Network Flow-Control using Asynchronous Stochastic Approximation

Mohammed Shahid Abdulla and Shalabh Bhatnagar
Department of Computer Science and Automation
Indian Institute of Science, Bangalore, INDIA
{shahid,shalabh@csa.iisc.ernet.in}

***Abstract*— We propose several stochastic approximation implementations for related algorithms in flow-control of communication networks. First, a discrete-time implementation of Kelly's primal flow-control algorithm is proposed. Convergence with probability 1 is shown, even in the presence of communication delays and stochastic effects seen in link congestion indications. This ensues from an analysis of the flow-control algorithm using the asynchronous stochastic approximation (ASA) framework. Two relevant enhancements are then pursued: *a)* an implementation of the primal algorithm using second-order information, and *b)* an implementation where edge-routers rectify misbehaving flows. Next, discrete-time implementations of Kelly's dual algorithm and primal-dual algorithm are proposed. Simulation results *a)* verifying the proposed algorithms and, *b)* comparing the stability properties are presented.**

Keywords

Network flow-control, Kelly's primal algorithm, asynchronous stochastic approximation, SPSA, edge-router based rectification.

I. Introduction

A. Problem Description

For a network with finite sets of sources R and links L, the network utility-maximization problem is given by

$$\max_{x_r} \sum_{r\in R} U_r(x_r) \quad \text{s. t.} \sum_{s\in R^l} x_s \le c_l, \forall l \in L, \qquad (1)$$

where $c_l > 0$ is the capacity of link l and x_r is the rate at which the source r sends traffic into the network. The utility function U_r is positive-valued, strictly concave, and differentiable, and is termed as 'elastic' utility due to its concavity. The set $R^l \subset R$ is the set of sources that transmit over link l. An approximation to this problem is the unconstrained maximization problem with the following objective function:

$$V(x) = \sum_{r\in R} U_r(x_r) - \sum_{l\in L}\int_0^{\sum_{s\in R^l} x_s} p_l(y)dy, \qquad (2)$$

where p_l are penalty functions that act as proxy for the link capacity constraints in (1). Also, the $|R|-$dimensional vector x above is $(x_r, r\in R)^T$.

Studied in [1] for a simpler case, the continuous-time primal flow-control algorithm tunes the sending rate $x_r(t)$ using the ODE:

$$\begin{aligned} \dot{x}_r(t) &= \kappa_r(x_r)(U'_r(x_r(t)) - q_r(t)), \ \forall r \in R \\ q_r(t) &\stackrel{\Delta}{=} \sum_{l\in L_r} p_l(\sum_{s\in R^l} x_s(t)) \end{aligned} \qquad (3)$$

where $\kappa_r(x_r) > 0$ is a suitable gain and $L_r \subset L$ is the set of links that carry the traffic of r. The contribution to $q_r(t)$ from each link $l \in L_r$ can be interpreted as a price based on the aggregate flow of all sources $s \in R^l$ being carried over l, whilst $U'_r(x_r(t))$ above is the derivative of U_r at $x_r(t)$. Also, $q_r(t)$ is not an explicit function of t since it depends only on $x(t)$. The algorithm (3) is implemented in a distributed manner, since each source r only knows prices charged by links l carrying its traffic. Under intuitive conditions on p_l and U_r, there exists a globally asymptotically stable equilibrium point x^*_r of the above ODE. Due to the penalty function method (2), however, x^*_r is only an 'approximately fair' equilibrium.

Delays in link congestion indications are experienced due to network congestion or propagation time. The price demanded of r by link l could be $p_l(\sum_{s\in R^l} x_s(t - \xi^l_r(t) - \psi^l_s(t)))$ and not the current value $p_l(\sum_{s\in R^l} x_s(t))$, where $\xi^l_r(t)$ (resp. $\psi^l_s(t)$) is the $\mathcal{R}_+-$valued feedback (resp. feedforward) delay from link l to source r (resp. from source s to link l). Since propagation delays are assumed higher than queueing delays in large-scale networks, $\xi^l_r(t)$ and $\psi^l_s(t)$ can be held constant in t. Using this assumption, and for $U_r(x_r) = w_r \ln x_r$ and $\kappa_r(x_r) = \kappa_r x_r$ where $\kappa_r, w_r > 0$, the impact of such delays is analyzed in [1] to guarantee local stability alone of (3), and modifications for faster convergence to equilibrium x^*_r may compromise this stability. Past work does not distinguish between links l either, and defines the delay as $T_r \stackrel{\Delta}{=} \max_{l\in L_r}(\xi^l_r + \psi^l_r)$ (which is the RTT seen by source r).

Whether (3) is stable despite delays T_r, and whether this stability is achieved with only minimal restrictions on any of the parameters involved, are questions of enduring interest. Work in [2] assumes homogeneous sources, i.e. $T_r \equiv T$ and $U_r \equiv U$ for all $r \in R$, and a sufficient condition relating gain κ and T is proposed. Heterogeneous users are considered in [3], and a sufficient condition proposed where the product $\kappa_r T_r$ is bounded by a constant and the stability assured about x^*_r is local. This bound assumes a fixed RTT T_r while [4] suggests that such an assumption is not always reasonable. The work of [5] proposes a delay-independent stability criterion provided certain relations between p_l and U_r hold. These results characterize invariance, i.e. the continued evolution of (3) within a region, and stability (convergence to x^*_r). In particular, any family of p_l and U_r that imply a stable market equilibrium are also sufficient for global stability of the system (3), when affected by delays T_r.

For the algorithm that we propose in §II, the basic assumptions from [1], viz. *a)* U_r are positive, strictly concave and differentiable, $\forall r$ and *b)* link price functions $p_l(y)$ are non-negative, continuous and increasing in y are appended with

Assumption 1: Functions U'_r and link costs p_l are Lipschitz continuous.

B. Proposed Implementation

The algorithm (3) can be implemented in discrete-time, and three such implementations are identified in the subsection *Time Lags* of [1]. Stability is also analysed in such a framework, e.g. [2] investigates the stability of delayed difference equations arising from one such implementation. The implementations, however, are not strictly asynchronous since a global clock tick t is assumed to index all the recursions. Instead, we propose a controller at source r that uses a private, *local* index t_r:

$$x_r[t_r+1] := x_r[t_r] + \kappa_r[t_r](U_r'(x_r[t_r]) - q_r[t_r]),$$

followed by $t_r := t_r + 1$.

We explain the asynchronous nature of such a scheme, which is captured in t_r, where $t_r \in \mathcal{Z}_+$. Assume a counting process where for $t \in \mathcal{R}_+$, $t_r(t)$ indicates the number of updates made at source r until time t, the elapsed time between updates t_r and t_r+1 being an estimate of RTT T_r. The quantity T_r can be assumed fixed as in [3], [5] and [6] but we permit T_r to be varying. Similar to $t_r(t)$, every link l is seen as having made $t^l(t)$ measurements of the (time-varying) aggregate flow $\sum_{s\in R^l} x_s(t)$ over it. These link measurements could also be update schemes, for example, exponential moving averages in RED. In §III-B, we provide an example of a link update scheme. Random variable ξ_r^l, which differs in interpretation from its $\mathcal{R}_+-$valued analog in continuous-time, represents the feedback delay. Abusing the notation for p_l, the price demanded of source r by link l is the $(t_r - \xi_r^l)-$th iterate $p_l[t_r - \xi_r^l]$. Thus, ξ_r^l takes values in the entire integer set $\mathcal{Z}$.

We also consider noise ϵ_r^l as an additive stochastic effect in the congestion indication scheme, making $p_l[t_r-\xi_r^l]+\epsilon_r^l$ as the price demanded at the t_r-th update of r. For instance, in [6] $p_l[t_r - \xi_r^l] + \epsilon_r^l$ is the number of Explicit Congestion Notification (ECN) marks $M_r[t_r]$ observed in the current epoch t_r of source r. In [1], a congestion notification scenario is considered where links l send a Poisson number of indications to sources r at the rate $p_l(\cdot)$ and therefore $\frac{M_r[t_r]}{T_r}$ is an RV with mean $p_l[t - \xi_r^l]$. TCP congestion controllers (in their more general form as minimum potential delay controllers) are modelled in [6] in like manner. If considered time-invariant, $\kappa_r[t_r] \equiv \kappa_r$ plays a role in characterizing stability about the equilibrium point x^* (cf. [3] and [7, §4]). Instead, in §II, we propose a $\kappa_r[t_r]$ that is an Ideal Tapering Stepsize (ITS) of [8].

In §III, we pursue enhancements to the proposed implementation of §II. Since (3) can be viewed as gradient ascent along (2), §III-A proposes using the second-order information via an efficient variant of the gradient-descent algorithm Simultaneous Perturbation Stochastic Approximation (SPSA) proposed in [9]. The algorithm proposed in §III-B considers the problem in [10] where a misbehaving flow r must be rectified by a router $\hat{r}$ placed at the edge of the network using suitably rigged costs $\hat{q}_r$. In both enhancements, the method of ASA is retained and we outline a proof of convergence.

In §IV we propose similar discrete-time implementations of the dual and primal-dual algorithms (cf. [7, §2]). In particular, the primal-dual algorithm is implemented using a two-timescale ASA method adapted from the constrained Markov Decision Process algorithm of [11]. A regime of step-sizes (known as the two-timescale condition) at links and sources permits simultaneous updates towards the respective optima.

In the numerical results of §V we perform two experiments. First, we verify the convergence of all the proposed algorithms in a single link-multiple nodes setting with simulated packet arrivals and departures. The rate allocations to which the proposed algorithms converge are only negligibly away from x^*. We then consider a system from [5] where the corresponding delayed-differential equation model of the system was shown to be unstable. But when flow-control is implemented using the proposed algorithm, convergence w.p. 1 is assured.

II. Primal Algorithm using ASA

It is necessary to motivate the use of ASA for at least the simple case of fixed, but heterogeneous, RTTs T_r. The proposed algorithm can, however, handle restrictions like varying RTTs T_r. Suppose that $t \in \mathcal{R}_+$ is the update instant of source r (i.e. $t = n_r T_r$ for some $n_r \in \mathcal{Z}_+$). Then, $t_s(t) \neq t_r(t)$, in general, due to the differing T_rs. Moreover, the values of x_s seen by the controller of source r via p_l, $l \in L_r$ need not be $x_s[t_s(t)]$ (i.e. the most recent value of x_s) and due to propagation delays, could be some past value of x_s. In terms of the *local* indices of s, this iterate may be $x_s[t_r - \xi_r^l - \psi_{s,r}^l]$ where both $\xi_r^l, \psi_{s,r}^l \in \mathcal{Z}$.

In the proposed algorithm, we require that ITS $\kappa_r[t_r]$ be *identical* for all r (i.e. $\kappa_r[t_r] \equiv \kappa[t_r]$). Thus we have the recursion:

$$x_r[t_r+1] := x_r[t_r] + \kappa[t_r](U_r'(x_r[t_r]) - q_r[t_r]) \quad (4)$$
$$q_r[t_r] := \sum_{l\in L_r} \left(p_l[t_r - \xi_r^l] + \epsilon_r^l[t_r]\right),$$

with index update $t_r := t_r + 1$. Here, the iterate $q_r[t_r]$ estimates the charge payable by source r with stochastic effects $\epsilon_r^l[t_r]$ being noise in the estimate. The ITS conditions on the stepsize κ, for all sources r, are:

$$\kappa[t_r] > 0 \ \forall t_r \quad , \quad \sum_{t_r} \kappa[t_r] = \infty,$$
$$\sum_{t_r} \kappa^{1+q}[t_r] < \infty \qquad \text{for some } q \in (0,1). \quad (5)$$

The proposed ITS nature of $\kappa[t_r]$ is an issue, since conventional flow control algorithms assume stationary behaviour independent of index t_r and we have not yet come across any algorithms using diminishing stepsizes. However, [7, §2], for instance, assumes that κ is a function of rate $x_r(t)$. The use of ITS is crucial to apply the theory of ASA in the current setting.

We introduce a notational simplification in the form of a global event-index n that denotes events: whether these be measurements of aggregate flow by links $l \in L$ or rate updates by sources $r \in R$. For every $t \in \mathcal{R}_+$, $n(t)$ will indicate the counting process of all events. We use parentheses $(\cdot)$ to identify a variable according to this global index whilst square brackets $[\cdot]$ will denote 'with respect to local indices' t_r. For each value of $n \geq 0$ there are corresponding local indices $t_r \stackrel{\Delta}{=} t_r(n)$ and $t^l \stackrel{\Delta}{=} t^l(n)$ such that $t_r, t^l \leq n$ and $\sum_{r\in R, l\in L} t_r + t^l = n$. The global index n thus can be constructed from local indices t_r and t^l. Also, if $x_r(n) \neq x_r(n-1)$ (implying that an update of x_r has taken place at $n-1$), then $x_r(n+k) = x_r(n)$, $\forall k : t_r(n+k) = t_r(n)$. Analogously define the process $p^l(n)$ for links l. Also note that neither links l nor sources r need to know this global clock n.

A. Convergence Analysis

For ease of analysis, assume that at a given instant t only one event of measurement or update occurs. This can be accomplished using a technique to unfold the recursions proposed in [8, §3]. These can, however, be multiple updates at any instant. For each source r, define two vectors $\bar{x}_r(n) = (x^l_{s,r}(n), \forall l \in L_r, \forall s \in R^l)^T$ and $\epsilon_r(n) = (\epsilon^l_r(n), \forall l \in L_r)^T$. Here, $x^l_{s,r}(n) = x_s(n - \xi^l_r - \psi^l_s)$ with $\mathcal{Z}_+-$valued random variables ξ^l_r, ψ^l_s being delays in terms of n. Though we re-use symbols ξ and ψ denoting delays, the parenthesis $(\cdot)$ makes the context clear. The total delay in receiving x_s by source r, via link l, is given by $\tau^l_{s,r} \stackrel{\Delta}{=} \xi^l_r + \psi^l_s$. Similarly, $\epsilon^l_r(n)$ is the noise seen at instant n in measurement p_l communicated by link l to source r. Thus, the update can be written as:

$$
\begin{aligned}
x_r(n+1) \quad := \quad & x_r(n) + \kappa(n,r) \cdot \\
& F_r(\bar{x}_r(n), x_r(n), \epsilon_r(n)) I_{\{\varphi_n = r\}},
\end{aligned}
$$

where $\{\varphi_n = r\}$ corresponds to the event that x_r is updated at n and $\sum_{k=1}^{n} I_{\{\varphi_k = r\}} = t_r$. Further, $\kappa(n,r) = \kappa[t_r]$ and F_r is the 'reinforcement' term $(U'_r(x_r[t_r]) - q_r[t_r])$ in the recursion (4). It is possible that $\varphi_n = \phi$, the empty set, if n corresponds to a link measurement instant.

The analysis follows the pattern and notation of [8]: [8, Lemma 3.3] is modified as Lemma 1 to accommodate the delays $\tau^l_{s,r}$. Treatment in [8] handles only a single delay $\tau_{s,r}$ per pair of components (s,r) whereas here, per source r we have $\sum_{l \in L_r} |R^l|$ such delays including 'self-delays' of the form $\tau^l_{r,r}$. We verify assumptions (A1)-(A6) of [8] and modify where necessary to accommodate $\tau^l_{s,r}$. Assumption (A1) follows from the ITS property of $\kappa[t_r]$. Further, (A2) holds due to $|R| < \infty$ and the upper-bound on Retransmission Time-out (RTO) (due to which component x_r is updated regularly). We define the sigma-algebra $\mathcal{F}_n \stackrel{\Delta}{=} \sigma(x_r(\tilde{m}), \tau^l_{s,r}(m), \epsilon^l_r(m))$, $\forall \tilde{m} \leq n, \forall m < n, \forall r \in R$, $l \in L_r$, and $s \in R^l$ in order to rephrase (A3):

Assumption 2: $\tau^l_{s,r}(n) \in \{0, 1, ..., n\}$, $\forall r \in R$, $l \in L_r$, $s \in R^l$ and $\exists b, C > 0$ s.t. $E((\tau^l_{s,r}(n))^b | \mathcal{F}_n) \leq C$ a.s.

It is sufficient to have $\tau^l_{s,r} < \infty$, w.p.1, to satisfy this assumption. Assumption 1 made above results in (A4) being satisfied. The next assumption is on the stochastic effects $\epsilon^l_r(n)$, we first define:

$$
f_r(\bar{x}_r(n), x_r(n)) = \int F_r(\bar{x}_r(n), x_r(n), \epsilon_r) P_r(d\epsilon_r).
$$

where $P_r(d\epsilon_r)$ is the law according to which ϵ_r is distributed and integration is over the positive orthant $\mathcal{R}_+^{|L_r|}$. Note that P_r may, in general, depend upon $\bar{x}_r(n)$ although what we require is:

Assumption 3: If $\tau^l_{s,r} = 0$, $\forall r \in R$, $l \in L_r$, $s \in R^l$ then $f_r(\bar{x}_r(n), x_r(n)) = U'_r(x_r(n)) - q_r(n)$.

Mean-zero $\epsilon^l_r(n)$ suffices to satisfy this assumption, and the Poisson congestion indication of [1] is an example. We abuse notation to consider the function $f_r(x(n)) = f_r(\bar{x}_r(n), x_r(n))$ in the absence of delays. Then, the system of ODEs asymptotically tracked by recursion (4) is $\dot{x}_r(t) = f_r(x(t))$, $\forall r \in R$. Thus (A5) is satisfied since the set J of equilibrium points required by (A5) contains the single element x^*. Further, (A6) is satisfied by the strict Liapunov function $V(x)$ defined in (2).

Let $\hat{F}(n) = E(F(n)|\mathcal{F}_n)$ and $f^s : \mathcal{R}^{|R| \times |R| \times |L|} \mapsto \mathcal{R}^{|R|}$ be a function which we describe as follows. Given $s \in R$, let $\tilde{x}_s(n)$ be an element of $\mathcal{R}^{|R| \times |L|}$ with components $(\tilde{x}_s(n))_{r,l} = x^l_{r,s}(n)$ if $r, s \in R^l$ and all other components zero. Consider $\hat{x}(n) \stackrel{\Delta}{=} ((x_s(n), \tilde{x}_s(n)), s \in R)^T$ and thus f^s is such that $f^s_r(\hat{x}(n)) = f_s(\bar{x}_s(n), x_s(n)) \delta_{s,r}$ for $r, s \in R$ where $\delta_{s,r}$ is the Kronecker delta function. Let the analogous $\bar{f}^s$ be defined when delays $\tau^l_{s,r}$ are zero $\forall r, s$. The stepsize $b(n) \stackrel{\Delta}{=} \max_r \kappa(n,r)$ for $n \geq 0$.

Lemma 1: For $\varphi_n \neq \phi$, almost surely, $\exists K_1 > 0$, and a random $N \geq 1$ s.t. for $n \geq N$:

$$
\|\bar{f}^{\varphi_n}(\hat{x}(n)) - \hat{F}(n)\| \quad \leq \quad K_1 b^q(n)
$$

Proof: We take K_2 as the upper bound on $\{\|f(x)\|_\infty\}$. Let $\tilde{F}_r = f_r^{\varphi_n}(\hat{x}(n))$, and $c = 1 - q$ (for q from (5) above). Modify (3.5) of [8] as follows: $|\bar{f}_r^{\varphi_n}(\hat{x}(n)) - \tilde{F}_r(n)|$

$$
\leq E[|\bar{f}_r^{\varphi_n}(\hat{x}(n)) - \tilde{F}_r(n)| I_{\{\forall l,s,r: \tau^l_{s,r}(n) \leq b^{-c}(n)\}} | \mathcal{F}_n]
$$
$$
+ E[|\bar{f}_r^{\varphi_n}(\hat{x}(n)) - \tilde{F}_r(n)| I_{\{\exists l,s,r: \tau^l_{s,r}(n) > b^{-c}(n)\}} | \mathcal{F}_n].
$$

By Assumption 2 (the b and C from which we use) and the modified Chebyshev inequality, the second term is a.s. bounded by $2K_2 C |R|^2 |L| b^{bc}(n)$. The remainder of the proof now follows as in Lemma 3.3 of [8]. In passing, we mention that the assumption $\tau_{r,r} = 0$ made in (A3) of [8], although intuitive, is not required. □

Theorem 1: Over trajectories where $\|x(n)\|_\infty < \infty$, $\forall n$, the algorithm (4) converges to x^*_r a.s..

Proof: We verify two conditions in the statement of Theorem 3.1 (a) in [8]. The first condition is to check if $\exists a > 0$ s.t. $\dot{x}(t) = f(x(t))$ is an $a-$robust system. To see this, we note that the strict Liapunov property of $V(x)$ and the distributed form of (3) ensures that $\bigtriangledown_r V(x) \cdot f_r(x) < 0$ for any point $x \neq x^*$. Thus, for *any* $a > 0$, $\bigtriangledown_r V(x) \cdot a \cdot f_r(x) < 0$, implying the system is $a-$robust.

Next, construct a sequence t_n where $t_n \stackrel{\Delta}{=} \sum_{k=0}^{n} \kappa(k, \varphi_k)$. For a given t, $\bar{\mu}^t(s)$, $s > 0$ is the Dirac measure φ_n for $s \in [t + t_n, t + t_{n+1})$. Further, $a-$thickness of the continuous-time process $\bar{\mu}^t$ is defined as $\bar{\mu}^t_r(s) > a$, $\forall r \in R$. We refer to [8, §3] for detailed definitions. The second condition of the theorem is to check that all limit points of $\bar{\mu}^t$ in U are $a-$thick a.s.. The limit points of $\bar{\mu}^t$ correspond to the update frequencies of the sources r at time $t \in \mathcal{R}^+$. Upper bounds on RTO mentioned earlier along with bounded delays $\tau^l_{s,r}$ ensure that as $t \to \infty$, $\bar{\mu}^t(r) > a$, $\forall r \in R$ for some $a > 0$. Thus limit points of $\bar{\mu}^t$ are $a-$thick, and the claim follows. □

The statement of the theorem is qualified, in that it assumes bounded trajectories $x_r(n)$. This can be implemented by clipping the iterates $x_r(n)$ against an interval $C_r \stackrel{\Delta}{=} [x_{r,\text{min}}, x_{r,\text{max}}]$ such that x^*_r is contained in this interval, a reasonable guess being $C_r = [0, \min_{l \in L_r} c_l]$. Consider $C = \Pi_{r \in R} C_r$, then the projected ODE that results will nevertheless be asymptotically stable. This is because either the asymptotically stable equilibrium point is contained within C or if it lies outside of C, the ODE will get trapped at a boundary point of C (thus introducing spurious fixed points).

III. ENHANCEMENTS

A. Second-order Primal Algorithm

The primal algorithm can be interpreted as a gradient ascent along the Liapunov function $V(x)$ of (2) above. Therefore, the diagonal elements of the Hessian matrix of $V(x)$ can also be computed in a distributed manner: $\bigtriangledown^2_{r,r} V(x) = U_r^{''}(x_r) - \sum_{l\in L_r} \bigtriangledown_r p_l(\sum_{s\in R^l} x_s)$. Apart from U_r, also assume that $p_l(\cdot)$ are C^2 (i.e. twice continuously differentiable). This is a pre-requisite since the gradient terms $\bigtriangledown_r p_l(\sum_{s\in R^l} x_s)$ are estimated using an efficient variant of SPSA proposed in [9]. Though $\bigtriangledown_r p_l(\sum_{s\in R^l} x_s)$ is of identical value $\forall r \in R^l$, each source needs to compute its own estimate.

This estimate is constructed by measuring p_l such that the rate of source r is not $x_r[t_r]$ but a perturbed value $x_r^+[t_r] \stackrel{\triangle}{=} x_r[t_r] + \delta_r[t_r]\Delta_r[t_r]$. The scalar δ_r is a positive, diminishing step-size with other properties explained later, while perturbations $\Delta_r[t_r]$ are independent, $\pm 1-$valued with probability 0.5. Both δ_r and Δ_r are the local information of source r. A crucial departure from SPSA-based gradient search occurs due to the asynchronous nature of the update scheme. The measurement $p_l(n-\xi_r^l)$ seen at update t_r does not reflect the rate $x_r^+[t_r-1]$ set at the last update $t_r - 1$.

Thus an asynchronous variant of SPSA is proposed here. This algorithm would also be of independent interest in the SPSA framework. Assume that current measurement of p_l has a contribution from $x_r^+[t_r - k_r^l]$ and that $k_r^l \le K_r, \forall l \in L_r$ with a non-zero probability. In the absence of precise methods to determine K_r, it can be taken as 1 if time between updates at r is distributed closely about the fixed RTT T_r. At each source r, we maintain an array of size K_r and store $\{\Delta_r[t_r-1], \Delta_r[t_r-2], ..., \Delta_r[t_r-K_r]\}$ while $\delta_r[t_r-k]$ can be computed due to its closed form.

Since at t_r the offset k_r^l is unknown, we pick $\Delta_r[t_r - k_r^l]$, $1 \le k_r^l \le K_r$ with probability $\frac{1}{K_r}$ and use $\delta_r[t_r - k_r^l]\Delta_r[t_r - k_r^l]$ in a manner described in the following. Let $\bar{x}_{l,r}(n)$ be the $|R^l|-$sized vector $(x_s(n-\tau_{s,r}^l), \forall s \in R^l)^T$ and similarly, let $\bar{\delta}_{l,r}(n)$ be $(\bar{\delta}_{l,r}(n,s) \stackrel{\triangle}{=} \delta_s(n-\tau_{s,r}^l)\Delta_s(n-\tau_{s,r}^l), \forall s \in R^l)^T$. Now we define $\bar{x}_{l,r}^+(n)$ as $\bar{x}_{l,r}(n) + \bar{\delta}_{l,r}(n)$. We choose from the array of Δ, and get a correspondence $\delta_r[t_r - k_r^l]\Delta_r[t_r - k_r^l] \equiv \delta_r(n-\tau_r^l)\Delta_r(n-\tau_r^l)$ where $\tau_r^l \in \mathcal{Z}_+$: this we abbreviate as $\tilde{\delta}_{l,r}(n)$. We also abbreviate the noisy measurement $p_l(\bar{x}_{l,r}^+(n)) + \epsilon_r^l(n)$ as $\tilde{p}_l(\bar{x}_{l,r}^+(n))$. Notation p_l is used loosely here, since p_l depends on $\bar{x}_{l,r}^+(n)$ through the sum $\sum_{s\in R^l} x_s(n-\tau_{s,r}^l)$.

Using a Taylor series expansion of $p_l(\bar{x}_{l,r}^+(n))$ we have:

$$\frac{\tilde{p}_l(\bar{x}_{l,r}^+(n))}{\tilde{\delta}_{l,r}(n)} = \frac{p_l(\bar{x}_{l,r}(n)) + \epsilon_r^l(n)}{\tilde{\delta}_{l,r}(n)} + \sum_{s\in R^l} \frac{\bigtriangledown_s p_l(\bar{x}_{l,r}(n))\bar{\delta}_{l,r}(n,s)}{\tilde{\delta}_{l,r}(n)} \quad (6)$$
$$+ \frac{\bar{\delta}_{l,r}^T(n) H_l(\bar{x}_{l,r}(n))\bar{\delta}_{l,r}(n)}{2\tilde{\delta}_{l,r}(n)} + O(\tilde{\delta}_{l,r}^2(n)), \quad (7)$$

where H_l corresponds to the Hessian of p_l. With probability $\frac{1}{K_r}$ our guess of $\tilde{\delta}_{l,r}(n)$ will result in (6) becoming:

$$\bigtriangledown_r p_l(\bar{x}_{l,r}(n)) + \sum_{s\neq r, s\in R^l} \frac{\bigtriangledown_s p_l(\bar{x}_{l,r}(n))\bar{\delta}_{l,r}(n,s)}{\tilde{\delta}_{l,r}(n)}.$$

The expected value of the second term w.r.t. Δ is 0. Further, the terms involving $\tilde{\delta}_{l,r}(n)$ in (6)-(7) are also mean-0 when the guess $\tilde{\delta}_{l,r}(n)$ is wrong. Thus, we conclude that:

$$E\left(\frac{\tilde{p}_l(\bar{x}_{l,r}^+(n))}{\tilde{\delta}_{l,r}(n)} | \bar{x}_{l,r}(n)\right) = \frac{1}{K_r}\bigtriangledown_r p_l(\bar{x}_{l,r}(n)) + O(\tilde{\delta}_{l,r}^2(n)).$$

This estimate, however, corresponds to a delayed measurement and proving convergence would further hinge on the ASA framework of [8] as applied in §II-A. The choice $\tilde{\delta}_{l,r}(n)$ need not be separate for each $l \in L_r$ and a single $\tilde{\delta}_r(n)$ suffices. These estimates are iterated for better averaging according to the following recursion:

$$h_r(n+1) := h_r(n) + \alpha(n,r)\cdot \left(\sum_{l\in L_r} \frac{\tilde{p}_l(\bar{x}_{l,r}^+(n))}{\tilde{\delta}_r(n)} - h_r(n)\right) I_{\{\varphi_n = r\}}, \quad (8)$$

with the properties: $\alpha(n,r) = \alpha[t_r]$, $\sum_k \alpha[k] = \infty$, $\sum_k \alpha^2[k] < \infty$, $\kappa[k] = o(\alpha[k])$ as $k\to\infty$. This last property is the characteristic property of two-timescale stochastic approximation. Further, the ratio of $\alpha[k]$ with $\delta_r[k]$ must be square-summable over k. As the selection of K_r may be imprecise, estimates $h_r(n)$ of (8) can be truncated against $[m, M]$ for some $M > m > 0$ so as to avoid arbitrarily small values.

The discrete-time implementation is now:

$$x_r(n+1) := x_r(n) + \kappa(n,r) h_r^{-1}(n)\cdot \left(U_r'(x_r(n)) - \sum_{l\in L_r} \tilde{p}_l(\bar{x}_{l,r}^+(n))\right) I_{\{\varphi_n = r\}}. \quad (9)$$

We show that using $\sum_{l\in L_r} \tilde{p}_l(\bar{x}_{l,r}^+(n))$ above in place of $\sum_{l\in L_r} \tilde{p}_l(\bar{x}_{l,r}(n))$ in (4) does not impact the convergence of the algorithm. Again, a Taylor series expansion of $\tilde{p}_l(\bar{x}_{l,r}^+(n))$ results in $\tilde{p}_l(\bar{x}_{l,r}(n)) + O(\delta_r^2(n-\tau_r^l))$. The latter bias term arises from $\frac{1}{2}\bar{\delta}_{l,r}^T(n) H_l(\bar{x}_{l,r}(n))\bar{\delta}_{l,r}(n)$ and cancels due to square-summability of the ratio of $\kappa[t_r]$ and $\delta_r[t_r]$. Once the t_r-th update is performed according to (8)-(9), the index t_r is increased by one, source r computes $\delta_r[t_r]$ and generates $\Delta_r[t_r]$ to send at rate $x_r^+(n+1) = x_r(n+1) + \delta_r[t_r]\Delta_r[t_r]$ until the next update.

We now efficiently implement (8) using the method of [9]. Define $\tau_r \in \mathcal{Z}_+$ as $\tau_r = \min_\tau I_{\{\varphi_{n-\tau}=r\}} = 1$. The past measurements of prices at $|L_r|$ links, $\tilde{p}_l(\bar{x}_{l,r}^+(n-\tau_r))$ are stored at r, with $|L_r|$ units of storage being needed for this. Then replacing $\tilde{p}_l(\bar{x}_{l,r}^+(n))$ in (8) with $\tilde{p}_l(\bar{x}_{l,r}^+(n)) - \tilde{p}_l(\bar{x}_{l,r}^+(n-\tau_r))$ results in improved performance of the algorithm, as analyzed in [9].

B. Rectification of misbehaving flows

Recently, [10] considered a misbehaving source of announced-utility $U_r(x_r)$ surreptitiously using $\hat{U}_r(x_r)$ for rate updates. An edge-router $\hat{r}$ was proposed to police r subject to certain information constraints. This $\hat{r}$ knows only current source flow-rate x_r, declared utility U_r and price $q_r(n)$ and uses these to produce a *reference* rate $\hat{x}_r$. Lacking direct control over r's sending rate, $\hat{r}$ computes

$\hat{x}_r$ to demand a price $\hat{q}_r(n)$ from r, in place of $q_r(n)$ that r assumes would suffice. The convergence (cf. [10]) is to the (approximate) proportionally-fair equilibrium x_r^* and not some optima desired by the malicious user. It is even possible for r to destabilize the network, by using a $\hat{U}_r$ that thwarts the convergence of (3).

To illustrate the proposed algorithm, we consider the following synchronous recursions. For the present, take φ_n to be set-valued in $R \cup L$ and assume that r and $\hat{r}$ both perform updates at the same indices n, i.e., $I_{\{r \in \varphi_n\}} = I_{\{\hat{r} \in \varphi_n\}}$. Their corresponding recursions are:

$$\begin{aligned} x_r(n+1) &= x_r(n) + \kappa(n,r)(\hat{U}_r'(x_r(n)) - \hat{q}_r(n)) \\ \hat{x}_r(n+1) &= x_r(n) + \kappa(n,r)(U_r'(x_r(n)) - \hat{q}_r(n)). \end{aligned}$$

It is now easy to penalize source r, since, subtracting the two equations, the quantity

$$\delta q_r(n+1) \quad \triangleq \quad \frac{x_r(n+1) - \hat{x}_r(n+1)}{\kappa(n,r)}$$

is the extra allocation gained by source r. Thus the price demanded of source r from epoch $n+1$ onwards till the next update is $\hat{q}_r(n+1) := q_r(n+1) + \delta q_r(n+1)$, composed of the cost for the present rate and the penalty for misrepresenting utility in the last epoch. However, complications arise when r and $\hat{r}$ update asynchronously. Suppose there is an n s.t. $I_{\{\varphi_n = \hat{r}\}} = 1$ and the local index is $t_{\hat{r}}$. Then, $\hat{x}_r(n) \triangleq x_r[t_{\hat{r}}]$ is available from the last update whereas the rate at which r is sending currently is $x_r(n)$.

The edge-router $\hat{r}$ takes the view that source r has changed its transmission rate from $\hat{x}_r(n)$ to $x_r(n)$ in a *single* update, and that r also has the same update frequency as itself. Thus, $\hat{r}$ assumes that at r the local index n_r is $n_{\hat{r}} + 1$. It assumes that the following recursion has taken place at source r:

$$x_r(n) \quad := \quad \hat{x}_r(n) + \kappa(n,\hat{r})(\hat{U}_r'(\hat{x}_r(n)) - \tilde{q}_r(n)),$$

where $\tilde{q}_r(n)$ is the modified cost payable by r computed (and stored) by $\hat{r}$ at local index $t_{\hat{r}} - 1$. Further, $\hat{r}$ computes the one-time reference quantity $\tilde{x}_r$:

$$\tilde{x}_r \quad := \quad \hat{x}_r(n) + \kappa(n,\hat{r})(U_r'(\hat{x}_r(n)) - \tilde{q}_r(n)).$$

Thus, an approximation to $\hat{U}_r'(\hat{x}_r(n)) - U_r'(\hat{x}_r(n))$ would be $\frac{x_r(n) - \tilde{x}_r}{\kappa(n,\hat{r})}$ and therefore

$$\begin{aligned} \delta q_r(n+1) &:= \frac{x_r(n) - \tilde{x}_r}{\kappa(n,\hat{r})}, \\ \tilde{q}_r(n+1) &:= q_r(n) + \delta q_r(n+1), \\ \hat{x}_r(n+1) &:= x_r(n). \end{aligned}$$

To see how the convergence analysis proceeds, we take a large enough n such that $I_{\{\varphi_n = r\}} = 1$ and assume that all other delays of the form $\tau_{s,r}^l$ are zero. Now suppose $n - \tau_r^1$ was the last instant at which $\hat{r}$ performed an update, and computed $\delta q_r(n - \tau_r^1 + 1)$ such that the current price is $\hat{q}_r(n) = q_r(n) + \delta q_r(n - \tau_r^1 + 1)$. Also assume that $n - \tau_r^1 - \tau_r^2$ was the previous update at $\hat{r}$ and define $\tau_r = \tau_r^1 + \tau_r^2$. The update at $n - \tau_r^1$ uses the older $x_r(n - \tau_r)$ as reference value $\hat{x}_r(n - \tau_r^1)$. Thus, the recursion at r is: $x_r(n+1) := x_r(n) + \kappa(n,r)(\hat{U}_r(x_r(n)) - \hat{U}_r(x_r(n-\tau_r)) + U_r(x_r(n-\tau_r)) - q_r(n))$. We again have a case where varying delays affect the same component of the recursion, viz. x_r is affected by delays 0 and τ_r. The analysis now follows largely on the same lines as that in §II-A.

We note here that the ASA framework requires the edge-router to adjust the costs $\hat{x}_r$ only a minimal number of times. It suffices for convergence if sampling of all sources occurs with non-zero relative frequencies.

IV. Dual and Primal-Dual Algorithms

We explain the continuous-time Kelly dual algorithm briefly. The source r infers its sending rate from the price signal $q_r(t)$ as $x_r(t) = (U_r')^{-1}(q_r(t))$. At link l, a first-order dynamic price update follows the ODE $\dot{p}_l(t) = \alpha_l(y_l(t) - c_l(p_l(t)))$ where $\alpha_l > 0$ is a scale factor, $y_l(t)$ is the flow $\sum_{s \in R^l} x_s(t)$ through l, while $c_l(p_l(t))$ is the flow at l for which $p_l(t)$ is payable. The corresponding ASA algorithm again uses an ITS $\alpha[t^l]$:

$$p_l[t^l+1] \quad := \quad p_l[t^l] + \alpha[t^l](y_l[t^l] - c_l[t^l])_{p_l}^+, \forall l \in L,$$

where measurement $y_l[t^l]$ is $\sum_{s \in R^l} x_s[t_s - \psi_s^l]$. Operator $(y_l[t^l] - c_l[t^l])_{p_l}^+$ signifies truncation to 0 if both $y_l[t^l] - c_l[t^l] < 0$ and $p_l[t^l] \leq 0$ (cf. [10]).

The proof techniques of §II hold. The Lipschitz continuity of the term $y_l[t^l] - c_l[t^l]$ which corresponds to $\sum_{s: l \in L_s} {U'}_s^{-1}(\sum_{l \in L_s} p_l[t_r - \xi_r^l]) - c_l[t^l]$ must be satisfied. Thus, we assume ${U'}_s^{-1}$ and $c_l(\cdot)$ are Lipschitz. The unique, globally stable equilibrium p_l^* is reached as $t \to \infty$ and has the property $q_r^* = \sum_{l \in L_r} p_l^* = U_r'(x_r^*)$.

Next we propose an ASA algorithm to solve the primal-dual problem. The inequality constraints of the network-utility maximization problem (1) can be folded-in as follows:

$$\min_{p_l > 0} \max_{x_r \geq 0} \sum_{r=1}^{|R|} U_r(x_r) - \sum_{l=1}^{|L|} p_l(\sum_{s: l \in L_s} x_s - c_l),$$

where the latter term involves a multiplication with the Lagrange multipliers p_l (not the *functions* $p_l(\cdot)$ used previously). The slower timescale recursion is performed at the links l with stepsizes $\alpha[t^l]$ whilst the faster is at sources r with stepsizes $\kappa_r[t_r]$

$$\begin{aligned} x_r[t_r+1] &= x_r[t_r] + \kappa_r[t_r]\left(U_r'(x_r[t]) - q_r[t_r]\right) \\ q_r[t] &= \sum_{l \in L_r} p_l[t_r - \xi_r^l] + \epsilon_r^l[t_r] \\ p_l[t^l+1] &= p_l[t^l] + \alpha[t^l](y_l[t^l] - c_l)_{p_l}^+. \end{aligned}$$

The conditions on the stepsizes, $\forall r \in R$ and $l \in L$, are: $\kappa[t_r], \alpha[t^l] > 0$,

$$\begin{aligned} \sum_{t_r} \kappa[t_r] &= \sum_{t^l} \alpha[t^l] = \infty, \\ \sum_{t_r} (\kappa[t_r])^2 &, \quad \sum_{t^l} (\alpha[t^l])^2 < \infty, \end{aligned}$$

and, the characteristic two-timescale condition:

$$\alpha[t^l] \quad = \quad o(\kappa[t^l]), \text{as } t^l \to \infty, \forall l \in L_r$$

V. Simulation Results

We conduct two sets of experiments with the proposed algorithms. First, we consider a simple setting of a single link into which four sources feed traffic. These sources $r = \{1,2,3,4\}$ send Poisson streams of packets at rates x_r. The value $y_1 = \sum_{r=1}^{4} x_r$ is estimated at the end of every slot of 5.0 seconds by the link by dividing the number of packets arrived in the slot with

Algorithm	$E(\hat{n})$	$\sigma_{\hat{n}}$	$\|\cdot\|_1-$error
Fixed Delay	7365	580	0.070
Noisy Delay	7234	726	0.072
Second-Order	4128	972	0.214
Rectification	7568	625	0.071

TABLE I: Performance of Primal Algorithm and variants

Algorithm	$E(\hat{n})$	$\sigma_{\hat{n}}$	$\|\cdot\|-$error
Dual	4780	506	0.101
Primal-Dual	961	153	0.102

TABLE II: Performance of Dual and Primal-Dual Algorithms

5.0. This aggregate measurement y_1 is then intimated to all sources r via the cost-function $p_1(y_1) = y_1^{0.8}$. The sources, however, update only at intervals of 1, 2, 3 and 6 slots, respectively. Therefore, the global index n now corresponds to the most frequent of all updates/measurements, viz. that of link 1 and source 1.

We choose the utility function $U_r(x) = w_r \ln x_r$ (from [1]) where $w_r = 1$, 2, 3, and 6, respectively. First, fixed delays were considered where both feedforward $\psi_r^1(n)$ and feedback delays $\xi_r^1(n)$ were set to 1, 2, 2 and 4, respectively, ensuring that no source r performs an update using the most recent measurements. Next, variable delays were considered as $\tilde{\psi}_r^1(n) = \psi_r^1(n) - P_r^1(n)$ where the random variable $P_r^1(n) \in \{0, 1\}$ with probability 0.5 each. Also, feedback delays $\tilde{\xi}_r^1(n)$ are defined analogously. This delay structure also helps in the experiment with second-order enhancement of §III-A since a choice of $K_r = 1, \forall r$ is valid. We also truncate against $[0.1, 10.0]$ the iterates $h_r(n)$ of (8). Further, to experiment with the rectification algorithm of §III-B we took $\hat{w}_1 = 2.0$, where source 1 has misrepresented its utility $\hat{U}_1(x_1) = \hat{w}_1 \ln x_1$. The edge-router that computes the modified price $\hat{q}_1(n)$ does so every 5 slots, as compared to source 1 which updates rate $x_1(n)$ in every slot n.

Our findings are tabulated in Table I. The algorithms were terminated at an $\hat{n}$ where $\max_{k\in\{1,2,\ldots,100\}} \|x(\hat{n}-k) - x(\hat{n})\|_1 \le 10^{-3}$. We used the stepsizes, $\kappa[t_r] = t_r^{-1}$ and $\alpha[t_r] = t_r^{-0.55}$ (applicable to the second-order algorithm). The optimum vector x^* is $(0.33, 0.66, 0.99, 1.98)^T$, and the error $\|x(\hat{n}) - x^*\|_1$ w.r.t. x^* is also recorded. The term $\sigma_{\hat{n}}$ denotes the standard deviation of $\hat{n}$ from its mean $E(\hat{n})$ over 100 runs of the algorithm.

We verified the dual and primal-dual algorithms proposed in §IV. The dual algorithm divided the bandwidth $C_1 = 5.0$ packets per second of the single link among the sources r. While the utility functions U_r were the same as above, note that the price charged by the link is always linear in the aggregate flow y_1. We chose stepsize $\alpha^1[t^l] = (t^l)^{-1}$. In the primal-dual algorithm, the same task is accomplished faster. For any system, once the regime $\alpha^l[k] = o(\kappa_r[k])$ is in place, the primal and dual algorithms at sources and links, respectively, can execute in tandem. A source r, for example, is unaffected by (and is also unaware of) an algorithm that changes the price-function at some link $l \in L_r$.

In the second set of experiments we chose the unstable system in §IX.A.2 of [5]. In this system, three sources $r \in R \equiv \{1, 2, 3\}$ use two links l_1, and l_2 such that $R^1 = \{1, 3\}$ and $R^2 = \{2, 3\}$. The utility functions are $U_r(x_r) = \frac{-1}{a_r}\frac{1}{x_r^{a_r}}$ (resulting in $U_r'(x_r) = x_r^{-(a_r+1)}$) with $a_1, a_2 = 3$ and $a_3 = 4$. Links charge using the function $p_l(y) = \left(\frac{y}{C_l}\right)^{b_l}$, $b_l = 3.5$ for $l = 1, 2$ where C_l are the link capacities at 5 and 4, respectively. Using the analysis of [5], feedback delays $(\xi_1^1, \xi_3^1)^T = (280, 770)^T$ and $(\xi_2^1, \xi_3^1)^T = (430, 770)^T$ seconds (all feedforward delays ψ_r^l are 0) result in a delayed differential equation (DDE) that does not possess a globally stable equilibrium point. For smaller delays or a different set of parameters a and b, this system converges, as evidenced in §IX.A.1 of [5].

Algorithm	$E(\hat{n})$	$\sigma_{\hat{n}}$	$E(x(\hat{n}))$	$\|\cdot\|_1-$error
Exact	-	-	$(1.65, 1.44, 1.22)^T$	0
Delay	91762	15774	$(1.63, 1.43, 1.22)^T$	0.040

TABLE III: Comparing with system in [5]

When implemented using the ASA framework of §II convergence is assured irrespective of delays, since it is the ODE with a globally stable equilibrium point that the ASA recursion tracks. The global index n represents a slot of 1 second. Thus, source 3, for example updates at $\{n : n\%770 = 0\}$. Since the updates are spaced farther apart, the convergence conditions are stricter, with $\hat{n} = \max_{k\in\{1,2,\ldots,5000\}} \|x(\hat{n}-k) - x(\hat{n})\|_1 \le 10^{-3}$.

Acknowledgments

This work was supported in part by Grant no. SR/S3/EE/43/2002-SERC-Engg from the Department of Science and Technology, Government of India.

REFERENCES

[1] F. Kelly, A. Maulloo, and D. Tan, "Rate control in communication networks: shadow prices, proportional fairness, and stability," *Journal of the Operational Research Society*, vol. 49, pp. 237–252, 1998.

[2] R. Johari and D. Tan, "End-to-End Congestion Control for the Internet: Delays and Stability," *IEEE/ACM Transactions on Networking*, vol. 9, no. 6, pp. 818–832, 2001.

[3] G. Vinnicombe, "On the stability of networks operating TCP-like congestion control," *Proceedings of the IFAC World Congress, Barcelona, Spain*, 2002.

[4] S. Liu, T. Basar, and R. Srikant, "Modeling of rtt variations in window-based congestion control," in *Proceedings of IEEE INFOCOM, Miami, FL*, 2005.

[5] P. Ranjan, R. La, and E. Abed, "Global Stability Conditions for Rate Control With Arbitrary Communication Delays," *IEEE/ACM Transactions On Networking*, vol. 14, no. 1, pp. 94–107, 2006.

[6] S. Deb and R. Srikant, "Rate-based versus queue-based models of congestion control," in *Proceedings of ACM SIGMETRICS*, 2004.

[7] R. Srikant, "Models and methods for analyzing internet congestion control algorithms," in *Advances in Communication Control Networks (series LCNCIS)*, C. Abdallah, J. Chiasson, and S. Tarbouriech, Eds. Springer-Verlag, 2004.

[8] V. Borkar, "Asynchronous stochastic approximations," *SIAM Journal on Control and Optimization*, vol. 36, pp. 840–851, 1998.

[9] M. S. Abdulla and S. Bhatnagar, "SPSA with measurement reuse," in *Proceedings of the Winter Simulation Conference, Dec 4-6, Monterey, CA, USA*, 2006.

[10] X. Fan, K. Chandrayana, M. Arcak, S. Kalyanaraman, and J. Wen, "A Two-Time-Scale Design for Edge-Based Detection and Rectification of Uncooperative Flows," *Accepted in IEEE/ACM Transactions on Networking, also in IEEE CDC'05*, 2006.

[11] V. Borkar, "An actor-critic algorithm for constrained markov decision processes," *Systems and Control Letters*, vol. 54, pp. 207–213, 2005.