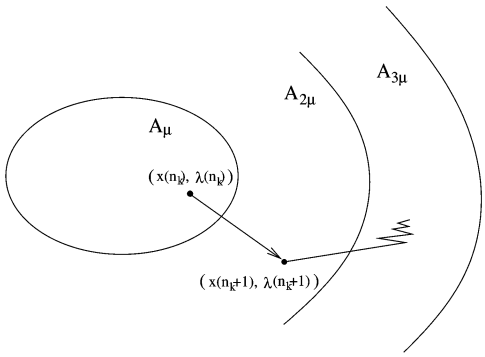

Fig. 13. A sketch of the basic idea in Step II.

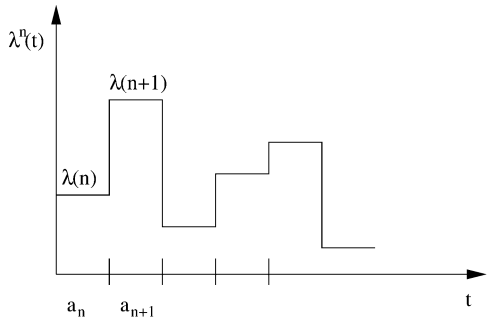

Fig. 14. An illustration of the shifted interpolated process of $\{\boldsymbol{\lambda}(n)\}$.

that can drive $(\mathbf{x}(n), \boldsymbol{\lambda}(n))$ out of $A_{3\mu}$ are the martingale difference noises $\epsilon_n \zeta_s(n)$ and $\epsilon_n \xi_l(n)$. However, it follows from the Martingale Inequality [22] that

$$P\left\{\sup_{k\geq m} |\sum_{n=m}^{k-1} \epsilon_n \zeta_s(n)| \geq \rho\right\} \leq \frac{\limsup_n E[\zeta_s(n)^2]}{\rho^2} \sum_{n=m}^{\infty} \epsilon_n^2$$

and hence

$$\lim_{m\to\infty} P\left\{\sup_{k\geq m} |\sum_{n=m}^{k-1} \epsilon_n \zeta_s(n)| \geq \rho\right\} = 0, \forall\, \rho > 0. \quad (65)$$

That is to say, $\sup_{k\geq m} |\sum_{n=m}^{k-1} \epsilon_n \zeta_s(n)| p_{\rightarrow} 0$ as $m \rightarrow \infty$. It can be further shown that this is equivalent to $\sup_{k\geq m} |\sum_{n=m}^{k-1} \epsilon_n \zeta_s(n)| \rightarrow 0$ almost surely (cf. Theorem 7.3.2 in [32]). Similarly, $\limsup_{m\to\infty} |\sum_{n=m}^{\infty} \epsilon_n \xi_l(n)| = 0$ almost surely. In a nutshell, these Martingale difference noises cannot drive $(\mathbf{x}(n), \boldsymbol{\lambda}(n))$ out of $A_{3\mu}$ for n_k sufficiently large.

Combining the above steps, we conclude that $\{(\mathbf{x}(n), \boldsymbol{\lambda}(n)), n = 1, 2, \ldots\}$ leaves $A_{3\mu}$ only finite often almost surely.

Since μ can be made arbitrarily small, it follows that $\{(\mathbf{x}(n), \boldsymbol{\lambda}(n),), n = 1, 2, \ldots\}$ converges w.p. 1 to the optimal solutions. This concludes the proof. □

B. Proof of Theorem 3.2

Proof: Theorem 3.2 can be proven along the same line as that of Theorem 3.1 above. In what follows, we outline a few key steps.

Define $A_{\mu,\eta} \triangleq A_\mu \cup A_\eta$. It can be shown that $A_{\mu,\eta}$ is compact. Following the same line as in the proof for Theorem 3.1, it can be shown that outside $A_{\mu,\eta}$

$$\begin{aligned} &E_n[V(\mathbf{x}(n+1), \boldsymbol{\lambda}(n+1))] \\ &\quad \leq V(\mathbf{x}(n), \boldsymbol{\lambda}(n)) \\ &\qquad + 2\epsilon_n(1-\eta) G(\mathbf{x}(n), \boldsymbol{\lambda}(n)) + O(\epsilon_n^2) \end{aligned} \quad (66)$$

where $V(\cdot,\cdot)$ and $G(\cdot,\cdot)$ are defined in Appendix VI-A. Recall that $G(\mathbf{x}(n), \boldsymbol{\lambda}(n)) < -\delta$ for some positive constant δ when $(\mathbf{x}(n), \boldsymbol{\lambda}(n)) \in A_\mu^c$. Since $A_{\mu,\eta}^c$ is a subset of A_μ^c, it follows that $G(\mathbf{x}(n), \boldsymbol{\lambda}(n)) < -\delta$ for $(\mathbf{x}(n), \boldsymbol{\lambda}(n)) \in A_{\mu,\eta}^c$. Accordingly, appealing to Lemma A.1, we conclude that the iterates return to $A_{\mu,\eta}$ infinitely often with probability one. Letting $\mu \rightarrow 0$, we have that $A_{\mu,\eta} \rightarrow A_\eta$, thereby concluding the proof. □

C. Proof of Theorem 5.3

Proof: To establish the stability of the stochastic two time-scale algorithm based on primal decomposition, it suffices to show that on the faster time scale, the iterates $\boldsymbol{\lambda}(n)$ follow asymptotically the trajectory of the mean ODE in (40) and (41) with fixed $\mathbf{p}$, and that on the larger time scale, the iterates $\mathbf{p}(n)$ follow the trajectory of the mean ODE (42) in the subgradient form. We note that due to the structure of this two time-scale algorithm, the establishment of the "passage" from the stochastic form to the mean ODE's is nontrivial, and it is intimately tied to the "amplified" noise in the gradient estimation and the curvatures of the utility functions.

As is standard [22], we define the continuous-time interpolation of $\{\boldsymbol{\lambda}(n)\}$ as

$$\boldsymbol{\lambda}^0(t) = \begin{cases} \boldsymbol{\lambda}(0), & \text{if } t \leq 0 \\ \boldsymbol{\lambda}(n), & \text{if } t_n^a \leq t < t_{n+1}^a \end{cases} \quad (67)$$

and define the shifted process $\boldsymbol{\lambda}^n(t) \triangleq \boldsymbol{\lambda}^0(t_n + t)$ (see Fig. 14). Define the interpolated (and shifted) process of $\mathbf{p}$ on the slower time scale analogously with $\mathbf{p}(n)$ in lieu of $\boldsymbol{\lambda}$.

We first show that on the faster time scale, the change of the shifted process $\mathbf{p}^n(t)$ is infinitesimal for any given t as $n \rightarrow \infty$. To see this, note that

$$\mathbf{p}^n(t) = \mathbf{p}(n) + \sum_{i=n}^{m^a(t+t_n)-1} b_i \mathbf{F}(i) + \sum_{i=n}^{m^a(t+t_n)-1} b_i \mathbf{N}(i)$$

where $\mathbf{F}_l(n) = \sum_{k\in L} \lambda_k^* \frac{\partial h_k(\mathbf{p}(n))}{\partial p_l}$. Thus, it suffices to show that the second term and the third term on the right hand side go to zero with probability one as $n \rightarrow \infty$ [22]. To this end, it is easy to see that the second term goes to zero since $\mathbf{F}$ is bounded above and that

$$\begin{aligned} \sum_{i=n}^{m^a(t+t_n)-1} b_i &\leq \sup_{i\geq n} \frac{b_i}{a_i} \sum_{i=n}^{m^a(t+t_n)-1} a_i \\ &\overset{(A)}{\leq} t \sup_{i\geq n} \frac{b_i}{a_i} n \rightarrow \infty \overset{(B)}{\longrightarrow} 0 \end{aligned} \quad (68)$$

where (A) is based on the definition of $m^a(t)$ and (B) follows from Condition **B1**. To examine the third term, observe that

$$\sum_{i=n}^{m^a(t+t_n)-1} b_i \mathbf{N}(i) = \mathbf{N}^b(T_{n,1}) - \mathbf{N}^b(T_{n,2}) \quad (69)$$

where $m^b(T_{n,1}) = m^a(t+t_n)$ and $m^b(T_{n,2}) = m^a(t_n)$. Based on the definition of $m^b(t)$, we conclude that as $n \to \infty$, $T_{n,i} \to \infty,\ i = 1, 2$, and that

$$T_{n,1} - T_{n,2} < \sum_{i=m^a(t_n)}^{m^a(t+t_n)-1} b_i < \sum_{i=m^a(t_n)}^{m^a(t+t_n)-1} a_i \le t.$$

It then follows from Condition $\mathbf{B2}$ that the third term goes to zero for any given t (note that in the above analysis, we have used the fact that if $\mathbf{B2}$ is valid for some positive T, then $\mathbf{B2}$ holds for all positive T [22]).

In summary, we have shown that on the faster time scale $\mathbf{p}^n(t)$ remains almost unchanged for any given t asymptotically. Next, we show that the shifted process $\boldsymbol{\lambda}^n(t)$ is equicontinuous in the extended sense and hence by the Arzela–Ascoli Theorem it contains a convergent subsequence whose limit follows the trajectory of the dual algorithm in (40) and (41). Specifically, rewrite $\boldsymbol{\lambda}_l^n(t)$ as follows:

$$\begin{aligned}\lambda_l^n(t) = \lambda_l(n) &- \sum_{i=n}^{m^a(t+t_n)} a_i \Bigg[c_l(\mathbf{p}(i)) \\ &- \sum_{s\in\mathcal{S}(l)} U_s'^{-1}\left(\sum_{l\in\mathcal{L}(s)} \lambda_l(i)\right)\Bigg] \\ &+ \sum_{i=n}^{m^a(t+t_n)} a_i \sum_{s\in\mathcal{S}(l)} U_s'^{-1}\left(\sum_{l\in\mathcal{L}(s)} \lambda_l(i) + M_s^x(i)\right) \\ &- \sum_{i=n}^{m^a(t+t_n)} a_i \sum_{s\in\mathcal{S}(l)} U_s'^{-1}\left(\sum_{l\in\mathcal{L}(s)} \lambda_l(i)\right) \\ &- \sum_{i=n}^{m^a(t+t_n)} a_i M_l^{\lambda}(i) + \sum_{i=n}^{m^a(t+t_n)} a_i Z_l^{\lambda}(i) \qquad (70)\end{aligned}$$

where $Z_l^{\lambda}(n)$ is the reflection term that keeps the shadow price nonnegative. Observe that

$$\begin{aligned}&\left|\sum_{i=n}^{m^a(t+t_n)} a_i \sum_{s\in\mathcal{S}(l)} U_s'^{-1}\left(\sum_{l\in\mathcal{L}(s)} \lambda_l(i) + M_s^x(i)\right)\right. \\ &\quad\left. - \sum_{i=n}^{m^a(t+t_n)} a_i \sum_{s\in\mathcal{S}(l)} U_s'^{-1}\left(\sum_{l\in\mathcal{L}(s)} \lambda_l(i)\right)\right| \\ &\quad\overset{(A)}{=} \left|\sum_{s\in\mathcal{S}(l)} \sum_{i=n}^{m^a(t+t_n)} a_i \vartheta_{s,i} M_s^x(i)\right| \xrightarrow{(B)} 0 w.p.1 \qquad (71)\end{aligned}$$

where (A) follows from the Mean Value Theorem and Condition $\mathbf{B3}$ (which indicates that $0 < \bar{\mu}_s \le \vartheta_{s,i} \le \bar{\nu}_s < \infty, \forall i, s$), and (B) from Condition $\mathbf{B2}$. Again, based on Condition $\mathbf{B2}$, we have that $|\sum_{i=n}^{m^a(t+t_n)} a_i M_l^{\lambda}(i)| \longrightarrow 0$. It then can be shown that $\sum_{i=n}^{m^a(t+t_n)} a_i [c_l(\mathbf{p}(i)) - \sum_{s\in\mathcal{S}(l)} U_s'^{-1}(\sum_{l\in\mathcal{L}(s)} \lambda_l(i))]$ and $\sum_{i=n}^{m^a(t+t_n)} a_i Z_l^{\lambda}(i)$ are equicontinuous in the extended sense [22]. As a result, the shifted process $\boldsymbol{\lambda}^n(t)$ is equicontinuous in the extended sense, and the limit of the convergent subsequence follows the trajectory of the mean ODE in (40) and (41).

Similarly, on the slower time scale, it can be shown that the interpolated process of $\mathbf{p}(n)$ follows the trajectory of the mean ODE in (42). This, together with the convergence property of the mean ODEs (40), (41), and (40). □

REFERENCES

[1] M. Adler, J.-Y. Cai, J. K. Shapiro, and D. Towsley, "Estimation of probabilistic congestion price using packet marking," in *Proc. IEEE INFOCOM'03*, San Francisco, CA, 2003, pp. 2068–2078.

[2] M. Andrews, "Joint optimization of scheduling and congestion control in communication networks," in *Proc. Conf. Information Science and Systems (CISS)*, Princeton, NJ, 2006, pp. 1572–1577.

[3] S. Athuraliya, V. H. Li, S. H. Low, and Q. Yin, "REM: Active queue management," *IEEE Network*, vol. 15, pp. 48–53, May/Jun. 2001.

[4] F. Baccelli and D. Hong, "AIMD, fairness and fractal scaling of TCP traffic," in *Proc. INFOCOM*, New York City, 2002, pp. 229–238.

[5] D. P. Bertsekas, *Nonlinear Programming*. Belmont, MA: Athena Scientific, 1995.

[6] D. P. Bertsekas and R. Gallager, *Data Networks*. , New Jersey: Prentice Hall, 1992.

[7] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*. Nashua, NH : Athena Scientific, 1996.

[8] P. Billingsley, *Probability and Measure*, 3rd ed. New York: Wiley, 1995.

[9] T. Bonald and L. Massoulie, "Impact of fairness on internet performance," in *Proc. Sigmetrics*, Cambridge, MA, 2001, pp. 82–91.

[10] V. Borkar, "Stochastic approximation with two time scales," *Syst. Contr. Lett.*, vol. 29, pp. 291–294, 1997.

[11] R. Buche and H. J. Kushner, "Rate of convergence for constrained stochastic approximation algorithms," *SIAM J. Contr. Optim.*, vol. 40, pp. 1011–1041, 2001.

[12] C. S. Chang and Z. Liu, "A bandwidth sharing theory for a large number of HTTP-like connections," *IEEE/ACM Trans. Netw.*, vol. 12, no. 5, pp. 952–962, Oct. 2004.

[13] L. Chen, S. H. Low, M. Chiang, and J. C. Doyle, "Jointly optimal congestion control, routing, and scheduling for wireless ad hoc networks," in *Proc. IEEE INFOCOM'06*, Barcelon, Catalonia, Spain, 2006, pp. 1–13.

[14] M. Chiang, "Balancing transport and physical layers in wireless multihop networks: Jointly optimal congestion control and power control," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 1, pp. 104–116, Jan. 2005.

[15] M. Chiang, S. H. Low, R. A. Calderbank, and J. C. Doyle, "Layering as optimization decomposition," *Proc. IEEE*, Jan. 2007.

[16] S. Deb, S. Shakkottai, and R. Srikant, "Asymptotic behavior of internet congestion controllers in a many-flow regime," *Math. Operations Research*, 2005.

[17] A. Eryilmaz and R. Srikant, "Joint congestion control, routing and MAC for stability and fairness in wireless networks," in *Proc. IZS'06*, Zurich, Switzerland, Feb. 2006.

[18] P. Gupta and A. L. Stolyar, "Optimal throughput allocation in general random-access networks," in *Proc. Conf. Information Science and Systems (CISS)*, Princeton, NJ, 2006, pp. 1254–1259.

[19] H. j. Kushner and J. Yang, "Analysis of adaptive step size sa algoithms for parameter tracking," *IEEE Trans. Autom. Control*, vol. 40, no. 8, pp. 1403–1410, 1995.

[20] F. P. Kelly, "Fairness and stability of end-to-end congestion control," *European J. Contr.*, pp. 159–176, 2003.

[21] F. P. Kelly, A. Maulloo, and D. K. H. Tan, "Rate control for communication networks: Shadow price, proportional fairness and stability," *J. Oper. Res. Soc.*, pp. 237–252, 1998.

[22] H. Kushner and G. Yin, *Stochastic Approximation and Recursive Algorithms and Applications*. New York: Springer, 2003.

[23] J. W. Lee, M. Chiang, and R. A. Calderbank, "Efficient and fair MAC for wireless network: Optimization framework, optimal algorithms, performance comparison," *IEEE Trans. Wireless Commun.*, 2007, to be published.

[24] J. Lee, M. Chiang, and A. Calderbank, "Jointly optimal congestion and medium access control in ad hoc wireless networks," *IEEE Commun. Lett.*, vol. 10, no. 3, pp. 216–218, Mar. 2005.

[25] X. Lin and N. Shroff, "The impact of imperfect scheduling on cross-layer rate control in multihop wireless networks," in *Proc. IEEE INFOCOM'05*, Miami, FL, 2005, pp. 1804–1814.

[26] X. Lin and N. B. Shroff, "Utility maximization for communication networks with multipath routing," *IEEE Trans. Autom. Control*, vol. 51, no. 5, pp. 766–781, May 2006.

[27] S. Low and R. Srikant, "A mathematical framework for designing a low-loss low-delay internet," *Netw. Spatial Econ.*, vol. 4, no. 1, pp. 75–101, 2004.
[28] M. Mehyar, D. Spanos, and S. Low, "Optimization flow control with estimation error," in *Proc. INFOCOM'04*, Hong Kong, China, 2004, pp. 984–992.
[29] J. Mo and J. Walrand, "Fair end-to-end window-based congestion control," *IEEE Trans. Netw.*, vol. 8, pp. 556–566, Nov. 2000.
[30] M. J. Neely, E. Modiano, and C. Li, "Fairness and optimal stochastic control for heterogeneous networks," in *Proc. INFOCOM*, Miami, FL, 2005, pp. 1723–1734.
[31] D. Palomar and M. Chiang, "A tutorial on decomposition methods and distributed network resource allocation," *IEEE J. Sel. Area Commun.*, vol. 24, no. 8, pp. 1439–1451, Aug. 2006.
[32] S. I. Resnick*, A Probability Path*. New York: Birkhauser, 1999.
[33] A. L. Stolyar, "Maximizing queuing network utility subject to stability: Greedy primal-dual algorithm," *Queue. Syst.*, vol. 50, no. 4, pp. 401–457, 2005.
[34] P. Tinnakornsrisuphap, R. La, and A. Makowski, "Modeling TCP traffic with session dynamics–many sources asymptotics under ECN/RED gateways," in *Proc. 18th Int. Teletraffic Congr.*, Berlin, Germany, 2003.
[35] X. Wang and K. Kar, "Cross-layer rate control for end-to-end proportional fairness in wireless networks with random access," in *Proc. MOBIHOC'05*, Urbana, IL, 2005, pp. 157–168.
[36] Y. Xi and E. M. Yeh, "Optimal capacity allocation, routing, and congestion control in wireless networks," in *Proc. IEEE Int. Symp. Information Theory*, Seattle, WA, 2006, pp. 2511–2515.
[37] J. Zhang, D. Zheng, and M. Chiang, "Impact of stochastic noisy feedback on distributed network utility maximization," Dept. Elec. Eng., Arizona State Univ., Tech. Rep. 2006.
[38] J. Zhang, D. Zhang, and M. Chiang, "Impact of stochastic noisy feedback on distributed network utility maximization," in *Proc. INFOCOM 2007*, Anchorage, AK, 2007, pp. 222–230.
[39] J. Zhang and D. Zheng, "A stochastic primal-dual algorithm for joint flow control and MAC design in multi-hop wireless networks," in *Proc. Conf. Information Science and Systems (CISS'06)*, Princeton, NJ, 2006, pp. 339–344.