Distributed Iterative Optimal Resource Allocation with Concurrent Updates of Routing and Flow Control Variables

Jayakrishnan Nair and D. Manjunath

***Abstract*—Consider a set of active elastic sessions over a network. Session traffic is routed at each hop (potentially through multiple network paths) based only on its destination. Each session is associated with a concave increasing utility function of its transfer rate. The transfer rates of all sessions and the routing policy define the operating point of the network. We construct a metric f of the goodness of this operating point. f is an increasing function of the session utilities and a decreasing function of the extent of congestion in the network. We define 'good' operating points as those that maximize f subject to the capacity constraints in the network. This paper presents a distributed, iterative algorithm for adapting the session rates and the routing policy across the network so as to converge asymptotically to the set of 'good' operating points. The algorithm updates session rates and routing variables *concurrently*, and is therefore amenable to distributed online implementation. The convergence of the concurrent update scheme is proved rigorously.**

***Index Terms*—Optimal routing, optimal rate control, multipath routing, two timescale iterations.**

I. Introduction

CONSIDER a network represented by an edge-capacitated connected digraph $\mathcal{G}(\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ denotes the set of nodes and $\mathcal{L}$ the set of links interconnecting the nodes. Link $l \in \mathcal{L}$ has capacity c_l. A set of end-to-end elastic sessions, $\mathcal{S}$ shares the resources (link capacities) of this network. We assume that each session $s \in \mathcal{S}$ submits traffic to the network according to a quasi stationary random process with (adaptive) transfer rate x_s. Each session s is associated with a concave increasing utility function $U_s(\cdot)$ of its transfer rate. We define for each link $l \in \mathcal{L}$ a convex increasing cost function $\phi_l(\cdot)$ of the mean flow w_l carried by it. In this paper, we propose an algorithm that iteratively adapts the session rates and the routing, seeking to maximize

$$f := \sum_{s \in \mathcal{S}} U_s(x_s) - \sum_{l \in \mathcal{L}} \phi_l(w_l) \tag{1}$$

subject to the capacity constraints of the network.

The first term of f in (1) represents the 'aggregate social utility,' a quantity many rate control algorithms proposed in the literature seek to maximize, e.g., [1]–[4]. It is easy to see that at an operating point of the network that maximizes this quantity, some of the links would operate at saturation. This of course is not acceptable in practice, as it might result in unbounded packet queues and delays. The second term of f in (1), which is an increasing function of the link utilizations represents the level of congestion in the network; see [5], [6] and the discussion in Section I-A. Thus our objective function f tempers the social utility by a penalty for congestion in the network. An alternative approach is to a priori specify an upper bound on the link utilizations (which would be strictly less than one). While this approach might help bound delays, our objective function explicitly captures the tradeoff between high data rates and network congestion.

In much of the recent literature on adaptive rate control and routing, it is assumed that each session is associated with a predefined set of network paths; each session source adapts the volume of flow along each of these paths. Such a routing scheme is clearly not scalable. In this paper, we employ the routing model of [6], [7]; each node i forwards a fraction $p^k_{(i,j)}$ of the traffic it receives for destination k to neighboring node j. This distributed, destination-based, multipath routing is obviously more amenable to implementation.

In this paper, we present and rigorously prove convergence of a distributed, iterative algorithm for the sessions and the network to attain respectively, the globally optimal transmission rates x_s and routing fractions $p^k_{(i,j)}$ that maximize f. Our algorithm allows for the session rates and routing variables to be *iterated simultaneously*, i.e., it avoids the 'two-level convergence structure' that is typical in solutions of non-convex network optimization problems with variables from multiples layers of the protocol stack; see [8], [9] for examples of such solutions. We discuss the issues with such two-level nested iterations in Section III.

To summarize the above discussion and delineate this work from other work in network optimization,

1) The objective function factors in social utility of the sessions as well as a cost for link congestion.
2) The routing algorithm we consider is scalable and hence more practical.
3) The iterations for the routing and the rate variables are concurrent.
4) The convergence of the iterative algorithm is proved rigorously.

In the following we first provide an interesting interpretation of (1) for a special choice of link cost functions $\phi_l(\cdot)$. We follow that up with a discussion of the related literature leading up to the problem formulation of this paper. Section II details our problem formulation. Section III outlines the solution

Jayakrishnan Nair is with the Department of Electrical Engineering, California Institute of Technology, Pasadena, CA 91125 USA (email: ujk@caltech.edu).

D. Manjunath is with the Department of Electrical Engineering, Indian Institute of Technology Bombay, Mumbai 400076 India (email: dmanju@ee.iitb.ac.in).

approach. Sections IV–VI describe the algorithm in detail and prove its convergence. Section VII discusses the mechanism for the information exchange required by the algorithm. Section VIII presents simulation results for our algorithm.

A. An interpretation of f

An appealing interpretation of the second term of f in (1) is as a penalty for the delay experienced by the packets in the network. Assume that the average number of packets, either queued at the input of link l or in transmission is a function $n_l(w_l)$ of its throughput w_l. Taking $\phi_l(w_l) = \alpha\ n_l(w_l)$ where $\alpha > 0$, we can rewrite (1) as follows.

$$
\begin{aligned}
f &= \sum_{s\in\mathcal{S}} U_s(x_s) - \alpha\left(\sum_{l\in\mathcal{L}} n_l\right) &(2)\\
&= \sum_{s\in\mathcal{S}} \left(U_s(x_s) - \alpha n^s\right) &(3)\\
&= \sum_{s\in\mathcal{S}} \left(U_s(x_s) - \alpha x_s D_s\right). &(4)
\end{aligned}
$$

Here n^s and D_s are, respectively, the average number of packets of session s in the network at any time and the average delay experienced per packet of s. (4) follows from (3) via Little's Theorem. Written in the form (2), maximizing f implies maximizing the sum total of user utilities subject to a penalty proportional to the average total number of packets, say n in the system at any time. n may be viewed as an indicator of congestion in the network. We remark here that this is the idea underlying isarithmic flow control [5]. The forms (3) and (4) indicate that we are maximizing the sum total of delay sensitive utilities $U_s(x_s) - \alpha\ x_s D_s$ of all sessions. The second term represents a penalty corresponding to the average delay experienced by a packet of the session. Such delay sensitive utility functions have been encountered in the reverse engineering of TCP-IP networks [10].

B. Background and Relation to Previous Work

A standard technique in network resource allocation is to define the desired operating point of the network as the solution of an optimization problem. Iterative procedures to attain the optimum are then derived. It is of course desirable that these iterative procedures be suitable for distributed implementation in the network. We too will follow this same course. Optimal rate control of elastic traffic is an old jungle problem and it is instructive to begin a survey with the following formulation.

$$
\begin{aligned}
\text{max.}\quad & \sum_{s\in\mathcal{S}} U_s(x_s) & & \\
\text{s.t.}\quad & w_l = \sum_{\text{sessions } s \text{ using link } l} x_s \le c_l & \forall \quad l \in \mathcal{L} & \qquad \text{(P1)}\\
& x_s \ge 0 & \forall \quad s \in \mathcal{S} &
\end{aligned}
$$

This assumes a single fixed route for each session. Distributed primal and dual algorithms for solving a relaxation of (P1) based on the o.d.e. method are described in [1]. For the case of logarithmic utility functions, [11] interprets the global optimum of (P1) as a Nash bargaining solution and presents a distributed gradient projection based solution of the dual of (P1). [2] presents a distributed gradient projection based solution of the dual of (P1) for a more general class of concave utility functions.

The generalization of (P1) to allow multipath routing has also received significant attention. It is easy to see that routing the flow of each session through multiple network paths (the terms path and route will be used interchangeably) can result in a better usage of the network resources and a higher aggregate utility $\sum_{s\in\mathcal{S}} U_s(x_s)$. However, if each session's flow is to be apportioned across multiple paths, then the network optimization has to determine not only the the session rates, but also the fractions of session flows to be routed along each path. Thus the multipath generalization of (P1) combines routing and rate control.

The most common approach towards a multipath generalization of (P1) is as follows. A set of routes R_s is defined for each session s. Let $R := \cup_{s\in S} R_s$. Denote the rate on route $r \in R$ by y_r. The following is a multipath generalization of (P1).

$$
\begin{aligned}
\text{max.}\quad & \sum_{s\in\mathcal{S}} U_s(x_s) = \sum_{s\in\mathcal{S}} U_s\left(\sum_{r\in R_s} y_r\right) & & \\
\text{s.t.}\quad & w_l = \sum_{\text{routes } r \text{ using link } l} y_r \le c_l & \forall \quad l \in \mathcal{L} & \qquad \text{(P2)}\\
& y_r \ge 0 & \forall \quad r \in R &
\end{aligned}
$$

Many distributed solutions to this problem have been proposed, e.g., [1], [3], [4], [12]. [1] suggests multipath generalizations of their algorithms for the single route per session case. [3] converts (P2) into an equivalent unconstrained problem by adding (non-differentiable) penalty terms to the objective function corresponding to link capacity constraints and presents a solution that solves it using subgradient projection.

Solving the dual of (P2) directly (i.e., a multipath generalization of the approach of [2]) is complicated by the fact that the objective function of (P2) is not strictly concave with respect to the path rates y_r. The objective function of the dual is therefore, not guaranteed to be differentiable. [12] suggests a subgradient projection based algorithm for solving the dual problem. It has however been observed that the session rates obtained by the algorithm of [12] may exhibit oscillatory behavior [3]. In [4], quadratic terms are added to the objective function of (P2) to make it strictly concave with respect to the path rates and the dual problem is solved using a proximal optimization algorithm [13].

Using the same routing as in (P2), Chapter 6 of [5] presents the following formulation where the objective function includes link congestion costs.

$$
\begin{aligned}
\text{min.}\quad & \sum_{s\in\mathcal{S}} e_s(x_s) + \sum_{l\in\mathcal{L}} \phi_l(w_l) & & \\
\text{s.t.}\quad & w_l = \sum_{\text{routes } r \text{ using link } l} y_r \le c_l & \forall \quad l \in \mathcal{L} & \qquad \text{(P3)}\\
& y_r \ge 0 & \forall \quad r \in R &
\end{aligned}
$$

Here, $e_s(x_s)$ is a convex decreasing penalty for the session rate x_s being too small. Interpreting $-e_s$ as a utility function for session s, (P3) is equivalent to a maximization of our objective function f subject to the same constraints. Various gradient based primal algorithms for solving (P3) are discussed in [5].

An important practical issue in the online implementation of a solution of (P2) and (P3) is that the source node of each session needs explicit knowledge of all the routes used by it. Furthermore, this route needs to be encoded in the packets, or all the nodes need to be capable of path-based routing. Such source routing or path-based routing schemes may be infeasible in a large network where there could potentially be a large number of paths for each session. In this work, we address this issue by adopting a destination-based multipath routing model, i.e., the forwarding rule for packets arriving at a node is a function of only the packet destination. As we will see, this routing model makes our resource allocation problem non-convex. As a result, our algorithm as well as its proof of convergence differ considerably from those of the references mentioned previously.

We now delineate our work from other research on cross layer optimizations. As must be evident from our earlier discussions, our interest in this paper is in the problem of rate control and routing of elastic traffic in a wireline network. We adopt the traffic model of [1], i.e., we model the session rates, but *not* the link queue lengths. In our case, link congestion is modeled (indirectly) via a convex function of the total flow carried by the link. There is considerable recent literature that focuses on rate control based on the occupancy of the link queues, especially in the context of wireless networks. For example, see [14]–[16].

II. Problem Formulation

We now describe the network model and the problem formulation. Recall that a set $\mathcal{S}$ of infinitely backlogged, elastic sessions shares the network, represented by the connected digraph $\mathcal{G}(\mathcal{N},\mathcal{L})$. Each session $s \in \mathcal{S}$ is defined by an origin or source node $o(s)$, a destination node $d(s)$ and a concave increasing utility function $U_s : \mathbb{R}_+ \to \mathbb{R}$ of its transfer rate x_s. Let $x = (x_s, s \in \mathcal{S})$ denote the vector [1] of session rates. Link $l \in \mathcal{L}$ has capacity c_l and $c = (c_l, l \in \mathcal{L})$ denotes the vector of link capacities.

For each node $i \in \mathcal{N}$, define $N(i,k)$ to be the set of neighbors of node i to which packets with destination $k \in \mathcal{N} \setminus \{i\}$ are forwarded. Node i forwards a fraction $p^k_{(i,j)}$ of the traffic it receives for destination k to neighbor $j \in N(i,k)$ on link (i,j). [2] The sets $N(i,k)$, defined for all $i,k \in \mathcal{N}$, $i \neq k$ determine the set of paths that the packets of each session can take. By appropriately restricting the sets $N(i,k)$, we can ensure loop-free paths for all sessions. We shall henceforth assume the following.

(A1) The sets $N(i,k)$ for all $i,k \in \mathcal{N}$, $i \neq k$ are defined (and known at node i) so that the flow between any source-destination pair gets routed through loop-free paths.

We note here that this a priori establishment of loop-free paths for all sessions may be achieved by a simple flooding of link weights. If $H(i,k)$ denotes the weight of the shortest path from i to k, we may restrict $N(i,k)$ to contain only those neighbors j of i that satisfy $H(j,k) < H(i,k)$.

(A1) implies that corresponding to each $k \in \mathcal{N}$, the sets $N(i,k)$ for $i \in \mathcal{N} \setminus \{k\}$ induce a directed acyclic subgraph $\mathcal{G}_k$ of $\mathcal{G}$ with sink k, over which traffic generated in the network with destination k flows.

For $i,k \in \mathcal{N},\ i \neq k$, let $p^k_i = (p^k_{(i,j)}, j \in N(i,k))$. The set of allowable values of p^k_i is

$$\Omega^k_i = \{p^k_i \geq 0 \mid \sum_{j \in N(i,k)} p^k_{(i,j)} = 1\}.$$

The routing vector p, obtained by concatenating the p^k_i is an element of the Cartesian product space

$$\Omega = \prod_{\substack{i,k \in \mathcal{N} \\ i \neq k}} \Omega^k_i.$$

Let $a_{l,s}(p)$ denote the fraction of the total flow of session s routed through link l. Clearly, $a_{l,s}$ is a sum of products of the routing fractions. Define the $|\mathcal{L}| \times |\mathcal{S}|$ routing matrix $A(p) = [[a_{l,s}(p)]]$. The vector of link flows $w(x,p) = (w_l(x,p), l \in \mathcal{L}) = A(p)x$. The capacity constraint on the network is thus $w(x,p) = A(p)x \leq c$.

Our optimal multipath rate control and routing is defined by the following optimization problem.

$$\begin{aligned} \text{max.} \quad & \sum_{s \in \mathcal{S}} U_s(x_s) - \sum_{l \in \mathcal{L}} \phi_l\left(w_l(x,p)\right) \\ \text{s.t.} \quad & A(p)x \leq c \\ & x \geq 0 \\ & p \in \Omega \end{aligned} \qquad \text{(P4)}$$

(P4) is not a convex program because the feasible region of the rate and routing variables is not convex. We present in this paper a distributed, iterative algorithm in which

1) the source $o(s)$ of each session $s \in \mathcal{S}$ adapts its transfer rate x_s, and
2) each node i adapts the routing fractions p^k_i, for $k \in \mathcal{N} \setminus \{i\}$,

so that (x,p) across the network converges asymptotically to a global optimum [3] of (P4). Note that the routing model is both destination-based and completely distributed.

III. Solution Approach

Our solution will involve transforming (P4) into an equivalent program (P5). In this section, we describe this transformation and the outline of our approach towards solving (P5). We begin by stating our assumptions on the functions representing the session utilities $U_s(\cdot)$ and the link costs $\phi_l(\cdot)$.

(A2) For all $s \in \mathcal{S}$, $U_s(\cdot)$ is monotonically increasing, strictly concave and twice continuously differentiable over $(0,\infty)$ with $U''_s(x_s) < 0$ for all $x_s > 0$. Also, $\lim_{x_s \to 0^+} U_s(x_s) = -\infty$.

(A3) For all $l \in \mathcal{L}$, $\phi_l(\cdot)$ is monotonically increasing, strictly convex and twice continuously differentiable over

[1] All vectors are taken to be column vectors.

[2] We refer to a link by a single index, e.g., l, or by the directed pair of nodes that it connects, e.g., (i,j).

[3] Henceforth, unless explicitly stated otherwise, an "optimum" refers to a global optimum.

$[-\epsilon_l, c_l)$ for some small $\epsilon_l > 0$. Additionally, $\phi_l''(w_l) > 0$ for all $w_l \in [-\epsilon_l, c_l)$ and $\lim_{w_l \to c_l-} \phi_l(w_l) = \infty$.

The need for the extension of the definition of the link cost function $\phi_l(\cdot)$ over the small negative range $[-\epsilon_l, 0)$ will be explained later in this section.

(A2) guarantees that at an optimum $(\tilde{x}, \tilde{p})$ of (P4), all session rates are strictly positive, i.e., $\tilde{x} > 0$. (A3) guarantees that all link utilizations are strictly less than one, i.e., $w(\tilde{x}, \tilde{p}) < c$. Both are clearly desired properties of the optimum point.

We introduce slack variables $z = (z_l, l \in \mathcal{L})$ defined by $z = c - w(x,p)$. Define $V_l(z_l) := -\phi_l(c_l - z_l)$. This allows us to rewrite (P4) as follows.

$$\begin{aligned} \text{max.} \quad & \sum_{s\in\mathcal{S}} U_s(x_s) + \sum_{l\in\mathcal{L}} V_l(z_l) \\ \text{s.t.} \quad & z + A(p)x = c \\ & x, z \geq 0 \\ & p \in \Omega \end{aligned} \qquad \text{(P4a)}$$

(A3) implies that for all $l \in \mathcal{L}$, $V_l(\cdot)$ is monotone increasing, strictly concave and twice continuously differentiable over $(0, c_l + \epsilon_l]$ with $V_l''(z_l) < 0$ for $z_l \in (0, c_l + \epsilon_l]$ and $\lim_{z_l \to 0^+} V_l(z_l) = -\infty$.

For each $s \in \mathcal{S}$, define $M_s := \sum_{j \in N(o(s),d(s))} c_{(o(s),j)}$. M_s is the sum total of the capacities of the links available to session source $o(s)$ to route its flow. Define

$$I := \{(x,z) \mid 0 \leq x_s \leq M_s,\ 0 \leq z_l \leq c_l + \epsilon_l\}.$$

Any feasible (x,z) of (P4a) must of course lie in I. However, it will aid our analysis to add to (P4a) the redundant constraint $(x,z) \in I$. Note that the extension in the definition of $\phi_l(\cdot)$ over $[-\epsilon_l, 0)$ allows the the objective function of (P4a) to be defined over I.

Since $V(z_l)$ is strictly increasing in z_l, we may further replace the equality constraint of (P4a) by an inequality to obtain the following equivalent program.

$$\begin{aligned} \text{max.} \quad & \sum_{s\in\mathcal{S}} U_s(x_s) + \sum_{l\in\mathcal{L}} V_l(z_l) \\ \text{s.t.} \quad & z + A(p)x \leq c \\ & (x, z) \in I \\ & p \in \Omega \end{aligned} \qquad \text{(P5)}$$

Consider a solution $(\tilde{x}, \tilde{z}, \tilde{p})$ of (P5). Clearly, $(\tilde{x}, \tilde{z}, \tilde{p})$ solves (P4a) and $(\tilde{x}, \tilde{p})$ solves (P4). Notice from the structure of (P5) that the slack variable z_l may be viewed as the volume of a 'phantom flow' on link l with the utility function $V_l(z_l)$. Following this interpretation, we shall henceforth refer to the tuple (x,z) as flow rates. Note that z_l can be controlled by the transmitting node of link l.

We now discuss the approach to solve (P5). If we keep the routing fractions p in (P5) frozen, we obtain the following program.

$$\begin{aligned} \text{max.} \quad & \sum_{s\in\mathcal{S}} U_s(x_s) + \sum_{l\in\mathcal{L}} V_l(z_l) \\ \text{s.t.} \quad & z + A(p)x \leq c \\ & (x, z) \in I \end{aligned} \qquad \text{(P5-TL)}$$

In this problem, the routing matrix A is a constant. This makes (P5-TL) a convex program parametrized by the routing vector p. (P5-TL) turns out to be a multipath generalization of the rate control problem studied in [2]. We will show in Section IV that its dual can be iteratively solved by a gradient projection method to obtain unique optimal Lagrange multipliers $\lambda^*(p)$ and the unique flow rates $(x^*(p), z^*(p))$. As in [2], we will see that the Lagrange multipliers λ will admit an interpretation as link prices.

Now, let $U(p)$ denote the optimum value of the objective function of (P5-TL). With this notation, (P5) is equivalent to:

$$\begin{aligned} \text{max.} \quad & U(p) \\ \text{s.t.} \quad & p \in \Omega \end{aligned} \qquad \text{(P5-NL)}$$

We will show in Section V that the gradient of $U(p)$ can be computed as a function of $\lambda^*(p)$ and $(x^*(p), z^*(p))$. This suggests the following abstract nested iterative algorithm for solving (P5). In the inner loop, keep the routing fractions fixed and perform iterations corresponding to (P5-TL) to obtain $\lambda^*(p)$ and $(x^*(p), z^*(p))$. In the outer loop, use these values to update the routing fractions using an ascent algorithm to solve (P5-NL). We will show in Section V that the sequence of session rates and routing fractions generated by such a scheme will converge to an optimum of (P5).

An online implementation of the above mentioned nested scheme in a distributed setting would imply that routing fractions are updated at a slower timescale than the session rates. That is, at a faster timescale, a transport layer algorithm updates flow rates (x,z) and link prices λ and at a slower timescale, a network layer algorithm updates the routing fractions p. Note that this is an approximation to the abstract scheme described above, since the network layer updates use only approximately equilibrated values of (x,z) and λ from the transport layer algorithm.

In optimization problems involving two sets of variables, this 'two timescale' solution technique of using a system of nested iterations, where the inner level of iterates of one set of variables is expected to approximately converge between iterations at the outer level of the other has been suggested in, among others, [8], [9]. However, in an online distributed implementation it is difficult to determine when to terminate the inner loop of iterations. Also, to prove the convergence of the iterative scheme, it is necessary to analyze the effect of point of termination of the inner loop on the convergence of the outer loop.

An alternative solution to such 'two timescale' iterative schemes that is both elegant and provably correct, has been described in [17] in the stochastic approximation framework. This scheme uses a single iterative loop with concurrent updates of both sets of variables, their stepsizes separated by an order of magnitude. We adapt this method to devise a synchronous iterative algorithm that updates all variables concurrently and converges to the set of solutions of (P5). This is the key contribution of this paper.

There has been some recent work that has explored ideas similar to ours. A problem formulation similar to ours has been reported in [9]. However, the update scheme proposed is based on the 'two timescale' technique of nested iterations described earlier. Further, to prove the convergence of our concurrent update scheme, we rigorously prove the convergence of the nested iterations.

More recently, a general network utility maximization problem involving two sets of variables in considered in [18]. The convergence of an iterative algorithm that makes concurrent updates to all variables using the idea of [17] is also proved. However, we note that our problem (P4) is not of the form considered in [18]. In particular, the problem considered in [18] is assumed to be convex, whereas (P4) is a non-convex optimization problem.

Sections IV and V describe respectively the transport layer and the network layer algorithms, in the abstract nested spirit, i.e., with the transport layer algorithm seeing a fixed p, and the network layer algorithm seeing equilibrated link prices and flow rates. We then use the developments of Sections IV and V to prove convergence of our proposed scheme of concurrent updates for all variables in Section VI.

IV. Transport Layer Algorithm

This section describes the transport layer algorithm which iteratively solves (P5-TL), treating the routing vector p as a parameter in Ω.

(P5-TL) is a convex program with no duality gap. Since the objective function is strictly concave in (x, z), and I is compact, (P5-TL) has a unique primal solution $(x^*(p), z^*(p))$. Further, (A2) and (A3) imply that $(x^*(p), z^*(p)) \in I^0$. Throughout this section, the parametrization of (P5-TL) by p will be recorded explicitly.

As in [2], our approach will be to solve the dual of (P5-TL) iteratively. In the following we will first formulate the dual program of (P5-TL) and prove uniqueness of its solution. We then introduce a gradient based projected o.d.e., trajectories of which converge to this solution. This o.d.e. inspires our discrete-time algorithm over the Lagrange multipliers of (P5-TL).

Let $\lambda = (\lambda_l, l \in L)$ denote the vector of Lagrange multipliers. The Lagrangian of (P5-TL) is:

$$\begin{aligned} L(x, z, \lambda, p) &= \sum_{s\in S} U_s(x_s) + \sum_{l\in\mathcal{L}} V_l(z_l) \\ &\quad - \lambda^T(z + A(p)x - c) \\ &= \sum_{s\in S} (U_s(x_s) - q^s(\lambda, p)x_s) \\ &\quad + \sum_{l\in\mathcal{L}} (V_l(z_l) - \lambda_l z_l) + \lambda^T c \end{aligned} \quad (5)$$

where $q^s(\lambda, p) = \sum_{l\in\mathcal{L}} a_{l,s}(p)\lambda_l$. If we interpret λ_l as the price per unit flow on link l, $q^s(\lambda, p)$ is the total cost per unit flow to be borne by session s.

Define

$$(\mathsf{x}(\lambda, p), \mathsf{z}(\lambda, p)) := \arg\max_{(x,z)\in I} L(x, z, \lambda, p).$$

Since L is strictly concave with respect to (x, z) and I is a compact convex set, $(\mathsf{x}(\lambda, p), \mathsf{z}(\lambda, p))$ is unique for any (λ, p). Further, it is easy to see from (5) that for each $s \in \mathcal{S}$ and $l \in \mathcal{L}$,

$$\mathsf{x}_s(\lambda, p) = \arg\max_{x_s\in[0,M_s]} U_s(x_s) - q^s(\lambda, p)x_s, \quad (6)$$

$$\mathsf{z}_l(\lambda, p) = \arg\max_{z_l\in[0,c_l+\epsilon_l]} V_l(z_l) - \lambda_l z_l. \quad (7)$$

Thus each session source $o(s)$ can compute $\mathsf{x}_s(\lambda, p)$ given its cost per unit flow $q^s(\lambda, p)$ and the transmitting node of each link l can compute $\mathsf{z}_l(\lambda, p)$ given the cost per unit link usage λ_l.

The dual of (P5-TL) is:

$$\min_{\lambda\geq 0} D(\lambda, p)$$

where

$$D(\lambda, p) = \max_{(x,z)\in I} L(x, z, \lambda, p) = L(\mathsf{x}(\lambda, p), \mathsf{z}(\lambda, p), \lambda, p)$$

$D(\lambda, p)$ is convex and continuously differentiable in λ; see [13], pp. 669. Let $G(\lambda, p) = \nabla_\lambda D(\lambda, p)$[4]. The components of $G(\lambda, p)$ are given by

$$G_l(\lambda, p) = \frac{\partial D(\lambda, p)}{\partial \lambda_l} = c_l - \mathsf{z}_l(\lambda, p) - \sum_{s\in S} a_{ls}(p)\mathsf{x}_s(\lambda, p). \quad (8)$$

We now characterize the dual of (P5-TL).

Lemma 1: The dual program of (P5-TL) has a unique solution $\lambda^*(p)$.

Proof: We know that the solution $(x^*(p), z^*(p))$ of the primal problem (P5-TL) lies in the interior of I. If $\lambda^*(p)$ denotes a solution of the dual program, then $(x^*(p), z^*(p), \lambda^*(p))$ must satisfy the KKT conditions; see Theorems 6.2.5 and 6.2.6, pp. 209 in [19]. Therefore, we have

$$V_l'(z_l^*(p)) = \lambda_l^*(p) \quad \forall \quad l \in \mathcal{L}. \quad (9)$$

The uniqueness of the primal solution $(x^*(p), z^*(p))$ thus implies that the solution $\lambda^*(p)$ of the dual is unique and is also strictly positive. ■

It follows that the objective function of (P5-NL) $U(p) = \min_{\lambda\geq 0} D(\lambda, p) = D(\lambda^*(p), p)$. Also, the optimal flow rates $(x^*(p), z^*(p)) = (\mathsf{x}(\lambda^*(p), p), \mathsf{z}(\lambda^*(p), p))$. We shall describe in this section an iterative algorithm on the Lagrange multipliers λ that converges to $\lambda^*(p)$.

Lemma 2: There exists a strictly positive $\varsigma \in \mathbb{R}_+^{|\mathcal{L}|}$ [5] such that $\lambda^*(p) < \varsigma$ for all $p \in \Omega$.

Proof: We prove in Appendix D that $\lambda^* : \Omega \to \mathbb{R}_+^{|\mathcal{L}|}$ is a continuously differentiable map. [6] Since Ω is a compact set, it follows that $\lambda^*(\Omega)$ is compact and therefore bounded. ■

Let Λ denote the hyperrectangle $\{\lambda \in \mathbb{R}^{|\mathcal{L}|} \mid 0 \leq \lambda \leq \varsigma\}$. It is assumed that the transmitting node of each link l knows the value of ς_l. Our iterative algorithm on λ will be constrained to the set Λ. We comment on how each link l may compute ς_l in Appendix C.

We are now ready to develop the projected o.d.e. and the iterative scheme.

[4] This denotes the gradient of D with respect to λ.

[5] $\mathbb{R}_+^k$ denotes the non-negative orthant of $\mathbb{R}^k$.

[6] This means that for any $\tilde{p} \in \Omega$, there exists an open set $\Phi_{\tilde{p}}$ containing $\tilde{p}$ and a continuously differentiable function $\lambda_{\tilde{p}}^*(\cdot) : \Phi_{\tilde{p}} \to \mathbb{R}^{|\mathcal{L}|}$ that agrees with $\lambda^*(\cdot)$ over $\Phi_{\tilde{p}} \cap \Omega$.

Lemma 3: $G : \Lambda \times \Omega \to \mathbb{R}^{|\mathcal{L}|}$ is Lipschitz continuous.
We prove this Lemma in Appendix C. Consider the following gradient descent projected o.d.e. (PrODE). We follow the notation of [20]; the reader is referred to Appendix A for an overview.

$$\dot{\lambda}(t) = \Pi_\Lambda(\lambda, -G(\lambda, p))^7 \tag{10}$$

Lemma 3 implies that for any fixed p, the map $G : \Lambda \to \mathbb{R}^{|\mathcal{L}|}$ is Lipschitz continuous. Thus (10) is well posed (see Appendix B and also [20]).

Lemma 4: $\lambda^*(p)$ is the unique equilibrium of (10).

Proof: Using Property 2 in Appendix A,

$$\Pi_\Lambda(\lambda, -G(\lambda, p)) = 0 \quad \text{iff} \quad G(\lambda, p)\cdot(\tilde{\lambda} - \lambda) \geq 0 \quad \forall \quad \tilde{\lambda} \in \Lambda$$

From the convexity of $D(\lambda, p)$ in λ, it follows that the equilibria of (10) are the global minimizers of $D(\lambda, p)$ over $\lambda \in \Lambda$. The claim then follows from Lemmas 1 and 2. ■

Theorem 1: $\lambda^*(p)$ is a globally asymptotically stable equilibrium of (10).

Proof: $D(\lambda, p)$ will serve as a Lyapunov function for (10). Using Property 3 in Appendix A,

$$\|\Pi_\Lambda(\lambda, -G(\lambda, p))\|^2 + G(\lambda, p)\cdot(\Pi_\Lambda(\lambda, -G(\lambda, p))) = 0.$$

For any $\lambda \in \Lambda \setminus \{\lambda^*(p)\}$, $\|\Pi_\Lambda(\lambda, -G(\lambda, p))\| > 0$ and therefore,

$$\begin{aligned}\nabla_\lambda D\cdot(\Pi_\Lambda(\lambda, -G(\lambda, p))) &= G(\lambda, p)\cdot(\Pi_\Lambda(\lambda, -G(\lambda, p))) \\ &< 0.\end{aligned}$$

Since Λ is compact, this implies global asymptotic stability of the equilibrium $\lambda^*(p)$ [21]. ■

The transport layer algorithm must 'track' the PrODE (10). The o.d.e. approach to stochastic approximation algorithms [22], [23] (see Appendix B) suggests the following algorithm.

$$\lambda[n+1] = \mathcal{P}_\Lambda\bigg(\lambda[n] - b[n]G(\lambda[n], p)\bigg)^8 \tag{11}$$

where n denotes the discrete time index and $b[n]$ is the stepsize sequence satisfying

$$b[n] > 0; \quad \sum_n b[n] = \infty; \quad \sum_n b[n]^2 < \infty.$$

As Λ is a hyperrectangle, the update of (11) can be performed in a distributed manner. The transmitting node of each link l can update λ_l given $G_l(\lambda[n], p)$. Each session source $o(s)$ can update its rate according to $x_s[n] = \mathsf{x}_s(\lambda[n], p)$, and the transmitting node of each link l can perform the update $z_l[n] = \mathsf{z}_l(\lambda[n], p)$.

Theorem 2: For any $\lambda[0] \in \Lambda$, the sequence $(x[n], z[n], \lambda[n])$ generated by the procedures described above converges to $(x^*(p), z^*(p), \lambda^*(p))$.

Proof: Since (10) has a strict Lyapunov function D, it follows from Theorem 8 in Appendix B that $\lambda[n] \to \lambda^*(p)$. It can be shown that $\mathsf{x}(\lambda, p)$ and $\mathsf{z}(\lambda, p)$ are continuous maps. This implies that $x[n] \to \mathsf{x}(\lambda^*(p), p) = x^*(p)$ and $z[n] \to \mathsf{z}(\lambda^*(p), p) = z^*(p)$. ■

[7] A dot over a function denotes its derivative with respect to time.

[8] $\mathcal{P}_\Lambda : \mathbb{R}^{|\mathcal{L}|} \to \Lambda$ denotes the projection operator onto Λ, i.e., $\mathcal{P}_\Lambda(\psi) = \arg\min_{\lambda\in\Lambda} \|\lambda - \psi\|$.

The (distributed) transport layer algorithm can now be described. At time step n ($n \geq 0$):

1) $q^s(\lambda[n], p)$ is conveyed to the source $o(s)$ of each session s, which computes $x_s[n] = \mathsf{x}_s(\lambda[n], p)$ according to (6). The transmitting node of each link l computes computes $z_l[n] = \mathsf{z}_l(\lambda[n], p)$ according to (7).
2) Each link l uses $x[n]$ and $z[n]$ to compute $G_l(\lambda[n], p)$ according to (8) and performs the update (11).

We comment on the mechanism by which the relevant information may be exchanged between the links and the sessions in Section VII.

V. Network Layer Algorithm

This section describes an iterative procedure for the routing fractions, that converges to the set of solutions of (P5-NL).

$$\begin{array}{ll}\text{max.} & U(p) \\ \text{s.t.} & p \in \Omega\end{array} \tag{P5-NL}$$

Recall that $U(p) = D(\lambda^*(p), p)$ is the optimum value of (P5-TL) for routing vector p. The network layer algorithm operates assuming that the optimal flow rates $(x^*(p), z^*(p))$ and link prices $\lambda^*(p)$ of (P5-TL) are available for each $p \in \Omega$.

As in Section IV, our approach will be to obtain a PrODE for p, trajectories of which converge to solutions of (P5-NL). From this o.d.e., we will derive the discrete-time update scheme for the routing vector. We first characterize the gradient of $U(p)$, in terms of $(x^*(p), z^*(p))$ and $\lambda^*(p)$. We remark here that $U(p)$ is in general not concave. In fact, as we demonstrate using an example in Appendix D, there can exist $p \in \Omega$ that is not a global maximizer of $U(\cdot)$, but, satisfies

$$\begin{aligned}\nabla U(p)\cdot(\tilde{p} - p) \leq 0 \quad \forall \quad \tilde{p} \in \Omega \\ \Leftrightarrow \Pi_\Omega(p, \nabla U(p)) = 0.\end{aligned} \tag{12}$$

The last step follows from Property 2, Appendix A. This implies that a purely gradient based PrODE will not suffice, as it did in Section IV. Our PrODE will however be inspired by the gradient characterization.

Lemma 5: $U(p)$ is continuously differentiable over Ω with $\nabla U(p) = \nabla_p L(x^*(p), z^*(p), \lambda^*(p), p)$. [9]
Note that $U(p)$ is defined over $p \in \Omega$, which is a closed set with an empty interior. The claim regarding continuous differentiability of $U(p)$ above means the following. For any $\tilde{p} \in \Omega$, there exists an open set $\Phi_{\tilde{p}}$ containing $\tilde{p}$ and a continuously differentiable map $U_{\tilde{p}}(\cdot) : \Phi_{\tilde{p}} \to \mathbb{R}$ that agrees with $U(\cdot)$ on $\Phi_{\tilde{p}} \cap \Omega$ and satisfies

$$\nabla U_{\tilde{p}}(\tilde{p}) = \nabla_p L(x^*(\tilde{p}), z^*(\tilde{p}), \lambda^*(\tilde{p}), \tilde{p}).$$

We prove Lemma 5 in Appendix D.

We now derive equations for the components of $\nabla U(p)$. Let us denote the component of $\nabla U(p)$ corresponding to routing fraction $p^k_{(i,j)}$ by $\frac{\partial U(p)}{\partial p^k_{(i,j)}}$. For link price vector λ and routing vector p, define $q^v_u(\lambda, p)$ as the cost per unit flow incurred in sending traffic from node u to node v. It is easy to see that

[9] Recall that $q^s(\lambda, p)$ has a polynomial form. We interpret $q^s(\lambda, p)$ as a function over $\mathbb{R}^{|\mathcal{L}|}_+ \times \mathbb{R}^\omega$ and $L(x, z, \lambda, p)$ as a function over $I \times \mathbb{R}^{|\mathcal{L}|}_+ \times \mathbb{R}^\omega$, where ω denotes the dimension of the routing vector.

$q_u^v(\lambda,p)$ has a polynomial form and that $q^s(\lambda,p)=q_{o(s)}^{d(s)}(\lambda,p)$. Let $q(\lambda,p)$ be the vector obtained by concatenating $q_u^v(\lambda,p)$ for $u,v\in\mathcal{N}$ in some order. $q(\lambda,p)$ uniquely satisfies

$$\begin{aligned} q_u^v(\lambda,p) &= 0 && \text{for } u=v,\\ q_u^v(\lambda,p) &= \sum_{j\in N(u,v)} p_{(u,j)}^v\left(\lambda_{(u,j)}+q_j^v(\lambda,p)\right) && \text{for } u\neq v. \end{aligned} \tag{13}$$

Recall that for any $v\in\mathcal{N}$, the sets $N(u,v)$ for $u\in\mathcal{N}\setminus\{v\}$ induce the acyclic digraph $\mathcal{G}_v$ of $\mathcal{G}$ over which traffic with destination v can flow. (13) shows that each node u can compute q_u^v given the q_j^v of all downstream nodes j in $\mathcal{G}_v$, i.e., by sequentially propagating values up the acyclic digraph $\mathcal{G}_v$, each node u can compute q_u^v.

For routing fraction $p_{(i,j)}^k$, it can be seen with some thought that

$$\frac{\partial q_u^v}{\partial p_{(i,j)}^k}=\begin{cases}0 & k\neq v\\ \alpha_{u,i}^k(p)\left(\lambda_{(i,j)}+q_j^k(\lambda,p)\right) & k=v\end{cases} \tag{14}$$

where $\alpha_{u,i}^k(p)$ denotes the fraction of flow of traffic originating at u destined for k arriving at i.

Using Lemma 5 and (14), we may express the gradient of $U(p)$ as follows.

$$\begin{aligned} \frac{\partial L}{\partial p_{(i,j)}^k} &= -\sum_{s:d(s)=k} x_s\frac{\partial q_{o(s)}^{d(s)}(\lambda,p)}{\partial p_{(i,j)}^k}\\ &= -\left(\lambda_{(i,j)}+q_j^k(\lambda,p)\right)\sum_{s:d(s)=k} x_s\alpha_{o(s),i}^k(p)\\ \Rightarrow \frac{\partial U(p)}{\partial p_{(i,j)}^k} &= -t_i^k(p)\left(\lambda_{(i,j)}^*(p)+q_j^k(\lambda^*(p),p)\right) \end{aligned} \tag{15}$$

where $t_i^k(p)=\sum_{s:d(s)=k} x_s^*(p)\alpha_{o(s),i}^k(p)$ is the total traffic with destination k arriving at node i. Define $F_i^k(\lambda,p):=\left(\lambda_{(i,j)}+q_j^k(\lambda,p), j\in N(i,k)\right)$. $F(\lambda,p)$ is the vector obtained by concatenating the $F_i^k(\lambda,p)$ in the same order as in the routing fraction vector p. Also, define $\nabla_i^k U(p):=\left(\frac{\partial U(p)}{\partial p_{(i,j)}^k}, j\in N(i,k)\right)$. From (15), it is clear that

$$\nabla_i^k U(p)=-t_i^k(p)F_i^k(\lambda^*(p),p). \tag{16}$$

Now, a gradient ascent PrODE for p would be the following.

$$\dot{p}(t)=\Pi_\Omega(p,\nabla U(p)) \tag{17}$$

Using Property 2 in Appendix A, we conclude that any local maximizer of $U(p)$ over Ω is an equilibrium point of (17). However, as we have remarked earlier, not all equilibria of (17) are necessarily global maximizers of $U(p)$. As it turns out, a sufficient condition for a point p to be a global maximizer is obtained if we replace $\nabla U(p)$ by $-F(\lambda^*(p),p)$.

Theorem 3: Any $p\in\Omega$ that satisfies

$$\Pi_\Omega(p,-F(\lambda^*(p),p))=0,$$

is a solution of (P5-NL).

We prove this theorem in Appendix D. This inspires us to choose our network layer PrODE to be

$$\dot{p}(t)=\Pi_\Omega(p,-F(\lambda^*(p),p)). \tag{18}$$

Note that since $\lambda^*(p)$ is continuously differentiable over Ω, so is $F(\lambda^*(p),p)$. This implies that $F(\lambda^*(p),p)$ is Lipschitz continuous over Ω and that (18) is well posed.

As per (18), the routing fractions at node i corresponding to destination k evolve as

$$\dot{p}_i^k(t)=\Pi_{\Omega_i^k}(p_i^k,-F_i^k(\lambda^*(p),p)).$$

If we interpret $\lambda_{(i,j)}^*(p)+q_j^k(\lambda^*(p),p)$ as the cost per unit flow at node i of routing traffic to k via j, then (18) continually increases the routing fractions along the 'cheaper' paths. We will prove in Appendix D that at an equilibrium point of (18), the routing fractions are non-zero along only those forwarding links that provide minimum cost paths to the destination.

We now analyze the asymptotic behavior of (18) using the Lyapunov function $U(p)$.

Lemma 6:

$$\nabla U(p)\cdot\Pi_\Omega(p,-F(\lambda^*(p),p))\geq 0\quad\forall\quad p\in\Omega$$

Proof: Since Ω is the Cartesian product of Ω_i^k for $i,k\in\mathcal{N},\ i\neq k$,

$$\begin{aligned} &\nabla U(p)\cdot\Pi_\Omega(p,-F(\lambda^*(p),p))\\ &=\sum_{\substack{i,k\in\mathcal{N}\\ i\neq k}}\nabla_i^k U(p)\cdot\Pi_{\Omega_i^k}(p_i^k,-F_i^k(\lambda^*(p),p))\\ &=\sum_{\substack{i,k\in\mathcal{N}\\ i\neq k}}-t_i^k(p)F_i^k(\lambda^*(p),p)\cdot\Pi_{\Omega_i^k}(p_i^k,-F_i^k(\lambda^*(p),p)). \end{aligned} \tag{19}$$

Using Property 3 in Appendix A, we have

$$F_i^k(\lambda^*(p),p))\cdot\Pi_{\Omega_i^k}(p_i^k,-F_i^k(\lambda^*(p),p))\leq 0.$$

■

Define $E:=\{p\in\Omega\mid\nabla U(p)\cdot\Pi_\Omega(p,-F(\lambda^*(p),p))=0\}$. Since $F(\lambda^*(p),p)$ is strictly positive, we can see from (16) and (19) that E consists precisely of the equilibrium points of (17).

Invoking Theorem 6 of Appendix A, we can say that trajectories of (18) converge to an invariant subset of E. Let $J=\{p\in\Omega\mid U(p)=\max_{\tilde{p}\in\Omega}U(\tilde{p})\}$ denote the set of solutions of (P5-NL). Obviously, $J\subseteq E$.

Lemma 7: The largest subset of E invariant under (18) is contained in J.

The proof is given in Appendix D. We argue there that a trajectory of (18) initiated at a point in E not in J cannot remain in E. We now know that the trajectories of (18) approach the set J of solutions of (P5-NL).

As before, the o.d.e. approach to stochastic approximation algorithms suggests the following algorithm.

$$p[\bar{n}+1]=\mathcal{P}_\Omega\left(p[\bar{n}]-m[\bar{n}]F(\lambda^*(p[\bar{n}]),p[\bar{n}])\right) \tag{20}$$

where $\bar{n}$ denotes the discrete time index of the network layer algorithm and $m[\bar{n}]$ is the stepsize sequence satisfying

$$m[\bar{n}]>0;\quad\sum_{\bar{n}}m[\bar{n}]=\infty;\quad\sum_{\bar{n}}m[\bar{n}]^2<\infty.$$

The structure of Ω allows for the update of (20) to be performed in a distributed manner. Each node i can perform the following update for every destination k.

$$p_i^k[\bar{n}+1] = \mathcal{P}_{\Omega_i^k}\left(p_i^k[\bar{n}] - m[\bar{n}]F_i^k(\lambda^*(p[\bar{n}]), p[\bar{n}])\right)$$

We comment on the mechanism of exchanging the relevant information between the nodes so that the above update can be performed across the network in Section VII.

Theorem 4: The sequence $p[\bar{n}]$ generated by (20) converges to J.

The proof follows easily from Lemmas 6, 7 and Theorem 8 in Appendix B.

VI. Concurrent Updates for Link Prices and Routing Fractions

In this section, we propose and prove convergence of an iterative scheme that updates the link prices and the routing fractions simultaneously. This scheme is inspired by the work of Borkar [17]. The stepsize sequence used by the p-update is made asymptotically negligible as compared to that used by the λ-update. As a result, the λ-update eventually sees p as quasi static, while the each p-update eventually sees λ as approximately equilibrated. The scheme is described by the following equations.

$$\begin{aligned} x[n] &= \mathsf{x}(\lambda[n], p[n]) \\ z[n] &= \mathsf{z}(\lambda[n], p[n]) \\ \lambda[n+1] &= \mathcal{P}_\Lambda\left(\lambda[n] - b[n]G(\lambda[n], p[n])\right) \\ p[n+1] &= \mathcal{P}_\Omega\left(p[n] - m[n]F(\lambda[n], p[n])\right) \end{aligned} \tag{21}$$

Here, n denotes the discrete time index for the updates. The stepsize sequences $b[n]$ and $m[n]$ satisfy

$$\begin{array}{c} b[n] > 0, \quad \sum_n b[n] = \infty, \quad \sum_n b[n]^2 < \infty, \\ m[n] > 0; \quad \sum_n m[n] = \infty; \quad \sum_n m[n]^2 < \infty, \\ \frac{m[n]}{b[n]} \stackrel{n\uparrow\infty}{\rightarrow} 0. \end{array} \tag{22}$$

We will now prove that the the sequence $(x[n], z[n], p[n])$ generated by (21) converges to solutions of (P5). The proof technique is an adaptation of the development in Chapter 6 of [23]. We proceed through the following sequence of lemmas.

Lemma 8: $(\lambda[n], p[n])$ generated by (21) approaches the set $\{(\lambda^*(p), p) : p \in \Omega\}$ as $n \to \infty$.

Proof: We rewrite the p-update of (21) as

$$p[n+1] = \mathcal{P}_\Omega\left(p[n] - b[n]\big(\frac{m[n]}{b[n]}F(\lambda[n], p[n])\big)\right).$$

Note that as F is bounded on $\Lambda\times\Omega$, $\|\frac{m[n]}{b[n]}F(\lambda[n], p[n])\| \to 0$. Theorem 7 of Appendix B implies that $(\lambda[n], p[n])$ tracks the PrODE

$$\begin{aligned} \dot{\lambda}(t) &= \Pi_\Lambda(\lambda, -G(\lambda, p)) \\ \dot{p}(t) &= 0. \end{aligned} \tag{23}$$

Now

$$\dot{D}(\lambda, p) = G(\lambda, p)\cdot\Pi_\Lambda(\lambda, -G(\lambda, p)) \ \leq\ 0,$$

with equality over the set $\{(\lambda^*(p), p) : p \in \Omega\}$ (see proof of Theorem 1) . The claim thus follows from Theorem 8 of Appendix B. ■

It is easy to see that Lemma 8 implies that

$$\|\lambda[n] - \lambda^*(p[n])\| \stackrel{n\uparrow\infty}{\rightarrow} 0, \tag{24}$$

i.e., $\lambda[n]$ asymptotically tracks $\lambda^*(p[n])$. Recall that J denotes the set of solutions of (P5-NL).

Lemma 9: $p[n]$ generated by (21) approaches the set J as $n \to \infty$.

Proof: We rewrite the p-update of (21) as follows.

$$p[n+1] = \mathcal{P}_\Omega\left(p[n] - m[n]\big(F(\lambda^*(p[n]), p[n]) + \rho[n]\big)\right)$$

where $\rho[n] = F(\lambda[n], p[n]) - F(\lambda^*(p[n]), p[n])$. Since F is Lipschitz continuous over $\lambda\times\Omega$,

$$\|\rho[n]\| \leq C_F\|\lambda[n] - \lambda^*(p[n])\| \stackrel{n\uparrow\infty}{\rightarrow} 0$$

where C_F denotes the Lipschitz constant of F. Invoking Theorem 7 of Appendix B, we conclude that $p[n]$ tracks the PrODE (18). It follows from Lemmas 6 and 7 and Theorem 8 of Appendix B that $p[n]$ converges to the set J. ■

We are now ready to prove the desired convergence result.

Theorem 5: $(x[n], z[n], p[n])$ generated by (21) converges to solutions of (P5).

Proof: Recall that the set of solutions of (P5) is $\{(x^*(p), z^*(p), p) \ : \ p \in J\}$. Lemma 9 tells us that $p[n]$ approaches J. Consider any convergent subsequence $p[k[n]] \to \grave{p} \in J$. From (24) and the continuity of $\lambda^*(p)$, it follows that $\lambda[k[n]] \to \lambda^*(\grave{p})$. The continuity of the map $\mathsf{x}(\lambda, p)$ implies that $x[k[n]] \to \mathsf{x}(\lambda^*(\grave{p}), \grave{p}) = x^*(\grave{p})$. It similarly follows that $z[k[n]] \to z^*(\grave{p})$.

Since every limit point of the bounded sequence $(x[n], z[n], p[n])$ lies in the set $\{(x^*(p), z^*(p), p) \ : \ p \in J\}$, the result follows. ■

It is interesting to note that the proof of the concurrent update scheme follows naturally from the developments of the 'abstract' nested scheme developed in Sections IV and V.

VII. Protocol for Information Exchange

In this section, we discuss the mechanism for relevant information to be exchanged in the network to enable the respective entities to perform the iterations (21).

A. Update of Link Price

Recall that the transmitting node of each link l, denoted by t_l updates z_l and λ_l according to

$$z_l[n] = \underset{z_l\in[0,c_l+\epsilon_l]}{\arg\max}\ V_l(z_l) - \lambda_l[n]z_l, \tag{25}$$

$$\lambda_l[n+1] = \mathcal{P}_{[0,\varsigma_l]}\left(\lambda[n] - b[n]G_l(\lambda[n], p[n])\right), \tag{26}$$

where $G_l(\lambda[n], p[n]) = c_l - z_l[n] - \sum_{s\in\mathcal{S}} a_{l,s}(p[n])x_s[n]$. The update for $z_l[n]$ requires the value of $\lambda[n]$, which is known at t_l. For the price update, each session s needs to convey to each link l that carries its flow the fraction $a_{l,s}$ of its flow routed through that link. This may be done as follows. The sets $N(i,k)$ define the acyclic digraph $\mathcal{G}^s$ of $\mathcal{G}$ over which the flow of session s can flow. As an example, Fig. 1 depicts this graph corresponding to Session 1 for the simulation scenario of Section VIII. It is easy to see how the values of $a_{l,s}(p)$ can propagate down this graph. The transmitting node of each link l in $\mathcal{G}^s$ conveys $a_{l,s}$ to its receiving node by multiplying the routing fraction along that link by the sum of the fractions it receives from its upstream neighbors.

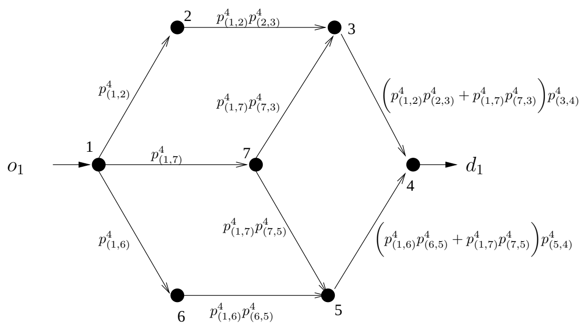

Fig. 1. $\mathcal{G}^1$

B. Update of Session Rates and Routing Fractions

Recall the following.

- The source $o(s)$ of each session updates its session rate x_s according to

$$x_s[n] = \arg\max_{x_s \in M_s} U_s(x_s) - q^s(\lambda[p], p[n])x_s. \tag{27}$$

- Every node i updates p_i^k for all $k \in \mathcal{N} \setminus \{k\}$ according to

$$p_i^k[n+1] = \mathcal{P}_{\Omega_i^k}\left(p_i^k[n] - m[n]F_i^k(\lambda^*(p[n]), p[n])\right). \tag{28}$$

As discussed in Section V, for any destination k, the values of $q_i^k(\lambda[n], p[n])$ can be computed at all nodes in $\mathcal{G}_k$ by propagating values up this graph. In this manner, each session s with destination k will be able to compute the value of $q^s(\lambda[p], p[n])$, enabling it to perform the update (27). Also, each node i in $\mathcal{G}_k$ will obtain the values of $q_j^k(\lambda[n], p[n])$ for all downstream nodes $j \in N(i,k)$, enabling it to update p_i^k according to (28).

Note that as is the case with most network utility maximization literature, we assume the existence of a transport layer mechanism for adapting the session rate. Routing fractions can be stored in the forwarding table of a router and a randomized strategy followed in determining the next hop of a packet.

VIII. Numerical Results

In this section, we present simulation results for our algorithm.

The network topology used for the simulation is depicted in Fig. 2. The network has 7 nodes, labeled 1 through 7 in the figure. A segment joining two nodes i and j indicates that there exist links (i,j) and (j,i), both having a capacity equal to the number shown in the middle of the segment. Session flows are routed only through minimum hop length paths. That is, if $H(i,k)$ denotes the hop-length of the shortest path from i to k, then $N(i,k)$ contains exactly those neighbors j of i that satisfy $H(j,k) < H(i,k)$.

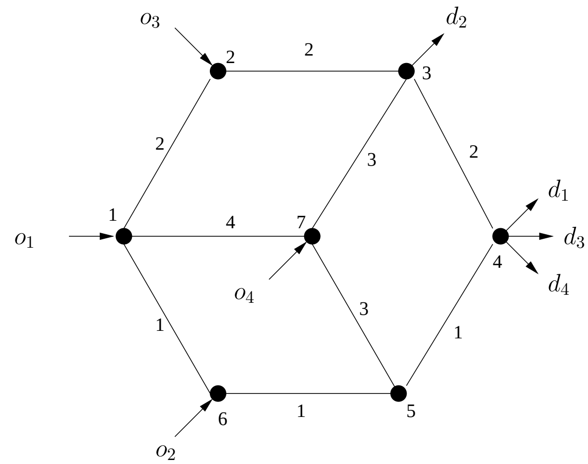

Fig. 2. Network topology

Four sessions, denoted by $s = 1,2,3,4$ use this network. The source node o_s and destination node d_s of each of these is marked in the figure. The utility functions of the sessions are taken to be

$$U_1(x_1) = 200\log(x_1),\ U_2(x_2) = 150\log(x_2),$$
$$U_3(x_3) = 150\log(x_3),\ U_4(x_4) = 100\log(x_4).$$

For each link l, $V_l(z_l) = \frac{z_l - c_l}{z_l}$. We use the stepsize schedule $m[n] = \frac{10}{n}$, $b[n] = \frac{10}{n^{2/3}}$.

The evolution of session rates and the routing fractions at Node 1 corresponding to destination Node 4 are shown respectively in Figures 3 and 4.

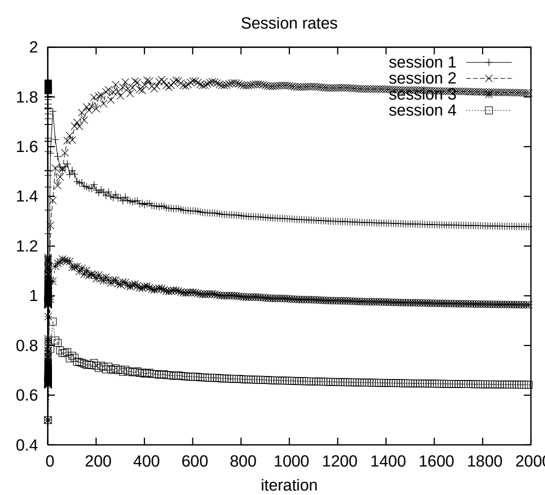

Fig. 3. Evolution of session rates

IX. Conclusion

This paper presents and rigorously proves convergence of a distributed, discrete time algorithm to solve the non-

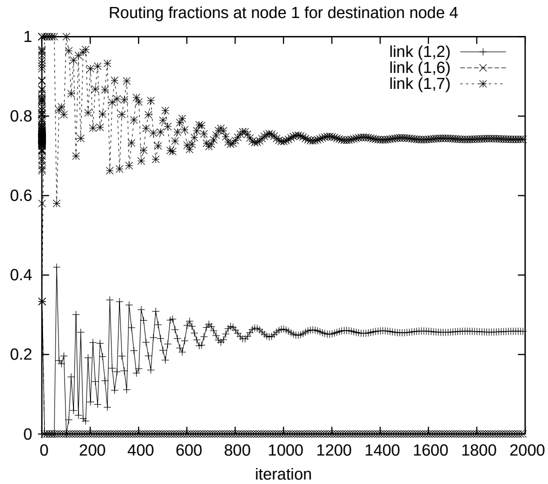

Fig. 4. Evolution of routing fractions at Node 1 corresponding to destination Node 4

convex network utility maximization problem (P4). The key feature of our problem formulation is a distributed, multipath, destination-based routing model. This model is more scalable than the multipath routing models conventionally used in cross-layer networking literature. Moreover, the algorithm we propose iterates routing and flow variables concurrently. By not requiring that different updates take place at different timescales, our algorithm is amenable to implementation in an online distributed setting.

Appendix A
Projected Ordinary Differential Equations

In this section, we introduce our notation for projected ordinary differential equations (PrODEs) and state some results that are used in this work.

Let K denote a non-empty convex polyhedral subset of $\mathbb{R}^k$. $\mathcal{P}_K : \mathbb{R}^k \to K$ denotes the projection operator onto K defined by $\mathcal{P}_K(v) = \arg\min_{u\in K} \|u-v\|$. For any $u \in K$ and $v \in \mathbb{R}^k$, the following limit is well defined [20].

$$\Pi_K(u,v) := \lim_{\delta\to 0^+} \frac{\mathcal{P}_K(u+\delta v) - u}{\delta} \tag{29}$$

For $u \in K^0$, [10] it is easy to see that $\Pi_K(u,v) = v$. Intuitively, when $u \in \partial K$, $\Pi_K(u,v)$ crops off the component of v pointing outward of the boundary at u (see Fig. 5). We make this intuition precise below.

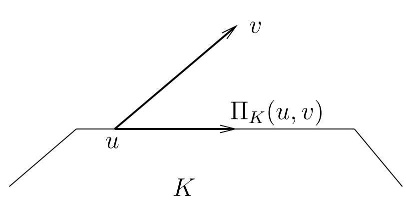

Fig. 5. An illustration of the operator Π_K

[10] K^0 and ∂K denote respectively the interior and the boundary of K.

Define the set of inward normals to K at $u \in \partial K$ as

$$\eta(u) = \{\gamma \in \mathbb{R}^k \;: \|\gamma\| = 1 \text{ and } \langle \gamma, \tilde{u} - u\rangle \geq 0 \;\forall\; \tilde{u} \in K\}.$$

Let $\gamma(u,v) = \arg\max_{\gamma\in\eta(u)} \langle v, -\gamma\rangle$. It can be shown that

$$\Pi_K(u,v) = v + (\langle v, -\gamma(u,v)\rangle \vee 0)\, \gamma(u,v) \tag{30}$$

where $a \vee b$ denotes the larger of the reals a and b (see [20]).

We now state some useful properties of the operator Π_K.

Property 1: For $u \in K$, $v \in \mathbb{R}^k$, and $\alpha \geq 0$,

$$\Pi_K(u, \alpha v) = \alpha \Pi_K(u,v)$$

This is of course obvious from (30).

Property 2: For $u \in K$, $v \in \mathbb{R}^k$,

$$\Pi_K(u,v) = 0 \quad \text{iff.} \quad v\cdot(\tilde{u} - u) \leq 0 \;\;\forall\;\; \tilde{u} \in K.$$

Proof: The result is obvious when $u \in K^0$ or $v = 0$. Otherwise, it follows from (30) that $\Pi_K(u,v) = 0$ iff. $-v/\|v\| \in \eta(u)$. ■

Property 3: For $u \in K$, $v \in \mathbb{R}^k$,

$$\|\Pi_K(u,v)\|^2 = v \cdot \Pi_K(u,v).$$

Proof: The equality holds trivially in either of the following cases.

1) $u \in K^0$.
2) $u \in \partial K$ and $\langle v, -\gamma(u,v)\rangle \leq 0$.

When $u \in \partial K$ and $\langle v, -\gamma(u,v)\rangle > 0$,

$$\Pi_K(u,v) = v + \langle v, -\gamma(u,v)\rangle\gamma(u,v)$$

It is easy to see $\Pi_K(u,v) \perp \gamma(u,v)$. The claim thus follows. ■

Let $g : K \to \mathbb{R}^k$ be a Lipschitz continuous vector field. We are interested in o.d.e.s of the form

$$\dot{x} = \Pi_K(x, g(x)). \tag{31}$$

(31) is called the projected o.d.e. (PrODE) associated with g and K. The projection on the right-hand side forces solution trajectories of (31) to evolve within the set K. If the vector field g drives a trajectory to the boundary of K and points outward of K, then the operator Π_K projects g back onto the boundary.

(31) is a non-classical autonomous o.d.e. since its right-hand side may be discontinuous on ∂K. It has been shown however that for Lipschitz continuous g, (31) is well posed, i.e., for any initial condition $x_0 \in K$, there is a unique absolutely continuous function $x(t)$ over $[0,\infty)$ with $x(0) = x_0$ that satisfies (31) almost everywhere. Moreover, this solution varies continuously with the initial condition x_0 (see [20]).

A set M is said to be invariant with respect to (31) if for any solution $x(t)$ of (31), $x(0) \in M \Rightarrow x(t) \in M \;\forall\; t \geq 0$. We now state a version of LaSalle's invariance theorem for PrODEs that is useful to us [22].

Theorem 6: Assume that K is compact. Let $V : K \to \mathbb{R}$ be a continuously differentiable function satisfying $\nabla V(x) \cdot \Pi_K(x, g(x)) \leq 0$ for all $x \in K$. [11]Define $E := \{x \in K \;\; :$

[11] $\nabla V(x) := \left(\frac{\partial V(x)}{\partial x_j}, j = 1, \cdots, k\right)$ denotes the gradient of V.

$\nabla V(x) \cdot \Pi_K(x, g(x)) = 0\}$. Then solutions of (31) approach the largest invariant subset of E.

APPENDIX B
CONSTRAINED STOCHASTIC APPROXIMATION

In this section, we state two convergence results related to constrained stochastic approximation algorithms. See Chapter 5 of [22] and Chapter 2 of [23] for proofs of these theorems.

Let K denote a non-empty compact convex polyhedral subset of $\mathbb{R}^k$. $g : K \to \mathbb{R}^k$ is a Lipschitz continuous vector field. We are interested in the convergence properties of the stochastic approximation algorithm

$$x[n+1] = \mathcal{P}_K\Big(x[n] + a[n]\big(g(x[n]) + \rho[n]\big)\Big) \tag{32}$$

where n denotes the discrete time index and $x[0] \in K$. Note that the projection operator $\mathcal{P}_K$ constrains the iterates to the set K. We make the following assumptions:

(a1) $a[n] > 0; \quad \sum_n a[n] = \infty; \quad \sum_n a[n]^2 < \infty$

(a2) $\rho[n]$ is a random sequence of 'errors' satisfying $\|\rho[n]\| \overset{n\uparrow\infty}{\to} 0$ almost surely (a.s.)

The o.d.e. approach to the analysis of (32) [22], [23] involves relating the asymptotics of (32) to the asymptotic properties of the PrODE

$$\dot{x} = \Pi_K\big(x, g(x)\big). \tag{33}$$

Define the sequence $t[n]$ as follows: $t[0] = 0$, $t[n] = \sum_{k=0}^{n-1} a[k]$ for $n \geq 1$. Clearly, $t[n] \overset{n\uparrow\infty}{\to} \infty$. Let $\bar{x}(t)$ denote the continuous time (piecewise linear) interpolation of $x[n]$ defined by: $x(t[n]) = x[n]$ for $n \geq 0$ and $x(t) = x[k] + \frac{t - t[k]}{t[k+1] - t[k]}(x[k+1] - x[k])$ for $t \in (t[k], t[k+1])$.

For any $s \geq 0$, let $x^s(t)$ denote the solution of (33) over $t \geq s$ satisfying $x(s) = \bar{x}(s)$.

Theorem 7: Under (a1) and (a2), the sequence $x[n]$ generated by (32) tracks the PrODE (33) a.s. in the sense that,

1) for any $T > 0$,

$$\lim_{s\to\infty} \sup_{t\in[s,s+T]} \|\bar{x}(t) - x^s(t)\| = 0$$

2) $x[n]$ converges to a compact invariant set of (33).

If (a2) holds always (as against a.s.), then the conclusions of Theorem 7 hold true for every sample path.

Theorem 8: Assume (a1) and (a2). Say there exists a continuously differentiable function $V : K \to \mathbb{R}$ such that $\nabla V(x) \cdot \Pi_K\big(x, g(x)\big) \leq 0$ for all $x \in K$. Define $E = \{x \in K \mid \nabla V(x) \cdot \Pi_K\big(x, g(x)\big) = 0\}$. Then $x[n]$ generated by (32) converges a.s. to a subset of E invariant under (33).

As before, if (a2) always holds, then so do the conclusions of Theorem 8.

APPENDIX C
TRANSPORT LAYER ALGORITHM

A. Computation of Λ

Recall that our link price iterations require each link l know ς_l that satisfies $\lambda_l^*(p) < \varsigma_l$ for all $p \in \Omega$. We show in Lemma 2 that such a constant exists. Computation of a ς_l will in general depend on the session utility functions and link cost functions. We now show how each link l may compute ς_l for a special case.

Let $\mathcal{S}_l$ denote the sessions that can route their flow through link l. Denote $s_l = |\mathcal{S}_l|$.

Lemma 10: Consider the case

$$\begin{aligned} U_s(x_s) &= \alpha \log(x_s) \quad \forall \quad s \in \mathcal{S}, \\ V_l(z_l) &= \frac{z_l - c_l}{z_l} \quad \forall \quad l \in \mathcal{L}, \end{aligned}$$

where $\alpha > 0$. Let $\epsilon_l \in (0, c_l)$. We may choose

$$\varsigma_l = \max\left\{\frac{\alpha(s_l+1)}{c_l - \epsilon_l}, \frac{c_l(s_l+1)^2}{(c_l-\epsilon_l)^2}\right\}.$$

Proof: Assume that $\lambda_l^*(p) \geq \varsigma_l$. Then, for $s \in \mathcal{S}_l$,

$$\begin{aligned} x_s^*(p) &= \arg\max_{x_s\in[0,M_s]} U_s(x_s) - q^s(\lambda^*(p), p)x_s \\ &\leq \frac{\alpha}{q^s(\lambda^*(p), p)} \leq \frac{\alpha}{a_{ls}(p)\varsigma_l} \leq \frac{c_l - \epsilon_l}{a_{ls}(p)(s_l+1)}. \end{aligned}$$

Similarly, it may be shown that $z_l^*(p) \leq \frac{c_l - \epsilon_l}{s_l+1}$. This implies that

$$w_l^*(p) := \sum_{s\in\mathcal{S}_l} a_{l,s}(p)x_s^*(p) + z_l^*(p) \leq c_l - \epsilon_l < c_l.$$

However, the optimal Lagrange multipliers $\lambda^*(p)$ of (P5-TL) must satisfy $\frac{\partial D(\lambda^*(p), p)}{\partial \lambda_l} = c_l - w_l^*(p) = 0$. We therefore have a contradiction. ∎

The intuitive explanation for the proof is that ς_l is a large enough link price that ensures the under-utilization of link l. Using this idea, one can extend Lemma 10 to cover the case where each session utility is of form $U_s(x_s) = \alpha_s \log(x_s)$ or $U_s(x_s) = -\alpha_s x_s^{-\beta_s}$ ($\alpha_s, \beta_s > 0$).

B. Proof of Lemma 3

Recall that for $l \in \mathcal{L}$

$$G_l(\lambda, p) = c - \mathsf{z}_l(\lambda, p) - \sum_{s\in\mathcal{S}} a_{ls}(p)\mathsf{x}_s(\lambda, p).$$

We will now prove that $G_l(\lambda, p)$ is Lipschitz. Let $C_L(K)$ denote the set of Lipschitz continuous functions that map the compact set K into $\mathbb{R}$. Since $q^s(\lambda, p)$ and $a_{ls}(p)$ are continuously differentiable, it follows that $q^s(\lambda, p) \in C_L(\Lambda \times \Omega)$ and $a_{ls}(p) \in C_L(\Omega)$.

Define, for all $s \in \mathcal{S}$ and $l \in \mathcal{L}$

$$\begin{aligned} \tilde{x}_s(t) &= \arg\max_{x_s\in[0,M_s]} U_s(x_s) - tx_s, \\ \tilde{z}_l(t) &= \arg\max_{z_l\in[0,c_l+\epsilon_l]} V_l(z_l) - tz_l. \end{aligned}$$

$\tilde{x}_s(t)$ is differentiable over $[0, \infty) - \{U_s'(M_s)\}$ with

$$\tilde{x}_s'(t) = \begin{cases} 0 & t < U_s'(M_s) \\ \frac{1}{U_s''(\tilde{x}_s(t))} & t > U_s'(M_s) \end{cases}$$

Since $|\frac{1}{U_s''(x_s)}|$ is bounded above over $(0, M_s]$, it follows that $\tilde{x}_s(t) \in C_L(\mathbb{R})$. Now $\mathsf{x}_s(\lambda, p) = \tilde{x}_s(q^s(\lambda, p))$. Since a composition of Lipschitz functions is Lipschitz, we conclude that $\mathsf{x}_s(\lambda, p) \in C_L(\Lambda \times \Omega)$. It can be similarly argued that $\mathsf{z}_l(\lambda, p) \in C_L(\Lambda \times \Omega)$.

As the product of two bounded Lipschitz functions is Lipschitz, $a_{ls}(p)\mathsf{x}_s(\lambda, p) \in C_L(\Lambda \times \Omega)$. This in turn implies that $G_l(\lambda, p) \in C_L(\Lambda \times \Omega)$. Therefore, $G : \Lambda \times \Omega \to \mathbb{R}^{|\mathcal{L}|}$ is Lipschitz continuous.

Appendix D
Network Layer Algorithm

A. Proof of Lemma 5

We recall that for any $\tilde{p} \in \Omega$, (P5-TL) has a unique primal solution $(x^*(\tilde{p}), z^*(\tilde{p}))$ and its dual has a unique solution $\lambda^*(\tilde{p})$. We will now invoke the Implicit Function Theorem [24] to deduce smoothness of the maps $x^*(p)$, $z^*(p)$ and $\lambda^*(p)$.

Let ω denote the dimension of the routing vector p. Treat the Lagrangian $L(x, z, \lambda, p)$ of (P5-TL) (Equation (5)) as a function over $I \times \mathbb{R}_+^{|\mathcal{L}|} \times \mathbb{R}^\omega$.

Lemma 11: For any $\tilde{p} \in \Omega$, the system of equations

$$\nabla_{(x,z,\lambda)} L(x, z, \lambda, \tilde{p})^{12} = 0 \tag{34}$$

has the unique solution $(x^*(\tilde{p}), z^*(\tilde{p}), \lambda^*(\tilde{p}))$ over $(x, z, \lambda) \in I \times \mathbb{R}_+^{|\mathcal{L}|}$.

Proof: Since $(x^*(\tilde{p}), z^*(\tilde{p})) \in I^0$, $(x^*(\tilde{p}), z^*(\tilde{p}), \lambda^*(\tilde{p}))$ must satisfy the KKT conditions for (P5-TL) for routing fraction $\tilde{p}$; see Theorems 6.2.5 and 6.2.6, pp. 209 in [19]. Since $\lambda^*(\tilde{p}) > 0$, it follows that

$$\nabla_{(x,z,\lambda)} L(x^*(\tilde{p}), z^*(\tilde{p}), \lambda^*(\tilde{p}), \tilde{p}) = 0.$$

Any tuple $(\tilde{x}, \tilde{z}, \tilde{\lambda})$ that satisfies (34) also satisfies the KKT conditions for (P5-TL), implying that $(\tilde{x}, \tilde{z})$ and $\tilde{\lambda}$ solve (P5-TL) and its dual respectively for routing fraction $\tilde{p}$. Since these solutions are known to be unique, the claim follows. ■

Lemma 12: For any $\tilde{p} \in \Omega$, the matrix $\nabla^2_{(x,z,\lambda)} L(x^*(\tilde{p}), z^*(\tilde{p}), \lambda^*(\tilde{p}), \tilde{p})$ [13] is non-singular.

Proof: Define the diagonal matrix $\mathcal{W}$ as follows.

$$\mathcal{W} := \mathrm{Diag}\Bigg(\big((U_s''(x_s^*(\tilde{p})), s \in \mathcal{S}); (V_l''(z_l^*(\tilde{p})), l \in \mathcal{L})\big)\Bigg)^{14}$$

With this notation, it may be verified that

$$\nabla^2_{(x,z,\lambda)} L(x^*(\tilde{p}), z^*(\tilde{p}), \lambda^*(\tilde{p}), \tilde{p}) = \left[\begin{array}{c|c} \mathcal{W} & -\mathcal{A}(\tilde{p})^T \\ \hline -\mathcal{A}(\tilde{p}) & 0 \end{array}\right] \tag{35}$$

where $\mathcal{A}(\tilde{p})$ is obtained by concatenating the $|\mathcal{L}| \times |\mathcal{L}|$ identity matrix to the right of the routing matrix $A(\tilde{p})$. (A2) and (A3) imply that the diagonal entries of $\mathcal{W}$ are negative. Using elementary row transformations, we can transform the right hand side of (35) into

$$\mathcal{B} = \left[\begin{array}{c|c} \mathcal{W} & -\mathcal{A}(\tilde{p})^T \\ \hline 0 & -\mathcal{A}(p)\mathcal{W}^{-1}\mathcal{A}(\tilde{p})^T \end{array}\right].$$

As $\mathcal{A}(\tilde{p})$ has full row rank, $-\mathcal{A}(\tilde{p})\mathcal{W}^{-1}\mathcal{A}(\tilde{p})^T$ is positive definite and hence invertible. This proves that $\mathcal{B}$ is non-singular, and the claim follows. ■

[12] This denotes the gradient of L with respect to the tuple (x, z, λ).

[13] This denotes the Hessian of L with respect to the tuple (x, z, λ).

[14] $(u; v)$ denotes the concatenation of the column vectors u and v.

For each $\tilde{p} \in \Omega$, Lemmas 11 and 12 allow us to invoke the Implicit Function Theorem [24] for the system of equations (34), i.e., there exists an open set $\Phi_{\tilde{p}} \subset \mathbb{R}^\omega$ containing $\tilde{p}$ and a $\mathcal{C}^1$ function $(x^*_{\tilde{p}}(\cdot), z^*_{\tilde{p}}(\cdot), \lambda^*_{\tilde{p}}(\cdot)) : \Phi_{\tilde{p}} \to I \times \mathbb{R}_+^{|\mathcal{L}|}$ such that

$$\nabla_{(x,z,\lambda)} L(x^*_{\tilde{p}}(p), z^*_{\tilde{p}}(p), \lambda^*_{\tilde{p}}(p), p) = 0 \quad \forall \quad p \in \Phi_{\tilde{p}}.$$

From Lemma 11, we conclude that

$$(x^*_{\tilde{p}}(p), z^*_{\tilde{p}}(p), \lambda^*_{\tilde{p}}(p)) = (x^*(p), z^*(p), \lambda^*(p))$$

for all $p \in \Phi_{\tilde{p}} \cap \Omega$.

Define $U_{\tilde{p}}(\cdot) : \Phi_{\tilde{p}} \to \mathbb{R}$ by

$$U_{\tilde{p}}(p) = L(x^*_{\tilde{p}}(p), z^*_{\tilde{p}}(p), \lambda^*_{\tilde{p}}(p), p).$$

Invoking the chain rule of differentiation and Lemma 11, it is easy to see that

$$\nabla U_{\tilde{p}}(\tilde{p}) = \nabla_p L(x^*(\tilde{p}), z^*(\tilde{p}), \lambda^*(\tilde{p}), \tilde{p})$$

B. Proof of Theorem 3

We will show that at an equilibrium of p of (18), the tuple $(x^*(p), z^*(p), p)$ is a solution of (P5). The proof approach will be to define a convex program equivalent to (P5) and show that conditions satisfied by $(x^*(p), z^*(p), p)$ are equivalent to the KKT conditions for optimality of the convex program.

Recall that the sets $N(i, k)$, defined for $i, k \in \mathcal{N}$, $i \neq k$ define the set of paths R_s that may be taken by the flow of each session s. Let $R := \cup_{s \in S} R_s$ and $y = (y_r, r \in R)$, where y_r denotes that flow routed via path r. The transfer rate of session s is then given by $x_s = \sum_{r \in R_s} y_r$. Consider the following convex formulation.

$$\begin{array}{ll} \text{max.} & \sum_{s \in \mathcal{S}} U_s(x_s) + \sum_{l \in \mathcal{L}} V_l(z_l) \\ \text{s.t.} & x_s = \sum_{r \in R_s} y_r \quad \forall \quad s \in \mathcal{S} \\ & z + Hy \leq c \\ & y, z \geq 0 \end{array} \tag{P6}$$

Here, $H = [[h_{l,r}]]$ is the $|\mathcal{L}| \times |R|$ binary routing matrix i.e, $h_{l,r} = 1$ if route r includes link l and $h_{l,r} = 0$ otherwise.

It can be seen that for any feasible $(\tilde{x}, \tilde{z}, \tilde{p})$ of (P5), there exists a $\tilde{y}$ such that $(\tilde{x}, \tilde{y}, \tilde{z})$ is feasible for (P6) with the same objective function value. Similarly, corresponding to any feasible point $(\hat{x}, \hat{y}, \hat{z})$ of (P6), there exists a feasible point $(\hat{x}, \hat{z}, \hat{p})$ of (P5) with the same objective function value.

Let $\grave{p}$ denote an element of Ω. Let us denote for simplicity $(x^*(\grave{p}), z^*(\grave{p}), \lambda^*(\grave{p}))$ by $(\grave{x}, \grave{z}, \grave{\lambda})$ and $q(\lambda^*(\grave{p}), \grave{p})$ by $\grave{q}$.

Lemma 13: $\grave{p}$ is an equilibrium of (18) iff.

$$\left.\begin{array}{c} \grave{q}_i^k \;\leq\; \grave{\lambda}_{(i,j)} + \grave{q}_j^k \\ \grave{p}^k_{(i,j)}\left(\grave{q}_i^k - \left(\grave{\lambda}_{(i,j)} + \grave{q}_j^k\right)\right) = 0. \end{array}\right\} \forall \;\; i, k \in \mathcal{N},\; i \neq k \tag{36}$$

Proof: Using Property 2 in Appendix B,

$$\begin{array}{l} F_i^k(\lambda^*(p), p)) = 0 \;\; \text{iff} \\ \sum_{j \in N(i,k)} (\grave{\lambda}_{(i,j)} + \grave{q}_j^k)(\tilde{p}^k_{(i,j)} - \grave{p}^k_{(i,j)}) \geq 0 \;\forall\; \tilde{p}_i^k \in \Omega_i^k. \end{array} \tag{37}$$

(36) is equivalent to (37). If we interpret $\grave{\lambda}_{(i,j)} + \grave{q}_j^k$ as the price per unit flow of routing traffic from i to k via link (i, j), then both (36) and (37) hold iff. routing fractions in Ω_i^k are

non-zero only along forwarding links that provide minimum price paths to k. ■

As $(\grave{x}, \grave{z}) \in I^0$, the tuple $(\grave{x}, \grave{z}, \grave{\lambda})$ satisfies the KKT conditions for (P5-TL) (see [19], pp. 211). This implies that

$$U_s(\grave{x}_s) = \grave{q}_{o(s)}^{d(s)} \qquad \forall \qquad s \in \mathcal{S}, \tag{38}$$

$$V_l(\grave{z}_l) = \grave{\lambda}_l \qquad \forall \qquad l \in \mathcal{L}. \tag{39}$$

For route $r \in R_s$ taken by the flow of session s, define $\grave{y}_r = \grave{x}_s \prod_{l \in r} \grave{p}_l^{d(s)}$. Taking $\grave{y} = (\grave{y}_r, r \in R)$, we see that $(\grave{x}, \grave{y}, \grave{z})$ is a feasible point of (P6). Furthermore, (38) and (36) imply that

$$\begin{array}{lll} \grave{y}_r > 0 & \Rightarrow & U_s'(\grave{x}_s) = \sum_{l \in r} \grave{\lambda}_l, \\ \grave{y}_r = 0 & \Rightarrow & U_s'(\grave{x}_s) \leq \sum_{l \in r} \grave{\lambda}_l. \end{array} \tag{40}$$

(40) and (39) are the KKT conditions for (P6). Thus $(\grave{x}, \grave{y}, \grave{z})$ is a solution of (P6) and hence, $(\grave{x}, \grave{z}, \grave{p})$ is a solution of (P5)

C. *Proof of Lemma 7*

It is clear from our decomposition of (P5) into (P5-TL) and (P5-NL) and from the development in Sections IV and V that the set of solutions of (P5) is $\{(x^*(p), z^*(p), p) \mid p \in J\}$, where J is the solution set of (P5-NL). We prove Lemma 7 through the following sequence of Lemmas.

Lemma 14: Over $p \in J$, the tuple $(x^*(p), z^*(p), \lambda^*(p))$ is a constant, say (x^*, z^*, λ^*). Further $p \in J$ iff

$$(x^*(p), z^*(p), \lambda^*(p)) = (x^*, z^*, \lambda^*). \tag{41}$$

Proof: It is easy to see that the optimal (x, z) of (P6) is unique. We can then conclude from the discussion in the proof of Theorem 3 above that $(x^*(p), z^*(p))$ is constant over J. That $\lambda^*(p)$ is constant over J follows from (9).

Since the objective function of (P5) is a function of only (x, z), we conclude that (41) implies that $p \in J$. ■

Lemma 15: With $p(\cdot)$ evolving as per (18)), if $p(t) \in E$, then

$$\begin{array}{rcl} \dot{x}^*(p(t)) & = & 0, \\ \dot{z}^*(p(t)) & = & 0, \\ \dot{\lambda}^*(p(t)) & = & 0. \end{array}$$

Proof: Intuitively, it is easy to see why this is true. When $p \in E$, then

$$t_i^k(p) \Pi_{\Omega_i^k}(p_i^k, -F_i^k(\lambda^*(p), p)) = 0$$

for all $i, k \in \mathcal{N}, i \neq k$. Thus p_i^k can evolve only at nodes i where $t_i^k(p) = 0$, i.e., at nodes that receive no traffic for destination k. This should allow the solution of (P5-TL) to remain constant.

This can be proved formally using the equation for the derivative of the implicit function in the Implicit Function Theorem [24].

Lemma 16: For any fixed, strictly positive $\lambda \in \mathbb{R}^{|\mathcal{L}|}$, solutions of the PrODE

$$\dot{p}(t) = \Pi_\Omega(p, -F(\lambda, p)) \tag{42}$$

converge to its set of equilibria.

Proof: Let M_λ denote the equilibrium set of (42). It is easy to see that this set is non-empty.

Consider the Lyapunov function $Q(p) = \sum_{u,v \in \mathcal{N}} q_u^v(\lambda, p)$. From (14), we may write the partial derivative of Q with respect to routing fraction $p_{(i,j)}^k$ as follows.

$$\frac{\partial Q}{\partial p_{(i,j)}^k} = \left(\sum_{u \in \mathcal{N} - \{k\}} \alpha_{u,k}^i \right) (\lambda_{(i,j)} + q_j^k(\lambda, p)),$$

where $\alpha_{u,k}^i$ is the fraction of flow originating at u destined for k arriving at i. Since $\alpha_{i,k}^i = 1$, $\tau_i^k(p) := \sum_{u \in \mathcal{N} - \{k\}} \alpha_{u,k}^i > 0$.

Let $\nabla_i^k Q(p) = (\frac{\partial Q}{\partial p_{(i,j)}^k}, j \in N(i,k))$. Clearly, $\nabla_i^k Q(p) = \tau_i^k(p) F_i^k(\lambda, p)$. Using Property 3 in Appendix A,

$$\begin{array}{l} \|\Pi_{\Omega_i^k}\big(p_i^k, -F_i^k(\lambda, p)\big)\| > 0 \\ \Rightarrow F_i^k(\lambda, p) \cdot \Pi_{\Omega_i^k}\big(p_i^k, -F_i^k(\lambda, p)\big) < 0 \\ \Rightarrow \nabla_i^k Q(p) \cdot \Pi_{\Omega_i^k}\big(p_i^k, -F_i^k(\lambda, p)\big) < 0. \end{array}$$

Therefore, for $p \notin M_\lambda$,

$$\begin{array}{l} \nabla Q(p) \cdot \Pi_\Omega\big(p, -F(\lambda, p)\big) \\ = \displaystyle\sum_{\substack{i,k \in \mathcal{N} \\ i \neq k}} \nabla_i^k Q(p) \cdot \Pi_{\Omega_i^k}\big(p_i^k, -F_i^k(\lambda, p)\big) < 0. \end{array}$$

The claim thus follows from the Invariance principle. ■

Now, consider a trajectory of (18) that is initiated at a point $\grave{p}$ in E not in J. We claim that the trajectory cannot remain in E for all $t > 0$. Let us assume the contrary. We conclude that $(x^*(p), z^*(p), \lambda^*(p))$ must remain constant (Lemma 15). $p(t)$ will then converge to its set of equilibrium points (Lemma 16), which must be optimal for (P5-NL) (Theorem 3). But this is a contradiction as

$$(x^*(\grave{p}), z^*(\grave{p}), \lambda^*(\grave{p})) \neq (x^*, z^*, \lambda^*)$$

(Lemma 14).

D. *Example showing non-concavity of $U(p)$*

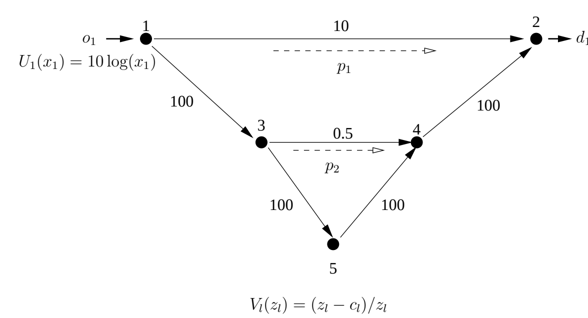

Fig. 6. Example topology to show non-concavity of $U(p)$.

Consider the network topology shown in Fig. 6. Node labels, link capacities and utility functions are marked on the figure. There is a single session, with source node 1 and destination node 2. The routing is characterized by the values of $p_{(1,2)}^2$, denoted by p_1 and $p_{(3,4)}^2$, denoted by p_2.

Consider the case $p_1 = p_2 = 1$. All session traffic is routed on the link $(1, 2)$. The solution (x, z, λ) of (P5-TL) for this

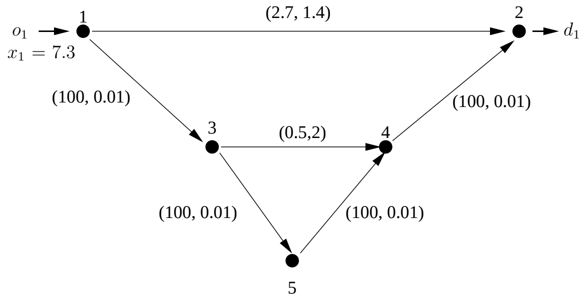

Fig. 7. (z_l, λ_l) is marked along each link l

routing is shown in Figure 7. Note that $\lambda_{(1,2)} < \lambda_{(1,3)} + q_3^2$. It is easy to verify that this routing satisfies (12). However, it is not optimal. The reader may verify that the routing $p_1 = p_2 = 0$, (all session traffic gets routed along the path 1-3-5-4-2) achieves a higher objective value.

Acknowledgment

The authors are grateful to Gopal Srinivasan and Vivek Borkar for many helpful discussions.

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, no. 3, pp. 237–252, 1998.

[2] S. H. Low and D. E. Lapsley, "Optimization flow control—I: Basic algorithm and convergence," *IEEE/ACM Transactions on Networking*, vol. 7, no. 6, pp. 861–874, 1999.

[3] K. Kar, S. Sarkar, and L. Tassiulas, "Optimization based rate control for multipath sessions," Institute for Systems Research, University of Maryland, Tech. Rep. Technical Report No. 2001-1, 2001.

[4] X. Lin and N. B. Shroff, "Utility maximization for communication networks with multi-path routing," *IEEE Transactions on Automatic Control*, vol. 51, no. 5, pp. 766–781, May 2006.

[5] D. P. Bertsekas and R. G. Gallager, *Data Networks*. Prentice Hall, 1992.

[6] R. G. Gallager, "A minimum delay routing algorithm using distributed computation," *IEEE Transactions on Communications*, vol. Com-25, no. 1, pp. 73–85, January 1977.

[7] V. S. Borkar and P. R. Kumar, "Dynamic Cesaro-Wardrop equilibration in networks," *IEEE Transactions on Automatic Control*, vol. 48, no. 3, pp. 382–396, 2003.

[8] X. Wang and K. Kar, "Cross-layer rate control for end-to-end proportional fairness in wireless networks with random access," in *MobiHoc '05: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, 2005.

[9] F. Paganini, "Congestion control with adaptive multipath routing based on optimization," in *Proceedings of the 40th Conference on Information Sciences and Systems (CISS 2006)*, Princeton, NJ, March 2006.

[10] J. Pongsajapan and S. H. Low, "Reverse engineering TCP/IP-like networks using delay-sensitive utility functions," in *Proceedings of IEEE INFOCOM*, Anchorage, Alaska, May 2007.

[11] H. Yaiche, R. Mazumdar, and C. Rosenberg, "A game theoretic framework for bandwidth allocation and pricing in broadband networks," *IEEE/ACM Transactions on Networking*, vol. 8, no. 5, pp. 667–678, 2000.

[12] S. H. Low, "Multipath optimization flow control," in *Proceedings of IEEE International Conference on Networks, Singapore*, September 2000.

[13] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1989.

[14] A. Stolyar, "Maximizing queueing network utility subject to stability: Greedy primal-dual algorithm," *Queueing Systems*, vol. 50, pp. 401–457, 2005.

[15] A. Eryilmaz and R. Srikant, "Joint congestion control, routing, and mac for stability and fairness in wireless networks," *IEEE Journal on Selected Areas in Communication*, vol. 24, no. 8, pp. 1514–1524, August 2006.

[16] L. Georgiadis, M. J. Neely, and L. Tassiulas, *Resource Allocation and Cross Layer Control in Wireless Networks*. NOW Publlishers, 2006.

[17] V. S. Borkar, "Stochastic approximation with two timescales," *Systems and Control Letters*, vol. 29, pp. 291–294, 1997.

[18] J. Zhang, D. Zheng, and M. Chiang, "The impact of stochastic noisy feedback on distributed network utility maximization," *IEEE Transactions on Information Theory*, vol. 54, no. 2, pp. 645–665, 2008.

[19] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming: Theory And Algorithms*. New York: Wiley, 1993.

[20] P. Dupuis and A. Nagurney, "Dynamical systems and variational inequalities," *Annals of Operations Research*, vol. 55, pp. 9–42, 1993.

[21] H. K. Khalil, *Nonlinear Systems*. Prentice-Hall, Inc., 2002.

[22] H. J. Kushner and G. G. Yin, *Stochastic Approximation and Recursive Algorithms and Applications*. Springer, 1997.

[23] V. S. Borkar, *Stochastic Approximation: A Dynamical Systems Viewpoint*. Hindustan Book Agency, New Delhi, India and Cambridge University Press, Cambridge, UK, 2008.

[24] W. Rudin, *Principles of Mathematical Analysis*. McGraw-Hill Publishing Co., 1976.

Jayakrishnan Nair received his BTech and MTech degrees from IIT Bombay in 2007. He is now a graduate student in the Dept of Electical Engineering at California Inst of Technology. His research interests include analysis and design of network protocols.

D. Manjunath received his BE from Mysore University, MS from Indian Institute of Technology, Madras and PhD from Rensselaer Polytechnic Inst, Troy NY in 1986, 1989 and 1993 respectively. He has worked in the Corporate R & D Center of General Electric in Scehenectady NY (1990), Computer and Information Sciences Dept of the University of Delaware (1992–93) and the Computer Science Dept, University of Toronto (1993–94). He was with the Department of Electrical Engineering of Indian Inst of Technology, Kanpur during 1994–98. He has been with the Electrical Engineering Dept of IIT-Bombay since July 1998 where he is now a Professor. His research interests are in the general areas of communication networks and performance analysis. His recent research has concentrated on network traffic and performance measurement, analysis of random networks with applications in wireless and sensor networks, network pricing and queue control. He is a coauthor of two texts from Morgan Kaufman publishers: *Communication Networking: An Analytical Approach* published in May 2004 and *Wireless Networking* published in April 2008.