

AN OPTIMAL DISTRIBUTED ROUTING ALGORITHM USING DUAL DECOMPOSITION TECHNIQUES*

PUNYASLOK PURKAYASTHA[†] AND JOHN S. BARAS[†]

Abstract. We consider the routing problem in wireline, packet-switched communication networks. We cast our optimal routing problem in a multicommodity network flow optimization framework. Our cost function is related to the congestion in the network, and is a function of the flows on the links of the network. The optimization is over the set of flows in the links corresponding to the various destinations of the incoming traffic. We separately address the single commodity and the multicommodity versions of the routing problem. We consider the dual problems, and using dual decomposition techniques, we provide primal-dual algorithms that converge to the optimal solutions of the problems. Our algorithms, which are subgradient algorithms to solve the corresponding dual problems, can be implemented in a distributed manner by the nodes of the network. For online, adaptive implementations of our algorithms, the nodes in the network need to utilize 'locally available information' like estimates of queue lengths on outgoing links. We show convergence to the optimal routing solutions for synchronous versions of the algorithms, with perfect (noiseless) estimates of the queueing delays. Every node of the network controls the flows on the outgoing links using the distributed algorithms. For both the single commodity and multicommodity cases, we show that our algorithm converges to a loop-free optimal solution. Our optimal solutions also have the attractive property of being multipath routing solutions.

1. Introduction. We consider in this paper the routing problem in wireline, packet-switched communication networks. Such a network can be represented by a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{L})$, where \mathcal{N} denotes the set of nodes and \mathcal{L} the set of directed links of the network. For such a network we are given a set of origin-destination (OD) node pairs, with a certain incoming traffic rate (in packets per sec) associated with each OD pair. The arriving packets at the origin node have to be transported by the network to the corresponding destination node of the OD pair. The routing objective is to establish a packet traffic flow pattern so that packets are directed to their respective destinations while, at the same time, minimizing congestion in the network. A typical framework for accomplishing the same involves setting up an optimization problem, with the cost being related to the network-wide congestion and the constraints being the natural flow conservation relations in the network. For large-scale networks which are of interest to us, it is desired that the routing solution be implementable in a decentralized manner by the nodes of the network.

*Dedicated to Roger Brockett on the occasion of his 70th birthday.

[†]The authors are with the Institute for Systems Research and the Department of Electrical and Computer Engineering, University of Maryland College Park, USA. Research supported by the U.S. Army Research Laboratory under the CTA C & N Consortium Coop. Agreement DAAD19-01-2-0011, the National Aeronautics and Space Administration under Coop. Agreement NCC8235, and by the U.S. Army Research Office under MURI01 Grant No. DAAD19-01-1-0465 and under Grant No. DAAD19-02-1-0319.

An early important work on optimal routing for packet-switched communication networks was Gallager [20]. The cost considered was the sum of all the average link delays in the network, and a distributed algorithm was proposed to solve the problem. The algorithm preserved loop-freedom after each iteration, and converged to an optimal routing solution. Bertsekas, Gafni, and Gallager [4] proposed a means to improve the speed of convergence by using additional information of the second derivatives of link delays. Both the algorithms mentioned above require that related routing information like marginal delays be passed throughout the network before embarking on the next iteration. Another cost function that has been considered in the literature is a sum of the integrals of queueing delays; see Kelly [22] and Borkar and Kumar [8]. The solution to this problem can be characterized by the so-called Wardrop equilibrium [31] - between a source-destination pair, the delays along routes being actively used are all equal, and smaller than the delays of the inactive routes. Another formulation of the optimal routing problem, called the path flow formulation (see Bertsekas [1] and Bertsekas and Gallager [3]), has been in vogue. Tsitsiklis and Bertsekas [29] considered this formulation, and used a gradient projection algorithm to solve the optimal routing problem in a virtual circuit network setting. They also proved convergence of a distributed, asynchronous implementation of the algorithm. This formulation was considered by Elwalid, Jin, Low, and Widjaja [18] to accomplish routing in IP datagram networks using the Multi-Protocol Label Switching (MPLS) architecture. The routing functionality is shifted to the edges of the network (a feature of the path flow formulation; this is also known as ‘source routing’), and requires measurements of marginal delays on all paths linking a source and a destination. Because the number of such paths could scale exponentially as the network size grows, it is not clear that the solution would scale computationally. Multipath routing and flow control based on utility maximization have also been considered, notably in Kar, Sarkar, and Tassiulas [21] and in Wang, Palaniswami, and Low [30]. Kar, Sarkar and Tassiulas propose an algorithm which uses congestion indicators to effect control of flow rates on individual links, but do not explicitly consider congestion indicators based on queueing delays as we do (their congestion indicators are based on flows on links). Their algorithm avoids the scalability issue mentioned above. On the other hand, Wang, Palaniswami, and Low consider a path flow formulation of the multipath routing and flow control problem, which as we have mentioned above has scalability problems. However, they do consider the effect of queueing delays, extracting related information by measuring the round trip times. Recently, dual decomposition techniques have been used by Eryilmaz and Srikant [19], Lin and Shroff [24], Neely, Modiano, and Rohrs [25], and Chen, Low, Chiang, and Doyle [15], to design distributed primal-dual algorithms for optimal routing and scheduling in wireless networks. Such techniques consider the dual to the primal optimization problem and exploit separable structures in the costs and/or constraints to come up with a decomposition which automatically points the

way towards distributed implementations. The seminal work of Kelly, Maulloo, and Tan [23] showed how congestion control can be viewed in this way. The approach (called Network Utility Maximization (NUM)) has gained currency and has been applied to a variety of problems in wireline and wireless communication networks, cutting across all layers of the protocol stack (see [16]).

We cast our optimal routing problem in a multicommodity network flow optimization framework. Our cost function is related to the congestion in the network, and is a function of the flows on the links of the network. The optimization is over the set of flows in the links corresponding to the various destinations of the incoming traffic. We separately address the single commodity and the multicommodity versions of the routing problem. Our approach is to consider the dual optimization problems, and using dual decomposition techniques we provide primal-dual algorithms that converge to the optimal solutions of the problems. Our algorithms, which are subgradient algorithms to solve the corresponding dual problems, can be implemented in a distributed manner by the nodes of the network. For online, adaptive implementations of our algorithms, the nodes in the network need to utilize ‘locally available information’ like estimates of queue lengths on outgoing links. We show convergence to the optimal routing solutions for synchronous versions of the algorithms, with perfect (noiseless) estimates of the queueing delays (these essentially are the convergence results of the corresponding subgradient algorithms). Our optimal routing solution is not an end-to-end solution (our formulation is not a path-based formulation) like many of the above-cited works [29], [18], [30]. Consequently, our algorithms would avoid the scalability issues related to such an approach. Every node of the network controls the total as well as the commodity flows on the outgoing links using the distributed algorithms. Our optimal solutions also have the attractive property of being multipath routing solutions. Furthermore, by using a parameter (β) we can tune the optimal flow pattern in the network, so that more flow can be directed towards the links with higher capacities by increasing the parameter (we observe this in our numerical computations).

The Lagrange multipliers (dual variables) can be interpreted as potentials on the nodes for the single commodity case, and as potential differences across the links for the multicommodity case. Then, with every link of the network there is associated a characteristic curve (see Bertsekas [1] and Rockafellar [27]), which describes the relationship between the potential difference across the link and the link flow, the potential difference being thought of as ‘driving the flow through the link’. Using the relationships between the potential differences and the flows, we then provide simple proofs showing that our algorithm converges to a loop-free optimal solution, which is a desirable property. Our techniques are related to those employed in the literature on NUM methods.

The methods, results and approach presented here make substantial contact with,

and indeed are inspired from, several ideas and results of Roger Brockett through the years. Indeed our costs are expressed as *path integrals* and our methods lead to gradient and subgradient optimization interpretation in a very concrete way. These ideas are related to the work of Brockett in [9, 13, 10]. Brockett often developed “electrical network” intuition for several diverse and seemingly unrelated problems – see for instance [13, 10, 14, 6, 11]. Our approach and solutions have very much such an “electrical network” intuition, see in particular the comments at the end of the first part of section 3 and the monographs of Dennis [17] and Rockafellar [27]. Network flow optimization is indeed a generalization of the assignment problem (see [1]). In this context our methods and approach result in gradient and subgradient algorithms (even distributed ones) to solve these network flow optimization problems. In this context our results are related to the work of Brockett and his research collaborators in [14, 6, 11]. As we state at the end of the first part of section 3 our methods amount to solving flow optimization problems via analog computation, an idea that Brockett developed even for discrete optimization problems [7]. The fact that the form of our costs was inspired by our earlier work on the convergence of the so called “Ant Routing Algorithm” [26], indicates the close interrelationships between the routing costs and the stability analysis of routing algorithms, in a spirit related to the early work of Brockett in [9]. Finally, recently Brockett has turned his attention to queuing networks – see [12] and more recent work. Indeed at a recent workshop he suggested to one of us (Baras) that our work on trust in communication networks and on the related problem area of collaboration among nodes in networks, could be given an “electrical network” interpretation and thus be treated via optimization methods and gradient-subgradient algorithms – a suggestion we are pursuing.

Our paper is organized as follows. In the next section, we discuss the formulation of our optimal routing problem in detail. In Section 3 we investigate the special case of the single commodity problem, and in Section 4 we consider the general problem with multiple commodities (corresponding to different destinations of the incoming traffic). For both cases a few examples that illustrate the computations are also provided.

2. General Formulation of the Routing Problem. We now describe in brief our formulation. Let $r_i^{(k)} \geq 0$ denote the rate of input traffic entering the network at node i and destined for node k ¹. The flow on link (i, j) corresponding to packets bound for destination k is denoted by $f_{ij}^{(k)}$. The total flow on link (i, j) is denoted by F_{ij} and is given by $F_{ij} = \sum_{k \in \mathcal{N}} f_{ij}^{(k)}$. All packet flows having the same node as destination are said to belong to one commodity, irrespective of where they originate. Let C_{ij} denote the total capacity of link (i, j) . At node i , for every outgoing link (i, j) , there is an associated queue which is assumed to be of infinite size. Let $D_{ij}(F_{ij})$ denote

¹The arrival process is usually modeled as a stationary stochastic process, and $r_i^{(k)}$ then refers to the time averaged rate of the process.

the average packet delay in the queue when the total traffic flow through (i, j) is F_{ij} , with F_{ij} satisfying the constraint $0 \leq F_{ij} < C_{ij}$. (Quantities $F_{ij}, r_i^{(k)}, f_{ij}^{(k)}$ and C_{ij} are all expressed in the same units of packets/sec.)

Let \mathbf{f} denote the (column) vector of commodity link flows $f_{ij}^{(k)}, (i, j) \in \mathcal{L}, k \in \mathcal{N}$, in the network. We consider the following optimal routing problem:

Problem (A) : Minimize the (separable) cost function

$$G(\mathbf{f}) = \sum_{(i,j) \in \mathcal{L}} G_{ij}(F_{ij}) = \sum_{(i,j) \in \mathcal{L}} \int_0^{F_{ij}} u [D_{ij}(u)]^\beta du,$$

subject to

$$\begin{aligned} (1) \quad & \sum_{j:(i,j) \in \mathcal{L}} f_{ij}^{(k)} = r_i^{(k)} + \sum_{j:(j,i) \in \mathcal{L}} f_{ji}^{(k)}, \quad \forall i, k \neq i, \\ (2) \quad & f_{ij}^{(k)} \geq 0, \quad \forall (i, j) \in \mathcal{L}, k \neq i, \\ (3) \quad & f_{ij}^{(i)} = 0, \quad \forall (i, j) \in \mathcal{L}, \\ (4) \quad & F_{ij} = \sum_{k \in \mathcal{N}} f_{ij}^{(k)}, \quad \forall (i, j) \in \mathcal{L}, \end{aligned}$$

$$\text{with } 0 \leq F_{ij} < C_{ij}, \quad \forall (i, j) \in \mathcal{L}.$$

In an earlier work on convergence of Ant-Based Routing Algorithms [26], we showed, for a simple network involving N parallel links between a source-destination pair of nodes, that the equilibrium routing flows were such that they solved an optimization problem with a similar cost function and with similar capacity constraints as above. The scheme also yielded a multipath routing solution. It was natural to look for a generalization for the network case that has similar attractive properties. We shall see, using dual decomposition techniques, that the solution to our (optimization) Problem (A) is also a multipath routing solution, which can be implemented in a distributed manner by the nodes in the network. Our cost function is related to the network-wide congestion as measured by the link delays, and is small if the link delays are small. (Other cost functions have been used in the literature: in Gallager [20] and Bertsekas, Gallager and Gafni [4] it is of the form (in our notation) $D(\mathbf{f}) = \sum_{(i,j) \in \mathcal{L}} D_{ij}(F_{ij})$; and in the Wardrop routing formulation (see Kelly [22]) it is of the form $W(\mathbf{f}) = \sum_{(i,j) \in \mathcal{L}} \int_0^{F_{ij}} D_{ij}(u) du$.) The parameter β in our cost can be used to change the overall optimal flow pattern in the network. Roughly speaking, a low value of β results in the flows being more ‘uniformly distributed’ on the paths, whereas a high value of β tends to make the flows more concentrated on links lying on higher capacity paths.

Constraints (1) above are the per-commodity flow balance equations at the network nodes (flow out of the node = flow into the node), and constraints (3) express the fact that once a packet reaches its intended destination it is not routed back into the network. The optimization is over the set of link flow vectors \mathbf{f} .

3. The Single Commodity Problem: Formulation and Analysis. We consider in this section the single commodity problem, which involves routing of flows to a common destination node, which we label as d . We restate the problem for this special case in the following manner:

$$\mathbf{Problem (B)} : \text{Minimize } G(\mathbf{F}) = \sum_{(i,j) \in \mathcal{L}} G_{ij}(F_{ij}) = \sum_{(i,j) \in \mathcal{L}} \int_0^{F_{ij}} u [D_{ij}(u)]^\beta du,$$

subject to

$$(5) \quad \sum_{j:(i,j) \in \mathcal{L}} F_{ij} = r_i + \sum_{j:(j,i) \in \mathcal{L}} F_{ji}, \quad \forall i \in \mathcal{N},$$

$$(6) \quad F_{dj} = 0, \quad \text{for } (d,j) \in \mathcal{L},$$

$$\text{with } 0 \leq F_{ij} < C_{ij}, \quad \forall (i,j) \in \mathcal{L}.$$

r_i is the incoming rate for traffic arriving at node i , and destined for d . The optimization is over the set of link flow vectors \mathbf{F} , whose components are the individual link flows $F_{ij}, (i,j) \in \mathcal{L}$. As usual, equations (5) give the flow balance equations at every node and equations (6) refer to the fact that once a packet reaches d , it is not re-routed back into the network.

We use a dual decomposition technique of Bertsekas [1] to develop a distributed primal-dual algorithm that solves the above-stated optimal routing problem. We carry out our analysis under the following fairly natural assumptions. These assumptions are also used, almost verbatim, for the multicommodity version of the problem in Section 4.

Assumptions:

(A1) $D_{ij}(u)$ is a nondecreasing, continuously differentiable, positive real-valued function of u , defined over the interval $[0, C_{ij})$.

(A2) $\lim_{u \uparrow C_{ij}} D_{ij}(u) = +\infty$.

(A3) There exists at least one feasible solution of the primal Problem (B).

Assumption (A1) is a reasonable one, because when the flow u through a link increases, the average queueing delay (which is a function of the flow u) increases too. Assumption (A2) is satisfied for most queueing delay models of interest. Assumption (A3) implies that there exists a link flow pattern in the network such that the incoming traffic can be accommodated without the flow exceeding the capacity in any link.

We start the analysis by attaching prices (Lagrange multipliers) $p_i \in \mathbb{R}$, to the flow balance equations (5) and form the Lagrangian function $L(\mathbf{F}, \mathbf{p})$

$$L(\mathbf{F}, \mathbf{p}) = \sum_{(i,j) \in \mathcal{L}} G_{ij}(F_{ij}) + \sum_{i \in \mathcal{N}} p_i \left(\sum_{j:(j,i) \in \mathcal{L}} F_{ji} + r_i - \sum_{j:(i,j) \in \mathcal{L}} F_{ij} \right),$$

a function of the (column) price vector \mathbf{p} and the link flow vector \mathbf{F} . We can rearrange the Lagrangian to obtain the following convenient form

$$(7) \quad L(\mathbf{F}, \mathbf{p}) = \sum_{(i,j) \in \mathcal{L}} \left(G_{ij}(F_{ij}) - (p_i - p_j)F_{ij} \right) + \sum_{i \in \mathcal{N}} p_i r_i.$$

Using the Lagrangian, the dual function $Q(\mathbf{p})$ can be found by

$$Q(\mathbf{p}) = \inf L(\mathbf{F}, \mathbf{p}),$$

where the infimum is taken over all vectors \mathbf{F} , such that the components F_{ij} satisfy $0 \leq F_{ij} < C_{ij}$.

From the form (7) for the Lagrangian function, we can immediately see that

$$(8) \quad \begin{aligned} Q(\mathbf{p}) &= \sum_{(i,j) \in \mathcal{L}} \inf_{\{F_{ij}: 0 \leq F_{ij} < C_{ij}\}} \left(G_{ij}(F_{ij}) - (p_i - p_j)F_{ij} \right) + \sum_{i \in \mathcal{N}} p_i r_i, \\ &= \sum_{(i,j) \in \mathcal{L}} Q_{ij}(p_i - p_j) + \sum_{i \in \mathcal{N}} p_i r_i, \end{aligned}$$

where $Q_{ij}(p_i - p_j) = \inf_{\{F_{ij}: 0 \leq F_{ij} < C_{ij}\}} \left(G_{ij}(F_{ij}) - (p_i - p_j)F_{ij} \right)$. We can extend the definition of the function G_{ij} to the whole of \mathbb{R} , by simply setting it to be $+\infty$ outside $[0, C_{ij})$. Then the function $-Q_{ij}(p_i - p_j) = \sup_{\{F_{ij} \in \mathbb{R}\}} \left((p_i - p_j)F_{ij} - G_{ij}(F_{ij}) \right)$ is just the conjugate or the Legendre transform [1, 27] of the function G_{ij} .

The dual optimization problem, to Problem (B), is

$$\begin{aligned} \text{Problem (BD):} \quad & \text{Maximize } Q(\mathbf{p}) \\ & \text{subject to no constraints on } \mathbf{p} \text{ (i.e., } \mathbf{p} \in \mathbb{R}^{|\mathcal{N}|}). \end{aligned}$$

The dual function is a concave function and the dual optimization problem is a convex optimization problem. According to our Assumption (A3) and from the fact that $G_{ij}(F_{ij})$ is differentiable for every F_{ij} in $[0, C_{ij})$, with the derivative being $G'_{ij}(F_{ij}) = F_{ij}[D_{ij}(F_{ij})]^\beta$, there exists a regular ² primal feasible solution to our primal problem, Problem (B). Then, by Proposition 9.3 of Bertsekas [1], if \mathbf{F}^* is an optimal solution of the primal problem, there exists an optimal solution \mathbf{p}^* of the dual problem that satisfies the following Complementary Slackness (CS) conditions together with \mathbf{F}^*

$$(9) \quad G_{ij}(F_{ij}^*) - (p_i^* - p_j^*)F_{ij}^* = \inf_{\{F_{ij}: 0 \leq F_{ij} < C_{ij}\}} \left(G_{ij}(F_{ij}) - (p_i^* - p_j^*)F_{ij} \right), \quad \forall (i, j) \in \mathcal{L}.$$

Also, by Proposition 9.4 of Bertsekas [1], the optimal primal and dual costs are equal - that is, the duality gap is zero ³. Consider the minimization problem in the CS-condition (9) (for each link (i, j))

²A flow vector is called regular if for every link (i, j) , the left derivative $G_{ij}^-(F_{ij}) < \infty$, and the right derivative $G_{ij}^+(F_{ij}) > -\infty$ [1].

³This fact is nontrivial and a proof requires the techniques of monotropic programming [27], [1].

$$\begin{aligned} \text{Minimize } & G_{ij}(F_{ij}) - (p_i - p_j)F_{ij} = \int_0^{F_{ij}} u[D_{ij}(u)]^\beta du - (p_i - p_j)F_{ij} \\ \text{subject to } & 0 \leq F_{ij} < C_{ij}. \end{aligned}$$

The second derivative of G_{ij} is $G''_{ij}(F_{ij}) = [D_{ij}(F_{ij})]^\beta + \beta F_{ij}[D_{ij}(F_{ij})]^{\beta-1} D'_{ij}(F_{ij})$. Under our Assumption (A1), $G_{ij}(F_{ij})$ is twice continuously differentiable and strictly convex on the interval $[0, C_{ij}]$, so that the minimization problems above are all convex optimization problems on convex sets. We can show that for any price vector \mathbf{p} (in particular, for an optimal dual vector \mathbf{p}^*), there exists a unique $F_{ij} \in [0, C_{ij}]$ (for every (i, j)) which attains the minimum in the above optimization problem (Lemma 3, Appendix).

Conditions equivalent to (9) that an optimal primal-dual pair $(\mathbf{F}^*, \mathbf{p}^*)$ must satisfy are given by (for each $(i, j) \in \mathcal{L}$)

$$(10) \quad F_{ij}^* [D_{ij}(F_{ij}^*)]^\beta \geq p_i^* - p_j^*, \text{ if } F_{ij}^* = 0,$$

$$(11) \quad F_{ij}^* [D_{ij}(F_{ij}^*)]^\beta = p_i^* - p_j^*, \text{ if } F_{ij}^* > 0.$$

We also make the following observation. Suppose $p_i^* - p_j^* \leq 0$; then because for any $F_{ij} > 0$, $G_{ij}(F_{ij}) - (p_i^* - p_j^*)F_{ij} = \int_0^{F_{ij}} u[D_{ij}(u)]^\beta du - (p_i^* - p_j^*)F_{ij} > G_{ij}(0) - (p_i^* - p_j^*).0 = 0$, $F_{ij}^* = 0$ must then be the unique global minimum. Now, consider the contrapositive of (10) : if $p_i^* - p_j^* > 0$ then $F_{ij}^* > 0$. Thus, if $p_i^* - p_j^* > 0$ then F_{ij}^* is positive, and is given by the solution to the nonlinear equation

$$F_{ij}^* [D_{ij}(F_{ij}^*)]^\beta = p_i^* - p_j^*.$$

Because D_{ij} is a nondecreasing and continuously differentiable function, the above equation has a unique solution for F_{ij}^* .

To summarize, an optimal primal-dual pair $(\mathbf{F}^*, \mathbf{p}^*)$ is such that the following relationships are satisfied for each link (i, j) ,

$$(12) \quad F_{ij}^* = 0, \text{ if } p_i^* - p_j^* \leq 0,$$

$$(13) \quad F_{ij}^* [D_{ij}(F_{ij}^*)]^\beta = p_i^* - p_j^*, \text{ if } p_i^* - p_j^* > 0,$$

and in this case $F_{ij}^* > 0$. In analogy with electrical networks, the relations above can be interpreted as providing the ‘terminal characteristics’ of the ‘branch’ (i, j) . The Lagrange multipliers p_i^* can be thought of as ‘potentials’ on the nodes, and the flows F_{ij}^* as ‘currents’ on the links. The branch can be thought of as consisting of an ideal diode in series with a nonlinear current-dependent resistance. The difference of the ‘potentials’ or ‘voltage’ $p_i^* - p_j^*$, when positive, drives the ‘current’ or flow F_{ij}^* through a nonlinear flow-dependent resistance according to the law defined by (13). This analogy with electrical circuit theory helps in developing intuition. It was known (for the case of a quadratic cost function) to Maxwell, and was exploited by Dennis [17], who suggested that flow optimization problems with separable convex costs can be solved by setting up an electrical network with arcs having terminal characteristics

that can be derived in the same way as for our case. Once the network reaches equilibrium (starting from some initial condition), the currents and potentials can be simply ‘read off’ and are the optimal solutions to the primal and dual optimization problems, respectively. This amounts to solving the flow optimization problem using analog computation.

3.1. Distributed Solution of the Dual Optimization Problem. We now focus on solving the dual problem using a distributed primal-dual algorithm. We first make a quick remark on the differentiability properties of the dual function $Q(\mathbf{p})$. It can be verified that, for each (i, j) and (j, i) , the partial derivatives $\frac{\partial Q_{ij}(p_i - p_j)}{\partial p_i}$ and $\frac{\partial Q_{ji}(p_j - p_i)}{\partial p_i}$ exist for all $p_i \in \mathbb{R}$. Then, at any point \mathbf{p} , the partial derivatives $\frac{\partial Q(\mathbf{p})}{\partial p_i}$ all exist and can be easily seen to be given by

$$(14) \quad \frac{\partial Q(\mathbf{p})}{\partial p_i} = \sum_{j:(i,j) \in \mathcal{L}} \frac{\partial Q_{ij}(p_i - p_j)}{\partial p_i} + \sum_{j:(j,i) \in \mathcal{L}} \frac{\partial Q_{ji}(p_j - p_i)}{\partial p_i} + r_i, \quad i \in \mathcal{N}.$$

The gradient vector $\nabla Q(\mathbf{p})$ can thus be evaluated at each point \mathbf{p} .

The dual optimization problem can now be solved by the following simple gradient algorithm starting from an arbitrary initial price vector \mathbf{p}^0

$$(15) \quad \mathbf{p}^{n+1} = \mathbf{p}^n + \alpha_n \nabla Q(\mathbf{p}^n), \quad n \geq 0,$$

where $\{\alpha_n\}$ is a suitably chosen step-size sequence that ensures convergence of the gradient algorithm to an optimal dual vector \mathbf{p}^* . We now try to simplify the expression (14), and get it into a form that is suitable for computational purposes.

We had shown earlier that the minimum in the equation

$$Q_{ij}(p_i - p_j) = \inf_{\{F_{ij}: 0 \leq F_{ij} < C_{ij}\}} \left(G_{ij}(F_{ij}) - (p_i - p_j)F_{ij} \right)$$

is uniquely attained for each scalar $p_i - p_j$ by the flow F_{ij} which satisfies relations (12) and (13). Let us denote such a flow using the notation $F_{ij}(p_i - p_j)$, emphasizing its functional dependence on the price difference $p_i - p_j$. Then

$$Q_{ij}(p_i - p_j) = G_{ij}(F_{ij}(p_i - p_j)) - (p_i - p_j)F_{ij}(p_i - p_j).$$

Notice that $\frac{\partial Q_{ij}(p_i - p_j)}{\partial p_i} = -F_{ij}(p_i - p_j)$, and that $\frac{\partial Q_{ji}(p_j - p_i)}{\partial p_i} = F_{ji}(p_j - p_i)$, so that

$$(16) \quad \frac{\partial Q(\mathbf{p})}{\partial p_i} = \sum_{j:(j,i) \in \mathcal{L}} F_{ji}(p_j - p_i) - \sum_{j:(i,j) \in \mathcal{L}} F_{ij}(p_i - p_j) + r_i, \quad i \in \mathcal{N}.$$

The right hand side in the above equation can be interpreted to be the surplus flow at node i . The gradient algorithm (15) can now be written down in terms of iterations on the individual components of the price vector \mathbf{p} as follows

$$(17) \quad p_i^{n+1} = p_i^n + \alpha_n \left(\sum_{j:(j,i) \in \mathcal{L}} F_{ji}(p_j^n - p_i^n) - \sum_{j:(i,j) \in \mathcal{L}} F_{ij}(p_i^n - p_j^n) + r_i \right),$$

where for each (i, j) ,

$$(18) \quad F_{ij}(p_i^n - p_j^n) = 0, \quad \text{if } p_i^n - p_j^n \leq 0,$$

$$(19) \quad F_{ij}(p_i^n - p_j^n) > 0, \quad \text{if } p_i^n - p_j^n > 0,$$

and can be determined by solving the equation $F_{ij}[D_{ij}(F_{ij})]^\beta = p_i^n - p_j^n$. The relations (17), (18), and (19), can be used as a basis for a distributed algorithm that converges to a solution of the dual optimization problem. We now describe a general, online version of such an algorithm. The algorithm can be initialized with an arbitrary price vector \mathbf{p}^0 . (Each node can choose a real number as the initial value of its price variable.) Let's suppose that at the start of a typical iteration, the dual vector is \mathbf{p} , with each node i having available information of its own price p_i as well as the prices p_j of its neighbor nodes j such that $(i, j) \in \mathcal{L}$. Each node i makes an estimate of the average queueing delay on each of its outgoing links (i, j) . This estimate can be made by taking measurements of the packet delays over a time window and taking an average, or by using a 'running' estimator like an exponential averaging estimator. The flows $F_{ij}(p_i - p_j)$ are then determined by using relations (18) and (19), and node i adjusts the flows on its outgoing links to these values. Each node i then broadcasts the updated flow values to its neighbors j . In this way, every node i has information regarding the flows $F_{ij}(p_i - p_j)$ and $F_{ji}(p_j - p_i)$ on the links (i, j) and (j, i) , respectively. Node i can then use (17) to update its own dual variable p_i , and broadcasts to all the neighbor nodes this updated value. A fresh iteration can now commence with these updated dual variables. The general algorithm, as described, is adaptive and the updates of the dual variables and the flow variables in general take place asynchronously. As in [26], the outgoing flow on a link depends on the inverse of the estimated queueing delay on that link.

Although we have described a general, asynchronous, distributed algorithm for the dual problem, we shall discuss convergence only for the special case of the synchronous version as given by equations (17), (18), and (19), with 'perfect' ('noiseless') measurements of the delays. We can view the gradient algorithm (15) as a special case of a subgradient algorithm, and employ the results of convergence analysis (see, for example, Shor [28]) for the latter algorithm. For simplicity, we confine ourselves to a discussion of the constant step-size case - $\alpha_n = \alpha$, for some small, positive α . The central result is that, if the gradient vector $\nabla Q(\mathbf{p})$ has a bounded norm (that is, $\|\nabla Q(\mathbf{p})\| \leq G$, for some constant G , and for all \mathbf{p}), then for a small positive number h ,

$$(20) \quad Q(\mathbf{p}^*) - \lim_{n \rightarrow \infty} Q^n < h,$$

where Q^n is the 'best' value found till the n -th iteration, i.e., $Q^n = \max(Q(\mathbf{p}^0), Q(\mathbf{p}^1), \dots, Q(\mathbf{p}^n))$. The number h is a function of the step-size α , and decreases with it. In

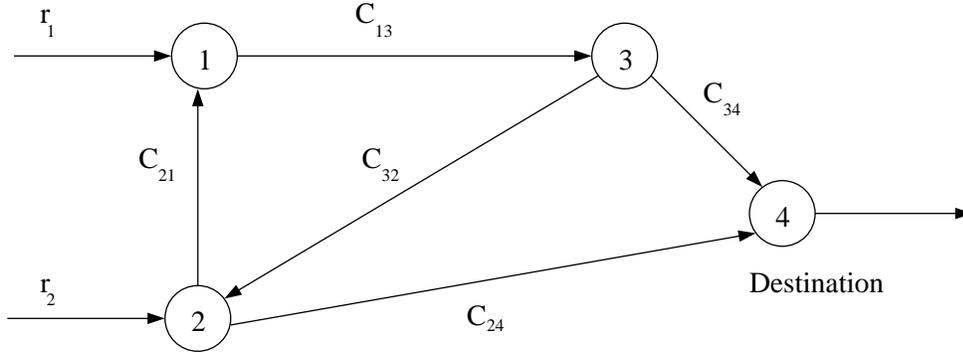


FIG. 1. The network topology and the traffic inputs : A single commodity example

our case, for any \mathbf{p} , the partial derivatives

$$\begin{aligned} \left| \frac{\partial Q(\mathbf{p})}{\partial p_i} \right| &= \left| \sum_{j:(j,i) \in \mathcal{L}} F_{ji}(p_j - p_i) - \sum_{j:(i,j) \in \mathcal{L}} F_{ij}(p_i - p_j) + r_i \right| \\ &\leq \sum_{j:(j,i) \in \mathcal{L}} C_{ji} + \sum_{j:(i,j) \in \mathcal{L}} C_{ij} + r_i \end{aligned}$$

are bounded, and so, the gradient vector $\nabla Q(\mathbf{p})$ is also (uniformly) upper bounded. The convergence result therefore holds in our case (with a constant step-size).

3.2. Loop Freedom of the Algorithm. Loop freedom is a desirable feature for any routing algorithm because communication resources are wasted if packets are routed in loops through the network. We now show, for the single commodity case under discussion, that the primal-dual algorithm converges to a set of optimal link flows, F_{ij}^* , $(i, j) \in \mathcal{L}$, which are loop free.

LEMMA 1. *An optimal link flow vector \mathbf{F}^* is loop free.*

Proof. Suppose that an optimal link flow vector \mathbf{F}^* is such that it forms a loop in the network. Then for some sequence of links $(i_1, i_2), (i_2, i_3), \dots, (i_n, i_1)$ that form a cycle, there is a positive flow on each of the links : $F_{i_1 i_2}^* > 0, F_{i_2 i_3}^* > 0, \dots, F_{i_n i_1}^* > 0$. This implies, by relation (11), that

$$p_{i_1}^* - p_{i_2}^* > 0, \quad p_{i_2}^* - p_{i_3}^* > 0, \dots, \quad p_{i_n}^* - p_{i_1}^* > 0,$$

which is impossible. (None of the nodes i_1, i_2, \dots, i_n above can be the destination node d , because of the condition (equation (6)) that flows are not re-routed back to the network once they reach the destination.) \square

3.3. An Example. We consider a simple example network in this subsection. We illustrate the computations and show how the allocation of flows to the links by the routing algorithm changes as the link capacities are changed, and that the optimal flow allocations avoid forming loops in the network.

TABLE 1
Optimal flows and ‘potentials’ ($C_{13} = 10, C_{21} = 4, C_{32} = 4, C_{34} = 14, C_{24} = 4$)

Optimal link flows	Optimal node potentials
$F_{13}^* = 6.89$	$p_1^* = 3.19$
$F_{21}^* = 0.89$	$p_2^* = 3.48$
$F_{24}^* = 3.11$	$p_3^* = 0.97$
$F_{34}^* = 6.89$	$p_4^* = 0.00$
$F_{32}^* = 0.00$	

The network that we consider is shown in Figure 1. There are incoming traffic flows at nodes 1 and 2, and the destination node is 4. The incoming traffic rates at nodes 1 and 2 are $r_1 = 6$ and $r_2 = 4$. C_{ij} denotes the capacities of the links (i, j) . In what follows, we assume that $\beta = 1$ and that the delay functions are of the form $D_{ij}(F_{ij}) = \frac{1}{C_{ij} - F_{ij}}$. (This is the commonly made $M/M/1$ approximation, also referred to as “Kleinrock’s independence assumption” [3].) This delay function satisfies our assumptions (A1) and (A2). We now set up the gradient algorithm.

For this case, the relations (18) and (19) reduce to

$$F_{ij}(p_i^n - p_j^n) = 0, \quad \text{if } p_i^n - p_j^n \leq 0,$$

$$F_{ij}(p_i^n - p_j^n) = \frac{(p_i^n - p_j^n)C_{ij}}{1 + p_i^n - p_j^n}, \quad \text{if } p_i^n - p_j^n > 0.$$

We set up the gradient algorithm with a small constant step-size, i.e., $\alpha_n = \alpha$, for some small positive α , and start with an arbitrary initial dual vector \mathbf{p}^0 . Each dual vector component is updated using the equation

$$p_i^{n+1} = p_i^n + \alpha \left(\sum_{j:(j,i) \in \mathcal{L}} F_{ji}(p_j^n - p_i^n) - \sum_{j:(i,j) \in \mathcal{L}} F_{ij}(p_i^n - p_j^n) + r_i \right),$$

where $F_{ij}(p_i^n - p_j^n)$ and $F_{ji}(p_j^n - p_i^n)$ are computed using the above equations.

The capacities of the links are $C_{13} = 10, C_{21} = 4, C_{32} = 4, C_{34} = 14, C_{24} = 4$, and α is set equal to 0.05. The optimal flows and potentials with this setting are tabulated in Table 1.

Keeping the capacities of the other links fixed, we can increase the capacity C_{24} until the entire flow that arrives at node 2 goes through the link $(2, 4)$ and no fraction traverses $(2, 1)$. This happens when C_{24} is increased to 8. The optimal flows and potentials that result are tabulated in Table 2.

If we now further increase the capacity C_{24} , because the flow coming in at node 3 now sees an additional available path that goes through node 2 to node 4 and which has high capacity, a fraction of the flow arriving at 3 goes through this path. The optimal flows and potentials when C_{24} is set at 16 are tabulated in Table 3.

The optimal flow allocations on links thus vary as the capacities vary relative to each other. In fact the routing algorithm can be seen as accomplishing a form of re-

TABLE 2

Optimal flows and ‘potentials’ ($C_{13} = 10, C_{21} = 4, C_{32} = 4, C_{34} = 14, C_{24} = 8$)

Optimal link flows	Optimal node potentials
$F_{13}^* = 6.00$	$p_1^* = 2.25$
$F_{21}^* = 0.00$	$p_2^* = 1.00$
$F_{24}^* = 4.00$	$p_3^* = 0.75$
$F_{34}^* = 6.00$	$p_4^* = 0.00$
$F_{32}^* = 0.00$	

TABLE 3

Optimal flows and ‘potentials’ ($C_{13} = 10, C_{21} = 4, C_{32} = 4, C_{34} = 14, C_{24} = 16$)

Optimal link flows	Optimal node potentials
$F_{13}^* = 6.00$	$p_1^* = 2.11$
$F_{21}^* = 0.00$	$p_2^* = 0.41$
$F_{24}^* = 4.67$	$p_3^* = 0.61$
$F_{34}^* = 5.33$	$p_4^* = 0.00$
$F_{32}^* = 0.67$	

source allocation (the resources being the link capacities) that helps relieve congestion in the network. Also the optimal flow allocations are always loop free.

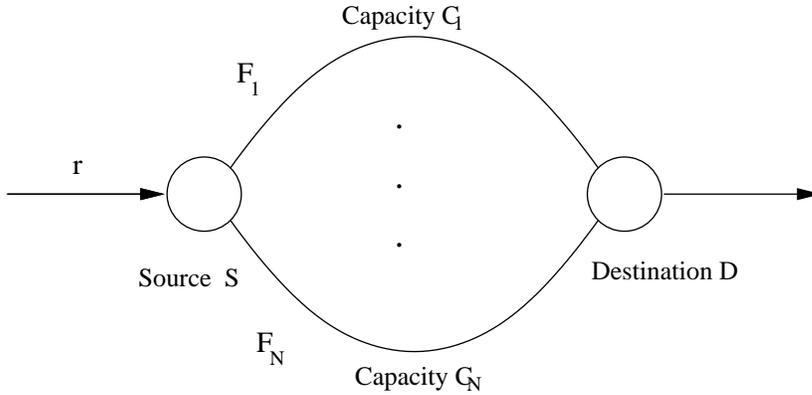
3.4. Effect of the parameter β . The positive constant β in the cost function ($\sum_{(i,j) \in \mathcal{L}} \int_0^{F_{ij}} u [D_{ij}(u)]^\beta du$) modulates the optimal outgoing flows on the links as can be seen from equations (12) and (13). Specifically, as β increases, more flow is diverted towards the outgoing links that lie on paths having higher link capacities. In this subsection, we show this for the special case of a network with one source and one destination, with N parallel paths (of capacities $C_i, i = 1, \dots, N$) joining them (Figure 2). The source has an incoming flow of r units which must be split between the N paths. The optimal routing problem is to find the optimal flows F_i^* that minimize the cost $\sum_{i=1}^N \int_0^{F_i} u [D_i(u)]^\beta du$, subject to the constraint $r = \sum_{i=1}^N F_i$. For this subsection, we denote the delay along the i -th path by $D_i(F_i) = D(F_i, C_i)$, the notation emphasizing the fact that the queueing delay function D (for instance, the $M/M/1$ delay function) is the same for all the paths. We assume that this function has the following properties: it is positive, and it is a strictly increasing function of F_i when C_i is held fixed, and a strictly decreasing function of C_i when F_i is held fixed.

It is easy to see that the optimal flows $F_i^*, i = 1, \dots, N$, satisfy the following equations

$$F_1^* \cdot D(F_1^*, C_1)^\beta = \dots = F_N^* \cdot D(F_N^*, C_N)^\beta,$$

$$F_1^* + \dots + F_N^* = r,$$

as well as the inequalities (queue stability conditions) $0 \leq F_i^* < C_i, i = 1, \dots, N$.

FIG. 2. The N parallel paths network

We denote this (unique) solution by $F^*(\beta) = (F_1^*(\beta), \dots, F_N^*(\beta))$, emphasizing its dependence on β .

Suppose now that $C_1 > C_2 = \dots = C_N$. Then using the relations

$$F_1^*(\beta)[D(F_1^*(\beta), C_1)]^\beta = \dots = F_N^*(\beta)[D(F_N^*(\beta), C_N)]^\beta,$$

one can check that

$$F_1^*(\beta) > F_2^*(\beta) = \dots = F_N^*(\beta)$$

and consequently that

$$(21) \quad D(F_1^*(\beta), C_1) < D(F_2^*(\beta), C_2) (= \dots = D(F_N^*(\beta), C_N)).$$

In this subsection, let's assume that β is a nonnegative real number (instead of being a positive integer). To arrive at a contradiction, let's suppose, for some small positive $\delta\beta$ that $F_1^*(\beta + \delta\beta) < F_1^*(\beta)$; then we also have $F_2^*(\beta + \delta\beta) > F_2^*(\beta)$. This implies that

$$(22) \quad \frac{F_1^*(\beta + \delta\beta)}{F_2^*(\beta + \delta\beta)} < \frac{F_1^*(\beta)}{F_2^*(\beta)}.$$

Using the relationships with the delays, we then have

$$\begin{aligned} \left[\frac{D(F_2^*(\beta + \delta\beta), C_2)}{D(F_1^*(\beta + \delta\beta), C_1)} \right]^{\beta + \delta\beta} &< \left[\frac{D(F_2^*(\beta), C_2)}{D(F_1^*(\beta), C_1)} \right]^\beta, \\ \left[\frac{D(F_2^*(\beta + \delta\beta), C_2)}{D(F_2^*(\beta), C_2)} \cdot \frac{D(F_1^*(\beta), C_1)}{D(F_1^*(\beta + \delta\beta), C_1)} \right]^{\beta + \delta\beta} &< \left[\frac{D(F_1^*(\beta), C_1)}{D(F_2^*(\beta), C_2)} \right]^{\delta\beta}. \end{aligned}$$

Using the hypothesis and the monotonicity property of the delay function with respect to the flow, it is easy to see that the left hand side of the above inequality is greater than one, which implies that

$$D(F_1^*(\beta), C_1) > D(F_2^*(\beta), C_2),$$

which contradicts the relation (21).

We now show that in fact $\frac{dF_1^*(\beta)}{d\beta} > 0$, when $C_1 > C_i, i = 2, \dots, N$. Then we have

$$F_1^*(\beta) > F_i^*(\beta), \quad i = 2, \dots, N,$$

and consequently that

$$(23) \quad D(F_1^*(\beta), C_1) < D(F_i^*(\beta), C_i), \quad i = 2, \dots, N.$$

From the equation $F_1^*(\beta) + F_2^*(\beta) + \dots + F_N^*(\beta) = r$, we have

$$(24) \quad \frac{dF_1^*(\beta)}{d\beta} = - \left(\frac{dF_2^*(\beta)}{d\beta} + \dots + \frac{dF_N^*(\beta)}{d\beta} \right).$$

Differentiating with respect to β the equation

$$F_1^*(\beta)[D(F_1^*(\beta), C_1)]^\beta = F_2^*(\beta)[D(F_2^*(\beta), C_2)]^\beta,$$

we have

$$(25) \quad \begin{aligned} & \frac{dF_1^*(\beta)}{d\beta} \left([D(F_1^*(\beta), C_1)]^\beta + \beta \cdot F_1^*(\beta) \cdot [D(F_1^*(\beta), C_1)]^{\beta-1} \cdot \frac{\frac{dD(F_1^*, C_1)}{dF_1^*}}{D(F_1^*(\beta), C_1)} \right) \\ & + F_1^*(\beta) \cdot [D(F_1^*(\beta), C_1)]^\beta \cdot \log D(F_1^*(\beta), C_1) \\ & = \frac{dF_2^*(\beta)}{d\beta} \left([D(F_2^*(\beta), C_2)]^\beta + \beta \cdot F_2^*(\beta) \cdot [D(F_2^*(\beta), C_2)]^{\beta-1} \cdot \frac{\frac{dD(F_2^*, C_2)}{dF_2^*}}{D(F_2^*(\beta), C_2)} \right) \\ & + F_2^*(\beta) \cdot [D(F_2^*(\beta), C_2)]^\beta \cdot \log D(F_2^*(\beta), C_2). \end{aligned}$$

Differentiating each of the equations

$$F_1^*(\beta)[D(F_1^*(\beta), C_1)]^\beta = F_i^*(\beta)[D(F_i^*(\beta), C_i)]^\beta, \quad i = 3, \dots, N,$$

we obtain $N - 2$ similar equations as equation (25) above. Adding all the equations we obtain the following relation

$$(26) \quad \begin{aligned} & (N-1) \frac{dF_1^*(\beta)}{d\beta} \left([D(F_1^*(\beta), C_1)]^\beta + \beta \cdot F_1^*(\beta) \cdot [D(F_1^*(\beta), C_1)]^{\beta-1} \cdot \frac{\frac{dD(F_1^*, C_1)}{dF_1^*}}{D(F_1^*(\beta), C_1)} \right) \\ & + (N-1) F_1^*(\beta) \cdot [D(F_1^*(\beta), C_1)]^\beta \cdot \log D(F_1^*(\beta), C_1) \\ & = \sum_{i=2}^N \left(\frac{dF_i^*(\beta)}{d\beta} \left([D(F_i^*(\beta), C_i)]^\beta + \beta \cdot F_i^*(\beta) \cdot [D(F_i^*(\beta), C_i)]^{\beta-1} \cdot \frac{\frac{dD(F_i^*, C_i)}{dF_i^*}}{D(F_i^*(\beta), C_i)} \right) \right. \\ & \left. + F_i^*(\beta) \cdot [D(F_i^*(\beta), C_i)]^\beta \cdot \log D(F_i^*(\beta), C_i) \right). \end{aligned}$$

Now let

$$\Gamma = \min_{i=2, \dots, N} \left([D(F_i^*(\beta), C_i)]^\beta + \beta \cdot F_i^*(\beta) \cdot [D(F_i^*(\beta), C_i)]^{\beta-1} \cdot \frac{\frac{dD(F_i^*, C_i)}{dF_i^*}}{D(F_i^*(\beta), C_i)} \right).$$

Γ is positive, because each of the terms in the minimum above is positive under our assumptions. Consequently, we can write

$$(27) \quad (N-1) \frac{dF_1^*(\beta)}{d\beta} \left([D(F_1^*(\beta), C_1)]^\beta + \beta \cdot F_1^*(\beta) \cdot [D(F_1^*(\beta), C_1)]^\beta \cdot \frac{\frac{dD(F_1^*, C_1)}{dF_1^*}}{D(F_1^*(\beta), C_1)} \right) \\ + (N-1) F_1^*(\beta) \cdot [D(F_1^*(\beta), C_1)]^\beta \cdot \log D(F_1^*(\beta), C_1) \\ \geq \Gamma \left(\frac{dF_2^*(\beta)}{d\beta} + \dots + \frac{dF_N^*(\beta)}{d\beta} \right) + \sum_{i=2}^N F_i^*(\beta) \cdot [D(F_i^*(\beta), C_i)]^\beta \cdot \log D(F_i^*(\beta), C_i).$$

Using equation (24) and transposing terms we obtain

$$(28) \quad \frac{dF_1^*(\beta)}{d\beta} \left((N-1) [D(F_1^*(\beta), C_1)]^\beta + \beta \cdot (N-1) \cdot F_1^*(\beta) \cdot [D(F_1^*(\beta), C_1)]^\beta \cdot \frac{\frac{dD(F_1^*, C_1)}{dF_1^*}}{D(F_1^*(\beta), C_1)} + \Gamma \right) \\ \geq \sum_{i=2}^N F_i^*(\beta) \cdot [D(F_i^*(\beta), C_i)]^\beta \cdot \log \frac{D(F_i^*(\beta), C_i)}{D(F_1^*(\beta), C_1)}.$$

We observe now that the coefficient of $\frac{dF_1^*(\beta)}{d\beta}$ on the left hand side of the inequality is positive. Also, using the fact that $F_1^*(\beta) [D(F_1^*(\beta), C_1)]^\beta = \dots = F_N^*(\beta) [D(F_N^*(\beta), C_N)]^\beta$ the expression on the right hand side can be written as

$$F_1^*(\beta) \cdot [D(F_1^*(\beta), C_1)]^\beta \cdot \sum_{i=2}^N \log \frac{D(F_i^*(\beta), C_i)}{D(F_1^*(\beta), C_1)},$$

and then using the relation (23), we have the desired result, $\frac{dF_1^*(\beta)}{d\beta} > 0$.

Thus, we can conclude that as $\beta \geq 0$ increases, more and more flow is diverted towards the outgoing links with more capacity, which shows that β can be seen as a tuning parameter that can be used to change the overall optimal flow pattern in the network. Increasing β tends to make the flows more concentrated on links lying on higher capacity paths, and decreasing β tends to make the flows more ‘spread out’ or ‘uniformly distributed’ on the paths (for instance, when $\beta = 0$, for the N parallel paths example, the outgoing flows on all the paths are equal: $F_1^*(0) = \dots = F_N^*(0) = \frac{r}{N}$). In our numerical studies on more general networks we have observed this property of our routing solution.

4. Analysis of the Optimal Routing Problem: The Multicommodity Case. Our approach takes as a starting point a decomposition technique that can be found, for example, in Rockafellar [27]. This technique decomposes a multicommodity flow optimization problem into a set of per link simple nonlinear convex flow problems and a set of per commodity linear network flow problems. We propose a primal-dual

approach to solve our optimal routing problem, Problem (A), utilizing this decomposition, aiming in particular to provide a solution that can be implemented in a completely distributed manner by the nodes themselves.

We carry out our analysis under the same assumptions (A1), (A2) and (A3) of Section 3, with a slight modification of Assumption (A3), whereby we now require that the primal Problem (A) has at least one feasible solution (we still refer to this assumption as Assumption (A3)).

We start by attaching Lagrange multipliers $z_{ij} \in \mathbb{R}$ to each of the constraints (4), and construct the Lagrangian function

$$L(\mathbf{f}, \mathbf{z}) = \sum_{(i,j) \in \mathcal{L}} G_{ij}(F_{ij}) + \sum_{(i,j) \in \mathcal{L}} z_{ij} \left(-F_{ij} + \sum_{k \in \mathcal{N}} f_{ij}^{(k)} \right),$$

where \mathbf{z} is the (column) vector of dual variables z_{ij} , $(i, j) \in \mathcal{L}$. The above equation can be rewritten in the following convenient form

$$(29) \quad L(\mathbf{f}, \mathbf{z}) = \sum_{(i,j) \in \mathcal{L}} \left(G_{ij}(F_{ij}) - z_{ij} F_{ij} \right) + \sum_k \sum_{(i,j) \in \mathcal{L}} z_{ij} f_{ij}^{(k)}.$$

The dual function $Q(\mathbf{z})$, which is a concave function, can then be written down as

$$(30) \quad Q(\mathbf{z}) = \min_{\mathbf{f}} L(\mathbf{f}, \mathbf{z}),$$

where the minimization is over all vectors \mathbf{f} satisfying the constraints (1), (2), and (3), and the capacity constraints $0 \leq F_{ij} < C_{ij}$, $(i, j) \in \mathcal{L}$. The form (29) of the Lagrangian function simplifies the computation of $Q(\mathbf{z})$, and we can obtain the following form for $Q(\mathbf{z})$

$$(31) \quad Q(\mathbf{z}) = Q_N(\mathbf{z}) + \sum_{k \in \mathcal{N}} Q_L^{(k)}(\mathbf{z}).$$

$Q_N(\mathbf{z})$ involves the solution of a set (one per link) of simple one-dimensional nonlinear optimization problems and is given by

$$(32) \quad Q_N(\mathbf{z}) = \sum_{(i,j) \in \mathcal{L}} \min_{0 \leq F_{ij} < C_{ij}} \left(G_{ij}(F_{ij}) - z_{ij} F_{ij} \right),$$

and for each commodity k , $Q_L^{(k)}(\mathbf{z})$ involves the solution of a linear network flow optimization problem with the costs associated with links (i, j) being the Lagrange multipliers z_{ij} ,

$$(33) \quad Q_L^{(k)}(\mathbf{z}) = \min_{\substack{f_{ij}^{(k)} \geq 0, \quad (i,j) \in \mathcal{L}, \\ \sum_j f_{ij}^{(k)} = r_i^{(k)} + \sum_j f_{ji}^{(k)}, \quad i \in \mathcal{N}}} \sum_{(i,j) \in \mathcal{L}} z_{ij} f_{ij}^{(k)},$$

the constraints being the commodity flow balance equations. An interesting interpretation of the above decomposition in terms of marginal costs and the notion of Wardrop equilibrium is provided in Rockafellar [27].

Once the dual function is available, the dual optimization problem can be cast as

Problem (AD): Maximize $Q(\mathbf{z})$
 subject to no constraint on \mathbf{z} (i.e., $\mathbf{z} \in \mathbb{R}^{|\mathcal{L}|}$).

Under our assumptions (A1) and (A3), a regular primal feasible (see [27]) solution to the optimization problem, Problem (A), exists. (This is because the function $G_{ij}(F_{ij})$ is differentiable, and the derivative $G'_{ij}(F_{ij}) = F_{ij}[D_{ij}(F_{ij})]^\beta$ is finite for F_{ij} in $[0, C_{ij})$.) Then it can be shown [27] that strong duality holds - the optimal primal and dual costs are equal ⁴. Suppose further that \mathbf{z}^* is an optimal solution to the dual optimization problem, and \mathbf{f}^* is an optimal solution to the primal optimization problem. Then $(\mathbf{f}^*, \mathbf{z}^*)$ solves the set of commodity linear optimization problems (33) (with the z_{ij} being set to z_{ij}^*). Also, for each $(i, j) \in \mathcal{L}$, the optimal total flow $F_{ij}^* = \sum_k f_{ij}^{(k)*}$, and satisfies along with z_{ij}^* the relation (equation (32))

$$(34) \quad G_{ij}(F_{ij}^*) - z_{ij}^* F_{ij}^* = \min_{0 \leq F_{ij} < C_{ij}} (G_{ij}(F_{ij}) - z_{ij}^* F_{ij}).$$

Now, for a given dual vector \mathbf{z} , let $\mathbf{F}(\mathbf{z})$ and $\mathbf{f}(\mathbf{z})$ be a pair of vectors that attain the minimum in (32) and (33), respectively. The components of $\mathbf{F}(\mathbf{z})$ are the flows $F_{ij}(\mathbf{z})$, and the components of $\mathbf{f}(\mathbf{z})$ are the flows $f_{ij}^{(k)}(\mathbf{z})$. We discuss in the following subsection how to compute $\mathbf{F}(\mathbf{z})$ and $\mathbf{f}(\mathbf{z})$ in a completely distributed manner, given a dual vector \mathbf{z} . We shall use this in Section 4.2 to develop a distributed primal-dual algorithm that solves the dual optimization problem. Because of strong duality and the discussion in the preceding paragraph, we shall have also obtained alongside the optimal flows \mathbf{f}^* .

4.1. Flow Vector Computations. For a given dual vector \mathbf{z} , we first turn our attention towards the problem of obtaining $\mathbf{F}(\mathbf{z})$. It is clear from the form of the expression in the right hand side of (32) that the computation can be arranged in a distributed manner, with each node i computing the flows $F_{ij}(\mathbf{z})$ on its outgoing links (i, j) by solving the problem

$$\begin{aligned} \text{Minimize } & G_{ij}(F_{ij}) - z_{ij} F_{ij} = \int_0^{F_{ij}} u [D_{ij}(u)]^\beta du - z_{ij} F_{ij}, \\ \text{subject to } & 0 \leq F_{ij} < C_{ij}. \end{aligned}$$

Under assumption (A1) this problem is a minimization problem of a strictly convex function over a convex set (arguments as in Section 3). Also, Lemma 3 of the Appendix shows that for every \mathbf{z} , there exists a unique minimum $F_{ij}(\mathbf{z})$ for the problem. An equivalent (necessary and sufficient) set of conditions that $F_{ij}(\mathbf{z})$ must satisfy are :

$$(35) \quad F_{ij}(\mathbf{z}) [D_{ij}(F_{ij}(\mathbf{z}))]^\beta \geq z_{ij}, \text{ if } F_{ij}(\mathbf{z}) = 0,$$

$$(36) \quad F_{ij}(\mathbf{z}) [D_{ij}(F_{ij}(\mathbf{z}))]^\beta = z_{ij}, \text{ if } F_{ij}(\mathbf{z}) > 0.$$

⁴The proof of this fact can be accomplished also by using the techniques of monotropic programming [27].

The relations (35) and (36) imply that

$$(37) \quad F_{ij}(\mathbf{z}) = 0, \text{ if } z_{ij} \leq 0,$$

$$(38) \quad F_{ij}(\mathbf{z})[D_{ij}(F_{ij}(\mathbf{z}))]^\beta = z_{ij}, \text{ if } z_{ij} > 0,$$

and in this case $F_{ij}(\mathbf{z}) > 0$ (arguments are similar to those in Section 3).

The relations (35) and (36) hold also for an optimal total flow and dual vector pair $(\mathbf{F}(\mathbf{z}^*), \mathbf{z}^*)$. At every node i the optimal total flows on its outgoing links could be positive or zero, depending on the capacities of the links. We can thus, in general, have a multipath routing solution to our optimal routing problem. The outgoing *total* flow $F_{ij}(\mathbf{z}^*)$, when positive, depends on the inverse of the average link delay.

For a given vector \mathbf{z} , we now focus on solving the commodity linear flow optimization problems (33). For each commodity k , solving the optimization problem gives the flows $f_{ij}^{(k)}(\mathbf{z})$, $(i, j) \in \mathcal{L}$. We use the ϵ -relaxation method (Bertsekas and Tsitsiklis [5], Bertsekas and Eckstein [2]), because it can be implemented in a purely distributed manner by the nodes in the network. The method is an algorithmic procedure to solve the dual to the primal linear flow optimization problem and is based on the notion of ϵ -complementary slackness. ϵ -complementary slackness involves a modification of the usual complementary slackness relations of the linear optimization problem by a small amount ϵ . At every iteration, the algorithm changes the dual prices and the incoming and outgoing link flows at every node i , while maintaining ϵ -complementary slackness and improving the value of the dual cost at the same time. We briefly provide an overview in the following paragraphs (for details see, for example, Bertsekas and Tsitsiklis [5]).

Consider the linear network flow problem for commodity k : Minimize the cost $\sum_{(i,j) \in \mathcal{L}} z_{ij} f_{ij}^{(k)}$, subject to the flow balance constraints $\sum_j f_{ij}^{(k)} = r_i^{(k)} + \sum_j f_{ji}^{(k)}$, for each node i , and the constraints $f_{ij}^{(k)} \geq 0$, for each link (i, j) . We also add the constraint $f_{ij}^{(k)} \leq C_{ij}$ (which must be satisfied at optimality), which enables us to apply the method without making any modifications. The dual problem is formulated by first attaching Lagrange multipliers (prices) $p_i \in \mathbb{R}$ to the balance equations at each node i , and forming the Lagrangian $M = \sum_{(i,j) \in \mathcal{L}} (z_{ij} f_{ij}^{(k)} - (p_i - p_j) f_{ij}^{(k)}) + \sum_{i \in \mathcal{N}} r_i^{(k)} p_i$. For a price vector \mathbf{p} and given $\epsilon > 0$, a set of flows and prices satisfies the ϵ -complementary slackness conditions if the flows satisfy the capacity constraints, and if the following are satisfied

$$\begin{aligned} f_{ij}^{(k)} < C_{ij} &\implies p_i - p_j \leq z_{ij} + \epsilon, \\ f_{ij}^{(k)} > 0 &\implies p_i - p_j \geq z_{ij} - \epsilon. \end{aligned}$$

The ϵ -relaxation method uses a fixed ϵ , and tries to solve the dual optimization problem using distributed computation. The procedure starts by considering an arbitrary initial price vector \mathbf{p}^0 , and then finds a set of flows on the links such that the flow-price pair satisfies the ϵ -complementary slackness conditions. At each iteration, the flow

surplus at nodes i , $g_i = \sum_j f_{ji}^{(k)} + r_i^{(k)} - \sum_j f_{ij}^{(k)}$, are computed. A node i with positive surplus is chosen. (If all nodes have zero surplus, then the algorithm terminates, because the ϵ -complementary slackness conditions are satisfied and the flow balance conditions, too, are satisfied. The corresponding flow vector \mathbf{f} is optimal.) The surplus g_i is driven to zero at the iterative step, and another flow-price pair satisfying ϵ -complementary slackness is produced. At the same time the dual function ($Q(\mathbf{p})$) value is increased, by changing the i -th price coordinate p_i . At the iteration, except possibly for price p_i at node i , the prices of the other nodes are left unchanged. It can be shown (even for z_{ij} and C_{ij} that are not necessarily integers, which is the case of interest to us; see [5]) that, if ϵ is chosen small enough, the algorithm converges to an optimal flow-price pair.

Besides the ϵ -relaxation method, there exist other distributed algorithms, like the auction algorithm [1], that solve linear flow optimization problems. Any such algorithm can be used to solve the linear flow optimization problems at hand.

4.2. Distributed Solution of the Dual Optimization Problem. We now focus on solving the dual problem using a distributed primal-dual algorithm. To that end, we first note that the dual function $Q(\mathbf{z})$ is a non-differentiable function of \mathbf{z} . This suggests that we can use a subgradient based iterative algorithm to compute an optimal dual vector \mathbf{z}^* . We shall see that the computations can be made completely distributed.

We first compute a subgradient for the concave function $Q(\mathbf{z})$ at a point \mathbf{z} (in $\mathbb{R}^{|\mathcal{L}|}$). Recall that a vector $\boldsymbol{\delta}(\mathbf{z})$ is a subgradient of a concave function Q at \mathbf{z} if

$$(39) \quad Q(\mathbf{w}) \leq Q(\mathbf{z}) + \boldsymbol{\delta}(\mathbf{z})^T (\mathbf{w} - \mathbf{z}), \quad \forall \mathbf{w} \in \mathbb{R}^{|\mathcal{L}|}.$$

Recall that for a given vector \mathbf{z} , $\mathbf{F}(\mathbf{z})$ and $\mathbf{f}(\mathbf{z})$ denote a pair of vectors of total flows and commodity flows that attain the minimum in (32) and in (33), respectively. For vectors \mathbf{z} , \mathbf{w} , we have

$$\begin{aligned} Q(\mathbf{w}) - Q(\mathbf{z}) &= \sum_{(i,j) \in \mathcal{L}} \left(G_{ij}(F_{ij}(\mathbf{w})) - w_{ij} F_{ij}(\mathbf{w}) \right) + \sum_k \sum_{(i,j) \in \mathcal{L}} w_{ij} f_{ij}^{(k)}(\mathbf{w}) \\ &\quad - \sum_{(i,j) \in \mathcal{L}} \left(G_{ij}(F_{ij}(\mathbf{z})) - z_{ij} F_{ij}(\mathbf{z}) \right) - \sum_k \sum_{(i,j) \in \mathcal{L}} z_{ij} f_{ij}^{(k)}(\mathbf{z}). \end{aligned}$$

Because

$$\begin{aligned} &\sum_{(i,j) \in \mathcal{L}} \left(G_{ij}(F_{ij}(\mathbf{w})) - w_{ij} F_{ij}(\mathbf{w}) \right) + \sum_k \sum_{(i,j) \in \mathcal{L}} w_{ij} f_{ij}^{(k)}(\mathbf{w}) \\ &\leq \sum_{(i,j) \in \mathcal{L}} \left(G_{ij}(F_{ij}(\mathbf{z})) - w_{ij} F_{ij}(\mathbf{z}) \right) + \sum_k \sum_{(i,j) \in \mathcal{L}} w_{ij} f_{ij}^{(k)}(\mathbf{z}), \end{aligned}$$

we have

$$\begin{aligned} Q(\mathbf{w}) - Q(\mathbf{z}) &\leq \sum_{(i,j) \in \mathcal{L}} \left(G_{ij}(F_{ij}(\mathbf{z})) - w_{ij}F_{ij}(\mathbf{z}) \right) + \sum_k \sum_{(i,j) \in \mathcal{L}} w_{ij} f_{ij}^{(k)}(\mathbf{z}) \\ &\quad - \sum_{(i,j) \in \mathcal{L}} \left(G_{ij}(F_{ij}(\mathbf{z})) - z_{ij}F_{ij}(\mathbf{z}) \right) - \sum_k \sum_{(i,j) \in \mathcal{L}} z_{ij} f_{ij}^{(k)}(\mathbf{z}). \end{aligned}$$

and so

$$Q(\mathbf{w}) - Q(\mathbf{z}) \leq \sum_{(i,j) \in \mathcal{L}} (w_{ij} - z_{ij}) \left(\sum_k f_{ij}^{(k)}(\mathbf{z}) - F_{ij}(\mathbf{z}) \right),$$

which shows that a subgradient of Q at \mathbf{z} is the $|\mathcal{L}|$ -vector $\boldsymbol{\delta}(\mathbf{z})$ with components $\sum_k f_{ij}^{(k)}(\mathbf{z}) - F_{ij}(\mathbf{z})$.

Consequently, in order to solve the dual optimization problem, we can set up the following subgradient iterative procedure, starting with an arbitrary initial vector of Lagrange multipliers \mathbf{z}^0 ,

$$(40) \quad \mathbf{z}^{n+1} = \mathbf{z}^n + \gamma_n \boldsymbol{\delta}(\mathbf{z}^n), \quad n \geq 0,$$

where $\{\gamma_n\}$ is a suitably chosen step-size sequence that ensures convergence of the above subgradient iterations to an optimal dual vector \mathbf{z}^* . The vector $\boldsymbol{\delta}(\mathbf{z}^n)$, with components $\sum_k f_{ij}^{(k)}(\mathbf{z}^n) - F_{ij}(\mathbf{z}^n)$, is a subgradient of Q at \mathbf{z}^n . In terms of the individual components, the subgradient algorithm (40) can be written as (for each $(i, j) \in \mathcal{L}$)

$$(41) \quad z_{ij}^{n+1} = z_{ij}^n + \gamma_n \left(\sum_k f_{ij}^{(k)}(\mathbf{z}^n) - F_{ij}(\mathbf{z}^n) \right).$$

The equation (41) above shows that the subgradient iterative procedure can be implemented in a distributed manner at the various nodes of the network. At each node i , to update the dual variables z_{ij} for the outgoing links (i, j) , the quantities that are required are the optimal commodity flows $f_{ij}^{(k)}(\mathbf{z})$ and total flows $F_{ij}(\mathbf{z})$. In Section 4.1 we showed how the above quantities can be computed in a completely distributed manner by every node, given \mathbf{z} . $F_{ij}(\mathbf{z}^n)$ can be computed (exactly as in Section 3.1) using estimates of average queueing delays on the outgoing links and using (37) and (38). Computation of the flows $F_{ij}(\mathbf{z}^n)$ and $f_{ij}(\mathbf{z}^n)$ require message exchange with neighbor nodes, and local information-like estimates of the outgoing links' queue lengths. The updated dual variables z_{ij} are broadcast to the neighbor nodes, which utilize this information in the execution of their iterations. In general, the updates of the dual variables and the flows take place asynchronously, so that the algorithm is asynchronous and adaptive.

We now briefly discuss the convergence behavior of the subgradient algorithm (40). As for the single commodity case, we restrict our attention to the synchronous

version of the algorithm as given by equation (40), and we consider only the constant step size case here - $\gamma_n = \gamma$, for all n and some small, positive γ . If the subgradient vector is bounded in norm, then the subgradient algorithm converges arbitrarily closely to the optimal point. As in Section 3.1, the sense in which convergence takes place is the following : for a small positive number h , we have

$$Q(\mathbf{z}^*) - \lim_{n \rightarrow \infty} Q^n < h,$$

where Q^n is the ‘best’ value found till the n -th iteration, i.e., $Q^n = \max(Q(\mathbf{z}^0), \dots, Q(\mathbf{z}^n))$. In our case, because the commodity flows $f_{ij}^{(k)}(\mathbf{z})$ and total flows $F_{ij}(\mathbf{z})$ are always bounded (because of the capacity constraints), the subgradients $\delta(\mathbf{z})$ are bounded in norm, and the subgradient algorithm converges. Shor [28] contains other attractive (albeit more involved) step-size rules, including diminishing step-size rules, which have more attractive convergence properties. This is an avenue for future exploration. Upon convergence, the algorithm yields simultaneously, the optimal dual vector \mathbf{z}^* , as well as the optimal flow vectors $\mathbf{F}(\mathbf{z}^*)$ and $\mathbf{f}(\mathbf{z}^*)$.

4.3. Loop Freedom of the Algorithm. In this subsection we show that an optimal flow vector $\mathbf{f}(\mathbf{z}^*)$ is loop free.

LEMMA 2. *An optimal flow vector $\mathbf{f}(\mathbf{z}^*)$ is loop free.*

Proof. Suppose that an optimal flow vector $\mathbf{f}(\mathbf{z}^*)$ is not loop free. Then for some commodity k , and for some sequence of links $(i_1, i_2), (i_2, i_3), \dots, (i_n, i_1)$ that form a cycle, there is a positive flow on each of the links : $f_{i_1 i_2}^{(k)}(\mathbf{z}^*) > 0, f_{i_2 i_3}^{(k)}(\mathbf{z}^*) > 0, \dots, f_{i_n i_1}^{(k)}(\mathbf{z}^*) > 0$. Consequently, for the total flows we have $F_{i_1 i_2}(\mathbf{z}^*) > 0, F_{i_2 i_3}(\mathbf{z}^*) > 0, \dots, F_{i_n i_1}(\mathbf{z}^*) > 0$. This implies, by relation (36), that

$$z_{i_1 i_2}^* > 0, z_{i_2 i_3}^* > 0, \dots, z_{i_n i_1}^* > 0.$$

On the other hand, the optimal flows $f_{ij}^{(k)}(\mathbf{z}^*), (i, j) \in \mathcal{L}$, constitute a solution to the linear programming problem: Minimize the cost $\sum_{(i,j) \in \mathcal{L}} z_{ij}^* f_{ij}^{(k)}$, subject to the constraints $\sum_j f_{ij}^{(k)} = r_i^{(k)} + \sum_j f_{ji}^{(k)}, i \in \mathcal{N}$, and the constraints $0 \leq f_{ij}^{(k)} < C_{ij}, (i, j) \in \mathcal{L}$. Attach Lagrange multipliers $p_i \in \mathbb{R}$ to the balance equations at each node i , and form the Lagrangian $N = \sum_{(i,j) \in \mathcal{L}} \left(z_{ij}^* f_{ij}^{(k)} - (p_i - p_j) f_{ij}^{(k)} \right) + \sum_{i \in \mathcal{N}} r_i^{(k)} p_i$. An optimal primal-dual vector pair $(\mathbf{f}(\mathbf{z}^*), \mathbf{p}^*)$ satisfies the following Complementary Slackness conditions (the derivation is similar to the derivation for equations (35) and (36))

$$\begin{aligned} z_{ij}^* &\geq p_i^* - p_j^*, & \text{if } f_{ij}^{(k)}(\mathbf{z}^*) = 0 \\ \text{and } z_{ij}^* &= p_i^* - p_j^*, & \text{if } f_{ij}^{(k)}(\mathbf{z}^*) > 0. \end{aligned}$$

From the foregoing it is clear that we must have

$$p_{i_1}^* - p_{i_2}^* > 0, p_{i_2}^* - p_{i_3}^* > 0, \dots, p_{i_n}^* - p_{i_1}^* > 0,$$

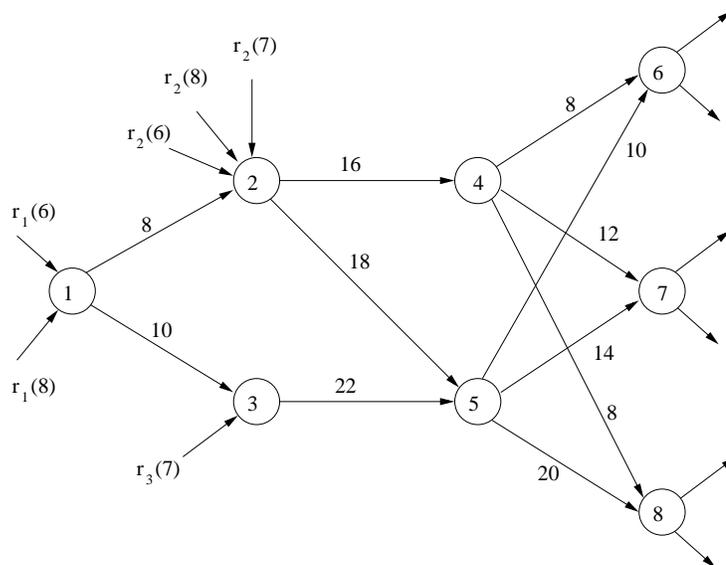


FIG. 3. The network topology and the traffic inputs : A Multicommodity Example

which is a contradiction. \square

4.4. An Illustrative Example. We consider an example network in this section and illustrate the computations. The network consists of eight nodes interconnected by multiple directed links. Figure 3 shows the network topology. The numbers beside the links are the capacities of the links. There are multiple sources and multiple sinks of traffic. The rates of input traffic at the sources are given by $r_1^{(6)} = 6$, $r_1^{(8)} = 8$, $r_2^{(6)} = 8$, $r_2^{(8)} = 6$, $r_2^{(7)} = 10$, $r_3^{(7)} = 10$. There are three commodities in the network corresponding to the three destinations for the traffic flows in the network. The capacities are such that the network is able to accommodate the incoming traffic to the network.

As in the single commodity example we assume that the delay functions $D_{ij}(F_{ij})$ are explicitly given by the formula $D_{ij}(F_{ij}) = \frac{1}{C_{ij} - F_{ij}}$. We carry out the numerical computations for the case when $\beta = 1$.

We set up the subgradient iterative algorithm (41) starting from an arbitrary initial vector \mathbf{z}^0

$$z_{ij}^{n+1} = z_{ij}^n + \gamma_n \left(\sum_k f_{ij}^{(k)}(\mathbf{z}^n) - F_{ij}(\mathbf{z}^n) \right), \quad (i, j) \in \mathcal{L},$$

where the flow vectors $\mathbf{F}(\mathbf{z}^n)$ and $\mathbf{f}(\mathbf{z}^n)$ are computed as outlined in Section 4.1. As we had noted in that section, computing $\mathbf{F}(\mathbf{z}^n)$ translates to satisfying the relations

(37) and (38), which in our example are the equations

$$F_{ij}(\mathbf{z}^n) = 0, \text{ if } z_{ij}^n \leq 0,$$

$$\frac{F_{ij}(\mathbf{z}^n)}{C_{ij} - F_{ij}(\mathbf{z}^n)} = z_{ij}^n, \text{ if } z_{ij}^n > 0.$$

The latter equation gives $F_{ij}(\mathbf{z}^n) = \frac{z_{ij}^n C_{ij}}{1+z_{ij}^n}$, a simple expression, showing that the flow is proportional to the capacity.

We use a constant step-size algorithm ($\gamma_n = \gamma$, for all n) with the step-size $\gamma = 0.01$. (This choice of small γ slows down the convergence of the algorithm. As pointed out in Section 4.2, other choices of step-size sequences can potentially improve the speed of convergence.) The ϵ chosen for the ϵ -relaxation method is 0.01.

The subgradient algorithm converges and the optimal flows (upon convergence) are tabulated in Table 4. As in Section 3 we note that the optimal routing solution allocates a higher fraction of total incoming flows at every node to outgoing links that lie on paths consisting of higher capacity links.

The optimal routing solution splits the total incoming flow at each node among the outgoing links. The solution also describes how the total flow on each link is split among the commodity flows. We also note that our routing solution is a multipath routing solution. It is well-known [3] that multipath routing solutions improve the overall network performance by avoiding routing oscillations (shortest-path routing solutions, for instance, are known to lead to routing oscillations), and by providing better throughput for incoming connections, while at the same time reducing the average network delay.

Our routing solution is not an end-to-end routing solution, as for example [18]. The control is not effected from the end hosts, but every node i in the network controls both the total as well as the commodity flows on its outgoing links (i, j) , using the distributed algorithm.

Appendix A. LEMMA 3. *Under our Assumptions (A1) and (A2), there exists a unique solution to the following minimization problem (for any given w_{ij}),*

$$\text{Minimize } G_{ij}(F_{ij}) - w_{ij}F_{ij} = \int_0^{F_{ij}} u[D_{ij}(u)]^\beta du - w_{ij}F_{ij},$$

$$\text{subject to } 0 \leq F_{ij} < C_{ij}.$$

Proof. For any given w_{ij} , $H_{ij}(F_{ij}) = G_{ij}(F_{ij}) - w_{ij}F_{ij}$ increases to $+\infty$ as $F_{ij} \uparrow C_{ij}$ (Assumption (A2)). Consequently, there exists an $M \in [0, C_{ij})$, such that $H_{ij}(F_{ij}) > H_{ij}(0)$ whenever $F_{ij} > M$. The function $H_{ij}(F_{ij})$ restricted to the domain $[0, M]$ attains the (global) minimum at the same point as the function considered on the set $[0, C_{ij})$. The set $[0, M]$ is compact; applying Weierstrass theorem to the continuous function $H_{ij}(F_{ij})$ on this set gives us the required existence of a minimum. Uniqueness follows from the fact that $H_{ij}(F_{ij})$ is strictly convex on $[0, C_{ij})$. \square

TABLE 4
Optimal flows in links

Link (i, j)	Optimal total flow F_{ij}^*	Optimal commodity flows
(1, 2)	5.77	$f_{12}^{(6)*} = 0, f_{12}^{(8)*} = 5.77$
(1, 3)	8.23	$f_{13}^{(6)*} = 6.00, f_{13}^{(8)*} = 2.23$
(2, 4)	14.31	$f_{24}^{(6)*} = 5.86, f_{24}^{(7)*} = 8.19, f_{24}^{(8)*} = 0$
(2, 5)	15.77	$f_{25}^{(6)*} = 2.14, f_{25}^{(7)*} = 1.82, f_{25}^{(8)*} = 11.77$
(3, 5)	18.23	$f_{35}^{(6)*} = 6.00, f_{35}^{(7)*} = 10.00, f_{35}^{(8)*} = 2.23$
(4, 6)	5.86	$f_{46}^{(6)*} = 5.86$
(4, 7)	8.19	$f_{47}^{(7)*} = 8.19$
(4, 8)	0	$f_{48}^{(8)*} = 0$
(5, 6)	8.23	$f_{56}^{(6)*} = 8.14$
(5, 7)	11.84	$f_{57}^{(7)*} = 11.82$
(5, 8)	14.00	$f_{58}^{(8)*} = 14.00$

REFERENCES

- [1] D. P. BERTSEKAS, *Network Optimization: Continuous and Discrete Models*, Athena Scientific, Belmont, MA, 1998.
- [2] D. P. BERTSEKAS AND J. ECKSTEIN, *Dual Coordinate Step Methods for Linear Network Flow Problems*, Math. Programming, Series B, 42(1988), pp. 203-243.
- [3] D. P. BERTSEKAS AND R. G. GALLAGER, *Data Networks*, Second Edition, Prentice Hall, Englewood Cliffs, NJ, 1992.
- [4] D. P. BERTSEKAS, E. GAFNI, AND R.G. GALLAGER, *Second Derivative Algorithms for Minimum Delay Distributed Routing in Networks*, IEEE Trans. on Communications, 32(1984), pp. 911 – 919.
- [5] D. P. BERTSEKAS AND J. N. TSITSIKLIS, *Parallel and Distributed Computation : Numerical Methods*, Prentice-Hall, Englewood Cliffs, NJ, 1989.
- [6] A. M. BLOCH, R. W. BROCKETT, AND T. S. RATIU, *Completely Integrable Gradient Flows*, Communications in Mathematical Physics, 147(1992), pp. 57 – 74.
- [7] A. M. BLOCH, R. W. BROCKETT, AND P. CROUCH, *Double Bracket Flows and Geodesic Equations on Symmetric Spaces*, Communications on Mathematical Physics, pp. 1 – 16, 1997.
- [8] V. S. BORKAR AND P. R. KUMAR, *Dynamic Cesaro-Wardrop Equilibration in Networks*, IEEE Trans. on Automatic Control, 48:3(2003), pp. 382-396.
- [9] R. W. BROCKETT, *Path Integrals, Liapunov Functions, and Quadratic Minimization*, Proc. of the 4th Allerton Conference, Urbana, IL, University of Illinois, pp. 685 – 697, 1966.
- [10] R. W. BROCKETT, *Finite Dimensional Linear Systems*, New York: John Wiley and Sons, 1970.
- [11] R. W. BROCKETT, *Differential Geometry and the Design of Gradient Algorithms*, in: Differential Geometry, Robert Green and S.T. Yau, eds., Proceedings of Symposia in Pure Math, American Math. Soc., pp. 69 – 92, 1992.
- [12] R. W. BROCKETT, *Stochastic Analysis for Fluid Queuing Systems*, Proc. of the 1999 CDC Conference, Phoenix AZ, pp. 3077 – 3082, 1999.
- [13] R. W. BROCKETT AND R. A. SKOOG, *A New Perturbation Theory for the Synthesis of Nonlinear Networks*, Mathematical Aspects of Electrical Network Analysis, SIAM-AMS Proc., Vol. III, pp. 17 – 33, 1970.
- [14] R. W. BROCKETT AND W. S. WONG, *A Gradient Flow for the Assignment Problem*, Progress

- in System and Control Theory, G. Conte and B. Wyman, eds., pp. 170–177, Birkhauser, 1991.
- [15] L. CHEN, S. H. LOW, M. CHIANG, AND J. C. DOYLE, *Cross-Layer Congestion Control, Routing and Scheduling Design in Ad hoc Wireless Networks*, Proc. IEEE INFOCOM, pp. 1–13, 2006.
- [16] M. CHIANG, S. H. LOW, A. R. CALDERBANK, AND J. C. DOYLE, *Layering as Optimization Decomposition : A Mathematical Theory of Network Architectures*, Proc. of the IEEE, 95:1(2007), pp. 255–312.
- [17] J. B. DENNIS, *Mathematical Programming and Electrical Networks*, Technology Press of M.I.T, Cambridge, MA, 1959.
- [18] A. ELWALID, C. JIN, S. LOW, AND I. WIDJAJA *MATE: MPLS Adaptive Traffic Engineering*, Computer Networks, 40:6(2002), pp. 695–709.
- [19] A. ERYILMAZ AND R. SRIKANT, *Joint Congestion Control, Routing, and MAC for Stability and Fairness in Wireless Networks*, IEEE J. on Sel. Areas of Comm. , 24:8(2006), pp. 1514-1524.
- [20] R. G. GALLAGER, *A Minimum Delay Routing Algorithm Using Distributed Computation*, IEEE Trans. on Communications, 23(1977), pp. 73-85.
- [21] K. KAR, S. SARKAR, AND L. TASSIULAS, *Optimization Based Rate Control for Multipath Sessions*, Proc. 17-th Intl. Teletraffic Congress, December, 2001.
- [22] F. P. KELLY, *Network Routing*, Phil. Trans. R. Soc. Lond. A: Physical Sciences and Engineering (Complex Stochastic Systems), Vol. 337, No. 1647, pp. 343–367, 1991.
- [23] F. P. KELLY, A. K. MAULLOO, AND D. K. H. TAN, *Rate Control in Communication Networks : Shadow Prices, Proportional Fairness and Stability*, J. of Oper. Res. Soc., 49(1998), pp. 237–252.
- [24] X. LIN AND N. B. SHROFF, *Joint Rate Control and Scheduling in Multihop Wireless Networks*, Proc. IEEE Conf. on Dec. and Cont., 2(2004), pp. 1484-1489.
- [25] M. J. NEELY, E. MODIANO, AND C. E. ROHRS, *Dynamic Power Allocation and Routing for Time Varying Wireless Networks*, IEEE J. on Sel. Areas of Comm., Special Issue on Wireless Ad-Hoc Networks, 23:1(2005), pp. 89-103.
- [26] P. PURKAYASTHA AND J. S. BARAS, *Convergence of Ant Routing Algorithms via Stochastic Approximation and Optimization*, Proc. IEEE Conf. on Dec. and Cont., pp. 340–345, December 2007.
- [27] R. T. ROCKAFELLAR, *Network Flows and Monotropic Optimization*, Athena Scientific, Belmont, MA, 1998.
- [28] N. Z. SHOR, *Minimization Methods for Non-Differentiable Functions*, Springer Series in Computational Mathematics, Springer-Verlag, Berlin, Heidelberg, 1985.
- [29] J. N. TSITSIKLIS AND D. P. BERTSEKAS, *Distributed Asynchronous Optimal Routing in Data Networks*, IEEE Trans. on Aut. Cont., 31:4(1986), pp. 325-332.
- [30] W.-H. WANG, M. PALANISWAMI, AND S. H. LOW, *Optimal Flow Control and Routing in multipath networks*, Perf. Evaluation Jl., Vol. 52, No. 2–3, pp. 119–132, Elsevier, 2003.
- [31] J. G. WARDROP, *Some Theoretical Aspects of Road Traffic Research*, Proc. Inst. Civil Engineers, Vol. 1, pp. 325–378, 1952.