

# Decentralized Optimal Traffic Engineering in the Internet<sup>\*</sup>

Constantino Lagoa  
Department of Electrical Engineering  
Pennsylvania State University  
University Park, PA 16802  
lagoa@enr.psu.edu

Hao Che  
Department of Electrical Engineering  
Pennsylvania State University  
University Park, PA 16802  
hxc30@psu.edu

## ABSTRACT

Distributed optimal traffic engineering in the presence of multiple paths has been found to be a difficult problem to solve. In this paper, we introduce a new approach in an attempt to tackle this problem. This approach has its basis in nonlinear control theory. More precisely, it relies on the concept of Sliding Modes. We develop a family of control laws, each of them having the property that the steady-state network resource allocation yields the maximum of the given utility function, subject to the network resource constraints. These control laws not only allow each ingress node to independently adjust its traffic sending rate but also provide a scheme for optimal traffic load redistribution among multiple paths. The only nonlocal information needed is binary feedback from each congested node in the path. Moreover, the algorithms 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 rate adaptive traffic with multiple paths, but also to assured service traffic with multiple paths. Preliminary case studies show that this technique is potentially very useful for optimal traffic engineering in a multiple-class-of-service and multiple-path enabled Internet, e.g., differentiated services enabled multi-protocol label switching networks.

## Categories and Subject Descriptors

C.2.1 [Computer-Communication Networks]: Network Architecture and Design

## Keywords

Traffic Engineering, Distributed Control, Sliding Modes, Nonlinear Control, MPLS

---

<sup>\*</sup>Partial funding for this research was provided by the National Science Foundation under Grant ECS-9984260

## 1. INTRODUCTION

The focal point of this paper is a network (e.g., an autonomous system) which has several paths available for routing flows between edge node pairs. The objective is to provide distributed control algorithms that will accommodate incoming traffic in an optimal way.

Network traffic management and control have become increasingly important as the Internet evolves into a global commercial communication infrastructure. The increase in complexity leads to the need for the development of the so-called automated Traffic Engineering (TE). TE improves network resource utilization and thus leads to an increase of revenue as well as satisfaction of diversified application requirements. As value-added services are being added to the Internet, TE algorithms should be able to address not only elastic Classes of Service (CoSs) but also inelastic CoSs in the presence of multiple paths.

Different methodologies have been proposed for TE. In [1], TE methodologies are classified into two basic types: time-dependent and state-dependent. In time-dependent TE, traffic control algorithms are used to optimize network resource utilization in response to long time scale traffic variations. They make no attempt to adapt to random short term traffic variations or changing network conditions. An example of time-dependent algorithm is the optimization-based centralized control algorithm proposed by Mitra, et al. [2]. In [2], the TE problem with multiple CoSs and multiple paths is formulated as a multi-commodity linear programming problem under the assumption that global information is available.

In contrast to time-dependent TE, in state-dependent TE, traffic control algorithms adapt to relatively fast network state changes. An example of state-dependent TE that operates in a relatively long time scale is constraint-based routing; e.g., see [3]. An example that operates in a relatively short time scale is load-balancing among multiple paths; e.g., see [4]. Our paper focuses on the state-dependent methodologies and explores the control issues for *globally optimal TE* in the presence of both multiple CoSs and multiple paths.

### 1.1 Decentralized Traffic Engineering

It has been widely recognized that there should be a development of distributed traffic control mechanisms for state-dependent TE which can quickly react to changes of the

network situation. There is extensive literature on distributed traffic control. It includes both empirical algorithms (e.g., see [5, 6]) and algorithms based on control theory; e.g., see [7, 8, 9]. These algorithms focus exclusively on congestion control issues for TCP (transport control protocol) or ABR (available bit rate) types of traffic. Moreover, they assume a single path and the approaches taken are not optimization-based. A distributed TE tool for load balancing of the best-effort traffic in an Multi-Protocol Label Switching (MPLS) network has been proposed in [4]. The objective of the proposed algorithm is to minimize the sum of the delays for multiple-path flows between an ingress-egress node pair. However, proof of optimality of the proposed algorithms was not presented.

Recently, several approaches have been proposed which address the distributed optimal TE problem using nonlinear optimization techniques. Their starting point is very similar to the one of this paper, i.e., maximization (minimization) of a utility (cost) function, subject to network resource constraints. Since different traffic types (i.e., flows with different ingress-egress node pairs) share the same network resources, the key challenge in the design of decentralized control laws is the fact that there is a high degree of interaction between different traffic types. The design of optimal control laws can be formalized as a nonlinear optimization problem where the constraints represent the interaction between different types of traffic. The high degree of “entanglement” of these constraints makes this a very complex optimization problem.

This obstacle has been circumvented in a number of different ways. In a paper by Golestani, et al. [10], 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 fed back from each of the links along the flow forwarding paths. The results presented by Kelly, et al. [11] 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 the “price,” which is done by each link in the network. The rate control at individual sources is based on the “prices” fed back from all the links in the data paths. Low [12] uses a technique which converts a constrained problem into a non-constrained dual problem. This reformulation results in a similar distributed control scheme. Iterative algorithms were also proposed and their convergence is proven for the single-path case. An attempt was also made in [12] to design a distributed multi-path algorithm. However, convergence of the algorithm was not proven.

## 1.2 A New Approach to Decentralized TE

To the best of our knowledge, no set of optimal distributed control laws and algorithms are available which address the case of multiple paths and multiple CoSs. There are two major obstacles to the application of nonlinear programming techniques to solve this kind of problem. First, the dual problem with multiple paths is non-differentiable [12] and is therefore difficult to solve using nonlinear programming

techniques. Second, Lagrangian approach 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. Since each link contributes at least one constraint, *every* link in the network has to periodically update and distribute one or more “prices” to all the source nodes using it, regardless of whether the link is congested or not. This raises the concern about possible high computational overhead imposed on the network nodes and large control traffic overhead in the network. Obviously, the problems will become more severe if multiple CoSs or multiple virtual private networks are to be supported in the same physical network.

To circumvent the problems mentioned above, we propose a new approach to the design of control algorithms for computer networks which relies on nonlinear control theory. Although some of the results are of a preliminary nature, 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 and/or redistribute traffic load among multiple paths, solely based on *binary* feedback information from the *congested nodes*. Moreover, the control law design technique allows for a large class of utility functions, namely utility functions that can be represented as a sum of concave terms. The technique can be used for TE of both rate adaptive and assured service CoSs. This technique is potentially very useful for optimal TE in a differentiated services enabled MPLS network [14].

## 1.3 Sliding Modes

As mentioned above, our 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 [15]. Many successful applications of this theory have been documented, from the control of robotic manipulators to the control of underwater vehicles.

The main idea behind sliding mode control is the use of discontinuous control laws to limit the evolution of the state of the system to a surface called “sliding surface.” Of particular interest to the problem addressed in this paper is the use of sliding modes in mathematical programming. The results in [13] indicate that sliding mode theory can be a powerful tool for optimizing convex functions subject to a large number of convex constraints. Motivated by this result, in this paper, we provide a class of adaptation control laws that converge to the optimal network resource allocation. These adaptation 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. A possible extension of the proposed approach to the control of aggregated inelastic traffic with multiple paths is also discussed.

## 1.4 The Sequel

The remainder of this paper is organized as follows. In Section 2, the proposed approach is described and the control laws presented. Section 3 is dedicated to the proofs of the results presented in Section 2 and it can be skipped by the reader who is solely interested in the application of the results of this paper. In Section 4, two examples of application of the results in this paper are presented. Finally, the conclusions and future work are given in Section 5.

## 2. MAIN RESULTS

To simplify the exposition, in this paper we assume that network resources are not shared by different CoSs and each CoS sees a separate logical network with dedicated resources. This implies that the control laws can be separately designed for individual CoSs. In our model, the traffic flows are assumed to be described by a fluid flow model and the only resource considered is the link bandwidth. This paper will not address practical implementation issues such as discretization, finite granularity, asynchronous updates and delays. Before presenting the main results, let us first introduce the notation that is used throughout this paper.

### 2.1 Notation

In the remainder of this paper we will use the terms call and flow interchangeably. Now, consider a computer network where calls of different types are present. Here, different types correspond to calls with different ingress/egress node pairs. Moreover, for a call of a given type there are several possible 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.

More precisely, consider a computer network whose set of links is denoted by  $\mathcal{L}$  and let  $c_l$  be the capacity of link  $l \in \mathcal{L}$ . Let  $n$  be the number of types of calls and  $n_i$  be the number of paths available for calls of type  $i$ . Finally, let  $\mathcal{L}_{i,j}$  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,

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

denotes the vector containing the data rates allocated to the different paths taken by calls of type  $i$ . Finally, the vector

$$\mathbf{x} \doteq [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n] \in \mathbf{R}^{n_1+n_2+\dots+n_n}$$

contains all the data rates allocated to different call types and respective paths.

We aim at maximizing utility functions of the form

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

subject to the network constraints, where  $f_i$  are differentiable concave functions, non decreasing in each one of the arguments,  $i = 1, 2, \dots, n$ . More precisely, we want to maximize the above utility function subject to

$$h_l(\mathbf{x}) \doteq \sum_{i,j:l \in \mathcal{L}_{i,j}} x_{i,j} - c_l \leq 0$$

for all  $l \in \mathcal{L}$  and  $x_{i,j} \geq 0$  for all  $i = 1, 2, \dots, n$  and  $j = 1, 2, \dots, n_i$ . Also, let  $b_{i,j}$  denote the number of bottleneck links encountered by calls of type  $i$  taking path  $j$ , i.e., the number of saturated links along the path taken by  $x_{i,j}$ . Finally, we use the notation  $g'(y^*)$  to denote the derivative of a function  $g$  evaluated at point  $y^*$ ; i.e.,

$$g'(y^*) \doteq \left. \frac{dg}{dy} \right|_{y=y^*}$$

and  $u(x)$  to denote the step function; i.e.,

$$u(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x < 0. \end{cases}$$

### 2.2 Rate Adaptive CoS with Multiple Paths

We are now ready to present the results that indicate how an edge node can adapt the sending rates so that they converge to the optimum values. These results indicate that, for utility functions of the form described in the previous section, there exists a family of adaptation laws that converges to the maximum. From this family, one can choose the one that best suits the problem at hand. Furthermore, the control law obtained is *decentralized*. By decentralized we mean that the adaptation law for the data rate  $x_{i,j}$  only depends on the data rates of calls of the same type  $i$  and the number of bottleneck links encountered in its path. We now provide a precise statement of a first result that gives the simplest member of this family.

**THEOREM 1.** *The adaptation law*

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

*converges to the maximum of the utility function  $U(\mathbf{x})$  for any constants  $\alpha > 0$  and  $\delta_{i,j} > 0$  satisfying*

$$\alpha - \left| \left. \frac{\partial f_i}{\partial x_{i,j}} \right|_{\mathbf{x}_i} \right| > \varepsilon,$$

*for some  $\varepsilon > 0$  and for all admissible  $\mathbf{x}_i$  and*

$$\delta_{i,j} > B_{i,j}\alpha.$$

*where  $B_{i,j}$  is the number of links used by calls of type  $i$  taking path  $j$ .*

Notice that the adaptation law provided above is decentralized. Adaptation of data rate  $x_{i,j}$  only uses knowledge of the number of congested nodes in path  $j$  and the value of  $\mathbf{x}_i$ ; i.e., to compute  $x_{i,j}$ , except for information on congestion, one only uses the data rates of calls of the same type. The term  $\delta_{i,j} u(-x_{i,j})$  is used to ensure that the sending rates are positive.

The result above only provides one of the possible adaptation laws. We now present an extension of the theorem above which describes a family of adaptation laws that converge to the optimal data rates.

THEOREM 2. *The adaptation law*

$$\dot{x}_{i,j} = \left[ \frac{\partial f_i}{\partial x_{i,j}} \Big|_{\mathbf{x}_i} - \alpha b_{i,j}(t) \right] [g'_{i,j}(g_{i,j}^{-1}(x_{i,j}))]^2 + \delta_{i,j} u(-x_{i,j}) |g'_{i,j}(g_{i,j}^{-1}(x_{i,j}))|$$

converges to the maximum of the utility function  $U(\mathbf{x})$  for any convex functions  $g_{i,j}(y_{i,j})$  whose inverse exists,  $g'_{i,j}(y_{i,j}) \neq 0$  and for which

$$f_i(g_{i,1}(y_{i,1}), g_{i,2}(y_{i,2}), \dots, g_{i,n_i}(y_{i,n_i}))$$

is a concave function of  $y_{i,1}, y_{i,2}, \dots, y_{i,n_i}$ ,  $i = 1, 2, \dots, n$ ,  $j = 1, 2, \dots, n_i$  and for any constants  $\alpha > 0$  and  $\delta_{i,j} > 0$  satisfying

$$\alpha - \left| \frac{\partial f_i}{\partial x_{i,j}} \Big|_{\mathbf{x}_i} \right| > \varepsilon,$$

for some  $\varepsilon > 0$  and

$$\delta_{i,j} > B_{i,j} \alpha |g'_{i,j}(g_{i,j}^{-1}(x_{i,j}))|.$$

for all admissible  $\mathbf{x}_i$ , where  $B_{i,j}$  is the number of links used by calls of type  $i$  taking path  $j$ .

### 2.3 Remarks

The above theorem provides a family of continuous rate adaptation laws concerning calls of type  $i$ . Note that type  $i$  refers to a given ingress/egress node pair. Therefore, the control laws above are distributed control laws in the sense that individual ingress nodes independently adjust their traffic sending rates and redistribute their traffic among multiple 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 nodes along flow forwarding paths. We can make use of the ideas of the binary congestion feedback signaling schemes proposed for ABR service [16] to convey  $b_{i,j}$  information back to the source. For instance, one possible scheme [17] is to let the congested nodes along the forwarding path periodically generate backward congestion messages. 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  $j$ . The effectiveness of different schemes together with other implementation issues, such as discretization, finite granularity, and delay, are subject to future investigation.

### 2.4 Assured Service with Multiple Paths

The approach above can be easily modified to accommodate assured service traffic. By assured service we mean that a target rate for a call should be guaranteed in average sense. In this case, if  $\Lambda_i$  is the target rate for calls of type  $i$ , the problem of bandwidth allocation can be easily reformulated leading to maximizing our utility function subject to link capacity constraints and

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

These extra constraints lead only to a small modification of the control law. The control law is given in the next theorem.

THEOREM 3. *The adaptation law*

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

where

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

converges to the maximum of the utility function  $U(\mathbf{x})$  for all constants  $\alpha > 0$ ,  $\beta_i > 0$  and  $\delta_{i,j} > 0$  satisfying

$$\alpha - \left| \frac{\partial f_i}{\partial x_{i,j}} \Big|_{\mathbf{x}_i} \right| > \varepsilon,$$

$$\beta_i - \left| \frac{\partial f_i}{\partial x_{i,j}} \Big|_{\mathbf{x}_i} \right| > \varepsilon,$$

for some  $\varepsilon > 0$  and for all admissible  $\mathbf{x}_i$  and

$$\delta_{i,j} > B_{i,j} \alpha + \beta_i.$$

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

### 2.5 Implementation Issues

If one uses continuous control, the control law described above guarantees that the trajectories of  $x_{i,j}$  will satisfy both link capacity constraints and assured service constraints. Hence, the target rate for the call of type  $i$  can be perfectly achieved. However, practical implementation requires discrete time, finite granular control, which will introduce "imperfections." Hence, the actual  $x_{i,j}$  values will oscillate and the target rate can be achieved only in average sense. As the control time interval and control granularity reduces, it can be proven that the oscillation subsides.

### 2.6 Remark on Non-Uniqueness

It should be noted that the data rate values that achieve the optimum value might not be unique. In other words, there might exist several solutions for the optimization problem studied in this paper. The control laws presented will converge to one of the achievers of the optimum. Moreover, once the optimum value of the utility function has been achieved, there will be no further adaptation of the data rates. For further details on the convergence properties of laws based on Sliding Mode Theory see [13].

## 3. PROOFS OF THEOREMS 1, 2 AND 3

In this section we provide the proofs of the theorems presented in the previous section. It can be skipped by the reader who is mainly interested in the application of the results provided in this paper. The proofs rely on the results presented in [13], which indicate how Sliding Mode Theory can be successfully applied to convex optimization problems. Consider the problem

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

where  $U(\cdot)$  is a concave function, subject to inequality constraints

$$h_l(\mathbf{x}) \leq 0; \quad l = 1, 2, \dots, m$$

and equality constraints

$$h_l(\mathbf{x}) = 0; \quad l = m+1, m+2, \dots, L.$$

The functions  $h_l(\cdot)$  are assumed to be convex for  $l = 1, 2, \dots, m$  and linear for  $l = m+1, m+2, \dots, L$ . This results in a convex optimization problem; i.e., the maximization of a concave function subject to convex constraints.

As it is proven in [13], adaptation laws of the following form

$$\dot{\mathbf{x}} = \nabla U(\mathbf{x}) - H(\mathbf{x}) \mathbf{v}(\mathbf{x}),$$

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

$$H(\cdot) = [\nabla h_1(\cdot) \quad \nabla h_2(\cdot) \quad \dots \quad \nabla h_L(\cdot)]$$

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

$$v_i(\mathbf{x}) = \begin{cases} \alpha_i & \text{if } h_i(\mathbf{x}) > 0 \\ 0 & \text{if } h_i(\mathbf{x}) < 0, \end{cases}$$

for  $i = 1, 2, \dots, m$  and

$$v_i(\mathbf{x}) = \begin{cases} \beta_i & \text{if } h_i(\mathbf{x}) > 0 \\ -\beta_i & \text{if } h_i(\mathbf{x}) < 0, \end{cases}$$

for  $i = m+1, m+2, \dots, L$ . Finally,  $\alpha_i > 0$ ,  $i = 1, 2, \dots, m$  and  $\beta_i > 0$ ,  $i = m+1, m+2, \dots, L$  are design constant parameters. To achieve convergence to the optimum these constants should satisfy the following condition: Let  $\mathbf{x}^*$  be a point that achieves the optimum of the problem above and let  $\mathcal{I}$  be the set of essential constraints at the point  $\mathbf{x}^*$ ; i.e.,

$$\mathcal{I} \doteq \{i \in \{1, 2, \dots, L\} : h_i(\mathbf{x}^*) = 0\}.$$

Then, the adaptation law converges to the optimal if there exists a vector  $\mathbf{v}^0 = [v_1^0, v_2^0, \dots, v_L^0]^T$  satisfying

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

and

$$\begin{aligned} v_i^0 &= 0 && \text{for } i \notin \mathcal{I}; \\ 0 \leq v_i^0 &< \alpha_i && \text{for } i \in \mathcal{I} \cap \{1, 2, \dots, m\} \\ -\beta_i &< v_i^0 < \beta_i && \text{for } i \in \mathcal{I} \cap \{m+1, m+2, \dots, L\}. \end{aligned}$$

This is a difficult condition to check, since we do not know a priori the optimal points  $\mathbf{x}^*$ . Instead, in this paper we use a sufficient condition which can also be found in [13]. Namely, the optimum value is achieved if there exists a constant  $\gamma < 0$  such that

$$\mathbf{v}^T(\mathbf{x}) H^T(\mathbf{x}) [\nabla U(\mathbf{x}) - H(\mathbf{x}) \mathbf{v}(\mathbf{x})] \leq \gamma < 0$$

for all  $\mathbf{x}$  leading to the violation of at least one of the constraints. We are now ready to present the proofs of the theorems stated in the previous section.

### 3.1 Proof of Theorem 1

Recall that in this case the function  $U(\mathbf{x})$  has the form

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

Furthermore, the constraints are of the form

$$h_l(\mathbf{x}) = \sum_{i,j:l \in \mathcal{L}_{i,j}} x_{i,j} - c_l \leq 0$$

for  $l \in \mathcal{L}$  and

$$\tilde{h}_{i,j}(\mathbf{x}) = -x_{i,j} \leq 0$$

for  $i = 1, 2, \dots, n$  and  $j = 1, 2, \dots, n_i$ . We now apply the algorithm introduced in [13] (presented at the beginning of this section). The algorithm indicates that

$$\begin{aligned} \dot{x}_{i,j} &= \frac{\partial U}{\partial x_{i,j}} \Big|_{\mathbf{x}} - \sum_{l \in \mathcal{L}} \frac{\partial h_l}{\partial x_{i,j}} \Big|_{\mathbf{x}} v_l(\mathbf{x}) \\ &\quad - \sum_{k=1}^n \sum_{m=1}^{n_i} \frac{\partial \tilde{h}_{k,m}}{\partial x_{i,j}} \Big|_{\mathbf{x}} \tilde{v}_{k,m}(\mathbf{x}). \end{aligned}$$

Note that

$$\frac{\partial U}{\partial x_{i,j}} \Big|_{\mathbf{x}} = \frac{\partial f_i}{\partial x_{i,j}} \Big|_{\mathbf{x}_i}.$$

Furthermore,

$$\frac{\partial h_l}{\partial x_{i,j}} \Big|_{\mathbf{x}} = \begin{cases} 1 & \text{if } l \in \mathcal{L}_{i,j} \\ 0 & \text{if } l \notin \mathcal{L}_{i,j} \end{cases}$$

for all  $l \in \mathcal{L}$  and

$$\frac{\partial \tilde{h}_{k,m}}{\partial x_{i,j}} \Big|_{\mathbf{x}} = \begin{cases} -1 & \text{if } k = i \text{ and } m = j \\ 0 & \text{otherwise.} \end{cases}$$

Now, recall that in this optimization problem there are only inequality constraints. However, instead of using the general adaptation law presented at the beginning of this section, we use

$$v_l(\mathbf{x}) = \begin{cases} \alpha & \text{if } h_l(\mathbf{x}) > 0 \\ 0 & \text{if } h_l(\mathbf{x}) < 0, \end{cases}$$

for all  $l \in \mathcal{L}$ . In other words, we take  $\alpha_l$  to be equal to a constant  $\alpha$  for all  $l \in \mathcal{L}$ . Also, take

$$\tilde{v}_{k,m}(\mathbf{x}) = \begin{cases} \delta_{k,m} & \text{if } \tilde{h}_{k,m}(\mathbf{x}) > 0 \\ 0 & \text{if } \tilde{h}_{k,m}(\mathbf{x}) < 0, \end{cases}$$

This implies

$$\sum_{l \in \mathcal{L}} \frac{\partial h_l}{\partial x_{i,j}} \Big|_{\mathbf{x}} v_l(\mathbf{x}) = \alpha b_{i,j}$$

where  $b_{i,j}$  is the number of bottleneck links for  $x_{i,j}$ . Furthermore,

$$\sum_{k=1}^n \sum_{m=1}^{n_i} \frac{\partial \tilde{h}_{k,m}}{\partial x_{i,j}} \Big|_{\mathbf{x}} \tilde{v}_{k,m}(\mathbf{x}) = -\delta_{i,j} u(-x_{i,j}).$$

Hence, the adaptation law

$$\dot{x}_{i,j} = \frac{\partial f_i}{\partial x_{i,j}} \Big|_{\mathbf{x}_i} - \alpha b_{i,j}(t) + \delta_{i,j} u(-x_{i,j}),$$

$i = 1, 2, \dots, n$  and  $j = 1, 2, \dots, n_i$ , will converge to the optimum if the constants  $\alpha$  and  $\delta_{i,j}$  satisfy the conditions for convergence. Recall that  $\delta_{i,j} > \alpha$  and

$$\alpha - \left| \frac{\partial f_i}{\partial x_{i,j}} \Big|_{\mathbf{x}_i} \right| > \varepsilon,$$

for some  $\varepsilon > 0$ . Let

$$\begin{aligned} w(\mathbf{x}) &\doteq H(\mathbf{x})v(\mathbf{x}) \\ &= [w_{1,1}(\mathbf{x}), \dots, w_{1,n_1}(\mathbf{x}), w_{2,1}(\mathbf{x}), \dots, w_{n,n_n}(\mathbf{x})]^T \end{aligned}$$

where

$$w_{i,j}(\mathbf{x}) = \alpha b_{i,j}(t) - \delta_{i,j}u(-x_{i,j}),$$

If at least one of the constraints is violated, we have

$$\begin{aligned} w(\mathbf{x})^T [\nabla U(\mathbf{x}) - w(\mathbf{x})] &= \sum_{i=1}^n \sum_{j=1}^{n_i} w_{i,j}(\mathbf{x}) \left( \frac{\partial f_i}{\partial x_{i,j}} - w_{i,j}(\mathbf{x}) \right) \\ &\leq -\varepsilon \min \left\{ \alpha, \min_{i,j} (\delta_{i,j} - \alpha) \right\} < 0. \end{aligned}$$

Hence, the sufficient condition for convergence is satisfied. Finally, the fact that

$$\delta_{i,j} > B_{i,j}\alpha \geq b_{i,j}(t)\alpha$$

implies that

$$\dot{x}_{i,j} > 0$$

whenever  $x_{i,j} < 0$ . Therefore, if the initial value of  $x_{i,j}(0)$  is nonnegative then  $x_{i,j}(t) \geq 0$  for all  $t > 0$ .

### 3.2 Comments on the Proof of Theorem 2

Without loss of generality assume that the functions  $g_{i,j}(\cdot)$  are increasing. A straightforward modification of the reasoning below can be applied to the case where some of the  $g_{i,j}(\cdot)$  are decreasing. Define

$$y_{i,j} \doteq g_{i,j}^{-1}(x_{i,j})$$

and consider the equivalent optimization problem on the variables  $y_{i,j}$

$$\max \sum_{i=1}^n f_i [g_{i,1}(y_{i,1}), g_{i,2}(y_{i,2}), \dots, g_{i,n_i}(y_{i,n_i})],$$

subject to

$$h_l(\mathbf{y}) = \sum_{i,j:l \in \mathcal{L}_{i,j}} g_{i,j}(y_{i,j}) - c_l \leq 0$$

for  $l \in \mathcal{L}$  and

$$\tilde{h}_{i,j}(\mathbf{y}) = -y_{i,j} + g_{i,j}^{-1}(0) \leq 0$$

for  $i = 1, 2, \dots, n$  and  $j = 1, 2, \dots, n_i$ . Since

$$f_i [g_{i,1}(y_{i,1}), g_{i,2}(y_{i,2}), \dots, g_{i,n_i}(y_{i,n_i})]$$

is a concave function of  $\mathbf{y}$ , we can apply the Sliding Mode optimization algorithm provided in [13]. Using the same line of reasoning as in the previous proof, we obtain the following adaptation law for the new variables  $y_{i,j}$

$$\begin{aligned} \dot{y}_{i,j} &= \frac{\partial f_i}{\partial x_{i,j}} \Big|_{\mathbf{x}_i=[g_{i,1}(y_{i,1}), g_{i,2}(y_{i,2}), \dots, g_{i,n_i}(y_{i,n_i})]^T} g'_{i,j}(y_{i,j}) \\ &\quad - \alpha b_{i,j}(t) g'_{i,j}(y_{i,j}) + \delta_{i,j}u[-y_{i,j} + g_{i,j}^{-1}(0)] \end{aligned}$$

where  $b_{i,j}(t)$  is the number of constraints involving  $y_{i,j}$  which are violated at time  $t$ . Again, a sufficient condition for convergence of the algorithm is

$$\alpha - \frac{\partial f_i}{\partial x_{i,j}} \Big|_{\mathbf{x}_i=[g_{i,1}(y_{i,1}), g_{i,2}(y_{i,2}), \dots, g_{i,n_i}(y_{i,n_i})]^T} > \varepsilon,$$

for some  $\varepsilon > 0$  and

$$\delta_{i,j}u[-y_{i,j} + g_{i,j}^{-1}(0)] - B_{i,j}\alpha g'_{i,j}(y_{i,j}) > 0$$

for all admissible  $\mathbf{y}_i = [y_{i,1}, y_{i,2}, \dots, y_{i,n_i}]^T$ , where  $B_{i,j}$  is the number of constraints that depend on  $y_{i,j}$ . Now, just use the fact that

$$x_{i,j} = g_{i,j}(y_{i,j})$$

and

$$\dot{x}_{i,j} = g'_{i,j}(y_{i,j})\dot{y}_{i,j}$$

to obtain the adaptation law presented in the theorem.

### 3.3 Remark

Note that if, for some  $i$  and  $j$ , the function  $g_{i,j}(\cdot)$  is decreasing then the constraint

$$\tilde{h}_{i,j}(\mathbf{y}) = -y_{i,j} + g_{i,j}^{-1}(0) \leq 0$$

should be substituted by

$$\tilde{h}_{i,j}(\mathbf{y}) = y_{i,j} - g_{i,j}^{-1}(0) \leq 0$$

giving rise to the term

$$\delta_{i,j}u(-x_{i,j}) |g'_{i,j}(g_{i,j}^{-1}(x_{i,j}))|$$

in the control law of  $x_{i,j}$ .

### 3.4 Comments on the Proof of Theorem 3

The main difference between the optimization problem addressed in this section and the previous ones is that we have additional equality constraints. In other words, one has constraints of the form

$$h_i(\mathbf{x}) = 0.$$

The Sliding Mode algorithm described at the beginning of this section also addresses this type of constraints. As we have seen in this case, the adaptation algorithm has an additional term. This additional term is of the form

$$\nabla h_i(\mathbf{x})v_i(\mathbf{x})$$

where

$$v_i(\mathbf{x}) = \begin{cases} \beta_i & \text{if } h_i(\mathbf{x}) > 0 \\ -\beta_i & \text{if } h_i(\mathbf{x}) < 0, \end{cases}$$

where  $\beta_i$  is a positive constant which must satisfy the convergence conditions. The remainder of the proof follows very closely the proof of Theorem 1. Just apply the rule for equality constraints taking into account their particular form and obtain the adaptation law presented in the statement of the theorem.

## 4. EXAMPLES

In this section we present two application examples of the results in this paper. The first one exemplifies how the framework presented in this paper generalizes the

linear-increase-exponential-decrease TCP-type control law to the case where multiple paths are available. The second example illustrates the use of the control law in the presence of assured traffic.

## 4.1 TCP-Type Congestion Control

As a simple example, consider the case where  $n_i$  paths are available for calls of type  $i$ . Let the utility function be of the form

$$U(x) = \sum_{i=1}^n k_i \log \left( \sum_{j=1}^{n_i} x_{i,j} \right),$$

where  $k_i > 0$  are constants. This is the so-called ‘‘law of diminishing returns.’’ The application of Theorem 2 to this problem yields the following family of adaptation laws that will converge to the maximum of the above utility function:

$$\dot{x}_{i,j} = \frac{k_i}{\sum_{j=1}^{n_i} x_{i,j}} [g'_{i,j}(g_{i,j}^{-1}(x_{i,j}))]^2 + \delta_{i,j} u(-x_{i,j}) |g'_{i,j}(g_{i,j}^{-1}(x_{i,j}))|$$

if there are no bottlenecks on path  $j$  and

$$\dot{x}_{i,j} = \left[ \frac{k_i}{\sum_{j=1}^{n_i} x_{i,j}} - \alpha b_{i,j} \right] [g'_{i,j}(g_{i,j}^{-1}(x_{i,j}))]^2 + \delta_{i,j} u(-x_{i,j}) |g'_{i,j}(g_{i,j}^{-1}(x_{i,j}))|$$

otherwise,  $i = 1, 2, \dots, n$ ,  $j = 1, 2, \dots, n_i$ . As we have seen in Section 2,  $\alpha > 0$  and  $\delta_{i,j}$  are design constants,  $g_{i,j}(\cdot)$  is a function that can be arbitrarily chosen within a given class and  $b_{i,j}$  is the number of bottleneck links encountered by calls of type  $i$  taking path  $j$ .

Notice that, if only one path is available for calls of type  $i$  (i.e,  $n_i = 1$ ,  $i = 1, 2, \dots, n$ ), and

$$g_{i,j}(y) = \frac{1}{4} y^2 \Rightarrow [g'_{i,j}(g_{i,j}^{-1}(x_{i,j}))]^2 = x_{i,j},$$

for high values of data rate the above law reduces to the well known linear-increase-exponential-decrease bandwidth adaptation law which has been shown to converge to the optimum of the ‘‘law of diminishing returns;’’ e.g., see [11].

## 4.2 Assured Service Traffic

As a demonstration of the power of the approach introduced in this paper, consider the network in Figure 1, where calls of type 1 have only one path available and calls of type 2, 3 and 4 have two paths available. Assume that the capacity of each link is 10. In this example, the following utility function is used

$$U(x) = \log(x_1) + \sum_{i=2}^4 \log(x_{i,1} + x_{i,2}).$$

All traffic is assumed to be of the assured service type, with target rate 10 for calls of types 2, 3 and 4.

At the beginning of the simulation, the sending rate of calls of type 1 is 5. Moreover, the sending rate of calls of types 2, 3 and 4 is equally partitioned between the available routes. More precisely, we have

$$\begin{aligned} x_1(0) &= 5 & x_{2,1}(0) &= 5 & x_{3,1}(0) &= 5 & x_{4,1}(0) &= 5 \\ & & x_{2,2}(0) &= 5 & x_{3,2}(0) &= 5 & x_{4,2}(0) &= 5 \end{aligned}$$

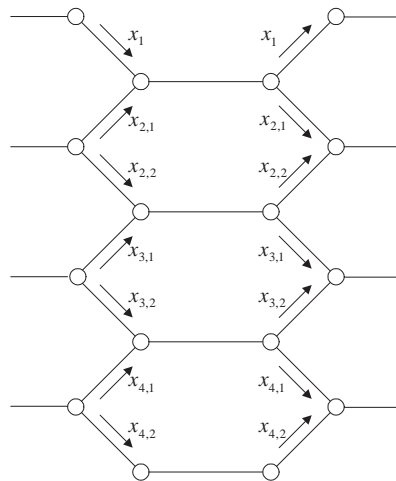


Figure 1: Sample Network

Therefore, the network is at a stationary point where all the demand is being met. Then, we introduce an additional demand of calls of type 1 of size 5. Using Theorem 3, we obtain the following adaptation laws

$$\dot{x}_1 = \frac{1}{x_1} - b_1(t) - r_1(t) + 4u(-x_1(t))$$

where  $r_1(t) = 1$  if  $x_1(t) > 10$  and  $r_1(t) = -1$  if  $x_1(t) < 10$  and

$$\dot{x}_{i,j} = \frac{1}{x_{i,1} + x_{i,2}} - b_{i,j}(t) - r_i(t) + 4u(-x_{i,j}(t))$$

for  $i = 2, 3, 4$ ,  $j = 1, 2$ , where  $r_i(t) = 1$  if  $x_{i,1}(t) + x_{i,2}(t) > 10$  and  $r_i(t) = -1$  if  $x_{i,1}(t) + x_{i,2}(t) < 10$ . In other words, we chose  $\alpha = 1$  and  $\delta_{i,j} = 4$  (each call uses 3 links).

As it can be seen in the simulation results presented in Figure 2, the traffic was reallocated so that all the demands were met. By the end of the simulation the distribution of the traffic was the following

$$\begin{aligned} x_1 &= 10 & x_{2,1} &= 0 & x_{3,1} &= 0 & x_{4,1} &= 0 \\ & & x_{2,2} &= 10 & x_{3,2} &= 10 & x_{4,2} &= 10 \end{aligned}$$

showing that this approach endows the network with the ability to adapt to changing demands. If enough resources are available, it will accept all the traffic. Otherwise, it will adapt the data rates in order to maximize the utility function. Note that we do not show the time scale in the plots in Figure 2. This information was omitted because the speed of convergence depends on the choice of the utility function. If one chooses a new utility function  $V(\mathbf{x})$  of the form

$$V(\mathbf{x}) = \gamma U(\mathbf{x})$$

with  $\gamma > 0$ , then the speed of convergence is  $\gamma$  times faster than the original one. The results presented are only indicative of the behavior of the network. The time scale associated with this behavior is a consequence of the design parameters.

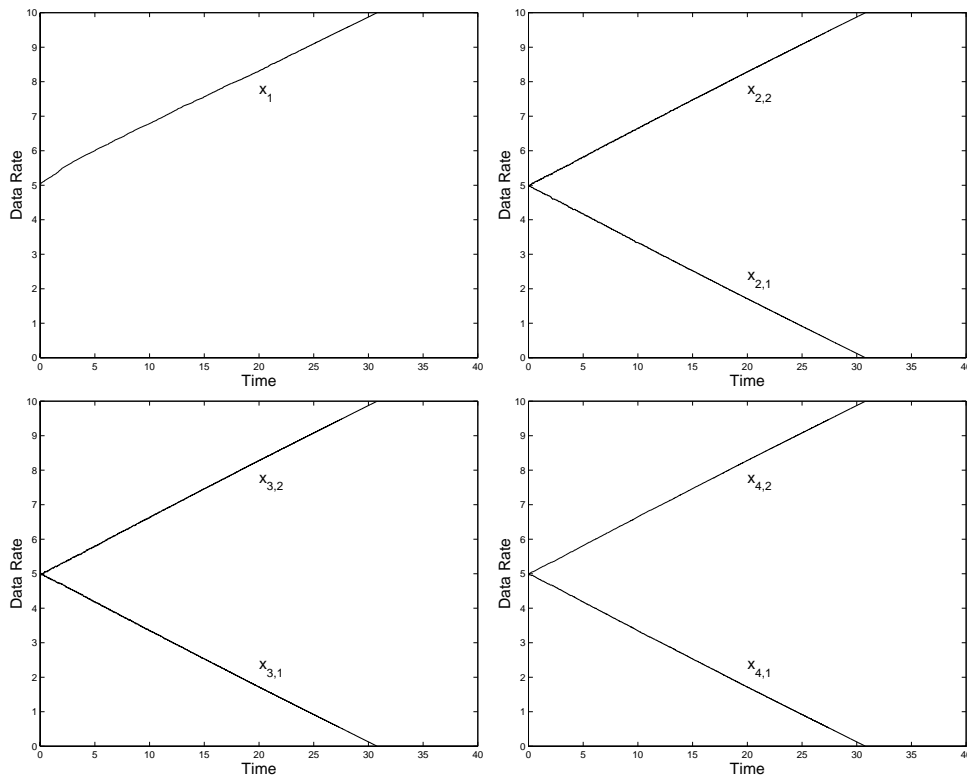


Figure 2: Simulation Results

## 5. CONCLUSIONS AND FUTURE WORK

In this paper, we introduced a new traffic control technique based on nonlinear control theory. The initial results indicate that the Sliding Mode concept is a powerful tool for computer network control and optimization. The technique provides a family of continuous adaptation laws for optimal TE for different CoSs in a multiple-path network. Moreover, the control laws obtained are decentralized and only require binary feedback from congested nodes. However, this paper is just the first step towards a comprehensive solution and many open issues are yet to be addressed.

### 5.1 Different CoSs

In this paper we assumed that resources are not shared by different CoSs and each CoS sees a separate logical network with dedicated resources so that the control laws can be separately designed for individual CoSs. In reality, however, network resources are to be shared among different CoSs and multiple resource constraints need to be imposed on individual links. Also the utility function needs to be carefully designed to include all the CoSs. To this end, resource allocation/sharing policies among different CoSs need to be incorporated as part of the algorithm design. This research issue is currently under investigation.

### 5.2 Discretization, Granularity and Delays

In a practical setting, one does not have continuous adaptation laws. Practical implementation requires individual ingress nodes to periodically adjust the traffic load at finite granularity. Moreover, link congestion feedback is triggered by the congestion itself and is subject to a delay. Part of

our future work is to study the consequences of discretization, since practical implementation requires discrete control laws. Not only is the adaptation not continuous with respect to time, it is granular and there is no synchronization between nodes. These implementation issues will introduce “imperfections” in the control algorithm that might result in a loss of performance or even instability. Therefore, an important part of our future work is to study the effects of discretization and to provide implementation algorithms that will minimize the adverse effects of this phenomenon.

### 5.3 Choosing the Right Control Law

Also of particular interest is the choice of the adaptation law. The approach presented in this paper provides a family of adaptation laws that will converge to the desired optimal value. However, each one of these laws will have a different transient response, in particular different convergence speeds. Therefore, one of the objectives of our future research is to study the properties of these adaptation algorithms so that one can choose the one that is best suited to solve the problem at hand.

### 5.4 Inelastic CoSs with Multiple Paths

Finally, we discuss a possible application of the control laws to optimal control of inelastic CoSs with multiple paths. Inelastic applications generally have stringent QoS (quality of service) requirements. For this kind of applications, multiple paths should not be used in its literal sense, simply because the received packets may be out-of-order and packet reordering at the receiver introduces undesirable variable delay and delay jitter. Also inelastic flow re-routing for the purpose



of load balancing is undesirable because a transition from one route to another may cause extra delay, delay jitter, and even packet reordering problems. Hence, at micro-flow level, only one path should be selected out of a given set of candidate paths for an inelastic flow. However, preliminary results indicate that the results in this paper can be applied to inelastic traffic when combined with a concept proposed in [14], called *trunk*. A trunk is a bundle of micro-flows of one or more CoSs who share the same forwarding behavior and the same edge node pair. Hence, we can treat each trunk as a fluid and apply the algorithms developed in this paper. The main difference resides in the fact that, in this case, we cannot modify the data rate of a micro-flow, change its path or terminate it. Therefore, we are looking into the possibility of directly controlling the call admission process. In other words, one controls the total rate occupied by a specific trunk by controlling the number of micro-flows that are accepted. Decreasing of the overall data rate is done by letting the calls that are in progress end and not replacing them with new ones. Preliminary simulations indicate that this is a very efficient way of handling inelastic traffic.

## Acknowledgement

The authors gratefully acknowledge the critical comments made by the reviewers which helped to substantially improve the quality of the paper.

## 6. REFERENCES

- [1] D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao, "A Framework for Internet Traffic Engineering," IETF Draft: draft-ietf-tewg-framework-01.txt, January 2000.
- [2] D. Mitra and K. G. Ramakrishnan, "A Case Study of Multiservice, Multipriority Traffic Engineering Design for Data Networks," Proceedings of GLOBECOM'99, Vol. 1B, pp. 1077-1083, December 1999, Brazil.
- [3] M. Kodialam and T. Lakshman, "Minimum Interference Routing with Applications to MPLS Traffic Engineering," *Proceedings of IEEE INFOCOM'2000*, March 2000, Israel.
- [4] I. Widjaja and A. Elwalid, "MATE: MPLS Adaptive Traffic Engineering," IETF Draft: draft-widjaja-mpls-mate-00.txt, August 1998.
- [5] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," *IEEE/ACM Trans. on Networking*, Vol. 1, No. 4, pp. 397, August 1993.
- [6] S. Floyd and T. Henderson, "The NewReno Modification to TCP's Fast Recovery Algorithm," IETF RFC 2582, April 1999.
- [7] F. Bonomi, D. Mitra, and J. Seery, "Adaptive Algorithms for Feedback-Based Flow Control in High-Speed, Wide-Area Networks," *IEEE Journal on Selected Areas in Communications*, Vol. 13, No. 7, Sept. 1995.
- [8] D. Chiu and R. Jain, "Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks," *Computer Networks and ISDN Systems*, Vol. 17, pp. 1, 1989.
- [9] G. Ramamurthy and A. Kolarov, "Application of Control Theory for the Design of Closed Loop Rate Control for ABR Service," *Proceedings of ITC 15*, pp. 751, 1997.
- [10] S. Golestani and S. Bhattacharyya, "A Class of End-to-End Congestion Control Algorithms for the Internet," *Proc. ICNP*, October 1998.
- [11] F. Kelly, A. Maulloo, and D. Tan, "Rate Control for Communication networks: Shadow Prices, Proportional Fairness and Stability," *Journal of the Operational Research Society*, 49, 1998.
- [12] S. Low, "Optimization Flow Control with On-line Measurement or Multiple Paths," <http://www.ee.mu.oz.au/staff/slow/papers/ofcb.ps>.
- [13] S. K. Korovin and V. I. Utkin, "Using sliding modes in static optimization and nonlinear programming," *Automatica*, 10, pp. 525-532, 1974 and the references therein.
- [14] T. Li and Y. Rekhter, "Provider Architecture for Differentiated Services and Traffic Engineering (PASTE)," IETF Draft: draft-li-paste-01.txt, September 1998.
- [15] J.-J. E. Slotine and W. Li, *Applied Nonlinear Control*, Prentice-Hall, Englewood Cliffs, 1991.
- [16] F. Bonomi and K. W. Fendick, "The Rate-Based Flow Control Framework for the Available Bit Rate ATM Service," *IEEE Network Mag.*, Vol. 9, No. 2, pp. 25-39, April 1995.
- [17] P. Newman, "Traffic Management for ATM Local Area Networks," *IEEE Commun. Mag.*, Vol. 32, No. 8, pp. 44-50, Aug. 1994.