Adaptive Control Algorithms for Decentralized Optimal Traffic Engineering in the Internet

Constantino Lagoa, Hao Che and Bernardo A. Movsichoff

***Abstract*— In this paper, we address the problem of optimal decentralized traffic engineering when multiple paths are available for each call. More precisely, given a set of possible paths for each call, we aim at distributing the traffic among the available paths in order to maximize a given utility function. To solve this problem, we propose a large family of decentralized sending rate control laws having the property that each of the members of this family "steers" the traffic allocation to an optimal operation point. The approach taken relies on the control theory concept of Sliding Modes. These control laws allow each ingress node to independently adjust its traffic sending rates and/or redistribute its sending rates among multiple paths. The only nonlocal information needed is binary feedback from each congested node in the path. The control laws presented are applicable to a large class of utility functions, namely, utility functions that can be expressed as the sum of concave functions of the sending rates. We show that the technique can be applied not only to usual rate adaptive traffic with multiple paths, but also to rate adaptive traffic with minimum service requirements and/or maximum allowed sending rate and to assured service with targeted rate guarantee, all allowing for multiple paths. It is also shown that these control laws are robust with respect to failures; i.e., they automatically reroute traffic if a link failure occurs. Finally, we provide some insight on how to choose the "right" control law. In particular, we provide a way of choosing a member of the family of control laws that reduces the sending rate oscillation caused by implementation constraints like delays and quantization. An example of application of the approach delineated in this paper is also presented. This example provides some insights on the implementation aspects and illustrates the robustness of the control laws developed in this paper.**

***Index Terms*— Traffic Engineering, Sliding Mode Control, Optimization**

I. Introduction

TRAFFIC ENGINEERING (TE) [2] has been considered as one of the vital components of an autonomous system required to achieve both high resource utilization and high quality of service for both real time and non real-time applications. The basic idea is to split the traffic flows among multiple paths or steer the traffic away from a shortest path found by the interior gateway protocols, so that the congestion is avoided and network resource utilization is maximized. In a connection-oriented network such as Multi-Protocol Label Switching (MPLS) and Asynchronous Transfer Mode (ATM) networks, a widely used approach is to set up multiple paths between an ingress-egress node pair and distribute traffic flows among those paths. This is known as load balancing.

The objective of this paper is to develop *distributed* algorithms for load balancing as well as rate adaptation that will result in the maximization of a given utility function. Moreover, this work also addresses the case where more than one class of service (CoS) is present in the network. In particular, these algorithms address the case where Assured Service (AS), Best-Effort (BE) as well as Minimum Rate Guaranteed Service (MRGS), Upper Bounded Rate Service (UBRS) and Minimum Rate Guarantee Upper Bounded Service (MRGUBS) share network resources. The flexible structure of the algorithms presented also allows us to address a critical issue intrinsic to practical implementation of algorithms of this kind — oscillation due to delay of information exchange and finite granularity of control.

A. Literature Background

There is extensive literature on distributed traffic control. It includes both empirical algorithms (for example, see [7], [8]) and algorithms based on control theory; e.g., see [4], [5], [24]. These algorithms are designed for single path rate adaptation and the approach taken is not optimization-based.

Recently, several methodologies have been proposed which address the optimization-based rate adaptation problem using nonlinear optimization techniques. Their starting point is similar to the one in this paper; i.e., maximization (minimization) of a utility (cost) function, subject to network resource constraints, where the constraints represent the interaction between different types of traffic; i.e., traffic with different ingress/egress nodes and/or different service requirements.

In the paper by Golestani, *et al.* [11], instead of using link resource constraints, a link congestion cost is incorporated into the overall utility function. The optimization problem was then solved using a gradient type algorithm. Iterative algorithms were proposed where individual sources periodically adjust their sending rates based on the congestion cost information periodically fed back from each of the links along the flow forwarding paths.

Kelly, *et al.* [15] use a Lagrangian multiplier technique to solve the optimization problem at hand. This results in a separation between the rate control executed at individual sources and the calculation of a "price" for each link in the

This work was supported by the National Science Foundation under grants ECS-9984260 and ANI-0125653.

Constantino Lagoa and Bernardo A. Movsichoff are with the Department of Electrical Engineering, The Pennsylvania State University. Email: lagoa@engr.ee.psu.edu; bernardo@gandalf.ee.psu.edu

Hao Che is with the Department of Computer Science and Engineering, University of Texas at Arlington. Email: hche@cse.uta.edu

network. The rate control at individual sources is based on the "prices" fed back from all the links in the data paths. The computation of the "price" is in turn based on the sending rate information fed back from all the sources using this link. Since only a relaxation of the original optimization problem is solved, it is not proven that the algorithm converges to the optimal traffic allocation. The same formulation is used in the work by La, *et al.* [16] where the algorithm provided is proven to converge to the optimum rate allocation in the single congested link case (low traffic networks).

Low [19] uses a technique which converts a constrained problem into a non-constrained dual problem. This reformulation results in a similar distributed control scheme to the one presented in [15]. Iterative algorithms were also proposed and their convergence is proven for the single-path case. A similar approach is taken by Sarkar, *et al.* [25] to address the optimization-based multi-rate multicast. A distributed control scheme is proposed and it degenerates to the one in [19] when there is only one path in the multicast tree.

An interesting work on dynamic multi-path routing which bears some common features with the present work was given by Su and de Veciana [27]. The authors considered dispersive routing of the packets in a given flow to multiple paths based on the information obtained from the routing link state updates. An attempt was made to achieve optimal dynamic multi-path dispersion. An asymptotic result for a simple multi-link network was obtained, which provides insights on the development of three dynamic multi-path routing schemes. One of the scheme leads to high network utilization suggesting that it is a robust dynamic multi-path routing scheme.

B. A New Approach to Decentralized TE

Most of the existing optimization-based algorithms mentioned in the previous section are based on the Lagrangian approach, which is not viable for problems with a large number of constraints; e.g., see [13]. Each resource constraint results in a Lagrangian multiplier (or "price") which needs to be periodically updated and distributed to the sending sources. In turn, the source also needs to periodically inform the links along the path of its updated rate. Recently, it has been shown [1], [9], [10], [14] that the price at each link can be calculated using the aggregated flow rate at that link instead of the flow rates from individual sources. This eliminates the need for the sources to periodically inform the links of their sending rates, reducing the control traffic overhead. However, as value added services (such as multiple CoSs or virtual private networks) are introduced, the more constraints the control law should take into account and, therefore, the greater the number of "prices" that has to be computed. This results in a large overhead, regardless of where these prices are estimated. This might be the reason why, up until now, one has not seen an implementation where a significant number of value added services is offered.

In parallel with our research effort, Kar *et al.* [12] have developed a technique which also leads to optimal traffic allocation using only binary feedback from the congested nodes. However, the approach presented addresses the case neither of multiple paths nor of multiple CoSs.

In view of the importance of traffic engineering in the presence of multiple CoSs in the next generation IP Internet, in this paper, we introduce a new approach to tackle this problem. This approach has its basis on nonlinear control theory. The proposed technique enables optimal distributed TE in a multiple-path and multiple-CoS network. It allows ingress nodes to independently adjust their CoS-based traffic sending rates or/and redistribute traffic load among multiple paths, solely based on *binary* feedback information from the *congested nodes*. The proposed technique can be applied to any utility function that can be represented as a sum of concave terms. The technique can be used for TE with multiple CoSs; e.g., minimum rate guarantee, and/or maximum allowed rate, and assured CoSs, as proposed in this paper.

This new approach is potentially very useful for optimal TE in a differentiated services enabled MPLS network where CoS-based TE is desirable [18]. The assured CoS is particularly useful for network resource optimization when the edge nodes coincide with the boundary nodes of an administrative domain. In this case, the gross traffic volume of a given CoS that can be injected into the domain is normally a given value based on the service level agreement between service providers. The administrative domain should not throttle the traffic as long as gross traffic volume does not exceed the negotiated maximum volume. Hence, the only means for the administrative domain to optimize its resource utilization is to do load balancing, not rate adaptation.

This paper also provides a way to address a critical issue: Oscillation. The results in [17] are generalized, giving rise to a much larger set of adaptive control laws. Members of this set can be chosen in such a way as to attenuate the adverse effects of delays and quantization.

C. Sliding Modes

As mentioned above, the approach has its basis in nonlinear control theory. More specifically, we use results from the theory of Sliding Mode Control (also known as Variable Structure Control). Sliding Mode Control has been widely used in the control of nonlinear systems both in the presence and absence of uncertainty; for an introduction on the use of Sliding Modes in control systems; e.g., see [26]. Many successful applications of this theory have been documented, from the control of robotic manipulators to the control of underwater vehicles.

Of particular interest to the problem addressed in this paper is the use of Sliding Mode theory in mathematical programming. The results in [13], [28] indicate that Sliding Mode theory can be a powerful tool for optimizing convex functions subject to a large number of convex constraints. Motivated by these results, in this paper, we develop a class of adaptive control laws that converge to the optimal network resource allocation. These adaptive control laws allow distributed multi-path flow rate adaptation and load balancing at individual sources. The only information required is binary feedback from each congested node in the path. We also show that the same technique can be applied to assured service traffic with given target rates (AS), traffic with minimum rate guarantee (MRGS) and/or upper bound allowed rates (MRGUBS/UBRS).

D. The Sequel

The remainder of this paper is organized as follows. In Section II, we introduce the notation that is used throughout the paper and provide an exact statement of the problem to be solved. Section III is dedicated to the presentation of the family of decentralized control laws. In Section V, we provide some insight on how to design the control parameters in order to address some of the problems that one encounters in a practical implementation. Examples of application of the proposed approach are presented in Section VI. Conclusions and directions for future research are given in Section VII. Finally, a proof of the main results is provided in the Appendix.

II. Problem Statement

Before presenting the main results, let us first introduce the notation that is used throughout this paper.

A. Notation

In our model, the traffic flows are assumed to be described by a fluid flow model and the only resource considered is link bandwidth. In the remainder of this paper we will use the terms call, session and flow interchangeably.

Consider a computer network where calls with different service requirements are present. We divide these calls into *types*. Types are aggregate of calls that, from the point a view of the traffic engineering algorithm, can be treated as a *unit*; i.e., these are calls with the same ingress and egress nodes and the service requirements are to be applied to the aggregate, not the individual calls. One can have call types with just one call. Moreover, one can have different call types with the same ingress/egress node pair. We assume that each call type might have several paths available. The objective is to find the allocation of the resources that leads to the maximization of a given utility function subject to the network resource constraints and CoS requirements.

More precisely, consider a computer network whose set of links is denoted by $\mathscr{L}$, with cardinality $\mathrm{card}(\mathscr{L})$ equal to its number of links. Let c_l be the capacity of link $l \in \mathscr{L}$, n be the number of types of calls, n_i be the number of paths available for calls of type i and $\mathscr{L}_{i,j}$ be the set of links used by calls of type i taking path j. Given calls of type i, let $x_{i,j}$ be the total data rate of calls of type i using path j. Also, let

$$\mathbf{x}_i \doteq [x_{i,1}, x_{i,2}, \ldots, x_{i,n_i}]^T \in \mathbf{R}^{n_i}$$

denote the vector containing the data rates allocated to the different paths taken by calls of type i and

$$\mathbf{x} \doteq \left[\mathbf{x}_1^T, \mathbf{x}_2^T, \ldots, \mathbf{x}_n^T\right]^T \in \mathbf{R}^{n_1+n_2+\cdots+n_n}$$

the vector containing all the data rates allocated to different call types and respective paths. Now, a link $l \in \mathscr{L}$ is said to be congested if the aggregated data rate of the calls using the link reaches its capacity c_l. Given this, define $b_{i,j}(\mathbf{x})$ as the number of congested links along the j-th path of calls of type i. Finally, let $u(x)$ be the unit step function; i.e., $u(x) = 1$ if $x \geq 0$ and $u(x) = 0$ otherwise.

B. Problem Statement

The results in this paper aim at solving the problem of maximizing utility functions of the form

$$U(\mathbf{x}) \doteq \sum_{i=1}^{n} f_i(\mathbf{x}_i) \doteq \sum_{i=1}^{n} f_i\left(x_{i,1}, x_{i,2}, \ldots, x_{i,n_i}\right),$$

subject to the network constraints and CoS requirements, where $f_i(\cdot)$, $i = 1, 2, \ldots, n$, are differentiable increasing concave functions.

We assume that five categories of CoSs share the network: Calls of type i, $i = 1, 2, \ldots, s_1$, are assumed, without loss of generality, to be of Assured Service (AS) CoS category. By AS we mean that a target rate for the call should be guaranteed in average sense. More precisely, assuming that the target rate for $\mathbf{x}_i$ is Λ_i, the objective is to allocate the data rates in such a way that

$$\sum_{j=1}^{n_i} x_{i,j} = \Lambda_i,$$

for all $i = 1, 2, \ldots, s_1$. Calls of type i, $i = s_1+1, s_1+2, \ldots, s_2$, are assumed to be of the Minimum Rate Guaranteed Service category (MRGS). More precisely, this type of calls should satisfy the following requirement

$$\sum_{j=1}^{n_i} x_{i,j} \geq \theta_i,$$

for some $\theta_i > 0$ and all $i = s_1+1, s_1+2, \ldots, s_2$.

Calls of type i, $i = s_2+1, s_2+2, \ldots, s_3$, belong to the Upper Bounded Rate Service (UBRS) category; i.e., there is an upper bound on the maximum bandwidth that can be used. More precisely, traffic should be allocated in such a way that calls of type i satisfy

$$\sum_{j=1}^{n_i} x_{i,j} \leq \Theta_i,$$

for some $\Theta_i > 0$ and all $i = s_2+1, s_2+2, \ldots, s_3$.

Next, calls of type i, $i = s_3+1, s_3+2, \ldots, s_4$, are defined to be of the CoS category consisting traffic with both a Minimum Service Guarantee and an Upper Bounded Rate (MRGUBS); i.e., traffic should be allocated so that

$$\theta_i \leq \sum_{j=1}^{n_i} x_{i,j} \leq \Theta_i,$$

for some $\theta_i > 0$, $\Theta_i > 0$ and all $i = s_3+1, s_3+2, \ldots, s_4$.

Finally, calls of type i, $i = s_4+1, s_4+2, \ldots, n$ are assumed to be of the Best Effort (BE) class. Calls of this class do not have any further service requirements.

Given this, the problem of optimal resource allocation can be formulated as the following optimization problem:

$$\max_{\mathbf{x}} U(\mathbf{x})$$

subject to the network capacity constraints

$$\sum_{i,j:\ l \in \mathscr{L}_{i,j}} x_{i,j} - c_l \leq 0; \qquad l \in \mathscr{L},$$

the AS requirements

$$\sum_{j=1}^{n_i} x_{i,j} = \Lambda_i; \qquad i = 1, 2, \ldots, s_1,$$

the MRGS requirements

$$\sum_{j=1}^{n_i} x_{i,j} \geq \theta_i; \qquad i = s_1+1, s_1+2, \ldots, s_2,$$

the UBRS requirements

$$\sum_{j=1}^{n_i} x_{i,j} \leq \Theta_i; \qquad i = s_2+1, s_2+2, \ldots, s_3,$$

the MRGUBS requirements

$$\theta_i \leq \sum_{j=1}^{n_i} x_{i,j} \leq \Theta_i; \qquad i = s_3+1, s_3+2, \ldots, s_4$$

and all data rates are nonnegative

$$x_{i,j} \geq 0; \qquad i = 1,2,\ldots,n;\ j = 1,2,\ldots,n_i.$$

Obviously, the optimization problem above is a convex problem; i.e., maximizing a concave function over a convex set. If global information is available then algorithms like gradient descent could be used to solve it. However, generally, global information is not available. The objective of this paper is to provide decentralized adaptation laws that converge to the solution of the problem stated above.

III. Family of Decentralized Control Laws

In this section we present the main result of this paper: A family of decentralized control laws which converge to the maximum of the given utility function subject to the network link capacity constraints and AS requirements. Before stating the main result, we describe the form of the control laws used in this paper.

A. The Family of Control Laws

Define the following family of control laws: For $i = 1,2,\ldots,s_1$ (AS calls), let

$$\dot{x}_{i,j} = z_{i,j}(t,\mathbf{x}) \left[\left.\frac{\partial f_i}{\partial x_{i,j}}\right|_{\mathbf{x}_i} - \alpha b_{i,j}(\mathbf{x}) - \beta_i r_i(\mathbf{x}_i) + \delta_{i,j} u(-x_{i,j}) \right],$$

where

$$r_i(\mathbf{x}_i) = \begin{cases} 1 & \text{if } \sum_{j=1}^{n_i} x_{i,j} > \Lambda_i \\ -1 & \text{if } \sum_{j=1}^{n_i} x_{i,j} < \Lambda_i \end{cases}.$$

For $i = s_1+1, s_1+2, \ldots, s_2$ (MRGS calls), let

$$\dot{x}_{i,j} = z_{i,j}(t,\mathbf{x}) \left[\left.\frac{\partial f_i}{\partial x_{i,j}}\right|_{\mathbf{x}_i} - \alpha b_{i,j}(\mathbf{x}) + \beta_i^m r_i^m(\mathbf{x}_i) + \delta_{i,j} u(-x_{i,j}) \right],$$

where

$$r_i^m(\mathbf{x}_i) = \begin{cases} 0 & \text{if } \sum_{j=1}^{n_i} x_{i,j} > \theta_i \\ 1 & \text{if } \sum_{j=1}^{n_i} x_{i,j} < \theta_i \end{cases}.$$

For $i = s_2+1, s_2+2, \ldots, s_3$ (UBRS calls), let

$$\dot{x}_{i,j} = z_{i,j}(t,\mathbf{x}) \left[\left.\frac{\partial f_i}{\partial x_{i,j}}\right|_{\mathbf{x}_i} - \alpha b_{i,j}(\mathbf{x}) - \beta_i^M r_i^M(\mathbf{x}_i) + \delta_{i,j} u(-x_{i,j}) \right],$$

where

$$r_i^M(\mathbf{x}_i) = \begin{cases} 1 & \text{if } \sum_{j=1}^{n_i} x_{i,j} > \Theta_i \\ 0 & \text{if } \sum_{j=1}^{n_i} x_{i,j} < \Theta_i \end{cases}.$$

For $i = s_3+1, s_3+2, \ldots, s_4$ (MRGUBS calls), let

$$\dot{x}_{i,j} = z_{i,j}(t,\mathbf{x}) \left[\left.\frac{\partial f_i}{\partial x_{i,j}}\right|_{\mathbf{x}_i} - \alpha b_{i,j}(\mathbf{x}) + \beta_i^m r_i^m(\mathbf{x}_i) - \beta_i^M r_i^M(\mathbf{x}_i) + \delta_{i,j} u(-x_{i,j}) \right],$$

where again

$$r_i^m(\mathbf{x}_i) = \begin{cases} 0 & \text{if } \sum_{j=1}^{n_i} x_{i,j} > \theta_i \\ 1 & \text{if } \sum_{j=1}^{n_i} x_{i,j} < \theta_i \end{cases}; \quad r_i^M(\mathbf{x}_i) = \begin{cases} 1 & \text{if } \sum_{j=1}^{n_i} x_{i,j} > \Theta_i \\ 0 & \text{if } \sum_{j=1}^{n_i} x_{i,j} < \Theta_i \end{cases}.$$

For $i = s_4+1, s_4+2, \ldots, n$ (BE calls), let

$$\dot{x}_{i,j} = z_{i,j}(t,\mathbf{x}) \left[\left.\frac{\partial f_i}{\partial x_{i,j}}\right|_{\mathbf{x}_i} - \alpha b_{i,j}(\mathbf{x}) + \delta_{i,j} u(-x_{i,j}) \right].$$

In the equations above $z_{i,j}(t,\mathbf{x})$, α, β_i, β_i^m, β_i^M and $\delta_{i,j}$ are design parameters and $b_{i,j}$ is (as defined in the previous section) the number of congested links encountered by calls of type i taking path j. Note that the laws above can be considered to be similar to the TCP congestion control laws used in today's Internet in the following sense: The data rate is increased until congestion occurs. Once congestion is detected, the data rates are decreased. This continues until an equilibrium is achieved.

Actually, the TCP congestion protocol is very close to a particular case of the control laws above. To see this, assume each type of calls has just one path available. Furthermore, assume that the data rates are bounded away from zero and that the utility function that one wants to maximize is

$$U(\mathbf{x}) = \sum_{i=1}^{n} \log(x_i).$$

Our framework gives similar control laws for all call types, so let us look at the control law for x_1. In this case, the control law above reduces to

$$\dot{x}_1 = z_1(\mathbf{x}) \left[\frac{1}{x_1} - \alpha b_1(\mathbf{x}) + \delta_1 u(-x_1) \right].$$

Note that, since x_1 is bounded away from zero, $\delta_1 u(-x_1)$ is equal to zero. To continue with this analysis, take $z_1(\mathbf{x}) = x_1$. Then,

$$\dot{x}_1 = 1 - \alpha x_1 b_1(\mathbf{x}).$$

In other words, when there are no congested links in the path, x_1 increases linearly. When a congestion occurs, x_1

decreases approximately exponentially. Obviously, there are differences between the behavior of this control law and the behavior of the TCP congestion protocol. However, in this particular instance of our control laws, both of them exhibit the linear increase, exponential decrease behavior and the results obtained should be similar especially for large values of x_1 and/or α.

Note that the control laws above do not require any kind of resource reservation to implement and comply with the AS, MRGS, UBRS and MRGUBS call requirements[1]. The proposed algorithm "scans" the network for available resources and allocates traffic in such a way as to, first, satisfy CoS requirements and, second, maximize the value of the utility function subject to these requirements. In particular, since there is no resource reservation at call level, it is compliant with the DiffServ architecture.

Although not immediately apparent, the optimum is indeed a stationary point for the control laws above. This is a consequence of the fact that the motion is given by an "average" of the behavior "below" and "above" the discontinuity surfaces; i.e., when $\mathbf{x}$ is on one of the surfaces defined by the constraints, its derivative is a convex combination of the derivatives "below" and "above" the surface; for more details, see [6]. This is also the reason why the value of $r_i(\mathbf{x}_i)$, $r_i^m(\mathbf{x}_i)$ and $r_i^M(\mathbf{x}_i)$ need not be defined at the discontinuity points. No matter which value is assumed at the discontinuity points, the behavior of the control law is always the same.

Remark: From the equations above, one can see that, besides congestion information, the adaptation of the data rates of calls of type i only depends on the data rates of calls of the same type i. Recalling that type i refers to a given ingress/egress node pair, one sees that the control laws above are distributed control laws; i.e., individual ingress nodes independently adjust their traffic sending rates and redistribute their traffic among the available paths. The quantity $b_{i,j}$ represents the only interaction between different types of traffic; i.e., the binary congestion information being fed back from the congested links along flow forwarding paths.

B. Robustness with Load Sharing

In practice, if one wishes to distribute the traffic among multiple paths, one should use a utility function of the form

$$U(\mathbf{x}) = \sum_{i=1}^{n} f_i(x_{i,1} + x_{i,2} + \cdots + x_{i,n_i}).$$

This means that we are not interested in how the traffic is distributed among the available paths, we are only concerned about how much traffic of a given type the network can handle.

If we look carefully at the form of the control law that one obtains in this special case, we see that we get an added benefit: Robustness with respect to failures. In the case of a link failure, the algorithm tries to reroute the traffic to the other available paths and provides an optimal traffic allocation for the new network configuration. In other words, in the case of a link failure, the algorithm provides a systematic (optimal) way of redistributing traffic[2].

[1] The provisioning of the aggregated resource for AS, MRGS, and MRGUBS traffic running between any pair of nodes does need to ensure that at least one feasible distribution exists, which is beyond the scope of this paper. Some optimization algorithms with global information such as the one proposed by Mitra [21] can be employed to serve this purpose during the network resource planning phase.

[2] Note that in the discussion of using the control laws to deal with link failure, we restrict ourselves to the control components running at the source. We consider neither mechanisms used for link failure detection nor the congestion detection mechanisms discussed in Section V-A which should not be used for link failure detection. Detection and notification to the sources of a link failure is beyond the scope of this paper and will be the subject of future research.

IV. Main Result

We now present the main result in this paper. This result tells us that the control laws introduced in Section III converge to the optimal traffic allocation.

Theorem 1: Assume that all data rates are bounded; i.e., there exists an $\rho \in \mathbf{R}$ such that the data rate vector $\mathbf{x}$ always belongs to the set

$$\mathscr{X} \doteq \big\{\mathbf{x} \in \mathbf{R}^{n_1+n_2+\cdots+n_n} : x_{i,j} \leq \rho;\ l \in \mathscr{L}_{i,j}; \\ j = 1,2,\ldots,n_i;\ i = 1,2,\ldots,n\big\}.$$

Also, assume that at the optimal traffic allocation, each congested link has at least one nonbinding class of service or a BE call with a nonzero data rate and that the components of the gradient of the utility function; i.e., $\nabla U(\mathbf{x})$, are bounded in $\mathscr{X}$. Let $\zeta > 0$ and $\varepsilon > 0$ be given (arbitrarily small) constants and let $z_{i,j}(t,\mathbf{x})$ be scalar continuous functions satisfying

$$z_{i,j}(t,\mathbf{x}) > \zeta,$$

for all $t > 0$ and $\mathbf{x} \in \mathscr{X}$. Furthermore, let α, β_i, β_i^m, β_i^M and $\delta_{i,j}$ be positive constants satisfying the following inequalities

$$\alpha > \alpha_{min} \doteq \max_{i,j,\mathbf{x}\in\mathscr{X}} \frac{\partial U(\mathbf{x})}{\partial x_{i,j}}$$

for all $\mathbf{x} \in \mathscr{X}$, $i = 1,2,\ldots,n$, $j = 1,2,\ldots,n_i$ and $\beta_i > \beta_{min}$, $\beta_i^m > \beta_{min}$, $\beta_i^M > \beta_{min}$, where

$$\beta_{min} \doteq \alpha_{min} \max_{i,j} B_{i,j}$$

and $B_{i,j}$ is the number of links used by calls of type i taking path j. Furthermore, let

$$\delta_{i,j} \geq \alpha_{min} \max_{i,j} B_{i,j} + \beta + \varepsilon,$$

where $\beta \in \{\beta_i, \beta_i^m, \beta_i^M\}$.

Then, the control laws presented above converge to the maximum of the utility function

$$U(\mathbf{x}) \doteq \sum_{i=1}^{n} f_i(\mathbf{x}_i) \doteq \sum_{i=1}^{n} f_i\left(x_{i,1}, x_{i,2}, \ldots, x_{i,n_i}\right),$$

subject to the network link capacity constraints and AS, MRGS, UBRS, MRGUBS and nonnegativity requirements.

Proof: See Appendix. ∎

V. Implementation Considerations

The result presented in the last section provides a family of control laws that converge to an optimal data rate allocation. This family is quite large and experiments suggest that the choice of the functions $z_{i,j}(t,\mathbf{x})$ can lead to quite different transient behaviors. In this section we discuss some of the issues related to practical implementation of the laws presented in this paper.

A. Congestion Detection

In the proposed control laws, the link congestion information is conveyed only through the parameter $b_{i,j}$. This allows for any link congestion definition and congestion detection mechanism. They can be incorporated into the control processes without any modification of the control laws. For example, if per CoS queueing is enabled at each output port of a router, a CoS based active queue management algorithm [23] (e.g., the average queue length threshold) can be implemented for each CoS to detect congestion for that specific CoS. Another example is when there is more than one priority based CoS sharing a single queue. In this case, a weighted random early detection (WRED) type of multi-threshold algorithm can be utilized for congestion detection.

Obviously, the way the congestion detection scheme is designed will have an impact on the performance of the control laws. For instance, for queue threshold based approaches, the time scale at which the queue length average is computed will affect the responsiveness of the control laws to link congestion. As we do not carry out packet level simulation in this paper, we do not study this implementation issue, which will be subject to future study.

We also note that link bandwidth constraints, such as bandwidth partitioning among multiple CoSs, do not generate new control parameters in the control laws. Nevertheless, the control laws will always drive the network to the optimal rate allocation under these constraints.

B. Congestion Information Feedback

There are several ways to implement the congestion information feedback. For example, one can make use of the ideas of the binary congestion feedback signaling schemes proposed for Available Bit Rate (ABR) service [3] to convey $b_{i,j}$ information back to the source. For instance, one possible scheme [22] is to let the congested nodes along the forwarding path periodically generate backward congestion messages for each CoS that is experiencing congestion. The flow sending source estimates $b_{i,j}$ value by periodically updating the number of counts of congestion messages received from the distinct congested nodes along forwarding path j. Ideally, one would like to have a very small period for the congestion feedback. However, the algorithms will work no matter which period one chooses. Obviously, the larger the period, the larger the oscillations (see next section). In other words, when designing the control laws parameters, one should decrease the speed of adaptation if the period of the congestion information update increases.

It should be noted that the proposed algorithm might converge to a nonoptimal equilibrium if the true number of congested links is not known. Nevertheless, experiments suggest that if a large value of α is used then the algorithm becomes more robust with respect to this loss of information. However, there is a trade-off when one does this since larger values of α will result in larger oscillations.

To be more precise, let us consider the case of a single path per call type and only BE service being provided. Furthermore, assume that all data rates are bounded away from zero and that the objective function that one is trying to maximize is

$$U(\mathbf{x}) = \sum_{i=1}^{n} f_i(x_i).$$

Now, the results in [20] can give us some insight on this matter. In the case of a single path, the control law proposed in our paper is

$$\dot{x}_i = z_i(t,\mathbf{x})\left[d_i(x_i) - \alpha b_i(\mathbf{x})\right],$$

where

$$d_i(x_i) = \frac{d f_i}{d x_i}.$$

Now, assume that we only know if the path is congested or not. In this case b_i is either zero or one. The results in [20] indicate that in this case one converges to the maximum of the utility function

$$\tilde{U}(\mathbf{x}) = \sum_{i=1}^{n} \int_0^{x_i} \log\left(\frac{\alpha}{\alpha - d_i(u_i)}\right) du_i.$$

The result above shows that we do not converge to the desired point. However, if α is large, then the two utility functions are approximately the same.

C. Discrete Time Control Law

The implementation of the above control laws in this paper has to be performed in discrete time when working on a real network. Therefore, we now describe a discrete-time approximation of the continuous–time control laws.

As is the case with discrete time controller design, there are different ways of obtaining difference state equations from the differential state equations. The approach used here is the forward rule approximation. This method avoids computational delays inherent to other discretization techniques such as trapezoidal or backward rule approximation. Let

$$\dot{x}_{i,j} = g_{i,j}(\mathbf{x},t)$$

denote the continuous time control law described in Section III. The discrete approximation that we propose is

$$x^d_{i,j}\left[(k+1)t_d\right] = x^d[kt_d] + t_d g_{i,j}\left(\mathbf{x}(kt_d), kt_d\right);\ k = 0,1,\dots$$

where t_d is the sampling period. Obviously, since this is not a continuous time law, Sliding Mode theory does not apply. However, one has the following result.

Proposition 2: Let $\mathbf{x}(t)$ be the trajectory obtained using the control laws in Section III and let $\mathbf{x}^d(t)$ be the corresponding discrete time trajectory obtained using the discretization algorithm above. Define the set $\mathscr{X}$ as in Theorem 1.

Given any time interval $[t_0, t_1]$ and constant $\varepsilon > 0$, there exists a $\delta > 0$ such that if

$$t_d z_{i,j}(t, \mathbf{x}) < \delta$$

for all $t > 0$ and $\mathbf{x} \in \mathscr{X}$, then

$$\left\| \mathbf{x}(t) - \mathbf{x}^d(t) \right\| < \varepsilon$$

for all $t \in [t_0, t_1]$.

Proof: Direct application of result 2, page 95 of [6]. ∎

D. Adaptive Control Law

When implementing the control laws developed in this paper, one is faced with several issues: First, one has to implement a discrete time version of the control algorithms, such as the one described above. Second, usually one uses finite word length which leads to a quantization of the possible data rate values. Finally, there is delay in the propagation of the congestion information. All of these lead to a well known phenomenon: Oscillation. Even in this case, the discretization of the control laws presented in this paper is approximately optimal. We now state the precise result.

Proposition 3: Let $\mathbf{x}(t)$ be the trajectory obtained using the control laws in Section III and let $\mathbf{x}^r(t)$ be the corresponding discrete time trajectory obtained using the discretization algorithm above and in the presence of delays in the propagation of the congestion information. Let t_r be an upper bound on the largest delay. Again, define $\mathscr{X}$ as in Theorem 1.

Given any time interval $[t_0, t_1]$ and constant $\varepsilon > 0$, there exists a $\delta > 0$ such that if

$$\max\{t_d, t_r\} z_{i,j}(t, \mathbf{x}) < \delta$$

for all $t > 0$ and $\mathbf{x} \in \mathscr{X}$, then

$$\|\mathbf{x}(t) - \mathbf{x}^r(t)\| < \varepsilon$$

for all $t \in [t_0, t_1]$.

Proof: Direct application of result 2, page 95 of [6]. ∎

Although we do have approximate optimality, the performance might degrade if the delays are too large and/or the value of $z_{i,j}(\cdot)$ is too low. Therefore, we now provide a method for choosing the functions $z_{i,j}(\cdot)$ which reduces the amplitude of oscillation of the data rates. The main idea is the following: Oscillation occurs when a link is congested. Hence, when there is no congestion one would like the rates to increase at a reasonably fast rate. Once congestion is about to occur, one would like to decrease the rate of change in order to reduce the amplitude of the oscillations.

Hence, we propose the following scheme for choosing the value of $z_{i,j}(\cdot)$: Let $T > 0$ be given. Initialize $z_{i,j}(0, \mathbf{x}) = k$, where $k > 0$ is a constant. If congestion is detected at time t_0 for calls of type i taking path j let

$$z_{i,j}(t, \mathbf{x}) = \omega(t - t_0)\,; \quad t_0 \le t < t_0 + T,$$

where $\omega : [0, T] \to [\zeta, k]$ is a decreasing function and ζ is some positive constant. Now, at $t_0 + T$ repeat the same reasoning. If there is no congestion, let $z_{i,j}(t, \mathbf{x}) = k$ until congestion is detected. Once congestion is detected let $z_{i,j}(t, \mathbf{x})$ be equal to a shifted version of $\omega(t)$. Examples of such functions are provided in the next section and the desired behavior is depicted in Fig. 1. Notice that the value $z_{i,j}(\cdot)$ is continuously reset every T seconds. This ensures that the network is able to adapt to changes in the traffic demand. Also, although we reduce the rate of change of $x_{i,j}$ when congestion is detected, the algorithm still converges to the optimum. This is a consequence of the fact that the functions $z_{i,j}(\cdot)$ mentioned above satisfy the conditions of Theorem 1.

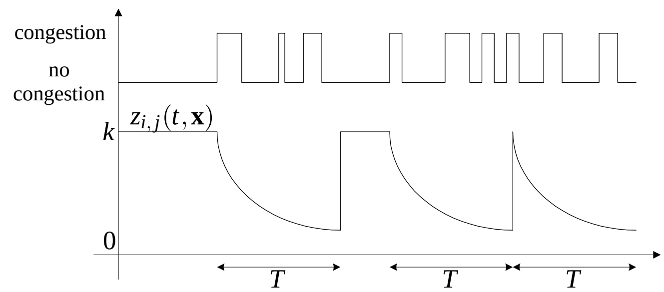


Fig. 1. Example of a scaling function $z_{i,j}(\cdot)$

Now, this choice for the functions $z_{i,j}(\cdot)$ leaves one with three free parameters that have to be selected. The period T of $z_{i,j}(\cdot)$ should be chosen taking into account the behavior of the demand on the network. More precisely, it should be equal to the interval of time in between substantial changes in the demand. Resetting $z_{i,j}(\cdot)$ in this form will enable the network to more rapidly adapt to the demand. Now, the parameters k and ζ of $z_{i,j}(\cdot)$ should be inversely proportional to the round trip time of the path j of calls of type i, which can be easily estimated. The reasoning behind this particular choice has to do with the fact that the largest delay for the propagation of the congestion information is the round trip time. Now, the exact value of these parameters depends on how one would like the network to behave. If they are high, then the network will quickly react to changes in the demand, but the oscillations will be large and the network might be operating far away from the optimal behavior. If the values k and ζ are low, then one will have small oscillations and a better steady state operating point. However, there will be a much larger transient behavior.

VI. Examples

In this section we present some simulation results which exemplify the behavior of the proposed family of control laws. In particular, we show the convergence of the algorithm under discretization, delays and link failure. Furthermore, these simulations show that the final data rate allocation results in a value of the utility function barely indistinguishable from the optimal one. In the simulations presented here, we did a flow approximation of the network behavior. The sources are assumed to be rate adaptive flows. The control laws determine the sending rate of each of the flows. The overall simulation was done using Matlab.

We concentrate on the case where both BE and AS traffic share the network since one would obtain similar results if one would also have MRGS, UBRS and MRGUBS CoS present. In the simulations described below, we use link full as congestion indicator for all CoSs. So a single binary feedback is used for all CoSs.

A. The Network Topology

The model of the network for our simulations is based on the one in [16] and is shown in Fig. 2, where we also show all the links' bandwidths and delays, as well as source and destination nodes. The key difference, however, is that in our setup we allow for multiple paths for each type of calls. Moreover, we also allow for multiple CoSs. We consider a total of $n = 8$ types of calls corresponding to 8 different combinations of source/destination nodes. The paths available for each pair of ingress/egress nodes are described in Table I. Recall that n_i indicates the number of paths available to calls of type i.

B. Utility Function and Control Law

We propose the utility function of the form given in Section III; i.e.,

$$U(\mathbf{x}) = \sum_{i=1}^{8} f_i(\mathbf{x}_i),$$

with

$$f_i(\mathbf{x}_i) \doteq \log\left(0.5 + \sum_{j=1}^{n_i} x_{i,j}\right); \qquad \begin{array}{l} i = 1, \ldots, 8 \\ j = 1, \ldots, n_i \end{array},$$

where n_i is given in Table I and $\log(\cdot)$ denotes the natural logarithm, although any base will serve the same purpose. The constant term 0.5 in the logarithm is included to avoid an infinite cost at startup and, as shown below, an infinite derivative of the data rates.

In these examples, calls of types 3 and 5 are assumed to be of AS type; i.e., calls of types 3 and 5 have target rates

$$\Lambda_3 = \Lambda_5 = 1\,\text{Mbps}.$$

Given the utility function and service requirement above, we use Theorem 1 to obtain the following family of control laws: For $i = 1, 3$ and $j = 1, \ldots, n_i$; i.e., AS calls

$$\dot{x}_{i,j} = z_{i,j}(t, \mathbf{x}) \left[\frac{1}{0.5 + \sum_{j=1}^{n_i} x_{i,j}} - \alpha b_{i,j}(\mathbf{x}) - \right.$$

$$\left. \beta_i r_i(\mathbf{x}_i) + \delta_{i,j} u(-x_{i,j}) \right]$$

and for $i = 1, 2, 4, 6, 7, 8$ and $j = 1, \ldots, n_i$; i.e., BE calls

$$\dot{x}_{i,j} = z_{i,j}(t, \mathbf{x}) \left[\frac{1}{0.5 + \sum_{j=1}^{n_i} x_{i,j}} - \alpha b_{i,j}(\mathbf{x}) - \delta_{i,j} u(-x_{i,j}) \right]$$

The design parameters α, β_i and $\delta_{i,j}$ and the functions $z_{i,j}(t, \mathbf{x})$ are chosen to satisfy the convergence conditions set forth by Theorem 1. Other practical considerations taken into account while choosing these parameters are explained in the following subsections when implementation issues are discussed. It is clear from the last equation, that the term 0.5 in the denominator imposes an upper bound on the derivatives, that otherwise will tend to infinity as all the data rates go to zero.

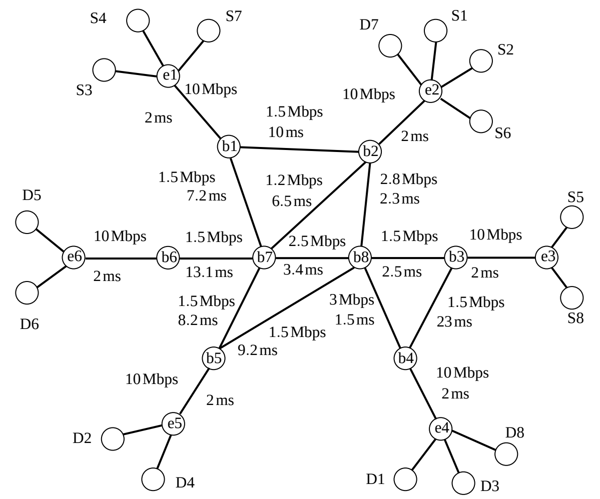


Fig. 2. Topology of the network

TABLE I
PATHS AVAILABLE FOR EACH TYPE OF CALLS

type 1 $n_1 = 4$	$x_{1,1} : e_2b_2b_8b_4e_4$ $x_{1,2} : e_2b_2b_8b_3b_4e_4$ $x_{1,3} : e_2b_2b_7b_8b_3b_4e_4$ $x_{1,4} : e_2b_2b_7b_8b_4e_4$	type 5 $n_5 = 2$	$x_{5,1} : e_3b_3b_8b_7b_6e_6$ $x_{5,2} : e_3b_3b_4b_8b_5b_7b_6e_6$
type 2 $n_2 = 3$	$x_{2,1} : e_2b_2b_8b_5e_5$ $x_{2,2} : e_2b_2b_7b_5e_5$ $x_{2,3} : e_2b_2b_1b_7b_5e_5$	type 6 $n_6 = 3$	$x_{6,1} : e_2b_2b_1b_7b_6e_6$ $x_{6,2} : e_2b_2b_8b_7b_6e_6$ $x_{6,3} : e_2b_2b_7b_6e_6$
type 3 $n_3 = 2$	$x_{3,1} : e_1b_1b_7b_8b_4e_4$ $x_{3,2} : e_1b_1b_2b_8b_4e_4$	type 7 $n_7 = 3$	$x_{7,1} : e_1b_1b_2e_2$ $x_{7,2} : e_1b_1b_7b_2e_2$ $x_{7,3} : e_1b_1b_7b_8b_2e_2$
type 4 $n_4 = 4$	$x_{4,1} : e_1b_1b_7b_5e_5$ $x_{4,2} : e_1b_1b_7b_8b_5e_5$ $x_{4,3} : e_1b_1b_2b_7b_5e_5$ $x_{4,4} : e_1b_1b_2b_8b_5e_5$	type 8 $n_8 = 2$	$x_{8,1} : e_3b_3b_4e_4$ $x_{8,2} : e_3b_3b_8b_4e_4$

C. Ideal Conditions

As a first step we test the adaptation laws in the almost ideal situation; i.e., very small delays and sampling time. The discretization of the continuous time control law was done using the scheme presented in Section V-C with a sampling time of $t_d = 0.1\,\text{ms}$. On the other hand the delays were all chosen equal to 0.1 ms, while the remaining parameters of the control law were taken to be $\alpha = 4$, $\beta_3 = \beta_5 = 22$ and $\delta_{i,j} = B_{i,j}\alpha + \beta_i + 0.0001$. The oscillation reducing function was kept constant; i.e., $z_{i,j}(t, \mathbf{x}) = 0.375$.

It can be seen in Fig. 3 that, under these conditions, the utility function converges to the optimal value, while the data rates for types 2, 3 and 5 show a clear sliding behavior, without any noticeable oscillation. Also, as it can be seen from the aggregate data rates for types 3 and 5 that the AS requirements are satisfied.

D. Effect of Delays

We now show the effects of the information propagation delays. For this purpose the delays were taken as indicated in Fig. 2. It can be seen in Fig. 4 that the existence of delays in the propagation of the congestion information results in the oscillation of the data rates. As expected, in this case we have convergence to a neighborhood of the optimum, not the

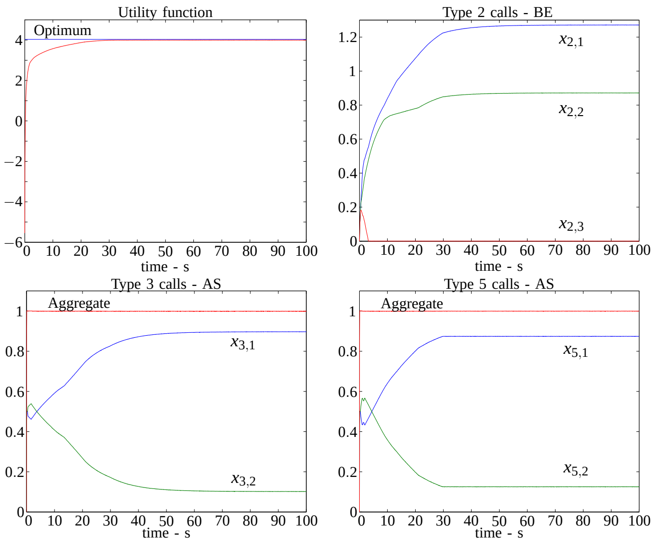


Fig. 3. Ideal Conditions: small delays and t_d

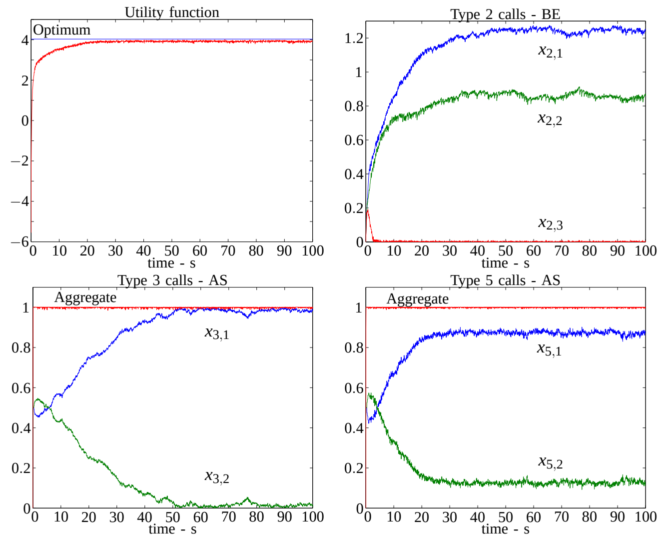


Fig. 4. Effect of larger congestion propagation delays

optimum itself. This is shown by the utility function in the same figure. Again, the AS requirements for types 3 and 5 were satisfied as exemplified by the aggregate data rates in the figure.

E. Effect of the Sampling Interval

We now concentrate on the performance changes caused by the sampling interval t_d utilized to obtain the discrete time control law. For this simulation the sampling interval was chosen as $t_d = 5\,\mathrm{ms}$ and the information propagation delays were again assumed to be very small.

Note that one can think of discretization as time varying delays. Hence, it is not surprising that one sees a similar behavior to the one shown in the previous simulation. Due to space constraints, these plots are not included.

F. Adaptive Oscillation Reduction

As shown in Sections VI-D and VI-E, oscillation degrades the performance of the adaptation laws. To reduce these oscillations and improve performance we resort to the scheme proposed in Section V.

However, the use of such a scheme introduces other undesirable effects on the data rates, mainly due to a reduction in the magnitude of the derivatives. In particular, the speed of convergence may be reduced and, as a consequence, a slower reaction to changing conditions can result. To exemplify this trade–off, Fig. 5 shows the utility function from Section VI-C but with a time varying $z_{i,j}(t,\mathbf{x}) = \omega(t-t_0)$, where

$$\omega(t) = 0.3(0.25 + 0.65^t).$$

The basic idea in choosing $z_{i,j}(\cdot)$ is to avoid large discontinuities at the on–off instants, since they introduce spikes in the response of the system and to keep those instants at some distance to avoid frequent spikes. To this end, once a congested link is detected, the oscillation reducing function is "active" for a time $T = 10\,\mathrm{s}$, as shown in Fig. 1.

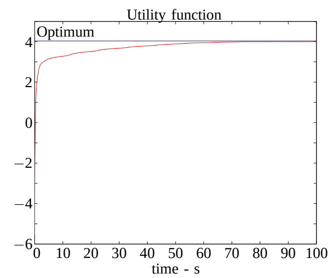


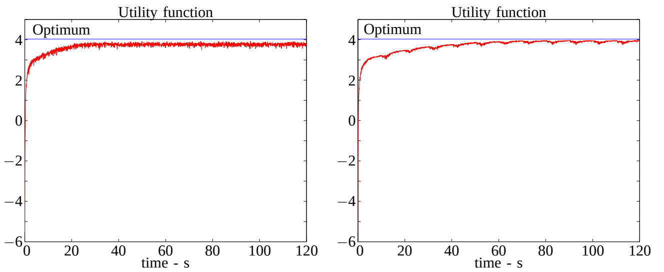
Fig. 5. Utility function for small delays and t_d, with time varying $z_{i,j}(\cdot)$

Note that even though $z_{i,j}(\cdot)$ is only piecewise continuous, this does not pose any problem. We can actually define a new function $\hat{z}_{i,j}(t,\mathbf{x})$ (interpolating) that is continuous and satisfies

$$\hat{z}_{i,j}\big(kt_d, \mathbf{x}(kt_d)\big) = z_{i,j}\big(kt_d, \mathbf{x}(kt_d)\big),$$

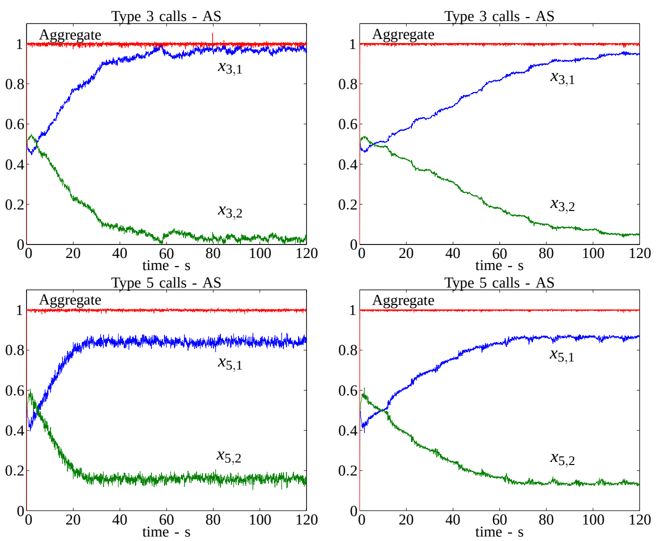
where $k = 0, 1, \ldots$ and kt_d are the sampling instants. Hence, the conditions of Theorem 1 are satisfied.

It can clearly be seen that the speed of convergence is reduced. Nevertheless, the benefit outweighs the loss. Now, let us examine the effects of a time varying $z_{i,j}(\cdot)$ on a network with a discrete control law and delays in the propagation of information. In Figs. 6, 7 and 8, we provide a comparison of the network behavior for a constant $z_{i,j}(\cdot)$ and a time varying one. Although both utility functions converge to a neighborhood of the optimum, the use of a time varying $z_{i,j}(\cdot)$ results in smaller oscillation in the data rates. Moreover, the time varying $z_{i,j}(\cdot)$ gives a better steady state performance. The introduction of an oscillation reduction scheme results in a smaller packet loss, with all the benefits thereof. This is the case because having less oscillation implies that the network is working above its capacity for shorter intervals of time. Finally, the aggregate data rates in the figure show that in both cases the AS requirements are being met. However for the time varying $z_{i,j}(\cdot)$ case, the departures from the average are smaller resulting in a better performance not only of the AS calls but also BE traffic.



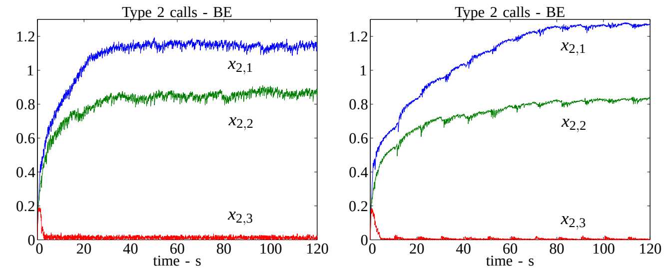
(a) Constant $z_{i,j}(t,\mathbf{x})$ (b) Time varying $z_{i,j}(t,\mathbf{x})$

Fig. 6. Oscillation reduction on the utility functions for larger delays and t_d



(a) Constant $z_{i,j}(t,\mathbf{x})$ (b) Time varying $z_{i,j}(t,\mathbf{x})$

Fig. 7. Oscillation reduction on AS calls for larger delays and t_d



(a) Constant $z_{i,j}(t,\mathbf{x})$ (b) Time varying $z_{i,j}(t,\mathbf{x})$

Fig. 8. Oscillation reduction on BE calls for larger delays and t_d

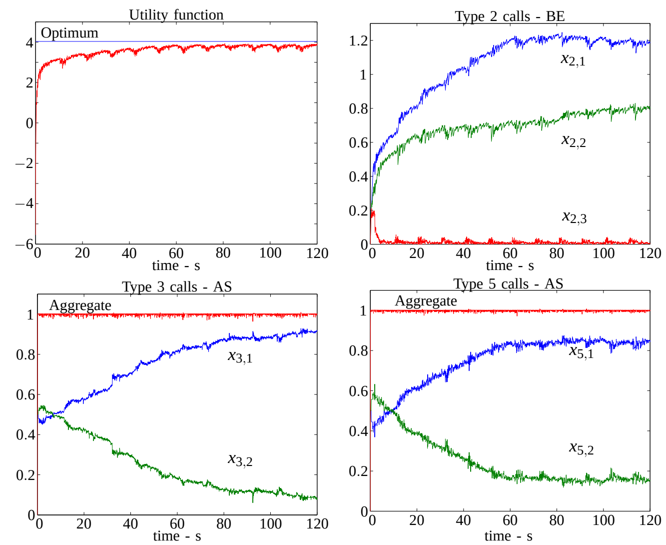


Fig. 9. Effect of larger values of α

G. Effect of α

Given the implementation constraints, the motion of the system will occur on some neighborhood of the feasible region. The value of α not only determines how fast the system goes back to the feasible region, once congestion is detected, but also indicates how "large" this neighborhood is. Ideally, one would like to combine fast return with a "small" neighborhood, but these conditions are contradictory.

To show this behavior we simulated the case $\alpha = 8$. These results are shown in Fig. 9. The magnitude of the oscillations increased significantly from the case $\alpha = 4$, shown in Figs. 6(b), 8(b) and 7(b). Nevertheless, convergence to a neighborhood of the optimum is still achieved and the AS requirements are met.

H. Effects of k, ζ and T

From the simulations above, it is clear that a larger value of k will produce more oscillations while a smaller one will yield slower convergence. To partially solve this compromise we use a time varying $z_{i,j}(\cdot)$ that introduces two new parameters T and ζ. The value of ζ affects oscillation and speed of convergence in the same manner as k. On the other hand, the value of T determines how much time we spend using low gains. This implies that a larger T can lead to lack of reaction to changing conditions like a link failure. However, a small T results in a large number of "spikes" in the data rates.

I. Link Failure

Finally, we show one of the salient features of the proposed control laws: Its robustness with respect to link failures. We claim that since the data rates are being updated adaptively, as soon as link failure is detected, the control law will reroute all the traffic away from the paths using this link. The control laws running at the sending nodes can handle this situation by treating link failure as congestion; i.e., the adaptation laws are oblivious to the failure. Instead, they simply reduce the sending rates because congestion is detected along the paths that include the broken link.

Note that, when a link failure occurs, knowledge of the exact number of congested links in the path is not necessary. In other words, one might think that the control laws proposed in this paper would not work since there might be a loss of the congestion information due to this failure; i.e., one would have a wrong estimate of $b_{i,j}$. However, once a failure occurs, one might use any value for $b_{i,j}$ as long as it is greater than or equal to one. By doing this, one ensures that the usage of the paths containing the broken link will decay to zero in finite time. A good choice for $b_{i,j}$ is the number of links in the path. A minor modification of the results in this paper can be done to prove that, in this case, one still converges to the optimal

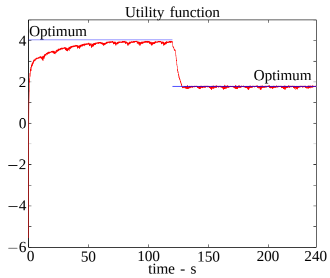


Fig. 10. Utility function: Response to a link failure

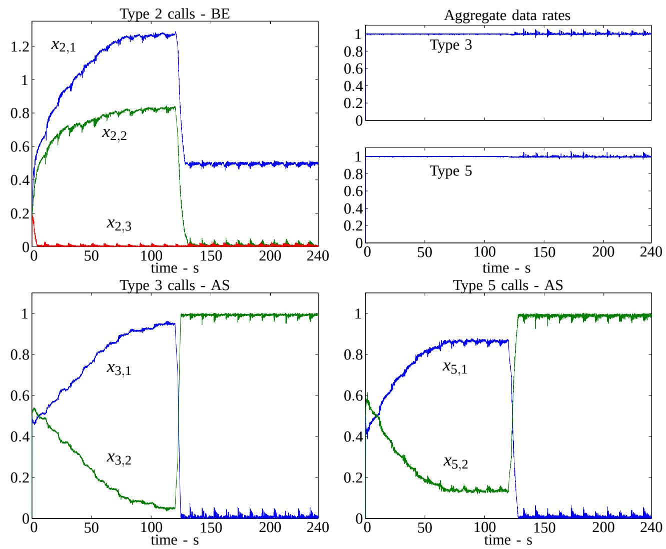


Fig. 11. Data rates: response to a link failure

resource allocation.

In order to test this feature, the link connecting nodes b_7-b_8 was opened at $t = 120$s. This link is particularly problematic because both type 3 and 5 (the AS calls) lose one of the two paths available to them. Furthermore, the path that remains for type 5 shares links with almost every other type, making it necessary for the other call types to also reroute some of their traffic and reduce their aggregated data rates.

All the variables involved were set to their nominal values; i.e., $\alpha = 4$, $\beta_3 = \beta_5 = 22$, $T = 10$s and delays from Fig. 2. As the simulations show, this control law excels at this task. In Fig. 10 one can see that the network reacts to this failure almost immediately. Furthermore, the utility function converges to its new optimal value (approximately 1.788) and the oscillations are small. In Fig. 11, the aggregate data rates of types 3 and 5 are shown. We observe that these aggregated data rates remain almost constant throughout the simulation. Finally, Fig. 11 also shows some data rates. In light of these plots we see that the proposed control laws quickly react to what can be considered a "very difficult situation." The link failure forces the network to reroute the AS traffic to the available paths. This means transferring the AS traffic to alternative paths that were almost unused by these types of traffic. This, in turn, forces the network to substantially reduce the resources allocated to BE traffic. For example, in Fig. 11 we see that data rates of BE traffic of type 2 had to be greatly reduced to make sure that AS traffic requirements were met. In other words, the control laws presented here endow the network with the capacity to quickly react to failures, always enforcing the AS requirements and distributing both the AS traffic and BE traffic among the available paths in such a way as to maximize the utilization of the network resources.

Remarks: The simulations above give us some clues on how to choose the values of α and β_i. Namely, their value should be on the low end of the admissible set. This will result in smaller oscillations. However, this choice will also result in slower convergence to the set of feasible rates. Nevertheless, this can be compensated by choosing a large enough value of the parameter k of the functions $z_{i,j}(\cdot)$.

VII. Conclusions and Directions for Further Research

In this paper, we introduced an optimization-based approach to allow distributed, optimal traffic engineering in the presence of both multiple CoSs and multiple paths. This approach results in a large family of control laws which ensures that the traffic rates between edge nodes converge to a distribution which maximizes a given utility function. The utility functions addressed in this paper are of a general form; i.e., they can be expressed as a sum of concave functions of the sending rates between a pair of edge nodes. The control laws proposed are decentralized and require minimum information exchange among different network elements. Only a binary link congestion notification from the congested nodes in the data path needs to be fed back to the edge nodes. In addition, this paper addresses the problem of data rate oscillation, typically seen in feedback control systems with delay and finite granularity of control.

Simulation studies showed that the proposed approach leads to rather high performance in terms of resource utilization, as measured by the utility function. In particular, the simulation results demonstrated that the approach can effectively reroute the traffic to an optimal state even in the presence of sudden link failures.

As mentioned before, the main focus of this paper is the introduction of new optimal decentralized TE control laws. Although some critical implementation issues are addressed (e.g., delay and granularity of control) there are many other implementation issues that have not been studied in the work presented in this paper. Therefore, a possible direction for future research is the detailed study of implementation issues and to test the implementation in large scale network settings. Due to its ability to deal with link failures and the importance of service protection against link failure, a major future research effort will be aimed at the design of optimal fast rerouting algorithms based on this theory. Note that no algorithms exist which can do fast rerouting in an optimal fashion. Only non-optimization based methodologies have been proposed in the literature.

Appendix
Proof of Theorem 1

In this section we present the proof of Theorem 1. We start by presenting an alternative formulation of the results presented in this paper.

A. Preliminaries

To simplify the exposition to follow, we now introduce some notation that enables us to present the proof in a more intuitive way. The problem at hand can be represented in the following form

$$\max_{\mathbf{x}} U(\mathbf{x})$$

subject to the inequality constraints

$$h_k(\mathbf{x}) \le 0; \qquad k = 1,2,\ldots,m$$

and the equality constraints

$$h_k(\mathbf{x}) = 0; \qquad k = m+1, m+2, \ldots, L,$$

where $U(\mathbf{x})$ is a concave differentiable increasing function and $h_k(\mathbf{x})$ are affine functions for all k. The inequality constraints correspond to the link capacity constraints, the restriction that the data rates have to be nonnegative and MRGS, UBRS and MRGUBS CoS constraints. On the other hand, the equality constraints correspond to the AS requirements. Now, define the matrix

$$Z(t,\mathbf{x}) \doteq \mathrm{diag}\big(z_{1,1}(t,\mathbf{x}),\ldots,z_{1,n_1}(t,\mathbf{x}), \ldots, z_{n,1}(t,\mathbf{x}),\ldots,z_{n,n_n}(t,\mathbf{x})\big).$$

Note that, given the special form of the constraints it is easy to see that the adaptation laws presented in Section III can be represented in the following form

$$\dot{\mathbf{x}} = Z(t,\mathbf{x})\big[\nabla U(\mathbf{x}) - H(\mathbf{x})\mathbf{v}(\mathbf{x})\big],$$

where $\nabla U(\cdot)$ denotes the gradient of the function $U(\cdot)$, $H(\cdot)$ is the following matrix

$$H(\cdot) \doteq \big[\nabla h_1(\cdot)\ \nabla h_2(\cdot)\ \cdots\ \nabla h_L(\cdot)\big]$$

and $\mathbf{v}(\cdot) = \big[v_1(\cdot), v_2(\cdot), \ldots, v_L(\cdot)\big]^T$ is an L-dimensional vector whose entries are of the form

$$v_k(\mathbf{x}) = \begin{cases} \gamma_k & \text{if} \quad h_k(\mathbf{x}) > 0 \\ 0 & \text{if} \quad h_k(\mathbf{x}) < 0, \end{cases}$$

for $k = 1,2,\ldots,m$, where $\gamma_k = \alpha$ for the constraints associated with link capacity; i.e., $k = 1,2,\ldots,\mathrm{card}(\mathscr{L})$, $\gamma_k = \delta_{i,j}$ for the constraints associated with nonnegativity of $x_{i,j}$, $\gamma_k = \beta_i^M$ for the maximum allowed rate constraints on calls of type i and $\gamma_i = \beta_i^m$ for the minimum service guarantee constraints on calls of type i. Also,

$$v_k(\mathbf{x}) = \begin{cases} \xi_k & \text{if} \quad h_k(\mathbf{x}) > 0 \\ -\xi_k & \text{if} \quad h_k(\mathbf{x}) < 0 \end{cases}$$

for $k = m+1, m+2, \ldots, L$, where $\xi_k = \beta_i$ for the AS constraint k associated with calls of type i. Also, let the admissible domain be the set

$$\mathscr{D} \doteq \Big\{\mathbf{x} \in \mathbf{R}^{\sum_{i=1}^n n_i} : h_k(\mathbf{x}) \le 0 \text{ for } k = 1,2,\ldots,m \text{ and } h_k(\mathbf{x}) = 0 \text{ for } k = m+1, m+2, \ldots, L\Big\}.$$

Essentially, the proof requires 4 steps. First, we prove that the adaptation law converges to the maximum of the function

$$\widehat{U}(\mathbf{x}) \doteq U(\mathbf{x}) - \Xi(\mathbf{x}),$$

where

$$\Xi(\mathbf{x}) \doteq \big[h_1(\mathbf{x}), h_2(\mathbf{x}), \ldots, h_L(\mathbf{x})\big]\mathbf{v}(\mathbf{x}).$$

Second, we provide necessary and sufficient conditions for the maximum of $\widehat{U}(\mathbf{x})$ to coincide with the maximum of $U(\mathbf{x})$. Third, we show that under this conditions this procedure converges to the solution of the optimization problem at hand. The final step is the realization that, under the conditions of Theorem 1, the necessary and sufficient conditions mentioned above are satisfied.

Lemma 4: The function $\widehat{U}(\mathbf{x})$ does not decrease along the trajectories.

Proof: If a sliding mode does not occur then

$$\begin{aligned} \frac{d\widehat{U}}{dt} &= \big[\nabla U - H(\mathbf{x})\mathbf{v}(\mathbf{x})\big]^T \dot{x} \\ &= \big[\nabla U - H(\mathbf{x})\mathbf{v}(\mathbf{x})\big]^T Z(t,\mathbf{x})\big[\nabla U - H(\mathbf{x})\mathbf{v}(\mathbf{x})\big] \\ &\ge 0 \end{aligned}$$

since the matrix $Z(t,\mathbf{x})$ is positive definite.

Now, assume that a sliding mode occurs in the intersection of the surfaces $h_k(\mathbf{x}) = 0$, $k \in \mathscr{I}$. Let $H_1(\mathbf{x})$ be the matrix whose columns are $\nabla h_k(\mathbf{x})$ for $k \in \mathscr{I}$ (and in the same order as in $H(\mathbf{x})$). Also, let $H_2(\mathbf{x})$ be the matrix with columns $\nabla h_k(\mathbf{x})$ for $k \notin \mathscr{I}$ (again in the same order as in $H(\mathbf{x})$). Then, given that a sliding mode occurs in the intersection of the surfaces $h_k(\mathbf{x}) = 0$, $k \in \mathscr{I}$, we have

$$H_1(\mathbf{x})^T Z(t,\mathbf{x})\big[\nabla U - H_1(\mathbf{x})\mathbf{v}_1(\mathbf{x}) - H_2(\mathbf{x})\mathbf{v}_2(\mathbf{x})\big] = 0$$

where $\mathbf{v}_1(\mathbf{x})$ is the vector containing $v_k(\mathbf{x})$, for $k \in \mathscr{I}$ and $\mathbf{v}_2(\mathbf{x})$ is the vector containing $v_k(\mathbf{x})$, for $k \notin \mathscr{I}$. Now, assume that $\det\big[H_1(\mathbf{x})^T Z(t,\mathbf{x}) H_1(\mathbf{x})\big] \ne 0$ (a reasoning similar to the one in [28] can be done to address the case where this does not happen). From now on, to simplify the exposition, we drop the dependency on $\mathbf{x}$. Then, the equivalent control is

$$\mathbf{v}_{1,\mathrm{eq}} = \big(H_1^T Z H_1\big)^{-1}\big(H_1^T Z \nabla U - H_1^T Z H_2 \mathbf{v}_2\big)$$

and the sliding motion is

$$\begin{aligned} \dot{\mathbf{x}} &= Z\nabla U - ZH_1\mathbf{v}_{1,\mathrm{eq}} - ZH_2\mathbf{v}_2 \\ &= Z\nabla U - ZH_1\big(H_1^T Z H_1\big)^{-1}\big(H_1^T Z\nabla U - H_1^T Z H_2\mathbf{v}_2\big) - ZH_2\mathbf{v}_2 \\ &= \Big[Z - ZH_1\big(H_1^T Z H_1\big)^{-1} H_1^T Z\Big]\big(\nabla U - H_2\mathbf{v}_2\big) \\ &= \sqrt{Z}P\sqrt{Z}\big(\nabla U - H_2\mathbf{v}_2\big), \end{aligned}$$

where

$$\sqrt{Z} \doteq \mathrm{diag}\Big(\sqrt{z_{1,1}(t,\mathbf{x})}, \sqrt{z_{1,2}(t,\mathbf{x})}, \ldots, \sqrt{z_{n,n_n}(t,\mathbf{x})}\Big)$$

and

$$P \doteq I - \sqrt{Z}H_1\big(H_1^T\sqrt{Z}\sqrt{Z}H_1\big)^{-1}H_1^T\sqrt{Z}.$$

Now, let Ξ_1 be the elements h_k of Ξ with $k \in \mathscr{I}$ (in the same order as in Ξ). Also, let Ξ_2 be the elements h_k of Ξ with $k \notin \mathscr{I}$ (again in the same order as in Ξ). Since a sliding mode occurs, during this motion we have

$$\widehat{U} = U - \Xi_2\mathbf{v}_2.$$

Now, since U and Ξ_2 are continuously differentiable and along this sliding motion $\mathbf{v}_2$ is constant, we have

$$\frac{d\widehat{U}}{dt} = \left(\nabla U - H_2\mathbf{v}_2\right)^T \dot{\mathbf{x}}.$$

Now, notice that

$$P = P^T = P^2.$$

Hence,

$$\begin{aligned}\frac{d\widehat{U}}{dt} &= \left(\nabla U - H_2\mathbf{v}_2\right)^T \sqrt{Z}P\sqrt{Z}\left(\nabla U - H_2\mathbf{v}_2\right)\\ &= \left\|P\sqrt{Z}\left(\nabla U - H_2\mathbf{v}_2\right)\right\|^2\\ &\geq 0.\end{aligned}$$

■

Lemma 5: The time derivative of $\widehat{U}$ is zero only when $\dot{\mathbf{x}} = 0$.

Proof: If a sliding mode does not occur we have

$$\frac{d\widehat{U}}{dt} = \left[\nabla U - H\mathbf{v}\right]^T Z\left[\nabla U - H\mathbf{v}\right]$$

and since Z is positive definite

$$\frac{d\widehat{U}}{dt} = 0 \Rightarrow \nabla U - H\mathbf{v} = 0 \Rightarrow Z\left[\nabla U - H\mathbf{v}\right] = 0 \Rightarrow \dot{\mathbf{x}} = 0.$$

Now assume that a sliding mode occurs in the intersection of the surfaces $h_k(\mathbf{x}) = 0$, $k \in \mathscr{I}$. In this case,

$$\frac{d\widehat{U}}{dt} = \left\|P\sqrt{Z}\left(\nabla U - H_2\mathbf{v}_2\right)\right\|^2.$$

Hence,

$$\begin{aligned}\frac{d\widehat{U}}{dt} = 0 \Rightarrow P\sqrt{Z}\left(\nabla U - H_2\mathbf{v}_2\right) = 0 \Rightarrow&\\ \sqrt{Z}P\sqrt{Z}\left(\nabla U - H_2\mathbf{v}_2\right) = 0 \Rightarrow \dot{\mathbf{x}} = 0.&\end{aligned}$$

■

Lemma 6: The stationary points of $\widehat{U}$ are the maximum points of $\widehat{U}$.

Proof: Let $\mathbf{x}_0$ be a stationary point on the intersection of surfaces given by $H_1 = 0$. In this case, we have

$$\nabla U(\mathbf{x}_0) - H_1(\mathbf{x}_0)\mathbf{v}_{1,\text{eq}}(\mathbf{x}_0) - H_2(\mathbf{x}_0)\mathbf{v}_2(\mathbf{x}_0) = 0.$$

Now, consider the function

$$\widehat{U}^*(\mathbf{x}) = U(\mathbf{x}) - H_1(\mathbf{x})\mathbf{v}_{1,\text{eq}}(\mathbf{x}_0) - H_2(\mathbf{x})\mathbf{v}_2(\mathbf{x}).$$

Given that the components of the equivalent control satisfy

$$0 \leq v_{1,\text{eq},k}(\mathbf{x}_0) \leq \gamma_k$$

for $1 \leq k \leq m$ and

$$-\xi_k \leq v_{1,\text{eq},k}(\mathbf{x}_0) \leq \xi_k$$

for $m < k \leq L$, we have

$$H_1(\mathbf{x})\mathbf{v}_{1,\text{eq}}(\mathbf{x}_0) \leq H_1(\mathbf{x})\mathbf{v}_1(\mathbf{x})$$

and, as a consequence

$$\widehat{U}^*(\mathbf{x}) \geq \widehat{U}(\mathbf{x}).$$

Now, since U is a concave function, h_k are convex functions for $1 \leq k \leq m$ and h_k are linear functions for $m+1 \leq k \leq L$ then $\widehat{U}^*$ is a concave function and hence it has a unique maximum. Furthermore, $\widehat{U}^*$ is continuously differentiable and

$$\nabla\widehat{U}^*(\mathbf{x}_0) = 0.$$

Therefore

$$\max_{\mathbf{x}} \widehat{U}^*(\mathbf{x}) = \widehat{U}^*(\mathbf{x}_0).$$

Now, since $\widehat{U}^*(\mathbf{x}) \geq \widehat{U}(\mathbf{x})$ and $\widehat{U}^*(\mathbf{x}_0) = \widehat{U}(\mathbf{x}_0)$, we conclude that $\widehat{U}(\mathbf{x})$ reaches its maximum at $\mathbf{x}_0$. Therefore, any stationary point of the optimization procedure is a maximum of $\widehat{U}(\mathbf{x})$. Now, assume that a maximum point $\mathbf{x}^*$ of $\widehat{U}(\mathbf{x})$ is not a stationary point. Then, we have

$$\frac{d\widehat{U}(\mathbf{x}^*)}{dt} > 0$$

and so $\widehat{U}$ will increase along the trajectory which contradicts the fact that $\mathbf{x}^*$ is a maximum point of $\widehat{U}(\mathbf{x})$. ■

Lemma 7: If the set of all maximum points is bounded (which is our case) then $\mathbf{x}$ will converge to this set from any initial condition.

Proof: The proof of this result is very similar to the one in [28], Chapter 15, Section 3. It makes use of the results from Lemmas 4, 4 and 4. Therefore, we refer the reader to it. ■

Theorem 8: Let $\mathbf{v}^0$ be a vector whose entries are of the form

$$\begin{aligned}0 \leq v_k \leq \gamma_k\,; &\quad k = 1,2,\ldots,m\\ -\xi_k \leq v_k \leq \xi_k\,; &\quad k = m+1, m+2,\ldots,L\,,\end{aligned}$$

where $v_k = 0$ for nonbinding constraints. Then, the maximum of $\widehat{U}(\mathbf{x})$ coincides with the optimal $U(\mathbf{x}^*)$ if and only if there exists $\mathbf{x}^*$ such that

$$\nabla U(\mathbf{x}^*) = H(\mathbf{x}^*)\mathbf{v}^0.$$

Proof: See [28], Chapter 5, Section 4. ■

Theorem 9: The control laws presented above converge to the set of maximum points of the utility function $U(\mathbf{x})$ if this set is bounded, the condition of Theorem 8 is satisfied and vector $\mathbf{v}^0$ is an inner point of the set defined in Theorem 8, except for the nonbinding constraints.

Proof: See [28], Chapter 5, Section 4. The proof follows by using Theorem 8 and Lemma 7. ■

Remark: The components of vector $\mathbf{v}^0$ are the Lagrange multipliers of the optimization problem at hand.

B. Proof of Theorem 1

The conditions on the parameters α, β_i, β_l^m, β_i^M and $\delta_{i,j}$ imposed in Theorem 1 imply that the necessary and sufficient conditions of Theorem 8 are satisfied for the optimization problem at hand. Indeed, if each congested link is used by a nonbinding CoS or a BE call, then in $\nabla U(\mathbf{x}^*) = H(\mathbf{x}^*)\mathbf{v}^0$ the components of $\mathbf{v}^0$ associated with capacity constraints; i.e., v_k^0 for $k = 1,2,\ldots,\text{card}(\mathscr{L})$, appear in a set of equations decoupled from the remaining components of $\mathbf{v}^0$. Then, the worst case (larger) value of v_k^0, $k = 1,2,\ldots,\text{card}(\mathscr{L})$ is

$$v_{k,max}^0 = \max_{i,j,\mathbf{x}\in\mathscr{X}} \frac{\partial U(\mathbf{x})}{\partial x_{i,j}}$$

Now, using this information in the remaining equations, it is possible to solve for $v^0_{k,max}$, $k = \mathrm{card}(\mathscr{L})+1,\ldots .L$. Since, $U(\mathbf{x})$ is an increasing function in all its variables $x_{i,j}$, the worst case for v^0_k associated with CoS constraints is

$$v^0_{k,max} = \sum_{k=1}^{\mathrm{card}\,L} v^0_{k,max} = \max_{i,j} B_{i,j} \max_{i,j,\mathbf{x}\in\mathscr{X}} \frac{\partial U(\mathbf{x})}{\partial x_{i,j}} = \alpha_{min} \max_{i,j} B_{i,j}.$$

Once these are determined, all that remains is to pick the worst case value $v^0_{k,max}$ associated with nonnegativity constraints. Since each one of these appears in a single equation, where all the other multipliers are already determined, the worst case value is given by

$$v^0_{k,max} = \alpha_{min} \max_{i,j} B_{i,j} + \beta, \qquad \beta \in \{\beta_i, \beta_i^m, \beta_i^m\}.$$

Hence, in order to satisfy the conditions in Theorem 8

$$\begin{aligned} v_k &\le v^0_{k,max} < \gamma_k\,, && k = 1,2,\ldots,m \\ |v_k| &\le \left|v^0_{k,max}\right| < \xi_k\,, && k = m+1,\ldots,L\,. \end{aligned}$$

Therefore, the family of adaptation laws proposed in this paper converge to the maximum of the utility function $U(\mathbf{x})$ subject to $\mathbf{x} \in \mathscr{D}$. In other words, they converge to the optimum of our optimization problem. ■

References

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

[2] D. O. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao, "Overview and principles of Internet traffic engineering," Internet Engineering Task Force, RFC 3272, May 2002.

[3] F. Bonomi and K. W. Fendick, "The rate-based flow control framework for the available bit rate ATM service," *IEEE Network*, vol. 9, no. 2, pp. 25–39, Mar./Apr. 1995.

[4] F. Bonomi, D. Mitra, and J. B. Seery, "Adaptive algorithms for feedback-based flow control in high-speed, wide-area networks," *IEEE J. Select. Areas Commun.*, vol. 13, no. 7, pp. 1267–1283, Sept. 1995.

[5] D. Chiu and R. Jain, "Analysis of the Increase/Decrease algorithms for congestion avoidance in computer networks," *Journal of Computer Networks and ISDN systems*, vol. 17, no. 1, pp. 1–14, June 1989.

[6] A. F. Filippov, *Differential Equations with Discontinuous Righthand Sides*, ser. Mathematics and Its Applications. Dordrecht, The Netherlands: Kluwer Academic Publishers, 1988.

[7] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Trans. Networking*, vol. 1, no. 4, pp. 397–413, Aug. 1993.

[8] S. Floyd and T. Henderson, "The new Reno modification to TCP's fast recovery algorithm," Internet Engineering Task Force, RFC 2582, Apr. 1999.

[9] R. J. Gibbens and F. P. Kelly, "Resource pricing and the evolution of congestion control," *Automatica*, vol. 35, no. 12, pp. 1969–1985, 1999.

[10] R. J. Gibbens and P. Key, "Distributed control and resource marking using best-effort routers," *IEEE Network*, pp. 54–59, May/June 2001.

[11] S. J. Golestani and S. Bhattacharyya, "A class of end-to-end congestion control algorithms for the Internet," in *Proc. IEEE Int. Conf. Network Protocols, ICNP*. IEEE Computer Society, Oct. 1998, pp. 137–150.

[12] K. Kar, S. Sarkar, and L. Tassiulas, "A simple rate control algorithm for max total user utility," in *Proc. IEEE INFOCOM*, vol. 1, Apr. 2001, pp. 133–141.

[13] S. K. Korovin and V. I. Utkin, "Using sliding modes in static optimization and nonlinear programming," *Automatica*, vol. 10, pp. 525–532, 1974.

[14] S. Kunniyur and R. Srikant, "Analysis and design of an adaptive virtual queue (AVQ) algorithm for active queue management," *SIGCOMM'01*, pp. 123–134, Aug. 2001.

[15] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan, "Rate control in communication networks: Shadow prices, proportional fairness and stability," *J. Oper. Res. Soc.*, vol. 49, no. 3, pp. 237–252, Mar. 1998.

[16] R. J. La and V. Anantharam, "Charge-sensitive TCP and rate control in the Internet," in *Proc. IEEE INFOCOM*, Mar. 2000, pp. 1166–1175.

[17] C. M. Lagoa and H. Che, "Decentralized optimal traffic engineering in the Internet," *SIGCOMM Computer Communication Review*, Oct. 2000.

[18] T. Li and Y. Rekhter, "Provider architecture for differentiated services and traffic engineering (PASTE)," Internet Engineering Task Force, RFC 2430, Oct. 1998.

[19] S. H. Low, "Optimization flow control with on-line measurement or multiple paths," in *Proc. 16th Int. Teletraffic Congress*, Edinburgh, U.K., June 1999.

[20] L. Massoulié and J. Roberts, "Bandwidth sharing: Objectives and algorithms," in *Proc. IEEE INFOCOM*, Mar. 1999, pp. 1395–1403.

[21] D. Mitra and K. Ramakrishnan, "A case study of multiservice, multipriority traffic engineering design for data networks," in *Proc. Globecom*, vol. 1b, Dec. 1999, pp. 1077–1083.

[22] P. Newman, "Traffic management for ATM local area networks," *IEEE Commun. Mag.*, vol. 32, no. 8, pp. 44–50, Aug. 1994.

[23] K. K. Ramakrishnan and S. Floyd, "A proposal to add explicit congestion notification (ECN) to IP," Internet Engineering Task Force, RFC 2481, Jan. 1999.

[24] G. Ramamurthy and A. Kolarov, "Application of control theory for the design of closed loop rate control for ABR service," in *Proc. Int. Test Conf., ITC*, Washington, USA, 1997, pp. 751–760.

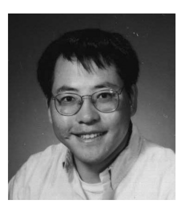
[25] S. Sarkar and L. Tassiulas, "Distributed algorithms for computation of fair rates in multirate multicast trees," in *Proc. IEEE INFOCOM*, vol. 1, Tel Aviv, Israel, Mar. 2000, pp. 52–61.

[26] J.-J. E. Slotine and W. Li, *Applied Nonlinear Control*. Englewood Cliffs, N.J.: Prentice-Hall, 1991.

[27] X. Su and G. de Veciana, "Dynamic multi-path routing: asymptotic approximation and simulations," in *Proc. 2001 ACM SIGMETRICS Int. Conf. Measurement and modeling of computer systems*. ACM Press, 2001, pp. 25–36.

[28] V. I. Utkin, *Sliding Modes in Control and Optimization*, ser. Communications and Control Engineering Series. Berlin, Heidelberg: Spriger-Verlag, 1992, vol. 66, no. 1.

Constantino Lagoa got his B.S. and M.Sc. degrees from the Instituto Superior Tecnico, Technical University of Lisbon, Portugal in 1991 and 1994 respectively and his Ph.D. degree from the University of Wisconsin at Madison in 1998. He joined the Electrical Engineering Department of The Pennsylvania State University in August 1998, where he currently holds the position of Assistant Professor. He has a wide range of research interests including robust control, controller design under risk specifications, control of computer networks and discrete event dynamical systems. In 2000 he received the NSF CAREER award for his proposal on system design under risk constraints.



Hao Che received the B.S. degree from Nanjing University, Nanjing, China, the M.S. degree in physics from the University of Texas at Arlington, TX, in 1994, and Ph.D. degree in electrical engineering from the University of Texas at Austin, TX, in 1998. He was an Assistant Professor of Electrical Engineering at the Pennsylvania State University, University Park, PA, from 1998 to 2001, and a System Architect with Santera Systems, Inc., Plano, TX, from 2000 to 2002. Since September 2002, he has been an Assistant Professor of Computer Science and Engineering at the University of Texas at Arlington, TX. He has also served as a Visiting Professor in the school of Information Science and Engineering, Yunnan University, China, since 2001. His current research interests include network architecture and design, network resource management, multiservice switching architecture, and network processor design.

Bernardo A. Movsichoff obtained his Ingeniero Electrónico degree from the Universidad de Buenos Aires, Argentina in December 1996. Since Fall 1999 he is at The Pennsylvania State University, where he is a Research Assistant working towards a Ph.D. degree in Electrical Engineering. His research interests include control and identification of uncertain systems and the application of control theory and optimization to computer networks.