Optimization-Based, QoS-aware Distributed Traffic Control Laws for Networks with Time-Varying Link Capacities

Wenjing Su ^a, Constantino Lagoa ^a, Hao Che ^b

^a*The Pennsylvania State University, University Park, PA, USA*

^b*University of Texas at Arlington, TX, USA*

Abstract

It is a challenge to design optimal, distributed traffic control mechanisms in a network where the link capacities and Class of Service (CoS) requirements may vary with time, such as a virtual network, an overlay network, or a wireless network. In this paper, we develop a family of optimization-based distributed traffic control laws to meet this challenge. This family of control laws enables Quality of Service (QoS), Traffic Engineering (TE), and Failure Recovery (FR) features simultaneously in a network where the link capacities and CoS requirement may vary with time. The approach taken relies on the concept of Sliding Modes to solve the resulting time-varying optimization problem. Running at the edge of a network, a set of control laws selected from this family enables class-of-service-based multi-path load balancing and/or rate adaptation to respond to network congestion, CoS requirement variation and link failures. The only nonlocal information needed as input to the control laws is the number of congested links along a forwarding path. This family of control laws is particularly viable to be implemented in a software-defined network (SDN) where the available underlying resources may not be accurately predictable and vary with time, due to, e.g., network virtualization.

Key words: overlay networks, traffic engineering, optimization, sliding modes control, quality of service.

1 Introduction

As communication networks become large, e.g., the Internet with global coverage, user traffic control must be distributed by design. Namely, the flow rates for the flows sending between various source-destination host pairs must be controlled by individual pairs independent of one another. The problem then arises as to how to control individual flows so that the network will converge to a stable state that achieves certain design objective. Mathematically, this problem is generally stated as follows: Find flow control laws that maximize the sum of individual user utilities as functions of flow rates, under network resource constraints. Significant research effort has been made in developing such traffic control laws. The aim is to provide theoretical underpinnings for the development of distributed traffic control protocols with provable properties, including scalability, stability, and optimality. This line of research has proven to be suc-

cessful and has resulted in the finding of large families of optimization-based, distributed control laws (e.g., [14,23,8,9,11,16]). This development is highly encouraging because by using these control laws, distributed traffic control protocols can now be developed based on *theory*, rather than ad hoc/empirical design.

However, all the existing control laws work under the assumption that the link bandwidths (or equivalently, link capacities) are fixed, which largely limits the applicability of such control laws. This is simply because the assumption may not hold true for some segments of the Internet, e.g., wireless networks, overlay networks, or virtual networks.

In a wireless network, the effective radio channel capacity is both time and space varying [12], as a result of several factors, including signal power attenuation, inter-channel interference, thermal noise, Doppler frequency shift, and multipath channel fading. An overlay network particularly enabled at the application layer, provides a convenient way to quickly deploy services with minimum interruption to the IP layer infrastructure, such as a P2P network [22] and a VoIP network [18]. In such networks, the underlying network topology and resources are hid-

* This work was partially supported by the National Science Foundation under grants CNS-1329422 and ECCS-1201973.

Email addresses: wenjing-su@hotmail.com (Wenjing Su), lagoa@engr.psu.edu (Constantino Lagoa), hche@cse.uta.edu (Hao Che).

den. Moreover, the network resources must be shared between the overlay traffic and the underlying network layer traffic. As a result, an overlay link capacity is time varying and the network layer path that forms an overlay link may change from time to time.

Network virtualization [17] is a natural extension of server virtualization to allow fully virtualized distributed computing; e.g., in a data center. Virtualization aims at emulating a hardware platform, such as a server, storage device or network resource, in software. Network virtualization has come into focus recently, with the true decoupling of the control and forwarding data planes, as advocated by software-defined networking (SDN) [1] and network functions virtualization (NFV) [2]. On one hand, the decoupling of the virtual networks from its physical counterparts allows the use of any general off-the-shelf physical platforms and multiple virtual networks sharing the same physical platform. On the other hand, the decoupling of the control and data planes makes it difficult to accurately predict the resource availability in virtual networks, which may change from time to time.

The main contribution of this paper is the development of a family of optimization-based, distributed, multipath enabled traffic control laws, which can function properly in the above mentioned networks with time-varying link bandwidths, hidden and/or changing topologies. These distributed traffic control laws enable rich QoS, TE, and FR features through QoS-aware multipath load balancing and rate adaptation, and require only a single-bit binary congestion feedback between two nodes interconnected through a link with time-varying link capacity. For example, in the case of an virtual/overlay network, it requires only a single-bit binary congestion feedback between two virtual/overlay nodes, which can be inferred by the virtual/overlay nodes themselves. This allows an overlay/virtual link to be truly virtual, in the sense that the network layer path that forms this virtual/overlay link is completely hidden from the overlay and may change from time to time.

Our approach has its basis in the theory of Sliding Mode Control. The results in [21] indicate that Sliding Modes theory can be a powerful tool for optimizing convex functions subject to a large number of convex constraints. Motivated by this result, we provide a family of optimal distributed traffic allocation algorithms for the time varying problem described above.

Finally, we note that our control laws are not meant to be a comprehensive solution, nor concerned with the protocol development and implementation. Instead, it is a theoretical underpinning upon which highly scalable and versatile traffic control mechanisms can be developed. It is quite similar to the distributed Bellman-Ford algorithm, which underpins any distance-vector routing protocols, rather than the protocols themselves. With

mathematical rigor, it simply states that with any given resources including given multipaths, given set of next hops and given workload such as x number of best-effort (BE) flows and y number of assured service (AS) flows, the family of control laws will drive the traffic allocation to track the time-varying link bandwidths and topology changes so that a given global utility function is maximized. Moreover, this paper is not concerned with the implementation issues, e.g., how to design the utility function, how to translate a control law associated with a given utility function into a window-based traffic control protocol, how to engineer the paths and allocate the link bandwidths, and so on. Interested readers are referred to [24] for the implementation of an end-to-end traffic control protocol based on the distributed controllers found in [11].

The remainder of this paper is organized as follows. In Section 2, we briefly discuss related work. In Section 3, we introduce the notation that is used throughout the paper and provide an exact problem statement. In Section 4, the family of optimal control laws is introduced. Examples of application of the proposed algorithm are presented in Section 5. Conclusions and directions for future research are given in Section 6. Finally, a sketch proof of the main results is provided in the Appendix.

2 Related Work

In this section, we focus exclusively on surveying literature relevant to the work presented in this paper, i.e., optimization-based, multipath-enabled, distributed traffic control schemes. But note that none of the existing solutions can deal with the case with time-varying link capacities.

The general approach taken to address the optimization-based, distributed traffic control problem is to formulate it, based on a fluid-flow model, as a nonlinear programming problem, taking into account of link bandwidth constraints. The aim is to find distributed traffic control laws with the property that by working independent of one another, they will drive the network to an operation point where a given utility function of flow rates is maximized. Since different flows share the link resources which are constrained, the key challenge in the design of distributed control laws is the fact that there is a high degree of interaction between different flows. This challenge has been tackled in three ways.

The first approach (e.g. [6,4]) is to incorporate link congestion costs into the overall utility function to convert a constrained problem into a non-constrained problem. The optimization problem was then solved using a gradient type of algorithm. Iterative algorithms were proposed where individual sources periodically adjust/balance their flow sending rates to multiple paths based on the congestion cost information fed back from

the (congested) links along each path. It is generally used for a network where each path is pinned, such as a label switched path [4], and, hence, not suitable for an overlay network where each physical path forming an overlay link is hidden.

The second approach (e.g., [8,9,7]) is to solve a relaxation of the original problem, by incorporating a price function into the overall utility function. Distributed control laws were found, which were proven to be locally stable in the presence of variable feedback delays. Working independently at a source, a control law adjusts/balances its flow sending rates to multiple paths based on periodic, cumulative price feedbacks from the destination node. Each component price is collected from the intermediate links along the forwarding path. The involvement of core nodes in conveying price information for each flow makes it difficult, if ever possible, to apply this scheme directly to an overlay environment where the underlying network topology is hidden.

The third approach is to solve the original problem directly (e.g., [23,10,11,15,16,20]). Using a duality model, an algorithm was provided in [23]. In [10,11,15,16], this problem was tackled by using a technique based on the Theory of Sliding Modes. Both end-to-end [10,11,16] and hop-by-hop [15] optimal control laws were found. Moreover, distributed multidomain optimal control laws are also found in [20]. These algorithms allow multipath forwarding and enable multiple Classes of Service (CoSes), and require minimum information feedback for control. For an overview of recent results in sliding mode theory and its applications see, for example, [3,25,26,13]

Due to the fact that the feasible set and the set of optimal solutions are time-varying, from mathematical viewpoint, the problem with time-varying link capacities is related to, but fundamentally different from the problems previously addressed in the literature, and it involves results on time varying optimization that are not available in the literature. To the best of our knowledge, it has not been shown that any of the algorithms available in the literature can track a time-varying optimal allocation. The novelty of this paper lies in the fact that it finds a new family of control laws to address the time-varying problem. It extends the preliminary work in [19] by including proofs and a more realistic example.

3 Problem Statement

3.1 Notation

Throughout this paper, traffic flows in a network are assumed to be described by a fluid flow model and the only resource considered is link bandwidth. Consider a network with time-varying link capacities, where flows with different service requirements are present. Divide these flows into types. Types are aggregates of flows that can

be treated as a unit, i.e., type is an aggregate of flows with the same ingress and egress nodes. Moreover, service requirements are to be applied to the aggregate, not individual flows. Note that one can have flow types with just one flow. Assume that each flow type has several paths available. The objective is to find an allocation of traffic that leads to the maximization of a given utility function subject to the network resource constraints and Class of Service (CoS) requirements.

More precisely, consider a network whose set of links is denoted by $\mathcal{L}$, with cardinality $\text{card}(\mathcal{L})$ equal to its number of links. Let $c_l(t)$ be the capacity of link $l \in \mathcal{L}$. Due to link resource uncertainty $c_l(t)$ is time-varying, and throughout this paper, for all $l \in \mathcal{L}$ and for all t , $c_l(t)$ is upper bounded, e.g., by the physical link capacity. Let $x_{i,j}$ be the data rate of flows type i taking path j , n_i be the number of available paths for type i , η be the number of types of flows. Let $\mathcal{L}_{i,j}$ be the set of links used by flows of type i taking path j . Also, let $\mathbf{x}_i \doteq [x_{i,1}, x_{i,2}, \dots, x_{i,n_i}]^T \in \mathcal{R}^{n_i}$ denote the vector containing the data rates allocated to the different paths by flows of type i , and $\mathbf{x} \doteq [\mathbf{x}_1^T, \mathbf{x}_2^T, \dots, \mathbf{x}_\eta^T]^T \in \mathcal{R}^n$ denote the vector containing all the data rates allocated to different types and respective paths, where $n = n_1 + n_2 + \dots + n_\eta$. A link $l \in \mathcal{L}$ is said to be congested if the aggregated data rate of the flows using the link is larger than or equal to $c_l(t)$. Given this, define $b_{i,j}(\mathbf{x})$ as the number of congested links along the j -th path of type i . Moreover, let $B_{i,j}$ be the number of links used by flows of type i taking path j .

$\mathcal{L}$	set of links of the network
η	number of types of flows
n_i	number of paths for flows of type i
$\mathcal{L}_{i,j}$	set of links used by flows of type i taking path j
$B_{i,j}$	cardinality of $\mathcal{L}_{i,j}$
$b_{i,j}$	number of congested links in $\mathcal{L}_{i,j}$
$x_{i,j}$	data rate of flows of type i taking path j
$\mathbf{x}_i$	$= [x_{i,1}, x_{i,2}, \dots, x_{i,n_i}]^T$
$\mathbf{x}$	$= [\mathbf{x}_1^T, \mathbf{x}_2^T, \dots, \mathbf{x}_\eta^T]^T$
$c_l(t)$	capacity of link l

Table 1
Notation Summary

3.2 Problem Statement

We assume that some of the types of calls have service requirements. Flows of type i , $i = 1, \dots, s_1$ are assumed to be Minimum Rate Guaranteed and Upper Bounded Rate Service (MRGUBS) category; i.e., there exist $0 < \theta_i(t) < \Theta_i(t)$ such that

$$\theta_i(t) \leq \sum_{j=1, \dots, n_i} x_{i,j} \leq \Theta_i(t).$$

Flows of type i , $i = s_1 + 1, \dots, \eta$ are assumed to be Best Effort (BE) category, type $\mathbf{x}_i$ of this class does not have any

further service requirements. Note that one may have other classes of service, like Assured Forwarding Service (AF) (a target rate to be guaranteed), Minimum Rate Guaranteed Service (MRGS), and Upper Bounded Rate Service (UBRS). These are straightforward modification of MRGUBS CoS by appropriately choosing $\theta_i(t)$ and $\Theta_i(t)$. Note that this paper handles time-varying CoS and link capacity. The time-varying CoS constraint and the time-varying link capacity constraint both are time-varying linear constraints, and are equivalent from the optimization viewpoint. It is assumed that $\theta_i(t)$, $\Theta_i(t)$ and $c_l(t)$ have bounded derivatives, i.e., there exists a finite constant $\dot{c}_{max}$ such that, for all t

$$\begin{aligned} \max_{t,i=1,\dots,s_1} |\dot{\theta}_i(t)| &< \dot{c}_{max}; & \max_{t,i=1,\dots,s_1} |\dot{\Theta}_i(t)| &< \dot{c}_{max} \\ \max_{t,l \in \mathcal{L}} |\dot{c}_l(t)| &< \dot{c}_{max} \end{aligned}$$

where $\dot{y}(t)$ denotes the derivative of $y(t)$ with respect to t . Assume also that all data rates are bounded, i.e., there exists a $\gamma \in \mathbf{R}$ such that $\mathbf{x}$ always belongs to the set

$$\mathcal{X} \doteq \{\mathbf{x} \in \mathbf{R}^n : x_{i,j} \leq \gamma; j = 1, 2, \dots, n_i; i = 1, 2, \dots, \eta\}$$

Let $U(\mathbf{x})$ be a given utility function representing the desired policy for assigning resources in the network

$$U(\mathbf{x}) = \sum_{i=1,\dots,\eta} U_i(\mathbf{x}_i)$$

where each $U_i(\cdot)$ is a differentiable strictly concave function in domain $\mathbf{R}^{n_i}$ and is increasing in each of its arguments. Moreover, it is assumed that the Hessian matrix H of $U(\mathbf{x})$ is invertible. The problem of optimizing utilization of the network resources can be formulated as:

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

subject to network capacity constraints, CoS constraints and non-negativity of the data rates; i.e.,

$$\sum_{i,j:l \in \mathcal{L}_{i,j}} x_{i,j} - c_l(t) \leq 0; \quad l \in \mathcal{L},$$

$$\theta_i(t) \leq \sum_{j=1,\dots,n_i} x_{i,j} \leq \Theta_i(t), i = 1, \dots, s_1$$

$$x_{i,j} \geq 0; \quad i = s_1 + 1, \dots, \eta, j = 1, \dots, n_i.$$

One can write this optimization problem as

$$\max_{\mathbf{x}} U(\mathbf{x}) \quad \text{subject to} \quad \mathbf{S}(\mathbf{x}, \mathbf{c}) \leq 0$$

where $\mathbf{S}(\mathbf{x}, \mathbf{c}) = \mathcal{G}\mathbf{x} - \mathbf{c}(t)$, $\mathcal{G}$ is the gradient of $\mathbf{S}(\mathbf{x})$ with respect to $\mathbf{x}$, and is an $L \times n$ constant matrix, i.e. there are L constraints. Clearly this is a time-varying parametric convex optimization problem since the $c_l(t)$

of link capacity constraint and $\theta_i(t)$ and $\Theta_i(t)$ of CoS constraint, are allowed to be time-varying. This problem is not a traditional optimization problem since the feasible set is time-varying. Correspondingly, the optimal solution of this problem, referred to as $\mathbf{x}_{opt}(t)$, is a function of t .

The objective of this paper is to develop decentralized rate adaptation laws that converge to and track the optimal data rates $\mathbf{x}_{opt}(t)$.

4 Optimal Control Laws

We propose the following distributed family of control laws: for MRGUBS flows:

$$\dot{\mathbf{x}}_i = -\zeta H_i^{-1} \left[\frac{dU_i}{d\mathbf{x}_i} - \alpha \mathbf{b}_i(\mathbf{x}) + \beta_i^m \mathbf{r}_i^m(\mathbf{x}_i) - \beta_i^M \mathbf{r}_i^M(\mathbf{x}_i) \right],$$

for BE flows:

$$\dot{\mathbf{x}}_i = -\zeta H_i^{-1} \left[\frac{dU_i}{d\mathbf{x}_i} - \alpha \mathbf{b}_i(\mathbf{x}) \right].$$

For both types of calls if $x_{i,j} = 0$ and $\dot{x}_{i,j} < 0$, then we force $\dot{x}_{i,j} = 0$. Here H_i^{-1} is the i -th block of the inverse of the Hessian matrix H (which is block diagonal), $\mathbf{b}_i(\mathbf{x})$, $\mathbf{r}_i^m(\mathbf{x}_i)$ and $\mathbf{r}_i^M(\mathbf{x}_i)$ are $n_i \times 1$ vectors:

$$\begin{aligned} \mathbf{b}_i(\mathbf{x}) &= [b_{i,1}(\mathbf{x}) \ b_{i,2}(\mathbf{x}) \ \dots \ b_{i,n_i}(\mathbf{x})]^T \\ \mathbf{r}_i^m(\mathbf{x}_i) &= \begin{cases} [0 \ 0 \ \dots \ 0]^T & \text{if } \sum_{j=1,\dots,n_i} x_{i,j} > \theta_i(t) \\ [1 \ 1 \ \dots \ 1]^T & \text{if } \sum_{j=1,\dots,n_i} x_{i,j} < \theta_i(t). \end{cases} \\ \mathbf{r}_i^M(\mathbf{x}_i) &= \begin{cases} [1 \ 1 \ \dots \ 1]^T & \text{if } \sum_{j=1,\dots,n_i} x_{i,j} > \Theta_i(t) \\ [0 \ 0 \ \dots \ 0]^T & \text{if } \sum_{j=1,\dots,n_i} x_{i,j} < \Theta_i(t). \end{cases} \end{aligned}$$

ζ , α , β_i^m and β_i^M are design parameters. Note that the control laws are *distributed* in the sense that flows are controlled independently and the only interactions among flows is through $b_{i,j}$, the number of congested links in a forwarding path j for flow i .

The terms in the control law are rather intuitive. The first term $\frac{dU_i}{d\mathbf{x}_i}$ is the one that forces the rates to increase if there are no other constraints active; i.e., no congestion and no binding CoS constraints. The term $\mathbf{r}_i^m$ is used to force the data rates to increase if they are below the lower bound of the CoS requirements. The term $\mathbf{r}_i^M$ is used to decrease data rates if the CoS upper bound is violated. Finally, $b_{i,j}$ is an indicator of path congestion

and forces decrease of the data rates that have a congested path. Since the utility function is monotone with respect to each of the entries, this behavior induces a sliding mode along the boundary of the set of allowable data rates. Hence, the sliding surfaces in this problem are determined by the link capacities. They are not a product of the controller design algorithm.

4.1 Additional notation and assumptions

Assume that at the optimal traffic allocation each congested link has at least one non-binding class of service or a BE flow with a nonzero data rate. Given any boundary point $\mathbf{x}_0$, one has an active constraint set $s_{\mathbf{x}_0}(\mathbf{x}) = [s_1(\mathbf{x}) \dots s_m(\mathbf{x})]^T$, where $s_i(\mathbf{x}_0) = 0$. Without loss of generality, assume that the rows of the gradient $G_{\mathbf{x}_0}$ are linear independent, where $G_{\mathbf{x}_0} = \frac{ds_{\mathbf{x}_0}(\mathbf{x})}{d\mathbf{x}}$. Define $Z = -H^{-1}$ and a subset of the set of boundary points:

$$X_{sm} = \{\mathbf{x}_0 : (G_{\mathbf{x}_0}Z(\mathbf{x}_0)G_{\mathbf{x}_0}^T)^{-1}G_{\mathbf{x}_0}Z(\mathbf{x}_0)\nabla U(\mathbf{x}_0) > 0\}$$

Assume that for all $\mathbf{x} \in X_{sm}$, each element of the vector $(G_{\mathbf{x}}Z(\mathbf{x})G_{\mathbf{x}}^T)^{-1}G_{\mathbf{x}}Z(\mathbf{x})\nabla U(\mathbf{x})$ is lower bounded:

$$\inf_{\mathbf{x} \in X_{sm}} ((G_{\mathbf{x}}Z(\mathbf{x})G_{\mathbf{x}}^T)^{-1}G_{\mathbf{x}}Z(\mathbf{x})\nabla U(\mathbf{x})) > \phi > 0 \quad (1)$$

Define $P > 0$, $M > 0$, such that

$$\begin{aligned} \max_{\mathbf{x} \text{ feasible}} \|(G_{\mathbf{x}}Z(\mathbf{x})G_{\mathbf{x}}^T)^{-1}\| &< P \\ \max_{\mathbf{x} \text{ feasible}} \|(G_{\mathbf{x}}Z(\mathbf{x})G_{\mathbf{x}}^T)\| &< M \end{aligned} \quad (2)$$

Remarks: 1) Condition (1) means that the unconstrained optimal solution is outside the feasible set. This condition usually holds true in computer networks.

2) Condition (2) guarantees that for $\mathbf{x} \in X_{sm}$, with appropriately chosen control parameters, the state variable $\mathbf{x}$ is able to “catch” and track the optimal solution.

We now show that the proposed control laws converge to the optimal traffic allocation.

Theorem 1 *If all the conditions in the Section 3 hold, there exists a constant $\alpha^*(\hat{c}_{max}) > 0$ such that if for all $\mathbf{x} \in \mathcal{X}$, $i = 1, 2, \dots, \eta$, $j = 1, 2, \dots, n_i$, the parameters ζ , α , β_i^m and β_i^M satisfy*

$$\zeta > \zeta^* = \frac{\hat{c}_{max}n^{1/2}P}{\phi}, \quad \alpha > \alpha^*, \beta_i^m, \beta_i^M = \beta \geq \alpha^* \max_{i,j} B_{i,j} \quad (3)$$

the family of control laws above converges to the optimal traffic allocation in the following sense:

$$\lim_{t \rightarrow \infty} \|\mathbf{x}(t) - \mathbf{x}_{opt}(t)\| = 0.$$

Fig. 1. Topology of the network

5 Example

The implementation of the proposed approach in a real network has to be performed in discrete time. In the simulations to follow, this is accomplished by obtaining a difference equation using the forward rule approximation

$$\mathbf{x}_{i,j}^d[(k+1)t_d] = \mathbf{x}_{i,j}^d[kt_d] + t_d \dot{\mathbf{x}}_{i,j}^d[(k+1)t_d]$$

for $k = 0, 1, 2, \dots$, where t_d is the integration step. The results in [5] imply that the control law above is “close” to the continuous time one as long as t_d is small.

The model of the network used in the simulation is based on the one in [11] and is shown in Fig. 1, where all the link bandwidths and delays, as well as source and destination nodes are shown. We assume that one has multiple paths and multiple CoSs. More precisely, there exists a total of $\eta = 8$ types of flows corresponding to 8 different combinations of source/destination nodes. The paths available for each pair of ingress/egress nodes are described in Table 2. Flows of type 3 and 5 are assumed to be of AF type, i.e. MRGUBS with $\theta_i = \Theta_i$, the target rates are $\theta_3 = \Theta_3 = 1Mb/s$, $\theta_5 = \Theta_5 = 0.8Mb/s$, flows of other types are assumed to be BE type. The capacity of the link between node b2 and node b7 is a random walk with bounded derivative. Moreover, to illustrate the algorithm behavior under link failure, at $t = 6s$, the link capacity of the link between node b2 and b8 becomes zero. The following utility function is assumed

$$U(\mathbf{x}) = \sum \log(x_{i,j} + 3), i = 1, \dots, 8, j = 1, \dots, n_i$$

As a first step, the control laws were tested in almost ideal conditions by setting network delays to 0 and sampling time to 0.1ms. It turns out that the conditions in Theorem 1 are too conservative and one can choose the control laws parameters in a less conservative way. The parameter ζ is chosen as 1. Also note that in this network, $\max_{i,j} B_{i,j} = 3$, so the other parameters are chosen as $\alpha = 0.5$, $\beta_i^m, \beta_i^M = \beta = 3\alpha$. The first figure in the left column of Fig. 2 gives the outcome of the time-varying link capacity between node b2 and b7. It can

Table 2
Paths Available for Each Type of Flows

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

be seen in the left column of Fig. 2 that, under these conditions, the utility function converges to an optimal utility function independently calculated by an Matlab optimization solver, and the network converges to an optimal operation point and keeps being optimal. Furthermore, the backbone links are fully utilized, and the AF requirements are satisfied.

The right column of Fig. 2 shows the network behavior with the delays given in Fig. 1 and sampling time $0.1ms$. The design parameters are chosen as $\alpha = 0.5$, $\beta_i^m, \beta_i^M = \beta = 3\alpha$, $\zeta = 1$. Here one observes the expected oscillation caused mainly by delays. The data rates, after a transient period, converge to a point close to the optimal point and the optimal traffic allocation is closely followed. Note that at $t = 6s$, the link capacity of the link between node b2 and b8 becomes 0, and the simulation shows that the proposed control laws can also handle link uncertainty/failure as good as our previous work [10], i.e., it converges to the new optimum in a short transient time.

6 Conclusions and Further Research Directions

In this paper, the network traffic control problem with time-varying link capacities and CoS requirements was formulated as a time-varying parametric convex optimization problem, and a family of distributed algorithms for optimal traffic allocation was developed, which addresses the case of utility functions with invertible Hessian. It was shown that the proposed control laws converge to optimal solution. Moreover, it follows the optimal traffic allocation if one starts from an optimal solution. The simulation results showed that the proposed algorithms lead to rather high performance in terms of resource utilization, as measured by utility function.

A Proof of Theorem 1

The algorithm in Section 4 can be written as:

$$\dot{\mathbf{x}} = \zeta Z \left(\frac{\partial U}{\partial \mathbf{x}} - \mathcal{G}^T \mathbf{u} \right)$$

Fig. 2. Simulation results

where $Z = -H^{-1}$ is block diagonal, and $\mathcal{G}$ is the gradient of the constraints, an $L \times n$ constant matrix. Since one only has linear constraints, $\mathcal{G}$ is constant. Moreover $\mathbf{u}$ is $L \times 1$ vector

$$u_i = \begin{cases} \alpha_i & \text{at } s_i(\mathbf{x}, t) > 0 \\ 0 & \text{at } s_i(\mathbf{x}, t) < 0 \end{cases}$$

for $i = 1, 2, \dots, L$, α_i satisfy following conditions:

$$\begin{aligned} &\text{for link capacity constraints: } \alpha_i \geq \alpha^* \\ &\text{for CoS constraints: } \alpha_i \geq \alpha^* \max_{i,j} B_{i,j} \\ &\text{for nonnegative rate constraints: } \alpha_i \geq 2\alpha^* \max_{i,j} B_{i,j} \end{aligned} \quad (\text{A.1})$$

We start the proof by showing that a sliding mode exists at $\mathbf{x} \in X_{sm}$. Since condition (3) holds (for simplicity, the index $\mathbf{x}$ is omitted), we have

$$\begin{aligned} &\inf_{\mathbf{x} \in X_{sm}} ((GZG^T)^{-1}(GZ\nabla U - \frac{1}{\zeta}\dot{\mathbf{c}}) \\ &> \phi - \frac{1}{\zeta^*} P n^{1/2} \frac{\zeta^* \phi}{n^{1/2} P} \geq 0 \end{aligned}$$

where $\zeta^* = \frac{\dot{c}_{max} n^{1/2} P}{\phi}$. This implies

$$(GZG^T)^{-1}(GZ\zeta\nabla U - \dot{c}) > 0 \quad (A.2)$$

If sliding mode occurs on active constraints $\bar{s} = 0$, the equivalent control satisfies [21]

$$\dot{\bar{s}} = 0 \Rightarrow \zeta \bar{G}Z \left(\frac{\partial U}{\partial \mathbf{x}} - \bar{G}^T \mathbf{u}_{eq} \right) - \dot{c} = 0$$

which leads to equivalent motion

$$\dot{\mathbf{x}}_{eq} = \zeta(Z - Z\bar{G}^T(\bar{G}Z\bar{G}^T)^{-1}\bar{G}Z)\nabla U + Z\bar{G}^T(\bar{G}Z\bar{G}^T)^{-1}\dot{c}$$

If $\mathbf{x}$ encounters a constraint s_g with gradient g at some $\mathbf{x} \in X_{sm}$, and g is not a linear combination of $\bar{G}^T$, by the condition (A.2)

$$\left(\begin{bmatrix} \bar{G} \\ g^T \end{bmatrix} Z \begin{bmatrix} \bar{G}^T & g \end{bmatrix} \right)^{-1} \left(\begin{bmatrix} \bar{G} \\ g^T \end{bmatrix} Z\zeta\nabla U - \begin{bmatrix} \dot{c} \\ \dot{c}_g \end{bmatrix} \right) > 0$$

which implies that

$$\begin{bmatrix} * & * \\ -C_2^{-1}g^T Z\bar{G}^T(\bar{G}Z\bar{G}^T)^{-1} & C_2^{-1} \end{bmatrix} \begin{bmatrix} \bar{G}Z\zeta\nabla U - \dot{c} \\ g^T Z\zeta\nabla U - \dot{c}_g \end{bmatrix} > 0$$

where $C_2 = g^T Zg - g^T Z\bar{G}^T(\bar{G}Z\bar{G}^T)^{-1}\bar{G}Zg > 0$.

$$g^T Z\zeta\nabla U - \dot{c}_g - g^T Z\bar{G}^T(\bar{G}Z\bar{G}^T)^{-1}(\bar{G}Z\zeta\nabla U - \dot{c}) > 0$$

If $s_g < 0$, $u_g = 0$, then

$$\dot{s}_g = g^T \zeta Z(\nabla U - \bar{G}^T(\bar{G}Z\bar{G}^T)^{-1}(\bar{G}Z\nabla U - \frac{1}{\zeta}\dot{c})) - \dot{c}_g > 0$$

Define

$$zgn(s) = \begin{cases} 0 & s \leq 0 \\ 1 & s > 0 \end{cases}$$

and $zgn(\mathbf{s}) = [zgn(s_1) \ zgn(s_2) \ \dots \ zgn(s_m)]^T$. The control law is:

$$\dot{\mathbf{x}} = \zeta Z(\nabla U - \bar{G}^T \mathbf{u}) = \zeta Z(\nabla U - \alpha \bar{G}^T zgn(\mathbf{s}))$$

the equivalent motion is:

$$\dot{\mathbf{x}}_{eq} = \zeta Z(\nabla U - \bar{G}^T \mathbf{u}_{eq}) = \zeta Z(\nabla U - \alpha \bar{G}^T zgn(\mathbf{s})_{eq})$$

also note that $zgn(\mathbf{s})_{eq,i} \in [0, 1]$. If $s_g > 0$, $u_g = \alpha$, and for α large enough

$$\begin{aligned} \dot{s}_g &= g^T \dot{\mathbf{x}}_{eq} - \dot{c}_g \\ &= \zeta g^T Z(\nabla U - \alpha \bar{G}^T \mathbf{u}_{eq}) - \alpha \zeta g^T Z\bar{G}^T zgn(\mathbf{s})_{\bar{G},eq} - \dot{c}_g \\ &< \zeta^* \min_{\mathbf{x}}(Z)(\max_{\mathbf{x}}(\nabla U) - \alpha) + \dot{c}_{max} \leq 0 \end{aligned}$$

and hence a sliding mode occurs where $\mathbf{x} \in X_{sm}$.

Original-problem and two-modified-problems: We now introduce two approximations of the proposed algorithm and show their convergence. The optimization problem given in Section 3 is referred to as Original-problem(OP):

$$\max U(\mathbf{x}) \text{ subject to } s_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, L \quad (A.3)$$

all L constraints $s_i(\mathbf{x}, t) = g_i^T \mathbf{x} - c_i(t) \leq 0$ are linear inequality constraints. The First-modified-problem(FMP) is defined:

$$\max \tilde{U}(\mathbf{x}, t) \quad (A.4)$$

where

$$\begin{aligned} \tilde{U}(\mathbf{x}, t) &= U(\mathbf{x}) - \sum_{i=1, \dots, m} u_i s_i(\mathbf{x}) = U(\mathbf{x}) - \sum f_i(s_i) \\ u_i &= \begin{cases} \alpha_i & \text{if } s_i > 0 \\ 0 & \text{if } s_i < 0 \end{cases} \end{aligned} \quad (A.5)$$

$\alpha_i, i = 1, \dots, L$ satisfies Condition (A.1). The objective function $\tilde{U}(\mathbf{x}, t)$ is continuous, strictly concave. It has unique time-varying optimal solution $\tilde{\mathbf{x}}^*(t)$. Moreover [21,11] $\tilde{\mathbf{x}}^*(t) = \mathbf{x}^*(t)$. The gradient of $\tilde{U}(\mathbf{x}, t)$ is:

$$\nabla \tilde{U}(\mathbf{x}, t) = \nabla U(\mathbf{x}) - \mathcal{G}^T \mathbf{u} \text{ where } \forall i, s_i(t) \neq 0$$

the i th column of $\mathcal{G}^T$ is g_i , the i th element of $\mathbf{u}$ is u_i . The Hessian matrix $\tilde{H}$ of $\tilde{U}(\mathbf{x}, t)$ is $\tilde{H}(t) = H$ where $\forall i, s_i(t) \neq 0$. Note that the $\nabla \tilde{U}(\mathbf{x}, t)$ and $\tilde{H}$ do not exist at $\mathbf{x}$, if for some i , $s_i(\mathbf{x}) = 0$. Let $\tilde{Z} = -\tilde{H}^{-1}$, propose the Slide-algorithm(SA) for FMP:

$$\dot{\mathbf{x}} = -\tilde{H}^{-1} \nabla \tilde{U}(\mathbf{x}, t) = Z[\nabla U(\mathbf{x}) - \mathcal{G}^T \mathbf{u}] \quad (A.6)$$

The Slide-algorithm is not defined where $s_i = 0$. Note that the Slide-algorithm is exactly the algorithm given in Section 4. The convergence of SA will be provided in next sections. Now we define the Second-modified-problem(SMP). Let

$$\begin{aligned} \bar{U}_\delta(\mathbf{x}, t) &= U(\mathbf{x}) - \sum v_{\delta,i}(s_i) s_i(\mathbf{x}) \\ &= U(\mathbf{x}) - \sum f_{\delta,i}(s_i) = U(\mathbf{x}) - f_\delta(\mathbf{x}, \mathbf{c}) \end{aligned}$$

where

$$v_{\delta,i}(s_i) = \begin{cases} \alpha_i & \text{if } s_i > \delta \\ v_i(\delta, s_i, \alpha_i) & \text{if } -\delta \leq s_i \leq \delta \\ 0 & \text{if } s_i < -\delta \end{cases} \quad (A.7)$$

such that for $f_{\delta,i}(s_i) = v_{\delta,i}(s_i)[g_i^T \mathbf{x} - c_i(t)]$, $\frac{df_{\delta,i}}{dt}, \frac{df_{\delta,i}}{d\mathbf{x}}, \frac{d^2 f_{\delta,i}}{d\mathbf{x}dt}$ exist. And for $f_\delta(\mathbf{x}, \mathbf{c}) = \sum f_{\delta,i}(s_i)$, $H_{f_\delta} \geq 0$ ex-

ists:

$$f_{\delta,i}(s_i) = \begin{cases} s_i \alpha_i & \text{if } s_i > \delta \\ \in (0, \delta \alpha_i) & \text{if } -\delta \leq s_i \leq \delta \\ 0 & \text{if } s_i < -\delta \end{cases}$$

$$\frac{df_{\delta,i}(s_i)}{ds_i} = \begin{cases} \alpha_i & \text{if } s_i > \delta \\ \in (0, \alpha_i) & \text{if } -\delta \leq s_i \leq \delta \\ 0 & \text{if } s_i < -\delta \end{cases}$$

$$\frac{d^2 f_{\delta,i}(s_i)}{ds_i^2} = \begin{cases} 0 & \text{if } s_i \notin [-\delta, \delta] \\ \geq 0 & \text{if } -\delta \leq s_i \leq \delta \end{cases}$$

The gradient of $\bar{U}_\delta(\mathbf{x}, t)$ is:

$$\nabla \bar{U}_\delta(\mathbf{x}, t) = \nabla U(\mathbf{x}) - \frac{df_\delta(\mathbf{x}, \mathbf{c})}{d\mathbf{x}} = \nabla U(\mathbf{x}) - \sum \frac{df_{\delta,i}(s_i)}{dx}$$

The Hessian matrix of $\bar{U}_\delta(\mathbf{x}, t)$, $\bar{H}_\delta = H - H_{f_\delta} < 0$. The Second-modified-problem(SMP) is defined:

$$\max \bar{U}_\delta(\mathbf{x}, t) \quad (\text{A.8})$$

Let $\bar{Z}_\delta = -\bar{H}_\delta^{-1}$, propose the Cont-algorithm(CA) for SMP:

$$\dot{\bar{\mathbf{x}}}_\delta = \bar{Z}_\delta [\nabla U(\mathbf{x}) - \frac{df_\delta(\mathbf{x}, \mathbf{c})}{d\mathbf{x}} - \frac{d^2 f_\delta(\mathbf{x}, \mathbf{c})}{d\mathbf{x}dt}] \quad (\text{A.9})$$

Note that $\lim_{\delta \rightarrow 0} \|\bar{U}_\delta(\mathbf{x}, t) - \bar{U}(\mathbf{x}, t)\| = 0$, moreover, due to $\bar{U}_\delta(\mathbf{x}, t)$ is continuous, strictly concave, it has unique time-varying optimal solution $\bar{\mathbf{x}}_\delta^*(t)$, one has $\lim_{\delta \rightarrow 0} \bar{\mathbf{x}}_\delta^*(t) = \bar{\mathbf{x}}^*(t)$.

Convergence of the Cont-algorithm: The Second-modified-problem is an unconstrained time-varying optimization problem. It has unique time-varying optimal solution $\bar{\mathbf{x}}_\delta^*(t)$. Consider the Lyapunov function:

$$L_\delta(t) = (1/2) \nabla \bar{U}_\delta^T(\mathbf{x}, t) \nabla \bar{U}_\delta(\mathbf{x}, t)$$

Since we have already proven that $\dot{\mathbf{x}}$ is fast enough to follow the time-varying boundary, the CA $\dot{\bar{\mathbf{x}}}_\delta$ and the SA $\dot{\mathbf{x}}$ are only different in the δ -layer, so the $\dot{\bar{\mathbf{x}}}_\delta$ is fast to follow the time-varying δ -layer. Then

$$\begin{aligned} dL_\delta(t)/dt &= \nabla \bar{U}_\delta^T(\mathbf{x}, t) \bar{H}_\delta \dot{\bar{\mathbf{x}}}_\delta + \nabla \bar{U}_\delta^T(\mathbf{x}, t) \frac{d\nabla \bar{U}_\delta}{dt} \\ &= -\nabla \bar{U}_\delta^T(\mathbf{x}, t) \nabla \bar{U}_\delta(\mathbf{x}, t) \leq 0 \end{aligned}$$

it is zero iff $\nabla \bar{U}_\delta(\mathbf{x}, t) = 0$, i.e., $\bar{\mathbf{x}}_\delta(t) = \bar{\mathbf{x}}_\delta^*(t)$. So the CA converges to the optimal solution of SMP.

Relationship between SA and CA: Assume $\mathbf{x}_\delta(t)$ is close to m constraints, i.e., $s_i = g_i^T \mathbf{x} - c_i \in (-\delta, \delta)$, $i =$

$1, \dots, m$. Let $G^T = [g_1 \dots g_m]$, without loss of generality, assume $m \leq n$, G has linear independent rows, and the first m columns of G are linear independent. Construct $n \times n$ matrix M as following:

$$M = \begin{bmatrix} G_0 & G_0 \Lambda \\ 0 & I \end{bmatrix}$$

where $G = [G_0 \ G_0 \Lambda]$, G_0 is $m \times m$ invertible matrix, Λ is $m \times (n-m)$, I is $(n-m) \times (n-m)$ identity matrix, so M is invertible. Do a linear transformation $\mathbf{y} = M\mathbf{x}$, and we use the index y to distinguish the terms in $\mathbf{x}$ -space and $\mathbf{y}$ -space, for example, U is the objective function in $\mathbf{x}$ -space, and $U_y = U(M^{-1}\mathbf{y})$ in $\mathbf{y}$ -space. The Slide-algorithm given in $\mathbf{y}$ -space:

$$\dot{\mathbf{y}} = Z_y [\nabla U_y(\mathbf{y}) - \mathcal{G}_y^T u]$$

the Cont-algorithm given in $\mathbf{y}$ -space:

$$\dot{\bar{\mathbf{y}}}_\delta = \bar{Z}_{y,\delta} [\nabla U_y(\mathbf{y}) - \frac{df_{y,\delta}(\mathbf{y}, \mathbf{c})}{d\mathbf{y}} - \frac{d^2 f_{y,\delta}(\mathbf{y}, \mathbf{c})}{d\mathbf{y}dt}]$$

Note that the following equation holds for any $\delta > 0$

$$\bar{H}_{y,\delta} \dot{\bar{\mathbf{y}}}_\delta + \frac{d\nabla \bar{U}_{y,\delta}}{dt} = -\nabla \bar{U}_{y,\delta}^T(\mathbf{y}, t)$$

$$H_y \dot{\mathbf{y}}_\delta = -\nabla U_y + \frac{df_{y,\delta}}{d\mathbf{y}}$$

also note that $\lim_{\delta \rightarrow 0} \frac{d^2 f_{y,\delta,i}}{dy_i^2} = \infty$, one has

$$\lim_{\delta \rightarrow 0} \left\| \left[\frac{d^2 f_{y,\delta,1}}{dy_1^2} (\dot{y}_1 - \dot{c}_1) \dots \frac{d^2 f_{y,\delta,m}}{dy_m^2} (\dot{y}_m - \dot{c}_m) \mathbf{0}^T \right] \right\| = 0$$

$$\dot{\bar{\mathbf{y}}}_\delta = Z_y [\nabla U_y(\mathbf{y}) + \frac{df_{y,\delta}}{d\mathbf{y}}] + \mathcal{O}(\delta)$$

that means in the δ -neighborhood of the set of points of discontinuity of the Slide-algorithm $\dot{\mathbf{y}}$, the Cont-algorithm $\dot{\bar{\mathbf{y}}}$ differs for almost all t by no more than $\xi(\delta)$ from some mean (with any nonnegative weights) values of the function $\dot{\mathbf{y}}$ in the δ -neighborhood of the point $(t, \bar{\mathbf{y}}_\delta(t))$, and $\lim_{\delta \rightarrow 0} \xi = 0$. More specifically,

$$\|\dot{\bar{\mathbf{y}}}_\delta(t) - \sum_{i=1,\dots,k} \eta_i \dot{\mathbf{y}}_i(t)\| \leq \xi \quad (\text{A.10})$$

$$\sum_{i=1,\dots,k} \eta_i = 1, \eta_i > 0, \|\mathbf{y}_i(t) - \bar{\mathbf{y}}_\delta(t)\| \leq \delta, i = 1, \dots, k$$

where η_i and $\mathbf{y}_i(t)$ may depend arbitrarily on $(t, \bar{\mathbf{y}}_\delta(t))$. The Condition (A.10) also holds in $\mathbf{x}$ -space.

Convergence of the Slide-algorithm: For any given $\epsilon_1 > 0$, one may construct a sequence $\{\epsilon_i\}_{i \geq 1}$, such that

$\epsilon_1 > \epsilon_2 > \dots > \epsilon_i > \epsilon_{i+1} > \dots$ Since $\bar{\mathbf{x}}_\delta^*(t) \rightarrow \tilde{\mathbf{x}}^*(t)$ as $\delta \rightarrow 0$, for $\epsilon_1 > 0$, there exists $\delta_1(\epsilon_1) > 0$, such that if $0 < \delta < \delta_1(\epsilon_1)$

$$\|\bar{\mathbf{x}}_\delta^*(t) - \tilde{\mathbf{x}}^*(t)\| < \epsilon_1, \quad \forall t \quad (\text{A.11})$$

Note that the Cont-algorithm with $\dot{\bar{\mathbf{x}}}_\delta$ converges to $\bar{\mathbf{x}}_\delta^*(t)$. Then, for any initial point $(t_0, \mathbf{x}_0)$ and $\delta < \delta_1(\epsilon_1)$, there exists $t_1(\epsilon_1) < \infty$, such that

$$\|\bar{\mathbf{x}}_\delta(t) - \bar{\mathbf{x}}_\delta^*(t)\| < \epsilon_1, \quad t > t_1(\epsilon_1) \quad (\text{A.12})$$

Moreover, for $\epsilon_1 > 0$, and any interval $t_0 \leq t \leq T_1$, where $t_1 < T_1 < \infty$, there exists $\xi_1(\epsilon_1) > 0$, $\delta_2(\epsilon_1) > 0$, such that if $\delta < \delta_2$, Condition(A.10) holds. Then with any same initial point $(t_0, \mathbf{x}_0)$ [5], $\bar{\mathbf{x}}_\delta(t)$, the trajectory of CA $\bar{\mathbf{x}}_\delta$, and $\mathbf{x}(t)$, the trajectory of SA $\dot{\mathbf{x}}(t)$, satisfy

$$\|\bar{\mathbf{x}}_\delta(t) - \mathbf{x}(t)\| < \epsilon_1, \quad t_0 \leq t \leq T_1 \quad (\text{A.13})$$

Then by the Conditions (A.11, A.12, A.13), one has:

$$\|\mathbf{x}(t) - \tilde{\mathbf{x}}^*(t)\| < 2\epsilon_1, \quad t_1(\epsilon_1) < t \leq T_1$$

Similarly to the above steps, for the sequence $\{\epsilon_i\}$, one may construct a time sequence $\{t_i, T_i\}_{i \geq 1}$:

$$t_1 < t_2 < T_1 < t_3 < T_2 < \dots < t_i < T_{i-1} < t_{i+1} < T_i < \dots$$

such that $\|\mathbf{x}(t) - \tilde{\mathbf{x}}^*(t)\| < 2\epsilon_i$, $t_i < t \leq T_i$ which implies $\|\mathbf{x}(t) - \tilde{\mathbf{x}}^*(t)\| < 2\epsilon_i$, $t_i < t \leq t_{i+1}$ so for $t > t_1$, $\|\mathbf{x}(t) - \tilde{\mathbf{x}}^*(t)\| < 2\epsilon_1$, $t > t_1$. Due to ϵ_1 can be arbitrarily small, $\mathbf{x}(t)$ given by the Slide-algorithm converges to $\tilde{\mathbf{x}}^*(t)$. Note that the Slide-algorithm is exactly the algorithm given in Section 4 and $\tilde{\mathbf{x}}^*(t) = \mathbf{x}^*(t)$, so the algorithm given in Section 4 converges to the time-varying optimal solution $\mathbf{x}^*(t)$.

References

- [1] <https://www.sdncentral.com/what-the-definition-of-software-defined-networking-sdn/>.
- [2] <https://www.sdncentral.com/whats-network-functions-virtualization-nfv>.
- [3] G. Bartolini, L. Fridman, A. Pisano, and E. Usai. Modern sliding mode control theory. *Lecture Notes in Control and Information Sciences*, 375(7):23–49, 2008.
- [4] A. Elwalid, C. Jin, S. Low, and I. Widjaja. Mate: Mpls adaptive traffic engineering. In *IEEE INFOCOM 2001*, 2001.
- [5] A. F. Filippov. *Differential Equation with Discontinuous Righthand Sides*. Kluwer Academic Publishers, 1988.
- [6] S. J. Golestani and S. Bhattacharyya. A class of end-to-end congestion control algorithms for the internet. In *IEEE International conference on Network Protocols (ICNP)*, pages 137–150, Oct 1998.
- [7] H. Han, S. Shakkottai, C. V. Hollot, R. Srikant, and D. Towsley. Overlay tcp for multi-path routing and congestion control. In *ENS-INRIA ARC-TCP Workshop*, Nov 2003.
- [8] F. Kelly, A. Maulloo, and D. Tan. Rate control in communication networks: Shadow prices, proportional fairness and stability. *Operations Research Society*, 1:237–252, Mar 1998.
- [9] F. Kelly and T. Voice. Stability of end-to-end algorithms for joint routing and rate control. *SIGCOMM Comput. Commun. Rev.*, 35(2):5–12, April 2005.
- [10] C. Lagoa and H. Che. Decentralized optimal traffic engineering for the internet. *ACM SIGCOMM Computer Communication Review*, Oct 2000.
- [11] C. M. Lagoa, H. Che, and B. A. Movsichoff. Adaptive control algorithms for decentralized optimal traffic engineering in the internet. *IEEE/ACM Transactions on Networking*, 12(3):415–428, Jun 2004.
- [12] C. Z. Li, H. Che, and S. Q. Li. Wireless channel capacity model for quality of service. *IEEE Transactions on Wireless Communications*, 6:356–366, Jan 2007.
- [13] F. Li, L. Wu, P. Shi, and C.-C. Lim. State estimation and sliding mode control for semi-markovian jump systems with mismatched uncertainties. *Automatica*, 51:385–393, 2015.
- [14] S. Low and D. Lapsley. Optimization flow control, i: Basic algorithm and convergence. *IEEE/ACM Transactions on Networking*, 7:861–874, Dec 1999.
- [15] B. Movsichoff, C. Lagoa, and H. Che. Decentralized optimal traffic engineering in connectionless networks. *IEEE Journal on Selected Areas in Communications*, 23:293–303, Feb 2005.
- [16] B. Movsichoff, C. Lagoa, and H. Che. End-to-end optimal algorithms for integrated qos, traffic engineering, and failure recovery. *IEEE/ACM Transactions on Networking*, Oct 2007.
- [17] R. Scroggins. Virtualization technology literature review. *GJCT*, pages 1–6, 2013.
- [18] R. Singh and R. Chauhan. A review paper: Voice over internet protocol. *International Journal of Enhanced Research in Management & Computer Applications*, 3(1):15–23, Jan 2014.
- [19] W. Su, C. Lagoa, and H. Che. A family of optimization-based traffic control laws for overlay networks. In *Decision and Control, 2007 46th IEEE Conference on*, pages 5710–5715. IEEE, 2007.
- [20] W. Su, C. Liu, C.M. Lagoa, H. Che, K. Xu, and Y. Cui. Integrated, distributed traffic control in multidomain networks. *IEEE Transactions on Control Systems Technology*, to be published. Early Access.
- [21] V. I. Utkin. *Sliding Modes in Control and Optimization*. Springer-Verlag, 1992.
- [22] C. Wang and B. Li. Peer-to-peer overlay networks: A survey. Technical report, 2003.
- [23] W.-H. Wang, M. Palaniswami, and S. H. Low. Optimal flow control and routing in multi-path networks. *Performance Evaluation*, 52(23):119 – 132, 2003.
- [24] L. Ye, Z. Wang, H. Che, and C. Lagoa. TERSE: A unified end-to-end traffic control mechanism to enable elastic, delay adaptive, and rate adaptive services. *Journal on Selected Areas in Communications*, 26:938–950, May 2011.
- [25] X. Yu and M. Ö. Efe. *Recent Advances in Sliding Modes: From Control to Intelligent Mechatronics*, volume 24. Springer, 2015.
- [26] G. Zhao, C. Zhao, and J. Cheng. Decoupled terminal sliding-mode control for a class of under-actuated mechanical systems with hybrid sliding surfaces. *International Journal of Innovative Computing, Information and Control*, 10(6):2011–2023, 2014.