On Optimal Multipath Rate-adaptive Routing

Gábor Rétvári, Gábor Németh
Dept. of Telecommunications and Media Informatics
Budapest University of Technology and Economics
Email: {retvari,nemethgab}@tmit.bme.hu

***Abstract*—A centralized rate-adaptive routing algorithm is presented that, in contrast to the distributed ones available in the literature, achieves provable *stability*, *optimalilty* with respect to optional linear or quadratic objective functions, and *feasibility* in that it can accommodate any admissible traffic matrix in the network without violating link capacities. We recast the routing problem in the framework of constrained optimal control theory to obtain optimal state feedback routing controllers, and we present simulations confirming that our routing controllers are viable in small- and middle-sized networks.**

***Index Terms*—traffic engineering, optimal control theory, model predictive control**

I. Introduction

A major challenge of Internet Traffic Engineering [1] is to provision forwarding paths in a network, so that the required Quality of Service is guaranteed to the users while the expensive network infrastructure is utilized cost-efficiently. Historically, forwarding paths were either not optimized at all, or they were optimized statically with respect to some measured and/or expected traffic matrix [2], [3]. Static routing, however, has become more and more counterproductive recently, as networks are beginning to face more dynamically changing traffic [4]. In response, various proposals have surfaced to reduce the significance of traffic matrices in intra-domain traffic engineering [5]–[9]. The most attractive approach is multipath rate-adaptive routing: distributed algorithms have been designed that can adapt dynamically to momentary traffic matrices and maximize users' aggregate utility, while also avoiding link oversubscription [10]–[16].

In many commercially operated networks, like transit, provider or enterprise networks, the task of traffic engineering is posed somewhat more sharply [9], [13], [15], [16]. Today's operational networks are beginning to see more and more inelastic multimedia traffic, and a growing share of customers requires the network to provide guaranteed flat rate, to abide to strict SLAs, and to deliver hard QoS. Unfortunately, satisfying these requirements is difficult with traditional *distributed* multipath rate-adaptive routing algorithms [17]. In this paper, therefore, we propose an alternative, *centralized* approach to intra-domain traffic engineering for ISP networks. Leveraging on the rich path diversity [18] and the broad range of routing information available in central network management software widely used for operating ISP networks, our scheme ensures provable stability, optimality with respect arbitrary linear or quadratic objective functions and adaptability to arbitrary user demands with strict QoS guarantees.

The first author was supported by the Janos Bolyai Fellowship of the Hungarian Academy of Sciences.

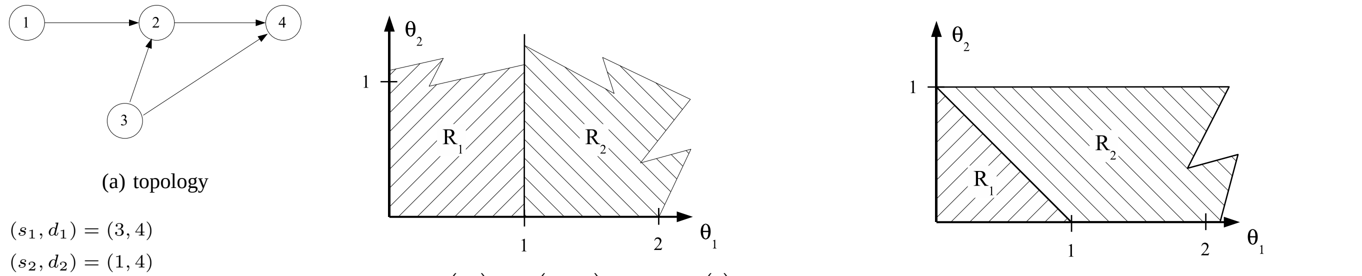
The main contribution of the paper is a formulation of the optimal rate adaptive routing problem in a control theoretic framework, which facilitates for building on firm theoretical foundations and a well-established numerical toolset. In Section II, we present a system model to describe the dynamic properties of networks and we design respective constrained optimal controllers. We evaluate our controllers in Section III and finally, Section IV concludes the paper.

II. Optimal routing control

The basic problem of rate-adaptive multipath routing can be formulated as follows. Given a network topology $G(V, E)$ consisting of n nodes and m edges; edge capacities $c = [c_{ij} : (i, j) \in E]$; and a set of source-destination pairs (or users) $(s_k, d_k) \in \mathcal{K}$, each one provisioned a set of paths $\mathcal{P}_k$ and each one presenting its momentary traffic demand θ_k to the network, the task is to adjust sending rates u_P along each path $p \in \mathcal{P}_k$ of each user $k \in \mathcal{K}$, so that no link becomes overloaded, that is, the aggregate flow sent to a link does not exceed its capacity. (See Table I for a list of notations.) Additionally, one may pose additional constraints on the routing algorithm, like complexity bounds, fairness in allocating network resources, or optimality with respect to some objective function that expresses the performance preferences of the network operator. In this paper, we deal with the latter case.

Consider the simple network depicted in Fig. 1. We give two routing controllers for this network in Fig. 1d and Fig. 1e. Our routing controllers are remarkably simple: they consist of a set of *regions* R_i and *affine routing functions* $S_i(\theta)$, so that the sending rate of users is set to $u = S_i(\theta)$ whenever the traffic matrix θ is in R_i, i.e., $\theta \in R_i$. (Note that affine functions take the form $f(x) = Ax + b$, where x is the vector of variables, A is a matrix of appropriate size and b is a constant column vector.) For instance, consider the controller in Fig. 1e and suppose that both user 1 and user 2 inserts 1 unit of traffic into the network. Then, since the traffic matrix $\theta = [1, 1]^T$ is in R_2, we apply routing function $S_2(\theta)$ corresponding to R_2 to obtain the rates $u_1 = 1$, $u_2 = 0$ and $u_3 = 1$. For $\theta = [2, 0]^T$ in the same region we get $u = [1, 1, 0]^T$.

This controller possesses some appealing properties. First, it is *feasible* in that the sending rate of the users is assigned so that no one link gets overprovisioned no matter what traffic matrix the users present to the network, as long as that traffic matrix is routable with *some* static routing (such traffic



(a) topology

$(s_1, d_1) = (3, 4)$
$(s_2, d_2) = (1, 4)$

(b) users

$p_1 = \{(3,4)\}$
$p_2 = \{(3,2),(2,4)\}$
$p_3 = \{(1,2),(2,4)\}$

(c) paths

$$\text{if } \theta_1 \le 1 \text{ then } \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad \text{(R1)}$$

$$\text{if } \theta_1 \ge 1 \text{ then } \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix} + \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix} \quad \text{(R2)}$$

(d) optimal routing controller (p_1 is cheaper than p_2)

$$\text{if } \theta_1 + \theta_2 \le 1 \text{ then } \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad \text{(R1)}$$

$$\text{if } \theta_1 + \theta_2 \ge 1 \text{ then } \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 0 & -1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix} + \begin{pmatrix} -1 \\ 1 \\ 0 \end{pmatrix} \quad \text{(R2)}$$

(e) optimal routing controller (p_2 is cheaper than p_1)

Figure 1: A sample network topology, source-destination pairs, a set of routes and two optimal routing controllers for the cases when path p_1 is preferred over p_2 and the other way around. Edge capacities all equal 1 unit.

matrices are called *admissible*). Second, it is *stable*, that is, to diminishing input it orders diminishing output. Third, it is *optimal*. The controllers in Fig. 1d and 1e were provisioned specifically to minimize the overall cost of the routing. In Fig. 1d, the cost per unit flow of path p_1 was smaller than that of p_2 (so this controller computes minimum hop-count paths), while in Fig.1e the cost was set the other way around. More complex objective criteria can be expressed as well.

In the rest of this section, we show that such stable, feasible and optimal routing controllers always exist, and each one takes the above form: a set of regions and the corresponding affine routing functions. First, we give a short introduction to optimal routing theory, then we discuss our network model and then we turn to controller design.

A. Optimal control theory

Suppose we are given a system characterized by the *state* x, *input* u and *output* y, whose evolution in time is governed by the linear system [19]:

$$\begin{aligned} x(t+1) =& Ax(t) + Bu(t) \\ y(t) =& Cx(t) + Du(t) \\ x(0) =& x_0 \end{aligned} \tag{S}$$

where A, B, C and D are constant matrices of proper size and $x(t)$, $u(t)$ and $y(t)$ are the values of the state(s), input(s) and output(s), respectively, at time t. Additionally, a set of constraints (C) can also be specified to which the system state and the input must obey at every time instance.

Suppose, in addition, that we are given an objective function, the *payoff* function, which prizes the evolution of the system in time as the function of the input and the initial state:

$$P(u(.), x(0)) = q_f^T x(N) + \sum_{t=0}^{N-1} (r^T u(t) + q^T x(t)) \tag{P}$$

Table I: Notations

$G(V,E)$	a directed graph, with the set of nodes V ($\|V\| = n$) and the set of directed edges E ($\|E\| = m$)
c	the column m-vector of edge capacities
(s_k, d_k)	the set of source-destination pairs (or users) for $k \in \mathcal{K} = \{1,\dots,K\}$
$\mathcal{P}_k$	the set of $s_k \to d_k$ paths assigned to some $k \in \mathcal{K}$
P_k	an $m \times \|\mathcal{P}_k\|$ matrix. The column corresponding to path $p \in \mathcal{P}_k$ holds the path-arc incidence vector of p
u_p	scalar, describing the traffic routed at path p
u_k	a column-vector, whose components are $u_p : p \in \mathcal{P}_k$ for some $k \in \mathcal{K}$ (whether we mean u_k or u_p will always be clear from the context)
u	a column vector representing a particular choice of u_ps (a "routing")
θ_k	the demand/throughput of some user $k \in \mathcal{K}$
θ	a column K-vector representing a particular combination of throughputs (a "traffic matrix")

where N is the *control horizon* and q_f^T, r^T and q^T (all constant row-vectors of proper size) are the *terminal cost* and *running payoffs*, respectively. For completeness, we note that the theory allows for linear [20], [21] as well as quadratic payoffs [22]. Finally, assume we are given a *terminal set* T which we would like our system to eventually settle down in.

Now, the basic problem of optimal control theory is to design a controller, which adjusts the input u so that the system (S), starting from some initial state $x(0)$, is regulated along an optimal trajectory to T obeying the constraints (C), as measured by the payoff function (P). In this setting, u is called an *optimal control*.

B. The Zero-buffer path-flow model

In our model, system state is the amount of traffic waiting to be served at the source nodes, output is simply this same state (which therefore we shall omit henceforth), and the control is the flow placed at individual paths of the users. Formally, let $x(t)$ be a column K-vector, whose kth component describes the amount of data to be delivered from s_k, and let u_p describe the flow routed at path $p \in \mathcal{P}_k, k \in \mathcal{K}$. Then, the *Zero-buffer path-flow (ZBPF) model* is characterized by the dynamics:

$$x_k(t+1) = x_k(t) - \tau \sum_{p \in \mathcal{P}_k} u_p(t) \qquad \forall k \in \mathcal{K} \qquad \text{(D)}$$

$$x_k(0) = \theta_k \qquad \forall k \in \mathcal{K} \qquad \text{(I)}$$

This model does not allow for buffering at intermediate nodes (hence the name). The state $x(t)$ integrates the data fed by the users at the source nodes into the network in time, minus the sum of flows carried away along the individual paths of the user within the discrete time step τ. In other words, $x_k(t)$ models the amount of traffic accumulated in the input buffer of source-destination pair k at time t, and the initial state $x_k(0)$ is simply the demand of user k represented by the data in the input buffer at the zeroth time instance. For the sake of simplicity, we shall assume henceforth that the discrete time step is 1 sec and θ is scaled accordingly, and so we shall omit τ in the equations. Additionally, we assume that no further traffic arrives within the time frame τN.

The control must respect certain operational constraints in assigning rates to the users. First, edge capacities may not be violated:

$$\sum_{k \in \mathcal{K}} P_k u_k(t) \leq c \ , \qquad \text{(C1)}$$

rates are non-negative:

$$u_k(t) \geq 0 \qquad \forall k \in \mathcal{K} \ , \qquad \text{(C2)}$$

and the controller can not remove more data from the source nodes than it is available there:

$$x_k(t) \geq 0 \qquad \forall k \in \mathcal{K} \ . \qquad \text{(C3)}$$

C. Optimal controller design

Next, we design an optimal controller for the network model described above. The controller's job will be to remove as much data from the input buffers as possible. In other words, the controller regulates the states towards the origin. The first step of controller design is to convince ourselves that our system is well-behaved and so a suitable controller exists. We say that a system is *controllable*, if there exists a control that drives it from any optional initial state into the origin in finite time.

Theorem 1: A capacitated network is controllable under the ZBPF model, according to the dynamics (D) and satisfying conditions (C1-3), if and only if the path set $\mathcal{P}$ contains at least one path of nonzero capacity for each user.

A way to prove this statement would be to a design a trivial controller, which puts some small, nonzero flow to the usable paths in each step, gradually consuming all the initial data at the source node.

Theorem 2: For any controllable capacitated network, any payoff (P), any $N > 0$, and any admissible initial state θ, there exists a controller that, starting from θ, regulates the ZBPF system to the origin in N steps, according to the dynamics (D) and satisfying conditions (C1-3), while optimizing (P). The control action $u(.)$ is a continuous and piece-wise affine (PWA) function of θ:

$$u(\theta) = F_i \theta + g_i \qquad \text{if } \theta \in R_i, i = 1, \ldots, r$$

with R_is being closed polyhedral sets in $\mathbb{R}^K$. Additionally, the set of initial states for which the controller converges in N step (the N-step feasible set) is convex.

Proof: Consider the linear program:

$$\max\ q_f^T x(N) + \sum_{t=0}^{N-1} r^T u(t) + q^T x(t) \qquad \text{(P)}$$

$$\text{s.t. } x_k(0) = \theta_k \qquad \forall k \in \mathcal{K} \qquad \text{(I)}$$

$$x_k(t+1) = x_k(t) - \sum_{p \in \mathcal{P}_k} u_p(t) \qquad \text{(D)}$$
$$\forall k \in \mathcal{K}, \forall t \in \{0, \ldots, N-1\}$$

$$x_k(N) = 0 \qquad \forall k \in \mathcal{K} \qquad \text{(T)}$$

$$\sum_{k \in \mathcal{K}} P_k u_k(t) \leq c \qquad \forall t \in \{0, \ldots, N-1\} \qquad \text{(C1)}$$

$$u(t) \geq 0 \qquad \forall t \in \{0, \ldots, N-1\} \qquad \text{(C2)}$$

$$x(t) \geq 0 \qquad \forall t \in \{1, \ldots, N\} \qquad \text{(C3)}$$

and solve it as a multi-parametric linear program as the function of θ. The claims of the theorem can then be proved based on the results in [23]. ■

This controller is called the *N-step Zero-buffer path-flow routing controller (N-RC)*. The above is an application of *model predictive control*: we prognosticate the evolution of the network using the ZBPF dynamics (D) within the control horizon N, starting from the initial state (I) and arriving in the Nth step into the terminal set (T), and we compute a sequence of control actions, $u(0)$, $u(1)$, ..., $u(N-1)$ that ensures constraint satisfaction (C1-3) and optimizes the payoff (P).

One way to obtain the control is to solve the above linear program on-site in each time step feeding the measured θ as input, but this would amount to online linear programming. Instead, we solve it as a multi-parametric linear program (mp-LP), which requires the solution of an LP as the function of some parameter. Here, the variables of the LP are the optimal evolution of the system and the control u, and the parameter is the initial state θ. The resultant series of control actions can then be stored in the controller and applied sequentially, $u(0)$ in the zeroth step, $u(1)$ in the first and so on, to regulate the system along the calculated trajectory. This scheme is called *open loop control*.

Unfortunately, if the system is subject to noise or the model is not perfectly precise, then the predicted state and the real state can diverge with time, which might deteriorate efficiency and even lead to instability. Therefore, it is a common practice to take only the first control $u(0)$ and apply it repeatedly to the system, yielding a closed-loop, state feedback controller. This scheme is called *receding horizon control*. For arbitrary

N the receding horizon control might not be stable (let alone optimal), but for $N = 1$ it is both stable and optimal.

Corollary 1: Given a controllable capacitated network, a user defined payoff and *any* admissible traffic matrix θ, 1-RC routes θ in the network without violating link capacities and optimizing the payoff.

A direct proof for this claim also appears in an accompanying paper of the authors [24].

In practice, routing control works as follows. One first needs to model the network using the ZBPF dynamics and solve the mp-LP of Theorem 2. For moderate sized networks, this is clearly viable thanks to recent advances in geometric multi-parametric programming algorithms [23], [22]. The result is a set of regions R_i and the corresponding (F_i, g_i) parameters that define the controller. These steps, while computationally rather involving due to the need to solve the mp-LP, can all be performed offline. Then, R_i and (F_i, g_i) are all downloaded to the on-site controller. This controller periodically scans the network, identifies the actual user demands θ, searches for a control region R_i containing θ and readjusts routing according to $u = F_i\theta + g_i$. This way, online activity reduces to solving a series of polyhedron inclusion problems and matrix multiplication.

Invoking an 1-RC for traffic engineering brings numerous benefits. First, it eliminates service disruptions due to link overprovisioning. Second, it is provably stable. Third, it is provably optimal, over *any* user defined payoff function. Note that these are hard guarantees that fulfill at any point in time. What is more, the control action is continuous, both inside and on the boundary of the control regions, which ensures that routing does not exhibit huge spikes and the controller transitions smoothly from one region to the other.

The applicability and complexity of our routing controllers fundamentally depends on the number of control regions I the controller comprises. This is because in every step the controller needs to solve $O(I)$ polyhedron inclusion problems in order to find $R_i : \theta \in R_i$. Additionally, the storage requirement also scales with $O(I)$. Unfortunately, constrained optimal control theory does not set a strict polynomial upper-bound on this number. Therefore, below we turn our attention to heuristic methods for reducing the online complexity of routing control.

A straightforward way to reduce controller complexity would be to increase the control horizon N. Recall that N connotes the time the controller is allowed to spend driving the system into the origin. Thus, the larger the control horizon the slower the controller. This is expected to yield larger control regions and hence to decrease complexity. It must be noted, however, that the ZBPF model, in its current form, assumes that no further traffic arrives into the input buffers within the time frame τN. When this assumption is violated, the system state as predicted by the controller and the real system state diverge[1]. This usually does not pose enormous problems in receding horizon control, since we consider only the first control action that is based on exact system state (the initial state).

[1]This problem only emerges for N-step controllers where $N > 1$. For $N = 1$, no prediction is needed as the system is cleared out in a single step.

Setting the control horizon not only affects controller complexity, but it also has profound impact on the set of states to which the controller orders control action. In general, the set of states from which a system can be controlled into the origin in N steps is called the *N-step feasible set*, it monotonically increases with increasing N and it precisely coincides with the set of states to which an N-RC orders control action. An 1-RC covers only the set of admissible traffic matrices, and increasing the control horizon has the useful consequence of broadening the range of traffic matrices our controller can handle.

Another way to exploit the performance-complexity trade-off is to weaken the terminal set constraints. Our controllers, defined in Theorem 2, explicitly require that the system settles in the origin. In practice, however, usually it is enough to merely move the system closer to the origin, and this can be achieved by dropping constraints (T) and penalizing divergence from the origin by setting a nonzero terminal cost q_f^T in the objective function. Note that a controller with weak terminal set conditions orders control to any non-zero traffic matrix.

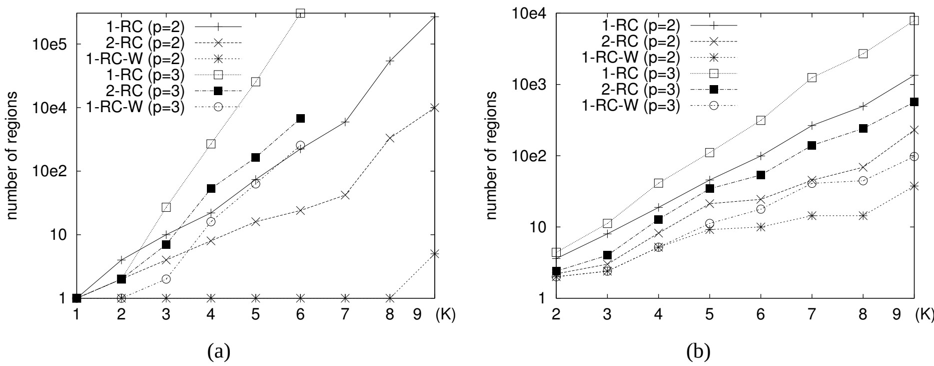
III. Performance evaluation

In order to see how useful these ideas prove in practice, we implemented routing control and we conducted several rounds of performance evaluation. The controller complexity was measured by the number of control regions. To avoid the bias caused by the different feasible sets, we explicitly constrained the state space of the controllers to the set of admissible traffic matrices.

Evaluating the performance, however, is slightly more difficult than measuring complexity. In theory, the largest state set into which the controller is guaranteed to drive the system from any admissible initial state in N steps (the *N-step reachable set*) would be a perfect measure of performance: the smaller the reachable set around the origin the better the performance. However, the reachable set is usually a convex polyhedron, whose size is hard to characterize with a single scalar. Using the polyhedron's volume would be straightforward, but that would make it impossible to rightfully compare controllers defined over different dimensions. Therefore, we estimate the size of a polyhedron by the edge length of its bounding hyper-cube, and we measure the performance of a controller by the ratio of the initial state-space size and the reachable-set size in terms of this quantity. Consider the following *N-step goodness* definition:

$$g_N = 1 - \frac{b(R_N)}{b(T)} \ , \tag{1}$$

where R_N is the N-step reachable set, T is the set of admissible traffic matrices and $b(X)$ denotes the edge length of the bounding hyper-cube of polytope X. This goodness definition connotes the ratio of demand a routing controller is



p	K	2-RC g_1	2-RC g_2	1-RC-W g_1	1-RC-W g_2
2	2	0.73	0.93	0.66	0.89
	3	0.71	0.92	0.66	0.89
	4	0.71	0.92	0.66	0.89
	5	0.70	0.92	0.66	0.89
	6	0.70	0.92	0.66	0.89
	7	0.70	0.92	0.66	0.89
	8	0.68	0.92	0.66	0.89
	9	0.64	0.91	0.60	0.86
3	2	0.78	0.95	0.75	0.93
	3	0.78	0.95	0.73	0.93
	4	0.75	0.95	0.71	0.92
	5	0.74	0.94	0.71	0.92
	6	0.65	0.93	0.71	0.92

(c)

Figure 2: Average number of control regions for (a) the NSF topology and (b) the AS 3257 (Tiscali, Europe) Rocketfuel topology as the function of the number of users K and the number of paths per user p. Table (c) gives routing controller performance in terms of (1) for the NSF topology.

guaranteed to clear out from the source nodes in N steps. Correspondingly, the goodness of an 1-RC equals 1, the goodness of the zero controller (which does not route any traffic at all) is 0, and the closer to 1 the goodness the better the performance.

The controllers we evaluated were ZBPF routing controllers with control horizon set to 1 (1-RC), 2 (2-RC), and the 1-step controller over weak terminal set conditions (1-RC-W). The objective function was set to minimize the cost of routing in terms of statically assigned path costs. We implemented the ZBPF model and the corresponding routing controllers in the excellent Multi-Parametric Toolbox (`MPT`, [25]) for Matlab[2].

In the first round of our evaluations, we used the NSFNET Phase II topology [26] and we observed how routing control's performance and complexity changes in increasingly complex scenarios. The number of users K was varied between 1 and 9, source-destination pairs were chosen according to the random bimodal distribution similarly to [27], and maximally node-disjoint paths were computed for the users. The complexity of the resultant controllers is given in Fig. 2a for 2 and 3 paths per user, and the goodness is indicated in Fig. 2c. Note that we omitted the column corresponding to 1-RC, as the goodness is always 1.

In the second round of the numerical studies, we conducted extensive evaluations on ISP data maps from the Rocketfuel dataset [28]. We used the same method as in [27] to obtain approximate POP-level topologies: we collapsed the topologies so that nodes correspond to cities, we eliminated leaf-nodes and we set link capacities inversely proportional to the link weights. Again, the number of users was increased from 1 to 9, source-destination pairs were chosen according to the bimodal distribution and paths were provisioned maximally node-disjoint. Six evaluation runs were conducted on different randomly chosen network samples and the results were averaged. Here, we only include the results for AS 3257 (Tiscali, Europe): Fig. 2b gives the complexity for 2 and 3 paths per user, respectively. The goodness was 65-75% in the first period and 85-90% in the second (results are not included here due to space limits). We obtained similar results on other Rocketfuel topologies.

The main observations are as follows: First, for networks serving only a couple of users, optimal routing control (1-RC) is clearly a viable option. However, complexity seems to increase exponentially with the number of users, and it becomes intractable when the total number of paths in the system surpasses about 30. Increasing the control horizon, i.e., 2-RC, looks more promising in this regard as it can clear out about 65-75% of any admissible traffic matrix in a single step, and more than 90% in the second, with complexity growing significantly slower. Finally, we found that weakening the terminal set conditions seems to be the most effective way of complexity reduction. Though, the performance penalty is larger due to the additive term in the objective function. A potential reason for this might be an inadequate setting of the terminal payoff in our evaluations.

Of course, we could not give a complete coverage on the extensive topic of controller complexity reduction in this paper. Many solutions exist that offer optimal control but significantly fewer control regions, at the price of increased offline computational cost [29]–[31]. Further complexity reduction could be achieved by weakening the capacity constraints (C1). For the sake of brevity, we completely ignored stability of reduced complexity routing controllers (although, in practice, we found that they are always stable). Stability is, however, easy to guarantee using additive, Lyapunov-type penalty in the objective function [32], [33].

IV. Conclusions

"One major challenge of Internet traffic engineering is the realization of automated control capabilities that adapt quickly and cost effectively to significant changes in a network's state,

[2]The code of the routing controllers, the network topologies and the path sets used throughout the paper are available at http://qosip.tmit.bme.hu/~retvari.

while still maintaining stability" (Overview and Principles of Internet Traffic Engineering, RFC 3272, page 6, [1]). This challenge has been open for a couple of years now. This paper is aimed at demonstrating that quick but stable adaptation of routing to changing operational conditions is certainly possible.

The majority of existing work on traffic engineering is based on a distributed scheme that tends to scale well with the increase of the user population. The present work explores the "centralized end" of the distributed-centralized spectrum. On this centralized end we find optimizability, strict feasibility and hard QoS guarantees, essential ingredients in commercially operated provider networks a distributed scheme hardly provides. On the other hand, centralized routing control seems to scale poorly. We demonstrated that the judicious increase of the control horizon and the weakening of terminal set constraints are effective ways to reduce the number of control regions without sacrificing performance, thus positioning routing control higher up scalability-wise. The solution to the complexity issues might be a hybrid centralized-distributed algorithm, as we demonstrate in an accompanying paper [24]

REFERENCES

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

[2] D. G. Cantor and M. Gerla, "Optimal routing in a packet-switched computer network," *IEEE Transactions on Computer*, vol. 23, no. 10, pp. 1062–1069, 1974.

[3] B. Fortz, J. Rexford, and M. Thorup, "Traffic engineering with traditional IP routing protocols," *IEEE Communications Magazine*, vol. 40, pp. 118–124, Oct 2002.

[4] S. Gunnar, M. Johansson, and T. Telkamp, "Traffic matrix estimation on a large IP backbone: a comparison on real data," in *IMC '04: Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, pp. 149–160, 2004.

[5] M. Roughan, M. Thorup, and Y. Zhang, "Traffic engineering with estimated traffic matrices," in *IMC '03: Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, pp. 248–258, 2003.

[6] C. Zhang, Y. Liu, W. Gong, J. Moll, and R. D. Towsley, "On optimal routing with multiple traffic matrices," in *INFOCOM 2005*, vol. 1, pp. 607–618, 2005.

[7] B. Fortz and M. Thorup, "Optimizing OSPF/IS-IS weights in a changing world," *IEEE Journal of Selected Areas in Communications*, vol. 20, pp. 756–767, May 2002.

[8] S. Suri, M. Waldvogel, and P. R. Warkhede, "Profile-based routing: a new framework for MPLS traffic engineering," in *Quality of Future Internet Services* (F. Boavida, ed.), vol. 2156 of *LNCS*, Springer, 2001.

[9] H. Wang, H. Xie, L. Qiu, Y. R. Yang, Y. Zhang, and A. Greenberg, "COPE: traffic engineering in dynamic networks," *SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 4, pp. 99–110, 2006.

[10] M. Chiang, S. Low, A. Calderbank, and J. Doyle, "Layering as optimization decomposition: A mathematical theory of network architectures," *Proceedings of the IEEE*, vol. 95, pp. 255–312, Jan 2007.

[11] F. Kelly and T. Voice, "Stability of end-to-end algorithms for joint routing and rate control," *SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 2, pp. 5–12, 2005.

[12] P. Key, J. J. T. Aveue, L. Massoulié, and J. J. T. Aveue, "Combining multipath routing and congestion control for robustness," in *in Conference on Information Sciences and Systems*, 2006.

[13] J. He, M. Bresler, M. Chiang, and J. Rexford, "Towards robust multilayer traffic engineering: Optimization of congestion control and routing," *Selected Areas in Communications, IEEE Journal on*, vol. 25, pp. 868–880, June 2007.

[14] C. M. Lagoa, H. Che, and B. A. Movsichoff, "Adaptive control algorithms for decentralized optimal traffic engineering in the internet," *IEEE/ACM Trans. Netw.*, vol. 12, no. 3, pp. 415–428, 2004.

[15] S. Kandula, D. Katabi, B. Davie, and A. Charny, "Walking the Tightrope: Responsive Yet Stable Traffic Engineering," in *ACM SIGCOMM'05*, August 2005.

[16] S. Fischer, N. Kammenhuber, and A. Feldmann, "REPLEX: dynamic traffic engineering based on wardrop routing policies," in *Proceedings of CoNEXT'06*, pp. 1–12, 2006.

[17] J.-W. Lee, R. R. Mazumdar, and N. B. Shroff, "Non-convex optimization and rate control for multi-class services in the Internet," *IEEE/ACM Trans. Netw.*, vol. 13, no. 4, pp. 827–840, 2005.

[18] R. Teixeira, K. Marzullo, S. Savage, and G. M. Voelker, "In search of path diversity in ISP networks," in *in Proceedings of the ACM SIGCOMM Internet Measurement Conference*, pp. 313–318, ACM Press, 2003.

[19] E. Sontag, *Mathematical Control Theory: Deterministic Finite Dimensional Systems*, vol. 6 of *Texts in Applied Mathematics*. Springer, 1998. available online: http://www.math.rutgers.edu/~sontag/FTP_DIR/sontag_mathematical_control_theory_springer98.pdf.

[20] A. Bemporad, F. Borrelli, and M. Morari, "Explicit solution of LP-based model predictive control," in *Proceedings of the 39th IEEE Conference on Decision and Control*, December 2000.

[21] A. Bemporad, F. Borrelli, and M. Morari, "Model predictive control based on linear programming – the explicit solution," *IEEE Transactions on Automatic Control*, vol. 47, no. 12, pp. 1974–1985, 2002.

[22] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, pp. 3–20, January 2002.

[23] F. Borrelli, A. Bemporad, and M. Morari, "Geometric algorithm for multiparametric linear programming," *Journal of Optimization Theory and Applications*, vol. 118, pp. 515–540, September 2003.

[24] G. Rétvári and G. Németh, "Demand-oblivious routing: distributed vs. centralized approaches," in *INFOCOM 2010*, March 2010.

[25] M. Kvasnica, P. Grieder, and M. Baotić, "Multi-Parametric Toolbox (MPT)," 2004. available online: http://control.ee.ethz.ch/~mpt/.

[26] B. Chinoy and H. W. Braun, "The national science foundation network." Tech. Rep., CAIDA, available online: http://www.caida.org/outreach/papers/1992/nsfn/nsfnet-t1-technology.pdf, Sep 1992.

[27] D. Applegate and E. Cohen, "Making intra-domain routing robust to changing and uncertain traffic demands: understanding fundamental tradeoffs," in *Proceedings of SIGCOMM '03*, pp. 313–324, 2003.

[28] R. Mahajan, N. Spring, D. Wetherall, and T. Anderson, "Inferring link weights using end-to-end measurements," in *IMW '02: Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*, pp. 231–236, 2002.

[29] P. Grieder and M. Morari, "Complexity reduction of receding horizon control," *Proceedings. 42nd IEEE Conference on Decision and Control*, vol. 3, December 2003.

[30] P. Grieder, P. Parrilo, and M. Morari, "Robust receding horizon control – analysis & synthesis," in *Proceedings. 42nd IEEE Conference on Decision and Control*, vol. 1, pp. 941–946, December 2003.

[31] P. Grieder, M. Kvasnica, M. Baotic, and M. Morari, "Low complexity control of piecewise affine systems with stability guarantee," in *Proceedings of the American Control Conference*, vol. 2, June 2004.

[32] D. Mayne, J. Rawlings, C. Rao, and P. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, pp. 789–814, June 2000.

[33] J. Rawlings and K. Muske, "The stability of constrained receding horizon control," *IEEE Transactions on Automatic Control*, vol. 38, no. 10, pp. 1512–1516, 1993.