Cooperative Content Distribution and Traffic Engineering in an ISP Network

Joe Wenjie Jiang[†], Rui Zhang-Shen[†], Jennifer Rexford[†], Mung Chiang[*]
[†]Department of Computer Science, and [*]Department of Electrical Engineering
Princeton University
{wenjiej, rz, jrex, chiangm}@princeton.edu

ABSTRACT

Traditionally, Internet Service Providers (ISPs) make profit by providing Internet connectivity, while content providers (CPs) play the more lucrative role of delivering content to users. As network connectivity is increasingly a commodity, ISPs have a strong incentive to offer content to their subscribers by deploying their own content distribution infrastructure. Providing content services in a provider network presents new opportunities for coordination between *traffic engineering* (to select efficient routes for the traffic) and *server selection* (to match servers with subscribers). In this work, we develop a mathematical framework that considers three models with an increasing amount of cooperation between the ISP and the CP. We both analytically and numerically study the stability and optimality conditions for these models. We show that separating server selection and traffic engineering leads to sub-optimal equilibria, even when the CP is given accurate and timely information about the ISP's network in a partial cooperation. More surprisingly, extra visibility results in a *less* efficient outcome and such performance degradation can be unbounded. Leveraging ideas from cooperative game theory, we propose an architecture based on the concept of *Nash bargaining solution* that significantly improves the fairness and efficiency of the joint system. Simulations on realistic backbone topologies are performed to quantify the performance differences between our models. We show that the joint design significantly improves the performance metrics of both the ISP and the CP, under a wide range of traffic conditions. This study is a step toward a systematic understanding of the interactions between those who provide and operate networks and those who generate and distribute content.

1. INTRODUCTION

Internet Service Providers (ISPs) and content providers (CPs) are traditionally independent entities. ISPs only provide connectivity, or the bandwidth "pipes" to transport content. As in most transportation businesses, connectivity and bandwidth are becoming commodities and ISPs find their profit margin shrinking [1]. At the same time, content providers generate revenue by utilizing existing connectivity to deliver content to ISPs' customers. This motivates ISPs to host and distribute content to their customers.Content can be enterprise-oriented like web-based services or residential-based like triple play as in AT&T's U-Verse [2] and Verizon FiOS [3] deployments. When ISPs and CPs operate independently, they optimize their performance without much cooperation, even though they influence each other indirectly. ISPs deploying content services makes the cooperation between ISP and CP possible, which begs the question of how much can be gained from such cooperation and what kind of cooperation is the most beneficial.

As a traditional service provider, an ISP's primary role is to deploy infrastructure, manage connectivity, and balance traffic load inside its network. In particular, an ISP solves the *traffic engineering* (TE) problem, i.e., adjusting the routing configuration to the prevailing traffic. The goal of TE is to ensure efficient routing to minimize congestion, so that users experience low packet loss, high throughput, and low latency, and that the network can gracefully absorb flash crowds.

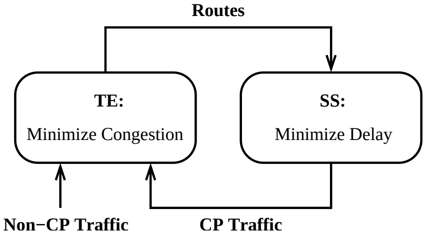


Figure 1: The interaction between traffic engineering (TE) and server selection (SS).

To offer its own content service, an ISP must deploy a content distribution infrastructure. In practice, as many content providers do, the state-of-the-art approach is to replicate content over a number of strategically-placed servers, and direct requests to different servers in the hope of balancing load and decreasing response time. Typical examples in the wide-area setting include YouTube, and content distribution networks (CDNs) like Akamai. The CP solves a *server selection* (SS) problem, i.e., determining which servers should deliver content to each end user. The goal of SS is to meet user demand, minimize network latency to reduce user waiting time, and balance server load to increase throughput.

To offer both network connectivity and content delivery, an ISP is faced with coupled TE and SS problems, as shown in Figure 1. TE and SS interact because TE affects the routes that carry the CP's traffic, and SS affects the offered load seen by the network. Actually, the degrees of freedom are also the "mirror-image" of each other: the ISP controls route selection, which is the constant parameter in the SS problem, while the CP controls server selection, which is the constant parameter in the TE problem.

In this paper, we study several approaches an ISP could take in managing traffic engineering and server selection, ranging from running the two systems independently to designing a joint system. We refer to CP as the part of the system that manages server selection, whether it is performed directly by the ISP or by a separate company that cooperates with the ISP. This study allows us to explore a migration path from the status-quo to a synergistic joint design that benefits both parties. We consider three scenarios with increasing amount of cooperation between traffic engineering and

	Optimality	Info. Exchange	Fairness	Architectural Change
Model I	Not Pareto-optimal	Measurement only	No	Current practice
Model II	Not Pareto-optimal in general Social-optimal in special case More info. may hurt the CP	Topology and Routing Background traffic	No	Minor CP changes
Model III	Pareto-optimal 5-30% performance improvement	Topology and Routing Link prices	Yes	Clean-slate design CP given more control Incrementally deployable

Table 1: Summary of results and engineering implications.

server selection, as summarized here:

- **Model I:** no cooperation (current practice).
- **Model II:** improved visibility (sharing information).
- **Model III:** a joint design (sharing control).

Model I. Content services could be provided by a CDN that runs independently on the ISP network. However, the CP has limited visibility into the underlying network topology and routing, and therefore has limited ability to predict the effect of its own actions. We model a scenario where the CP measures the end-to-end latency of the network and greedily assign each user to the servers with the lowest latency to the user, a strategy some CPs employ today [4]. We call this *greedy server selection*. In addition, TE assumes the offered traffic is unaffected by its routing decisions, despite that routing changes can affect path latencies and therefore the CP's traffic. When the TE problem and the SS problem are solved separately, their interaction can be modeled as a game where they settle in a Nash equilibrium, which may not be *Pareto optimal*.

Not surprisingly, performing TE and SS independently is often sub-optimal because (i) server selection is based on incomplete (and perhaps inaccurate) information about network conditions and (ii) the two systems, acting alone, may miss opportunities for a good joint selection of servers and routes. Models II and III capture these two issues, allowing us to understand which factor is more important in practice.

Model II. Greater visibility into network conditions should enable the CP to make better decisions. There are, in general, four types of information that could be shared: (i) physical topology information, e.g., P4PWG [5], (ii) connectivity information, e.g., routing in the ISP network, (iii) dynamic properties of links, e.g., OSPF link weights, background traffic, and congestion level, and (iv) dynamic properties of nodes, e.g., bandwidth and processing power that can be shared by the node. Our work focuses on a combination of these types of information, i.e., (i)-(iii), so that the CP is able to solve the SS problem more efficiently, i.e., to find the *optimal server selection*.

Sharing information requires minimal extensions to existing solutions for TE and SS, making it amenable to incremental deployment. In addition, we prove that, when the two systems have the same performance objective, TE and SS separately optimizing their own objectives converges to a global optimal solution. However, when the two systems have different performance objectives (e.g., SS minimizes end-to-end latency and TE minimizes congestion), the equilibrium is *not* optimal in general. In addition, we find that model II sometimes performs *worse* than model I—that is, extra visibility into network conditions sometimes leads to a *less efficient* outcome—and the performance degradation can be unbounded. The facts that both Model I and Model II in general do not achieve optimality, and that extra information (Model II) sometimes hurts the performance, motivate us to consider a clean-slate joint design for selecting servers and routes.

Model III. A joint design should achieve *Pareto optimality* for TE and SS. In particular, our design is based on the *Nash Bargaining Solution* [6] that arises from bargaining interactions between players. Thus the solution not only guarantees *efficiency*, but also *fairness* between synergistic or even conflicting objectives of two players, i.e., it is a point on the Pareto optimal curve where both TE and SS have better performance compared to the Nash equilibrium. We also propose a decomposed solution so that the joint design can be implemented in a distributed fashion with a limited amount of information exchange.

The analytical and numerical evaluation of these three models allows us to gain insights for designing a cooperative TE and SS system, summarized in Table 1. The conventional approach of Model I requires minimum information passing, but suffers from sub-optimality and unfairness. Model II requires only minor changes to the CP's server selection algorithm, but the result is still not Pareto optimal and the performance improvement is not guaranteed (in some cases the performance even *degrades*). Model III ensures optimality and fairness through a distributed protocol, requires only moderate increase in information exchange, and is incrementally deployable. Our results show that letting CP have some control over network routing is the key to TE and SS cooperation.

We perform numerical simulations on realistic ISP topologies, which allow us to observe the efficiency loss and paradoxes over a wide range of traffic conditions. The joint design shows significant improvement for both the ISP and the CP. The simulation results further reveal the impact of topologies on the efficiencies and fairness of the three system models.

The rest of the paper is organized as follows. Section 2 presents a standard model for traffic engineering. Section 3 presents our two models for server selection, when given minimal information (i.e., Model I) and perfect information (i.e., Model II) about the underlying network. We also describe the algorithms that implement greedy server selection and optimal server selection. Section 4 studies the interaction between TE and SS as a game and shows that they reach a Nash equilibrium. We also find a case where a social optimality is possible. Section 5 analyzes the efficiency loss of Model I and Model II in general. We show that the Nash equilibria achieved in both models are not Pareto optimal. In particular, we illustrate the counterintuitive observation that more information is not always helpful. Section 6 discusses how to jointly optimize TE and SS by implementing a Nash bargaining solution. We propose a distributed algorithm that allows practical and incremental implementation. We perform large-scale numerical simulations on realistic ISP topologies in Section 7. Finally, Section 8 presents related work, and Section 9 concludes the paper and discusses our future work.

$G:$	Network graph $G = (V, E)$
$V:$	Set of nodes
$S:$	$S \subset V$, the set of CP servers
$T:$	$T \subset V$, the set of users
$E:$	Set of links
$C_l:$	Capacity of link l
$r_l^{ij}:$	Proportion of flow $f: i \to j$ traversing link l, also noted as r_l^f
$R:$	The routing matrix $R: \{r_l^{ij}\}$
$x_{st}:$	Traffic rate from server s to user t
$X_{cp}:$	CP's decision variable $X_{cp} = \{x_{st}\}_{s \in S, t \in T}$
$M_t:$	User t's demand rate for content
$f_l:$	Total traffic on link l
$f_l^{cp}:$	CP's traffic on link l
$f_l^{bg}:$	Background traffic on link l
$D_p:$	Delay of path p
$D_l:$	Delay of link l
$g(\cdot):$	Cost function used in ISP traffic engineering
$h(\cdot):$	Cost function used in CP server selection

Table 2: Summary of notations.

2. TRAFFIC ENGINEERING BACKGROUND

In this section, we describe the network model and formulate the optimization problem that TE solves. (Note that our TE model follows a well-established formulation, and hence is not novel.) We also start introducing the notation used in this paper, which is summarized in Table 2.

Consider a network represented by graph $G = (V, E)$, where V denotes the set of nodes and E denotes the set of directed physical links. A node can be a router, a host, or a server. Let x_{ij} denote the rate of flow (i, j), from node i to node j, where $i, j \in V$. Flows are carried on end-to-end paths consisting of some links. Let $W = \{w_{pl}\}$ be the routing matrix, i.e., $w_{pl} = 1$ if link l is on path p. We do not limit the number of paths so W can include *all* possible paths. Alternatively, one can find out which paths actually carry traffic, and make W smaller by pruning the unused paths. The capacity of a link $l \in E$ is $C_l > 0$.

Given the traffic demand, traffic engineering changes routing to minimize network congestion. In practice, network operators control routing either by changing OSPF link weights [7] or by establishing MPLS label-switched paths [8]. In this paper we use the multi-commodity flow solution to route traffic, because a) it is optimal, i.e., it gives the routing with minimum congestion, and b) it can be realized by routing protocols that use MPLS tunneling, or as recently shown, in a distributed fashion by a new link-state routing protocol [9]. Formally, let $r_l^{ij} \in [0, 1]$ denote the proportion of traffic of flow (i, j) that traverses link l. To realize the multi-commodity flow solution, the network splits each flow over a number of paths. Let $R = \{r_l^{ij}\}$ be the routing matrix.

Let f_l denote the total traffic traversing link l, and we have $f_l = \sum_{(i,j)} x_{ij} \cdot r_l^{ij}$. Now traffic engineering can be formulated as the following optimization problem:

TE($R|X$):

$$\text{minimize} \quad TE = \sum_l g_l(f_l) \tag{1}$$

$$\text{subject to} \quad f_l = \sum_{(i,j)} x_{ij} \cdot r_l^{ij} \le C_l, \ \forall l$$

$$\sum_{l: l \in \text{In}(v)} r_l^{ij} - \sum_{l: l \in \text{Out}(v)} r_l^{ij} = I_{v=j}, \ \forall (i,j), \ \forall v \in V \backslash \{i\}$$

$$\text{variables} \quad 0 \le r_l^{ij} \le 1, \forall (i,j), \forall l$$

where $g_l(\cdot)$ represents a link's congestion cost as a function of the load, $I_{v=j}$ is an indicator function which equals 1 if $v = j$ and 0 otherwise, $\text{In}(v)$ denotes the set of incoming links to node v, and $\text{Out}(v)$ denotes the set of outgoing links from node v.

In this model, TE does not differentiate between the CP's traffic and background traffic. In fact, TE assumes a constant traffic matrix X, i.e., the offered load between each pair of nodes, which can either be a point-to-point background traffic flow, or a flow from a CP's server to a user. As we will see later, this common assumption is undermined when the CP performs dynamic server selection.

We only consider the case where cost function $g_l(\cdot)$ is a *convex*, *continuous*, and *non-decreasing* function of f_l. By using such an objective, TE penalizes high link utilization and balances load inside the network. We will discuss later the analytical form of $g_l(\cdot)$ which the ISP may use in practice. Since $g_l(\cdot)$ is convex and the constraint set is affine and compact, the TE problem (1) is a convex optimization problem. This implies that a local optimum is also a global optimum, and can be computed efficiently through standard algorithms such as the primal-dual interior point algorithm.

3. SERVER SELECTION MODELS

While traffic engineering usually assumes that demand is point-to-point and constant, both assumptions are untrue when some or all of the traffic is generated by the CP. A CP usually has many servers to serve the same content, and which server serves which user depends on the network conditions (e.g., congestion). In this section, we present two novel CP models which correspond to Model I and II introduced in Section 1. The first one models the current CP operation, where the CP relies on end-to-end *measurement* of the network condition in order to make server selection decisions; the second one models the situation when the CP obtains enough information from the ISP so that it can *calculate* the effect of its server selection actions.

3.1 Server Selection Problem

The CP solves the server selection problem to optimize the perceived performance of all its users. We first introduce the notation used in modeling server selection. In the ISP's network, let $S \subset V$ denote the set of CP's servers, which are strategically placed at different locations in the network. For simplicity we assume that all content is duplicated at all servers, and our results can easily be extended to the general case. Let $T \subset V$ denote the set of users who request content from the servers. A user $t \in T$ has a demand for content at rate M_t, which we assume to be constant during the time a CP optimizes its server section. We allow a user to simultaneously download content from multiple servers, because node t can be viewed as an edge router in the ISP's network that aggregates the traffic of many endhosts, which may be served by different servers. We further assume that the demand of a user node can be arbitrarily divided among the servers. Thus our analysis should give an upper bound on the CP's performance.

To differentiate the CP's traffic from background traffic, we denote x_{st} as the traffic rate from server s to user t. To satisfy the traffic demand, we need

$$\sum_{s \in S} x_{st} = M_t.$$

We denote $X_{cp} = \{x_{st}\}_{s \in S, t \in T}$ as the CP's decision variable. We assume that server capacity is not a bottleneck, so the load of server

s, i.e. $\sum_{t \in T} x_{st}$, is unconstrained.

One of the major goals in server selection is to optimize the overall performance of all CP's customers. We use an additive link cost for the CP that is inspired by modeling network latency, i.e., each link has a cost, and the end-to-end path cost is the sum of the link costs along the way. As an example, suppose the content is delay-sensitive (e.g., IPTV), and the CP would like to minimize the *average* end-to-end delay of all its users, which is the same as minimizing the *total* end-to-end delay. Let D_p denote the end-to-end latency of a path p, and $D_l(f_l)$ denote the latency of link l, modeled as a convex, non-decreasing, and continuous function of the amount of flow f_l on the link. By definition, $D_p = \sum_{l \in p} D_l(f_l)$. Then the overall latency experienced by all CP's users is

$$\begin{aligned} SS &= \sum_{(s,t)} \sum_{p \in P(s,t)} x_p^{st} \cdot D_p \\ &= \sum_{(s,t)} \sum_{p \in P(s,t)} x_p^{st} \cdot \sum_{l \in p} D_l(f_l) \\ &= \sum_{l} D_l(f_l) \cdot \sum_{(s,t)} \sum_{p \in P(s,t): l \in p} x_p^{st} \\ &= \sum_{l} f_l^{cp} \cdot D_l(f_l) \end{aligned} \tag{2}$$

where $P(s,t)$ is the set of paths serving flow (s,t) and x_p^{st} is the amount of flow (s,t) traversing path $p \in P(s,t)$.

Let $h_l(\cdot)$ represent the cost of link l, which we assume is convex, non-decreasing, and continuous. In this example, $h_l(f_l^{cp}, f_l) = f_l^{cp} \cdot D_l(f_l)$. Thus, the link cost $h_l(\cdot)$ is a function of the CP's total traffic f_l^{cp} on the link, as well as the link's total traffic f_l, which also includes background traffic.

Expression (2) provides a simple way to calculate the total user experienced end-to-end delay—simply sum over all the *links*, but it requires the knowledge of the load on each link, which is possible only in Model II. Without such knowledge (Model I), the CP has to rely on the end-to-end delay measurements.

We will use the overall user latency as the cost function for CP throughout this paper, and assume that the link delay function $D_l(\cdot)$ is convex and increasing. Our model can be readily extended to other additive costs by using a different convex, increasing link cost function. We can also use other cost functions such as the maximum user latency, and most of our results would hold.

3.2 Greedy Server Selection: Model I

In today's Internet architecture, a CP does not have access to an ISP's network information, such as routing, topology, link delay, and background traffic. Therefore a CP relies on measured or inferred information to optimize its performance. To minimize its users' latency, for instance, a CP can assign each user to servers with the lowest (measured) end-to-end latency to the user. In practice, content distribution networks like Akamai's DNS-based server selection algorithm use this approach [4]. We call it *greedy server selection* and use it as our first model.

To be precise, we assume that the CP monitors the latency from all servers to all users, and makes server selection decisions to minimize all users' total delay. Since the demand of a user can be arbitrarily divided, we can think of the CP as greedily assigning each infinitesimal demand to the best server. The placement of this traffic may change the path latency, which the CP monitors. Thus, at the equilibrium, the servers which send (nonzero) traffic to a user should have the same end-to-end latency to the user, because otherwise the server with low latency will be assigned more demand, causing its latency to increase, and the servers not sending traffic to a user should have higher latency than those that serve the user. This is sometimes called the *Wardrop equilibrium* [10]. The greedy server selection problem is very similar to selfish routing [11, 12], where each flow tries to minimize its average latency over multiple paths without coordinating with other flows. It is known that the equilibrium point in selfish routing can be viewed as the solution to a global convex optimization problem [11], so greedy server selection has a unique equilibrium point.

Although the equilibrium point is well-defined and is the solution to a convex optimization problem, in general it is hard to compute the solution analytically. Thus we leverage the idea of Q-learning [13] to implement a distributed iterative algorithm to find the equilibrium of greedy server selection. The algorithm is guaranteed to converge even under dynamic network environments with cross traffic and link failures, and hence can be used in practice by the CPs. The detailed description and implementation are in Appendix A.

As we will show, the greedy server selection is not optimal. We use it as a baseline for how well a CP can do with only the end-to-end latency measurements.

3.3 Optimal Server Selection: Model II

In this section, we describe how a CP can optimize server selection given *complete* visibility into the underlying network. That is, this is the best the CP can do without *changing* or directly influencing the routing of the network. We also present an optimization formulation that allows us to analytically study its performance.

Suppose that content providers are able to either obtain information on network conditions directly from the ISP, or infer it by its measurement infrastructure and technology. In the best case, the CP is able to obtain the complete information about the network, i.e., routing decision and link latency. This situation is characterized by problem (3), which is the best performance the CP can achieve without directly influencing the ISP. To optimize the overall user experience, the CP solves the following cost minimization problem:

SS$(X_{cp}|R)$**:**

$$\begin{aligned} \text{minimize} \quad & SS = \sum_{l} h_l(f_l^{cp}, f_l) \qquad (3) \\ \text{subject to} \quad & f_l^{cp} = \sum_{(s,t)} x_{st} \cdot r_l^{st}, \ \forall l \\ & f_l = f_l^{cp} + f_l^{bg} \leq C_l, \ \forall l \\ & \sum_{s \in S} x_{st} = M_t, \ \forall t \\ \text{variables} \quad & x_{st} \geq 0, \ \forall (s,t) \end{aligned}$$

where we denote $f_l^{bg} = \sum_{(i,j) \neq (s,t)} x_{ij} \cdot r_l^{ij}$ as the non-CP traffic on link l, which is a parameter to the optimization problem. If the cost function $h_l(\cdot)$ is increasing and convex on the variable f_l^{cp}, one can verify that (3) is a convex optimization problem, and hence has a unique global optimal value.

In practice, the optimal server selection (3) is amenable to an efficient implementation. The problem can either be solved centrally, e.g., at the CP's central coordinator, or via a distributed algorithm similar to that used for Model I. We solve (3) centrally in our simulations, since it is easy to solve analytically, and that we are more interested in the performance improvement brought by complete information than the algorithm that implements it.

4. ANALYZING TE-SS INTERACTION

In this section, we explore the interaction between the ISP and the CP when they operate independently without coordination, which applies to our Model I and Model II. We study their "interplay" in a game-theoretic framework. The game formulation allows us to analyze the stability condition, i.e., we show that alternate TE and SS optimizations will reach an equilibrium point. In addition, we find that when the ISP and the CP optimize the same system objective, their interaction achieve *global optimality* under Model II.

4.1 TE-SS Game and Nash Equilibrium

We start with the formulation of a two-player non-cooperative Nash game that characterizes the TE-SS interaction.

Definition 1. *The* TE-SS game *consists of a tuple* $[N,A,U]$. *The player set* $N = \{isp, cp\}$. *The action set* $A_{isp} = \{R\}$ *and* $A_{cp} = \{X_{cp}\}$, *where the feasible set of R and* X_{cp} *are defined by the constraints in (1) and (3) respectively. The utility functions are* $U_{isp} = -TE$ *and* $U_{cp} = -SS$.

Figure 1 shows the interaction between SS and TE. In both Model I and Model II, the ISP plays the best response strategy, i.e., the ISP always optimizes (3) given the CP's strategy X_{cp}. Similarly, the CP plays the best response strategy in Model II when given full information. However, the CP's strategy in Model I is not the best response, since it does not optimize (3) due to the lack of network visibility. Indeed, the utility the CP implicitly optimizes when doing greedy server selection is [11]

$$U_{cp} = -\sum_{l\in E}\int_0^{f_l} D_l(u)du$$

This later helps us understand the stability conditions of the game.

Consider a particular game procedure in which the ISP and the CP take turns to optimize their own objectives by varying its own decision variable, treating the that of the other player as constant. Specifically, in the k-th iteration, we have

$$\begin{aligned} R^{(k+1)} &= \underset{R}{\text{argmin}}\ TE(X_{cp}^{(k)}) \\ X_{cp}^{(k+1)} &= \underset{X_{cp}}{\text{argmin}}\ SS(R^{(k+1)}) \end{aligned} \tag{4}$$

Note that two optimization problems may be solved on different timescales. The ISP runs traffic engineering at the timescale of hours. Depending on the CP's design choices, server selection is optimized a few times a day, or at a smaller timescale like seconds or minutes of a typical content transfer duration. We assume that each player has fully solved its optimization problem before the other one starts.

Next we prove the existence of Nash equilibrium of the TE-SS game. We establish the stability condition when two players use general cost functions $g_l(\cdot)$ and $h_l(\cdot)$ that are continuous and convex. While TE's formulation is the same in Model I and Model II, we consider the two SS models, i.e., greedy server selection and optimal server selection, respectively.

Theorem 1. *The strategic game TE-SS has a Nash equilibrium for both Model I and Model II.*

Proof Sketch: It suffices to show that (i) each player's strategy space is a nonempty compact convex subset, and (ii) each player's utility function is continuous and quasi-concave on its strategy space, which follows the standard proof in [14]. The ISP's strategy space is defined by the constraint set of (1), which are affine equalities and inequalities, hence a convex compact set. Since $g_l(\cdot)$ is continuous and convex, we can easily verify that the objective of (1) is quasi-convex on $R = \{r_l^{ij}\}$. CP's strategy space is defined by the constraint set of (3), which is also convex and compact. Similarly, if $h_l(f_l^{cp})$ is continuous and convex, the objective of (3) is quasi-convex on X_{cp}. In particular, consider the special case in which CP minimizes latency (2). When CP is doing greedy server selection, $h_l(f_l) = \int_0^{f_l} D_l(u)du$. When CP is doing optimal server selection, $h_l(f_l^{cp}) = f_l^{cp} D_l(f_l)$. In both cases, if $D_l(\cdot)$ is continuous, non-decreasing, and convex, $h_l(\cdot)$ is also continuous and convex. One can again verify the quasi-convexity of the objective in (3). Details of the proof are omitted here due to the space limit. ■

The existence of a Nash equilibrium does not guarantee that the trajectory path (4) leads to one. In Section 7 we demonstrate the convergence of iterative player optimization in simulation. In general, the Nash equilibrium may not be unique, in terms of both decision variables and objective values. Next, we show a special case where the Nash equilibrium is unique and can be attained by alternate player moves (4).

4.2 Global Optimality under Same Objective

In the following, we study a special case of the TE-SS game, in which the ISP and the CP optimize the same objective function, i.e., $g_l(\cdot) = h_l(\cdot) = \Phi_l(\cdot)$, and there is no background traffic. One example is when the network carries only the CP traffic and both the ISP and the CP aim to minimize the average traffic latency, i.e., $\Phi_l(f_l) = f_l \cdot D_l(f_l)$. An interesting question that naturally arises is whether the two players' alternate optimization on each of their decision variables leads to a global optimum, i.e., when the (common) objective is optimized over all decision variables.

Consider a special case of the TE-SS game, where two players' objectives coincide:

$$TE = SS = \sum_l \Phi_l(f_l) \tag{5}$$

Namely, both the ISP and the CP want to minimize the average latency of the traffic carried on the network. The CP is given the visibility from the ISP, so it can fully solve the SS problem (Model II).

Then we define the notion of *global optimality*, which is the optimal point to the following optimization problem.

TE-SS-special(X):

$$\begin{aligned} \text{minimize} \quad & \sum_l \Phi_l(f_l) \\ \text{subject to} \quad & f_l = \sum_{(s,t)} x_l^{st} \le C_l,\ \forall l \\ & \sum_{s\in S}\left(\sum_{l:l\in \text{In}(v)} x_l^{st} - \sum_{l:l\in \text{Out}(v)} x_l^{st}\right) = M_t \cdot I_{v=t},\ \forall v \notin S,\ \forall t \in T \\ \text{variables} \quad & x_l^{st} \ge 0,\ \forall (s,t), \forall l \end{aligned} \tag{6}$$

where x_l^{st} denotes the traffic rate of flow (s,t) on link l. The variable x_l^{st} allows a global coordinator to route a user's demand from any server in any way it wants, thus problem (6) establishes a bound on how well one can do to minimize the CP traffic's average latency. A detailed discussion of such choice is left to the general case in Section 5.2.

We compare the Nash equilibrium of the TE-SS game with objective functions (5) to the optimal solution of (6). In particular, we study whether alternate player optimizations as in (4) lead to this optimal point. As good news, we show that the Nash equilibrium of TE-SS game is an optimal point, so the TE-SS interaction does not result in any efficiency loss when their objectives are identical. Actually, the equilibrium point can be achieved by alternate optimization steps of (4).

Lemma 1. *A special case TE-SS game in which two players' objectives are in the form of (5) has a Nash equilibrium, when the cost function $\Phi_l(f_l)$ is continuous, non-decreasing, and convex.*

The proof is similar to that of Theorem 1 and is omitted here. ■

Consider the following optimization problem:

$$\begin{aligned} \text{minimize} \quad & \sum_l \Phi_l(f_l) && (7) \\ \text{subject to} \quad & f_l = \sum_{(s,t)} x_{st} \cdot r_l^{st} \leq C_l, \ \forall l \\ & \sum_{l:l \in \text{In}(v)} r_l^{st} - \sum_{l:l \in \text{Out}(v)} r_l^{st} = I_{v=t}, \ \forall (s,t), \ \forall v \in V \backslash \{s\} \\ & \sum_{s \in S} x_{st} = M_t, \ \forall t \\ \text{variables} \quad & 0 \leq r_l^{st} \leq 1, \ x_{st} \geq 0 \end{aligned}$$

The alternate player moves in the special case TE-SS game is actually solving (7) by applying the non-linear Gauss-Seidel algorithm [15], which consists of iterative optimization in a round robin order with respect to each variable while keeping the rest fixed.

Note that (7) is a non-convex problem, since it involves the product of two variables r_l^{st} and x_{st}. However, it is equivalent to the convex problem (6).

Lemma 2. *The non-convex problem (7) that TE-SS game solves is equivalent to (6).*

Proof: We show that there is a one-to-one mapping between the feasible solutions of (6) and (7). Consider a feasible solution $\{x_l^{st}\}$ in (6). Let $x_{st} = \sum_{l:l \in \text{In}(t)} x_l^{st} - \sum_{l:l \in \text{Out}(t)} x_l^{st}$, $r_l^{st} = x_l^{st}/x_{st}$ if $x_{st} \neq 0$. To avoid the case of $x_{st} = 0$, suppose there is an infinitesimally small background traffic for every (s,t) pair, so the one-to-one mapping still holds. It is easy to show that this is a feasible solution of (7). On the other hand, for each feasible solution $\{x_{st}, r_l^{st}\}$ of (7), let $x_l^{st} = x_{st} \cdot r_l^{st}$, which is also a feasible solution of (6). Since two problems share the same objective, they are equivalent. ■

After proving the equivalence, it remains to show that the alternate player moves (4) actually converges to the optimal points of two equivalent problems.

Lemma 3. *The iterative TE and SS optimizations (4) converge to the Nash equilibrium of TE-SS game, which is also the optimal point of (6).*

Proof: The proof proceeds in two steps. We first show that the Nash equilibrium in the TE-SS game is also an optimal solution to (6), though not specifying how it is achieved. The basic idea is to check the KKT [16] conditions of TE and SS at Nash equilibrium, and then compare to the KKT condition of (6). Then we show that an algorithm like (4) converges to the Nash equilibrium. The complete proof is presented in Appendix B. ■

Theorem 2. *The Nash equilibrium of the special case TE-SS game (5) is also global optimal, which can be achieved by two players' alternate best response play.*

The proof follows Lemmas 1, 2, and 3. ■

The study of this special case allows us to estimate the lower bound on efficiency loss due to TE-SS interaction, which can be zero. Though such a special case is rare in general, it offers us some insights into when such efficiency loss may be low: (i) the ISP and the CP should have synergistic objectives, (ii) the CP's traffic is at high percentage, i.e., there is no background traffic. As we will see later, the lack of any one of the two conditions is possible to suffer great efficiency loss.

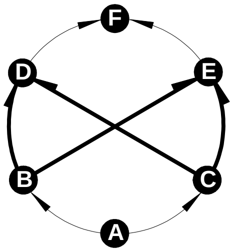


Figure 2: An Example of the Paradox of Extra Information

link	$l_1 : BD$	$l_2 : BE$	$l_3 : CD$	$l_4 : CE$
C_l	$1+\varepsilon$	$1+\varepsilon$	$1+\varepsilon$	$1+\varepsilon$
$D_l(f_l)$	f_1	$\frac{1}{1+\varepsilon-f_2}$	$\frac{1}{1+\varepsilon-f_3}$	f_4
$g_l(x)$	$g_1(\cdot) = g_2(\cdot) = g_3(\cdot) = g_4(\cdot)$			

Table 3: Link capacities, ISP's and CP's link cost functions in the example of Paradox of Extra Information.

5. EFFICIENCY LOSS

Though in the last section we show the stability of general TE-SS interaction, and global optimality in a special case, such interaction may result in serious efficiency loss in the general case. In this section, we conduct two case studies to show the optimality gap. We first present a toy network and show that under certain conditions the CP performs even worse in Model II than Model I, despite having more information about underlying network conditions. We next propose the notion of Pareto-optimality as the performance benchmark, and quantitatively demonstrate the suffering of efficiency loss in both Model I and Model II.

5.1 The Paradox of Extra Information

In the previous section, we present a special case when the ISP and the CP's interaction leads to a social optimal point. However, when their objectives are not aligned, the interaction may reach a sub-optimal point. To improve its performance, the CP can gain more knowledge about the underlying network by introducing accurate network measurement and inference. As proposed in Model II, ISPs and CPs can collaborate by passing information. Intuitively, the CP is able to achieve better performance, given more information from the network provider. However, as we will show in the following example, the argument is not always true.

Consider an ISP network illustrated in Figure 2. We designate an end user node, $T = \{F\}$, and two CP servers, $S = \{B, C\}$. The end user has a content demand of $M_F = 2$. We also allow two background traffic flows, $A \rightarrow D$ and $A \rightarrow E$, each of which has one unit of traffic demand. Edge directions are noted on the figure, so the ISP's routing decision space is self-evident, i.e., there are two possible paths for each traffic flow (clockwise and counterclockwise). To simplify the analysis and deliver the most essential message from this example, suppose that both TE and SS costs on four thin links are negligible so the four bold links constitute the *bottleneck* of the network. In Table 3, we list the link capacities, ISP's cost function $g_l(\cdot)$, and link latency function $D_l(\cdot)$. Suppose the CP aims to minimize the average latency of its traffic. We compare the Nash equilibrium of two situations when the CP optimizes its network by greedy server selection and optimal server selection.

The stability condition for the ISP at Nash equilibrium is $g_1'(f_1) = g_2'(f_2) = g_3'(f_3) = g_4'(f_4)$. Since ISP's link cost functions are identical, this implies the total traffic on each link must be identical. This can be viewed as one special case of ISP's optimization of minimizing the maximum link utilization, since links have the same capacities. On the other hand, the stability condition for the CP at Nash equilibrium is that (B,F) and (C,F) have the same latency, or marginal latency. Based on the observation, we can derive two Nash equilibrium points.

When the CP is using greedy server selection strategy, let

$$\text{Model I:}\begin{cases} X_{CP}:\left\{x_{BF}=1,\ x_{CF}=1\right\} \\ R:\left\{r_1^{BF}=1-\alpha,\ r_2^{BF}=\alpha,\ r_3^{CF}=\alpha,\ r_4^{CF}=1-\alpha,\right. \\ \qquad \left. r_1^{AD}=\alpha,\ r_3^{AD}=1-\alpha,\ r_2^{AE}=1-\alpha,\ r_4^{AE}=\alpha\right\} \end{cases}$$

One can check that this is indeed a Nash equilibrium solution, where $f_1=f_2=f_3=f_4=1$, and $D_{BF}=D_{CF}=1-\alpha+\alpha/\varepsilon$. The CP's objective $SS_{\mathrm{I}}=2(1-\alpha+\alpha/\varepsilon)$.

When the CP is using optimal server selection strategy, let

$$\text{Model II:}\begin{cases} X_{CP}:\left\{x_{BF}=1,\ x_{CF}=1\right\} \\ R:\left\{r_1^{BF}=\alpha,\ r_2^{BF}=1-\alpha,\ r_3^{CF}=1-\alpha,\ r_4^{CF}=\alpha,\right. \\ \qquad \left. r_1^{AD}=1-\alpha,\ r_3^{AD}=\alpha,\ r_2^{AE}=\alpha,\ r_4^{AE}=1-\alpha\right\} \end{cases}$$

This is a Nash equilibrium point, where $f_1=f_2=f_3=f_4=1$, and $d_{BF}=d_{CF}=\alpha(1+\alpha)+(1-\alpha)(1/\varepsilon+(1-\alpha)/\varepsilon^2)$. The CP's objective $SS_{\mathrm{II}}=2(\alpha+(1-\alpha)/\varepsilon)$.

It is interesting to see that when $0<\varepsilon<1, 0\le\alpha<1/2$, $SS_{\mathrm{I}}<SS_{\mathrm{II}}$, which means that more information may hurt CP's performance! In the worst case,

$$\lim_{\alpha\to 0,\varepsilon\to 0}\frac{SS_{\mathrm{II}}}{SS_{\mathrm{I}}}=\infty$$

i.e., the efficiency loss can be unbounded.

This is not surprising since the Nash equilibrium is generally not unique, both in terms of equilibrium solutions and equilibrium objectives. When ISP and CP's objectives are mis-aligned, ISP's decision may route CP's traffic on bad paths from the CP's perspective, though the ISP's objective is optimized given CP's traffic. In practice, such scenario is likely to happen, since the ISP cares about link congestion (link utilization), while the CP cares about latency, which not only correlates to link congestion, but also propagation delay. Thus ISP and CP's partial collaboration by only passing information is not sufficient for achieving global optimality.

5.2 Pareto Optimality and Illustration of Sub-Optimality

As we observe in the example in Figure 2, one of the major causes of sub-optimality is that TE and SS's objectives are not necessarily aligned. To measure efficiency in a system with multiple objectives, a common approach is to explore the operating region of the system and find the *Pareto curve*. Points on the Pareto curve are those of which we cannot improve one objective further without hurting the other. In particular, the Pareto curve characterizes the tradeoffs of conflicting (or at least not aligning) goals of different parties. One way to trace the tradeoff curve is to optimize a weighted sum of the objectives:

$$\text{minimize}\quad TE+\gamma\cdot SS \qquad (8)$$

$$\text{variables}\quad R\in\mathscr{R},\ X_{cp}\in\mathscr{X}_{cp}$$

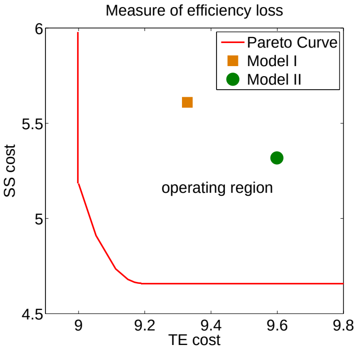


Figure 3: Illustration of sub-optimality.

where $\gamma\ge 0$ is a scalar representing the relative weight of the two objectives. $\mathscr{R}$ and $\mathscr{X}_{cp}$ are the feasible regions defined by the constraints in (1) and (3):

$$\mathscr{R}\times\mathscr{X}_{cp}=\Bigg\{r_l^{ij},\,x_{st}\mid 0\le r_l^{ij}\le 1,\ x_{st}\ge 0,$$
$$\sum_{l:l\in\mathrm{In}(v)} r_l^{ij}-\sum_{l:l\in\mathrm{Out}(v)} r_l^{ij}=I_{v=j},\forall v\in V\backslash\{i\},$$
$$f_l=\sum_{(i,j)} x_{ij}\cdot r_l^{ij}\le C_l,\ \sum_{s\in S} x_{st}=M_t\Bigg\}$$

The formulation of problem (8) is not easy to solve. In fact, the objective of (8) is no longer convex on the master variable $\{r_l^{st},x_{st}\}$, and the feasible region defined by constraints of (8) is not convex. One way to overcome this problem is to consider a relaxed decision space. Instead of restricting each player to its own operating domain, i.e., ISP controls routing and CP controls server selection, we introduce a joint routing and content delivery problem. Let x_l^{st} denote the *rate* of traffic carried on link l that belongs to flow (s,t), and denote the CP's decision variable as $X_{cp}=\{x_l^{st}\}_{s\in S,t\in T}$. Similar to the argument we made earlier, this can be viewed as a generalization of explicit multipath routing. Consider the following optimization problem:

TE-SS-weighted(X_{cp},R_{bg})

$$\begin{aligned} \text{minimize}\quad & TE+\gamma\cdot SS && (9)\\ \text{subject to}\quad & f_l^{cp}=\sum_{(s,t)} x_l^{st},\ \forall l \\ & f_l=f_l^{cp}+\sum_{(i,j)\notin S\times T} x_{ij}\cdot r_l^{ij}\le C_l,\ \forall l \\ & \sum_{l:l\in\mathrm{In}(v)} r_l^{ij}-\sum_{l:l\in\mathrm{Out}(v)} r_l^{ij}=I_{v=j},\ \forall(i,j)\notin S\times T,\forall v\in V\backslash\{i\} \\ & \sum_{s\in S}\left(\sum_{l:l\in\mathrm{In}(v)} x_l^{st}-\sum_{l:l\in\mathrm{Out}(v)} x_l^{st}\right)=M_t\cdot I_{v=t},\ \forall v\notin S,\ \forall t\in T \\ \text{variables}\quad & x_l^{st}\ge 0,\ 0\le r_l^{ij}\le 1 \end{aligned}$$

Denote the feasible space of the joint variable as $\mathscr{J}=\{X_{cp},R_{bg}\}$. If we vary γ and plot the achieved TE objectives versus SS objectives, we obtain the Pareto curve.

To illustrate the Pareto curve and efficiency loss in Model I and Model II, we plot in Figure 3 the Pareto curve and the Nash equilibria in the two-dimensional space (TE,SS) for an example network similar to Figure 2. The simulation shows that when the CP leverages the complete information to optimize (3), it is able to achieve lower delay, but the TE cost suffers. Though it is not clear which operating point is better, both equilibria are away from the Pareto curve, which shows that there is room for performance improvement in both dimensions.

6. A JOINT DESIGN

Motivated by the need for a joint TE and SS design, we propose the Nash bargaining solution to reduce the efficiency loss observed above. We also present an algorithm based on optimization decomposition theory to implement the solution distributively.

6.1 Motivation

An ISP providing content distribution service in its own network has control over both routing and server selection. So the ISP can take into consideration the characteristics of both types of traffic (background and CP) and jointly optimize a carefully chosen objective. The jointly optimized system should meet at least two goals: (i) optimality, i.e., it should achieve Pareto optimality so the network resources are efficiently utilized, and (ii) fairness, i.e., the tradeoff between two non-synergistic objectives should be balanced so both parties benefit from the cooperation.

One natural design is to optimize the weighted sum of the traffic engineering goal and server selection goal as shown in (9). However, solving (9) for each γ and adaptively tuning γ in a *trial-and-error* fashion is impractical and inefficient. First, it is hard to weigh the tradeoff between the two objectives, especially when they are measured in different units. Second, one needs to prepare for every set of background and CP traffic demand an appropriate weight parameter γ, which is not efficient. In addition, the offline computation does not adapt to dynamic changes of network conditions, such as cross traffic or link failures. Last, from a mathematical viewpoint, tuning γ to explore a broad region of system operating points is computationally expensive. Usually, exploring a large set of γ only produces a small operating region.

Apart from the system perspective, the economic consideration requires that the solution should be *fair*. Namely, such a joint design paradigm should benefit both TE and SS. In addition, such a model also applies to a more general case when the ISP and the CP are different entities. They cooperate only when the cooperation leads to a win-win situation, and the "division" of the benefits should be fair, i.e., one who makes greater contribution to the collaboration should be able to receive more reward, even when their goals are conflicting.

For the concern of a practical deployment, the new system should not require a significant change to the existing infrastructure. Though the joint system is designed from a clean state, it should allow an incremental deployment. In particular, we prefer that the functionalities of routing and server selection be separated, with minor changes to each component. The modularized design allows us to manage each optimization independently, with only a judicious amount of information exchange. Designing for scalability and evolvability is beneficial to both the ISP and the CP, and allows their cooperation either as a single entity or as different ones.

Based on all the above considerations, we borrow the concept of *Nash bargaining solution* [6, 17] from cooperative game theory. As we show next, it ensures that the joint system achieves an *efficient* and *fair* operating point. The solution structure also allows a distributed implementation that is amenable to building a practical system.

6.2 Nash Bargaining Solution

Consider a Nash bargaining solution which solves the following optimization problem:

$$\begin{aligned} \text{maximize} \quad & (TE_0 - TE)(SS_0 - SS) \\ \text{variables} \quad & \{x_l^{st}, r_l^{ij}\} \in \mathscr{I} \end{aligned} \qquad (10)$$

where (TE_0, SS_0) is a constant called the *disagreement point*, which represents the baseline to cooperate. Namely, (TE_0, SS_0) is the status-quo we observe before any cooperation. For instance, one can view the Nash equilibrium in Model I as a disagreement point, since it is the operating point the system would reach without any further optimization. By optimizing the product of performance improvements of TE and SS, the Nash bargaining solution guarantees the joint system is optimal and fair, due to the properties of a Nash bargaining solution. A Nash bargaining solution is uniquely defined by the following four axioms:

- *Pareto optimality*. A Pareto optimal solution ensures efficiency.

- *Symmetry*. The two players should get equal share of the gains through cooperation, if the two players' problems are symmetric, i.e., they have the same cost functions, and have the same objective value at the disagreement point. This is not true for the TE-SS game though.

- *Expected utility axiom*. The Nash bargaining solution is invariant under affine transformations. Intuitively, this axiom suggests that the Nash bargaining solution is insensitive to different units used in the objective and can be efficiently computed by affine projection.

- *Independence of irrelevant alternatives*. This means that adding in extra constraints in the feasible operating region does not change the solution, as long as the solution itself is feasible.

Note that Nash bargaining solution is the only solution that satisfies the above four axioms [6, 17].

The seminal idea of Nash bargaining solution comes from the modeling of bargaining arbitration between two players under which the solution is derived. The analysis is omitted here due to lack of space and we demonstrate the benefits through numerical evaluation. In practice, one can choose the disagreement point as the baseline performance requirement. In this work, we use the Nash equilibrium of Model I as the disagreement point, since it is readily accessible by both the ISP and the CP based on the empirical observation in current practice.

6.3 Distributed Algorithm

A good solution concept not only ensures efficiency and fairness, but should also have a distributed implementation. This is important because separate functionalities allows for a modularized design, in which legacy systems like CDNs can be leveraged with minor changes. Even if the ISP and the CP are two independent economic entities, the distributed structure offers an opportunity to cooperate with judicious information exchange.

In the following, we utilize the theory of optimization decomposition [18] to decompose problem (10) into two subproblems. The ISP solves a routing subproblem and the CP solves a server selection subproblem, with minimal information exchange on links.

The objective of (10) can be converted to

$$\text{maximize} \quad \log(TE_0 - TE) + \log(SS_0 - SS)$$

since the $\log(\cdot)$ function is monotonic and the feasible solution space is unchanged. We introduce two auxiliary variable $\overline{f_l^{cp}}$ and $\overline{f_l^{bg}}$. The above problem can be rewritten as

$$
\begin{aligned}
\text{max.} \quad & \log\Big(TE_0 - \sum_l g_l\big(f_l^{bg} + \overline{f_l^{cp}}\big)\Big) + \log\Big(SS_0 - \sum_l h_l\big(f_l^{cp} + \overline{f_l^{bg}}\big)\Big) \qquad (11)\\
\text{s.t.} \quad & f_l^{cp} = \sum_{(s,t)} x_l^{st},\ f_l^{bg} = \sum_{(i,j)\notin S\times T} x_{ij}\cdot r_l^{ij},\ \forall l\\
& \overline{f_l^{cp}} = f_l^{cp},\ \overline{f_l^{bg}} = f_l^{bg},\ f_l^{cp} + f_l^{bg} \le C_l,\ \forall l\\
& \sum_{l:l\in \mathrm{In}(v)} r_l^{ij} - \sum_{l:l\in \mathrm{Out}(v)} r_l^{ij} = I_{v=j},\ \forall (i,j)\notin S\times T, \forall v\in V\backslash\{i\}\\
& \sum_{s\in S}\left(\sum_{l:l\in \mathrm{In}(v)} x_l^{st} - \sum_{l:l\in \mathrm{Out}(v)} x_l^{st}\right) = M_t\cdot I_{v=t},\ \forall v\notin S,\ \forall t\in T\\
\text{var.} \quad & x_l^{st}\ge 0,\ 0\le r_l^{ij}\le 1,\ \forall (i,j)\notin S\times T,\ \overline{f_l^{cp}},\ \overline{f_l^{bg}}
\end{aligned}
$$

The partial Lagrangian of (11) is

$$
\begin{aligned}
L(x_l^{st}, &r_l^{ij}, \overline{f_l^{cp}}, \overline{f_l^{bg}}, \lambda_l, \mu_l, \nu_l)\\
&= \log\Big(TE_0 - \sum_l g_l\big(f_l^{bg} + \overline{f_l^{cp}}\big)\Big) + \sum_l \mu_l\big(f_l^{bg} - \overline{f_l^{bg}}\big)\\
&+ \log\Big(SS_0 - \sum_l h_l\big(f_l^{cp} + \overline{f_l^{bg}}\big)\Big) + \sum_l \nu_l\big(f_l^{cp} - \overline{f_l^{cp}}\big)\\
&+ \sum_l \lambda_l\big(C_l - f_l^{bg} - f_l^{cp}\big)
\end{aligned}
$$

where λ_l is the *link price*, and μ_l, ν_l are the *consistency prices*. Observe that f_l^{cp} and f_l^{bg} can be separated in the Lagrangian function. We take a dual decomposition approach, and (11) is decomposed into two subproblems:

SS-NBS$(x_l^{st}, \overline{f_l^{bg}})$**:**

$$
\begin{aligned}
\text{max.} \quad & \log\Big(SS_0 - \sum_l h_l\big(f_l^{cp} + \overline{f_l^{bg}}\big)\Big) + \sum_l \big(\nu_l f_l^{cp} - \mu_l \overline{f_l^{bg}} - \lambda_l f_l^{cp}\big) \qquad (12)\\
\text{s.t.} \quad & f_l^{cp} = \sum_{(s,t)} x_l^{st},\ \forall l\\
& \sum_{s\in S}\left(\sum_{l:l\in \mathrm{In}(v)} x_l^{st} - \sum_{l:l\in \mathrm{Out}(v)} x_l^{st}\right) = M_t\cdot I_{v=t},\ \forall v\notin S,\ \forall t\in T\\
\text{var.} \quad & x_l^{st}\ge 0,\ \overline{f_l^{bg}}
\end{aligned}
$$

and

TE-NBS$(r_l^{ij}, \overline{f_l^{cp}})$**:**

$$
\begin{aligned}
\text{max.} \quad & \log\Big(TE_0 - \sum_l g_l\big(f_l^{bg} + \overline{f_l^{cp}}\big)\Big) + \sum_l \big(\mu_l f_l^{bg} - \nu_l \overline{f_l^{cp}} - \lambda_l f_l^{bg}\big) \qquad (13)\\
\text{s.t.} \quad & f_l^{bg} = \sum_{(i,j)\notin S\times T} x_{ij}\cdot r_l^{ij},\ \forall l\\
& \sum_{l:l\in \mathrm{In}(v)} r_l^{ij} - \sum_{l:l\in \mathrm{Out}(v)} r_l^{ij} = I_{v=j},\ \forall (i,j)\notin S\times T, \forall v\in V\backslash\{i\}\\
\text{var.} \quad & 0\le r_l^{ij}\le 1, \forall (i,j)\notin S\times T,\ \overline{f_l^{cp}}
\end{aligned}
$$

The optimal solutions of (12) and (13) for a given set of prices μ_l, ν_l, and λ_l define the dual function $\mathrm{Dual}(\mu_l, \nu_l, \lambda_l)$. The dual problem is given as:

$$
\begin{aligned}
\text{minimize} \quad & \mathrm{Dual}(\mu_l, \nu_l, \lambda_l) \qquad (14)\\
\text{variable} \quad & \lambda_l \ge 0, \mu_l, \nu_l
\end{aligned}
$$

We can solve the dual problem with the following price updates:

$$\lambda_l(t+1) = \Big[\lambda_l(t) - \beta_l\big(C_l - f_l^{bg} - f_l^{cp}\big)\Big]^+,\ \forall l \qquad (15)$$

$$\mu_l(t+1) = \mu_l - \beta_l\big(f_l^{bg} - \overline{f_l^{bg}}\big),\ \forall l \qquad (16)$$

$$\nu_l(t+1) = \nu_l - \beta_l\big(f_l^{cp} - \overline{f_l^{cp}}\big),\ \forall l \qquad (17)$$

where β's are diminishing step sizes or small constant step sizes often used in practice [19].

In this new architecture, the ISP solves the modified version of TE, i.e., TE-NBS, and the CP solves the modified version of SS, i.e., SS-NBS. On information sharing, the CP learns the network topology from the ISP. They do not directly exchange information with each other. Instead, they report $\overline{f_l^{cp}}$ and $\overline{f_l^{bg}}$ variables to each link, which passes the computed price information back to them. This way, only one new component is required, e.g., the price updating function on each link, which can be potentially implemented in each router. Table 4 presents the algorithm that implements the Nash bargaining solution distributively.

	Distributed Algorithm for NBS
(1)	Link initialization: Set λ_l to be some nonnegative value, and set μ_l and ν_l to arbitrary real value.
(2)	The ISP solves (13) and makes routing decision r_l^{ij} for background traffic. The ISP passes f_l^{bg}, $\overline{f_l^{cp}}$ to each link l.
(3)	The CP solves (12) and makes decision x_l^{st} for CP traffic. The CP passes f_l^{cp}, $\overline{f_l^{bg}}$ to each link l.
(4)	Price update: Each link updates the link price λ_l according to (15), and passes λ_l to the ISP and the CP. Each link updates the consistency prices μ_l, ν_l according to (16) and (17), and passes μ_l, ν_l to the ISP and the CP.
(5)	Go to step (2) until the solution converges.

Table 4: Distributed algorithm for solving the Nash bargaining solution

We decouple the functionalities of a joint system, so they can be operated and deployed independently. Note that in our simulation, we actually solve the Nash bargaining solution centrally, without using the distributed solution, since we are primarily interested to see the performance of the solution. A practical protocol implementation and evaluation will be left as our future work.

7. PERFORMANCE EVALUATION

In this section, we use simulation to demonstrate several paradoxical examples that may occur for real network topologies and traffic models. We also compare the performance of the three models we proposed. Complementary to the theoretical analysis, the simulation results presented here allow us to gain a better understanding of the efficiency loss under realistic network environments. These simulation results also provide guides to network operators who need to decide which approach to take, sharing information or sharing control.

7.1 Simulation Setup

We evaluate our models under real ISP topologies obtained from Rocketfuel [20]. We use the backbone topology of the research network Abilene [21] and several major tier-1 ISPs in north America. The choice of these topologies also reflects different geometric properties of the graph. For instance, Abilene is the simplest graph with two bottleneck paths horizontally. The backbones of AT&T and Exodus are a hub-and-spoke structure with some shortcuts between nodes pairs. The topology of Level 3 is almost a complete mesh, while Sprint is in between these two kinds.

Obtaining an accurate traffic matrix is also essential to our evaluation. Unfortunately, we have no means to access real traffic patterns on these networks. So we simulate the traffic demand using a gravity model [22], which reflects the pairwise communication pattern on the Internet. The content demand of a CP user is assumed to be proportional to the node population.

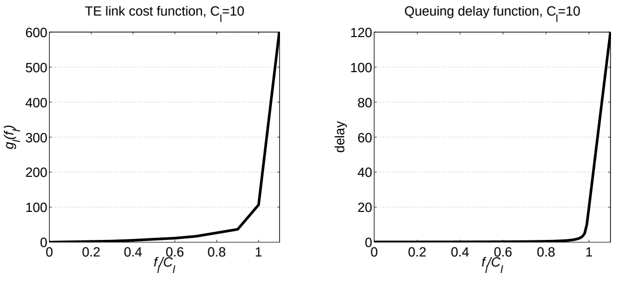
The TE cost function $g(\cdot)$ and the SS cost function $h(\cdot)$ are also carefully chosen. In particular, ISPs usually model congestion cost with a convex increasing function of the link load. The exact shape of the function $g_l(f_l)$ is not important, and we use the same piecewise linear cost function as in [7], given below:

$$g_l(f_l, C_l) = \begin{cases} f_l & 0 \leq f_l/C_l < 1/3 \\ 3f_l - 2/3C_l & 1/3 \leq f_l/C_l < 2/3 \\ 10f_l - 16/3C_l & 2/3 \leq f_l/C_l < 9/10 \\ 70f_l - 178/3C_l & 9/10 \leq f_l/C_l < 1 \\ 500f_l - 1468/3C_l & 1 \leq f_l/C_l < 11/10 \\ 5000f_l - 16318/3C_l & 11/10 \leq f_l/C_l < \infty \end{cases}$$

The CP's cost function can be the performance cost like latency, financial cost charged by ISPs, or something else. We consider the case where latency is the primary performance metric, i.e., the content traffic is delay sensitive like video conferencing or live streaming. So we let the CP's cost function $h_l(\cdot)$ be of the form given by Equation 2, i.e., $h_l(f_l) = f_l^{cp} \cdot D_l(f_l)$. A link's latency $D_l(\cdot)$ consists of queuing delay and propagation delay. The propagation delay is translated from geographical distances between nodes. The queuing delay is approximated by the M/M/1 model, i.e.,

$$D_{queue} = \frac{1}{C_l - f_l}, \quad f_l < C_l$$

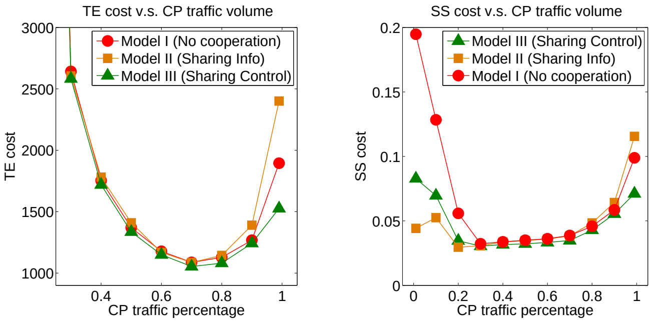
with a linear approximation when the link utilization is over 99%. We relax the link capacity constraints in both TE and SS and penalize traffic overshooting the link capacity with high costs. The shapes of the TE link cost function and queuing delay function are illustrated in Figure 4.



(a) TE link cost

(b) Link queuing delay

Figure 4: ISP and CP cost functions.



(a)

(b)

Figure 5: The TE-SS tussle v.s. CP's traffic intensity (Abilene topology)

Note that we intensionally choose the cost functions of TE and SS to be similar. This allows us to demonstrate the efficiency loss of Model I and Model II even when their objectives are not too conflicting, as well as the significant improvement brought by Model III.

7.2 Evaluation Results

7.2.1 *Tussle between background and CP's traffic*

We first demonstrate how CP's traffic intensity affects the overall network performance. We fix the total amount of traffic and tune the ratio between background traffic and CP's traffic. We evaluate the performance of different models when CP traffic grows from 1% to 100% of the total traffic. Figure 5 illustrates the results on Abilene topology.

The general trend of both TE and SS objectives for all three models is that the cost first decreases as CP traffic percentage grows, and later increases as CP's traffic dominates the network. The decreasing trend is due to the fact that CP's traffic is self-optimized by selecting servers close to a user, offloading the network. The increasing trend is more interesting, suggesting that when a higher percentage of total traffic is CP-generated, the negative effect of TE-SS interaction is amplified, even when the ISP and the CP share similar cost functions. Low link congestion usually means low end-to-end latency, and vice versa. However, their differ in the following: (i) TE might penalize high utilization before queueing delay becomes significant in order to leave as much room as possible to accommodate changes in traffic, and (ii) CP considers both propagation delay and queueing delay so it may choose a moderately-congested short path over a lightly-loaded long path. This explains why the optimization efforts of two players are at odds. As we will show in the following, such delicate differences also result in scenarios similar to what we found in the paradox of extra information.

7.2.2 *Network congestion v.s. performance improvement*

We now study the network conditions under which more performance improvement is possible. We evaluate three models on the Abilene topology. Again, we fix the total amount of traffic and vary the CP's traffic percentage. But we change link capacities and evaluate two scenarios: when the network is moderately congested and when the network is highly congested. We show the performance improvement of Model II and Model III over Model I (in percentages) and plot the results in Figure 6. Figure 6(a-b) show

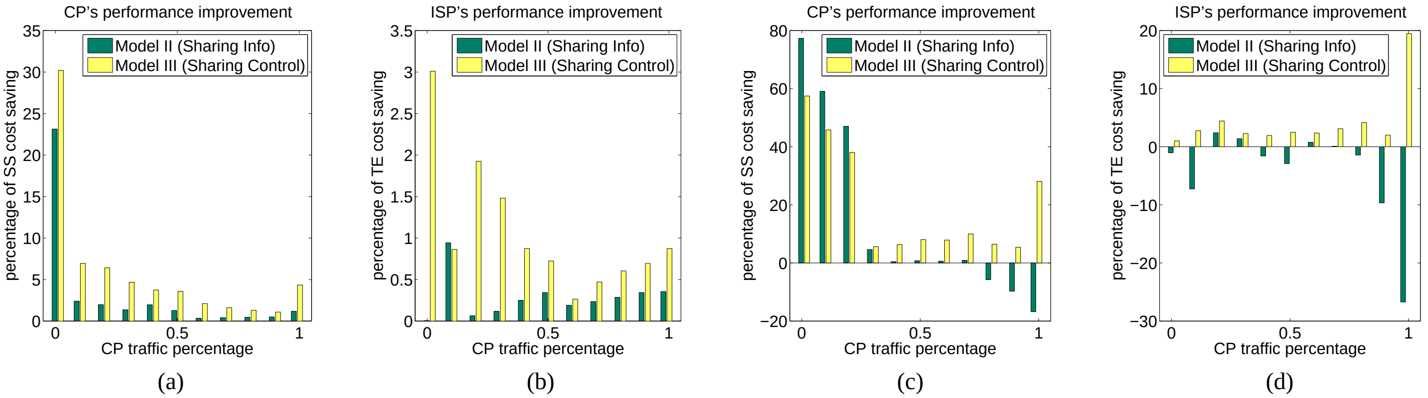


Figure 6: TE and SS performance improvement of Model II and III over Model I. (a-b) Abilene network under low traffic load: moderate improvement; (c-d) Abilene network under high traffic load: more significant improvement, but more information (in Model II) does not necessarily benefit the CP and the ISP (the paradox of extra information).

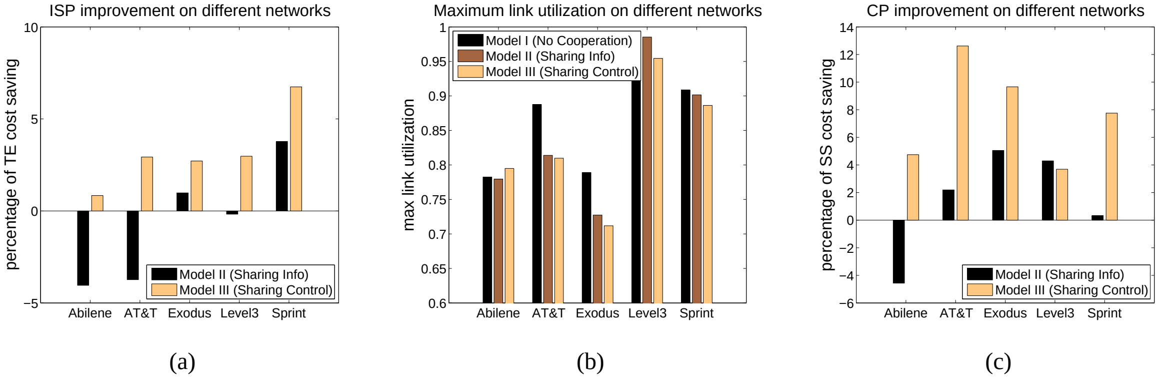


Figure 7: Performance evaluation over different ISP topologies. Abilene: small cut graph; AT&T, Exodus: hub-and-spoke with shortcuts; Level 3: complete mesh; Sprint: in between.

the improvement of the ISP and the CP when the network is under low load. Generally, Model II and Model III improve both TE and SS, though Model III outperforms Model II in almost all cases. However, except the case when the CP's traffic is little (1%), the CP's improvement is not significant. The ISP's improvement is not significant for any amount of CP traffic (note the different scales of *y*-axes). This is because when the network is under low load, the slopes of TE and SS cost functions are "flat," thus leaving little space for improvement. Figure 6(c-d) show the results when the network is under high load. Improvement becomes more significant, especially at the two extremes: when CP's traffic is little and is prevalent. However, we also observe that sometimes Model II performs worse than Model I: both the ISP and the CP do not benefit with more information.

7.2.3 *Impact of ISP topologies*

We evaluate the three models on different ISP topologies. The topological properties of different graphs are discussed earlier in the simulation setup. The CP's traffic is 80% of the total traffic and link capacities are set such that networks are under high traffic load. Our findings are depicted in Figure 7. Note that performance improvement is relatively more significant in more complex graphs. Simple topologies with small min-cut sizes are networks where the paradox is more likely to occur. Besides the TE and SS objectives, we also plot the maximum link utilization, which is another important metric that measures a network's congestion. Observe that the direction of change of the max link utilization metric does not always coincide with that of the TE metric. This suggests that Model I and Model II are also sensitive to a careful choice of the objective functions, while Model III is more robust under different objective models.

8. RELATED WORK

In [12], the authors show that selfish routing is close to optimal in Internet-like environments, while our work explores the optimality of strategic content distribution under the interaction with traffic engineering. [23] studies the problem of load balancing by overlay routing, and how to alleviate race conditions among multiple co-existing overlays. [24] studies the resource allocation problem at inter-AS level where ISPs compete to maximize their revenues. [25] leverages Nash bargaining solution to solve an inter-domain ISP peering problem. These pieces of work studied the interaction within ISPs or CPs themselves, but do not consider the tussle between network providers and content providers.

The need for cooperation between content providers and network providers is raising much discussion in both the research commu-

	CP no change	CP change
ISP no change	current practice	partial collaboration
ISP change	partial collaboration	joint system design

Table 5: To cooperate or not: possible strategies for content provider (CP) and network provider (ISP)

nity and the industry. [26] leverages price theory to reconcile the tussle between peer-assisted content distribution and ISP's resource management. [5] proposes a communication portal between ISPs and P2P applications so that both parties gain from cooperation. These pieces of work represent the approach of sharing information on one of the four dimensions as shown in Table 5. The possibility of sharing control has been unfortunately neglected.

Recent work [27] studied a similar problem on the interaction between content distribution and traffic engineering. Similar results on the possibility of the global optimality are reported, and their models are extended to multiple-ISP cases. This paper is a major extension of an earlier workshop paper [28], because we both qualitatively and quantitatively analyze conditions for optimality and efficiency loss. For example, the paradoxical example and its implication, the solution with distributed implementation, and large scale simulations were absent in the earlier version.

9. CONCLUSION AND FUTURE WORK

In this work, we examine the interplay between traffic engineering and content distribution. Though the problem has long existed, the dramatically increasing amount of content-centric traffic, e.g., CDN and P2P traffic, makes it more significant than ever. With the strong motivation for ISPs to provide content services, they are faced with the question of whether to stay with the current design or to adopt a joint system design. This work sheds light onto possible cooperations between CPs and ISPs.

This paper serves as a starting point of our future work in better understanding the interaction between ISPs and CPs. Traditionally, ISPs provide and operate the pipes, while content providers distribute contents over the pipes. In terms of what information can be shared between ISPs and CPs and what control can be jointly designed, there are four general categories as summarized in Table 5. The top left corner is the current practice, which may give an undesirable Nash equilibrium. The bottom right corner is the joint design, which achieves optimal operation points. The top right corner is the case where the CP receives extra information and adapts control accordingly, and the bottom left corner is the case of content-aware networking. This paper studies all four cases except content-aware networking. Starting from the current practice, to move along either direction of Table 5 when the two parties remain separate business entities would require unilaterally-actionable, backward-compatible, and incrementally-deployable migration paths yet to be discovered.

10. REFERENCES

[1] W. B. Norton, "Video Internet: The Next Wave of Massive Disruption to the U.S. Peering Ecosystem," Sept 2006. Eqinix white paper.

[2] AT&T, "U-verse." http://uverse.att.com/.

[3] Verizon, "FiOS." http://www.Verizon.com/fios/.

[4] A.-J. Su, D. R. Choffnes, A. Kuzmanovic, and F. E. Bustamante, "Drafting behind Akamai (Travelocity-based detouring)," *Proceedings of ACM SIGCOMM*, 2006.

[5] H. Xie, Y. R. Yang, A. Krishnamurthy, Y. Liu, and A. Silberschatz, "P4P: Provider Portal for (P2P) Applications," in *Proceedings of ACM SIGCOMM*, 2008.

[6] J. F. Nash, "The bargaining problem," *Econometrica*, vol. 28, pp. 155–162, 1950.

[7] B. Fortz and M. Thorup, "Internet traffic engineering by optimizing OSPF weights," in *Proceedings of IEEE INFOCOM*, pp. 519–528, 2000.

[8] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, and J.McManus, "RFC 2702: Requirements for Traffic Engineering Over MPLS," Sept. 1999.

[9] D. Xu, M. Chiang, and J. Rexford, "Like-state routing with hop-by-hop forwarding can achieve optimal traffic engineering," in *INFOCOM*, 2008.

[10] J. Wardrop, "Some theoretical aspects of road traffic research," *the Institute of Civil Engineers*, vol. 1, no. 2, pp. 325–378, 1952.

[11] T. Roughgarden and Éva Tardos, "How bad is selfish routing?," *J. ACM*, vol. 49, no. 2, 2002.

[12] L. Qiu, Y. R. Yang, Y. Zhang, and S. Shenker, "On selfish routing in Internet-like environments," in *Proceedings of ACM SIGCOMM*, pp. 151–162, 2003.

[13] M. Littman and J. Boyan, "A distributed reinforcement learning scheme for network routing," Tech. Rep. CMU-CS-93-165, Robotics Institute, Carnegie Mellon University, 1993.

[14] M. J. Osborne and A. Rubinstein, *A Course in Game Theory*. MIT Press, 1999.

[15] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Prentice Hall, 1989.

[16] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[17] K. Binmore, A. Rubinstein, and A. Wolinsky, "The Nash bargaining solution in economic modelling," *RAND Journal of Economics*, vol. 17, pp. 176–188, 1986.

[18] D. P. Palomar and M. Chiang, "A tutorial on decomposition methods for network utility maximization," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, pp. 1439–1451, 2006.

[19] J. He, R. Zhang-Shen, Y. Li, C.-Y. Lee, J. Rexford, and M. Chiang, "DaVinci: Dynamically Adaptive Virtual Networks for a Customized Internet," in *CoNEXT 08*, Dec 2008.

[20] N. Spring, R. Mahajan, D. Wetherall, and T. Anderson, "Measuring ISP topologies with Rocketfuel," *IEEE/ACM Trans. Netw.*, vol. 12, no. 1, pp. 2–16, 2004.

[21] "Abilene." http://www.internet2.edu.

[22] M. Roughan, M. Thorup, and Y. Zhang, "Performance of estimated traffic matrices in traffic engineering," *SIGMETRICS Perform. Eval. Rev.*, vol. 31, no. 1, pp. 326–327, 2003.

[23] W. Jiang, D.-M. Chiu, and J. C. S. Lui, "On the interaction of multiple overlay routing," *Perform. Eval.*, vol. 62, no. 1-4, pp. 229–246, 2005.

[24] S. C. Lee, W. Jiang, D.-M. C. Chiu, and J. C. Lui, "Interaction of ISPs: Distributed resource allocation and revenue maximization," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 2, pp. 204–218, 2008.

[25] G. Shrimali, A. Akella, and A. Mutapcic, "Cooperative interdomain traffic engineering using Nash bargaining and decomposition," in *Proceedings of IEEE INFOCOM*, 2007.

[26] M. J. Freedman, C. Aperjis, and R. Johari, "Prices are right: Managing resources and incentives in peer-assisted content distribution," in *IPTPS 08*, Feb. 2008.

[27] D. DiPalantino and R. Johari, "Traffic engineering vs. content distribution: A game theoretic perspective," Tech. Rep. 08-09-1129-44, Department of management science and engineering, Stanford University, 2008.

[28] Workshop paper, not disclosed due to anonymity.

[29] L. P. Kaelbling, M. L. Littman, and A. P. Moore, "Reinforcement learning: A survey," *Journal of Artificial Intelligence Research*, vol. 4, pp. 237–285, 1996.

[30] L. Qiu, Y. R. Yang, Y. Zhang, and H. Xie, "On self adaptive routing in dynamic environments - an evaluation and design using a simple, probabilistic scheme," in *Proceedings of IEEE ICNP*, 2004.

APPENDIX

A. DISTRIBUTED ALGORITHM FOR GREEDY SERVER SELECTION

In this section, we leverage Q-learning [29] in reinforcement learning to simulate the CP's server selection in model I. Basically, it is a distributed solution that drives the decision to the Wardrop equilibrium, as we will define later. Though it is not directly optimizing the objective function of (3), it is a distributed algorithm that is easily implementable and resembles the solution of many content providers today [4]. In fact, [12] points out that selfish-routing is close to optimal in Internet-like environments [12].

The application of Q-learning in network routing has been studied in some literatures [13] [30]. [30] applied Q-learning and proposed a probabilistic multi-path routing scheme that learns good routes adaptively. We model the CP's server selection as a similar Q-learning scheme that learns good servers adaptively, which we call Q-SS. Due to limited space, we only sketch the basic idea. Readers can refer to our technical report for more details.

We next define the algorithm more rigorously. Every user t is associated with a vector Q_t, where the s-th component $Q_t(s) \in [0,1]$ is the proportion of t's traffic demand that is served by server s. Hence, $\sum_{s\in S(t)} Q_t(s) = 1$. The basic idea of Q-SS works as follows. Initially, Q_t is set arbitrarily. Then it is updated according to the perceived end to end delay. Namely, if the delay from s to t is larger than the average perceived delay over all servers, $Q_t(s)$ is decreased so better servers will serve more demand. Otherwise, $Q_t(s)$ is increased. We show that server selection based on measured delay as above will reach a *Wardrop equilibrium* [10].

Let $D_{s,t}$ denote the end-to-end delay from s to t. The average perceived delay by user t is

$$D_t = \sum_{s\in S} Q_t(s)\cdot D_{s,t} \qquad (18)$$

We have that

Definition 2. *$Q_t, \forall t \in T$, is a Wardrop equilibrium, if $\forall s, s', s'' \in S$, $Q_t(s) > 0$, $Q_t(s') > 0$, and $Q_t(s'') = 0$ imply that $D_{s,t} = D_{s',t} \le D_{s'',t}$.*

Intuitively, at the equilibrium point, any server should have the same delay to a user, if the service rate is non-zero. And the delay is smaller than those with zero server rate. It turns out that the equilibrium point can be viewed as the solution to the following optimization problem, as studied in [11]:

$$\begin{aligned} \text{minimize} \quad & \sum_{l\in E}\int_0^{f_l} D_l(u)du && (19)\\ \text{subject to} \quad & \sum_s x_{st} = M_t,\ \forall t \\ \text{variable} \quad & x_{st} \ge 0 \end{aligned}$$

where $D_l(\cdot)$ is the link delay function, which consists of queueing delay and propagation delay. If $D_l(\cdot)$ is a strictly convex function, (19) has a unique optimal solution and hence, a unique Wardrop equilibrium. One can directly solve (19) to obtain the equilibrium point of greedy server selection.

The Q-SS algorithm proceeds in the following steps. Let $\tilde{D}_{s,t}$ be the measured end-to-end delay from s to t at some time, which can be obtained from active or passive probes. Whenever receiving a new measurement $\tilde{D}_{s,t}$, SS estimates new end-to-end delay in the future would be:

$$D_{s,t} = (1-\alpha)\cdot D_{s,t} + \alpha\cdot\tilde{D}_{s,t} \qquad (20)$$

```
Q-SS algorithm: Initialize $Q_t(s) = 1/|S(t)|$, $D_{s,t} = \tilde{D}_{s,t}$.
Repeat:
    Measure the end-to-end delay from server $s$ to user $t$ as $\tilde{D}_{s,t}$.
    Update the estimate of $D_{s,t}$ according to (20).
    Compute the average delay for user $t$ according to (21).
    Update $Q_t$ according to (22).
    Project $Q_t$ to $[0,1]^{|S(t)|}$ probability space.
Until $Q_t$ converges.
```

Table 6: SS Q-learning protocol with incomplete information.

wherein α is the delay learning factor. The average perceived delay by user t over all servers is

$$D_t = \sum_{s\in S(t)} Q_t(s)\cdot D_{s,t} \qquad (21)$$

Then we update Q_t as follows:

$$Q_t(s) = Q_t(s) + \beta[(D_t - D_{s,t})/D_t] \qquad (22)$$

where β is called the SS learning factor.

After the update, some entries may become negative or greater than one. Hence, Q_t is projected onto $[0,1]^{|S(t)|}$ to ensure that it is a valid probability vector.

Finally, we can use the normalized internal routing vector Q_t to compute the real routing vector P_t. Namely,

$$P_t(s) = (1-\varepsilon)Q_t(s) + \varepsilon/|S(t)| \qquad (23)$$

where ε is a small positive constant number. P_t is perturbed from Q_t by adding uniform routing probabilities to it. This is to make sure that all possible servers are probed, in case some entries of Q_t is zero.

Table 6 summarizes the Q-SS algorithm in model I. With appropriate choices of the learning factors α, β, i.e., diminishing step sizes, Q_t will converge to a stable point. So there is no self-oscillation, and SS eventually converges to Wardrop equilibrium.

B. PROOF OF LEMMA 3

Proof: The proof proceeds in two steps. We first show that the Nash equilibrium in the TESS game is also an optimal solution to (6), though not specifying how it is achieved. The basic idea is to check the KKT [16] conditions of TE and SS at Nash equilibrium, and then compare to the KKT condition of (6). Then we show that an algorithm like (4) converges to the Nash equilibrium.

Step I: Consider a feasible solution $\{x_{st}, r_l^{st}\}$ at Nash equilibrium, i.e., each one is the best response of the other. To assist our proof, we define $\phi_l(f_l) = \Phi'(f_l)$ as the marginal cost of link l.

We first show the optimality condition of SS. Let $\phi_{st} = \sum_l \phi_l(f_l)\cdot r_l^{st}$, which denotes the marginal cost of (s,t) pair. By the definition of Nash equilibrium, for any s such that $x_{st} > 0$, we have $\phi_{st} \le \phi_{s't}$ for any $s' \in S$, by inspecting the KKT condition of the SS optimization. This implies that servers with positive rate have the same marginal latency, which is less than those of servers with zero rate. Let $\phi_t = \phi_{st}$ for all $x_{st} > 0$.

We next check the optimality condition of TE. Consider an (s,t) server-user pair. Let δ_{sv} denote the average marginal cost from node s to v, which can be recursively defined as

$$\delta_{sv} = \begin{cases} \sum_{l:(u,v)\in \mathrm{In}(v)} (\delta_{su} + \phi_l)\cdot r_l^{st} / \sum_{l\in \mathrm{In}(v)} r_l^{st} & \text{if } v \ne s \\ 0 & \text{if } v = s \end{cases}$$

The KKT condition of the TE optimization is for $\forall v \in V$, $\forall l = (u,v), l' = (u',v) \in \text{In}(v)$, $r_l^{st} > 0$ implies $\delta_{su} + \phi_l \le \delta_{su'} + \phi_{l'}$. In other words, for any node v, the marginal cost accumulated from any incoming link with positive flow is equal, and less than those of incoming links with zero flow. So we can define $\delta_{sv} = \delta_{su} + \phi_l$, $\forall l = (u,v) \in \text{In}(v)$ with $r_l^{st} \ge 0$.

In fact, $\delta_{st} = \sum_{l:(u,t)} (\delta_{su} + \phi_l) \cdot r_l^{st} = \sum_l \phi_l \cdot r_l^{st} = \phi_{st}$, by inspecting flow conservation at each node and the fact that any (s,t) path has the same marginal latency as observed above. Combining the two KKT conditions together gives us the necessary and sufficient condition for Nash equilibrium:

$$\begin{cases} \delta_{su} + \phi_l \le \delta_{su'} + \phi_{l'} \text{ if } r_l^{st} > 0 & \forall s \in S,\ \forall v \in V,\ \forall l, l' \in \text{In}(v) \\ \delta_{st} \le \delta_{s't}, \text{ if } x_{st} > 0 & \forall s, s' \in S \end{cases} \tag{24}$$

An intuitive explanation is to consider the marginal latency of any path p that is realized by the routing decision. Let $P(t)$ be the set of paths that connect all possible servers and the user t. Let $\phi_p = \sum_{l:l\in p} \phi_l$. A path p is *active* if $r_l^{st} > 0$ for all $l \in p$, which means there is a positive flow between (s,t). Then the above condition can be translated into the following argument: for any path $p, p' \in P(t)$, $\phi_p \le \phi_{p'}$ if p is active. In other words, any active path has the same marginal latency, which is less than those of non-active paths.

On the other hand, we show the KKT condition for (6). Suppose $\{x_l^{st}\}$ is an optimal solution to (6). Similarly, we can define the marginal latency from node s to v as

$$\Delta_{sv} = \begin{cases} \sum_{l:(u,v)\in \text{In}(v)} (\Delta_{su} + \phi_l) \cdot x_l^{st} / \sum_{l \in \text{In}(v)} x_l^{st} & \text{if } v \ne s \\ 0 & \text{if } v = s \end{cases}$$

The KKT condition of (6) is the following:

$$\begin{cases} \Delta_{su} + \phi_l \le \Delta_{su'} + \phi_{l'} \text{ if } x_l^{st} > 0 & \forall s \in S,\ \forall v \in V,\ \forall l, l' \in \text{In}(v) \\ \Delta_{st} \le \Delta_{s't}, \text{ if } x_l^{st} > 0 \text{ for some } l & \forall s, s' \in S \end{cases} \tag{25}$$

One can readily check the equivalence of conditions (24) and (25). To be more specific, suppose $\{x_{st}, r_l^{st}\}$ is a Nash equilibrium that satisfies (24), we can construct $\{x_l^{st}\}$ as discussed the proof of Lemma 2, which one can easily verify that satisfies (25). Vice versa, given an optimal solution $\{x_l^{st}\}$ to (6), one can construct $\{x_{st}, r_l^{st}\}$ as advised Lemma 2, which satisfies (24).

Step II: In the final step, we prove alternate game moves, i.e., ISP and CP iteratively optimize their own objectives, will lead to the optimal point. To see this, the objective in (7) is a Lyapunov function. Since two players share the same objective, and each player move is the best response to the other, the resulting objective value in the game is a decreasing sequence. In addition, the objective of (7) is lower-bounded by the optimal value of (6). So the game sequence will eventually reach the optimal point, which is also a Nash equilibrium.

$\blacksquare$