Cooperative Content Distribution and Traffic Engineering

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 *server selection* (to match servers with subscribers) and *traffic engineering* (to select efficient routes for the traffic). In this work, we utilize a mathematical framework to show that separating server selection and traffic engineering leads to a sub-optimal equilibrium, even when the CP is given accurate and timely information about network conditions. Leveraging ideas from cooperative game theory, we propose that the system implements a Nash bargaining solution that significantly improves the fairness and efficiency of the joint system. This study is another step toward a systematic understanding of the interactions between those who generate and distribute content and those who provide and operate networks.

Categories and Subject Descriptors

C.4 [**Performance of Systems**]: [Performance attributes]

General Terms

Design, Economics, Performance

1. INTRODUCTION

Traditionally, Internet Service Providers (ISPs) and content providers (CPs) are independent entities. ISPs only provide connectivity, or the bandwidth "pipes" to transport content. As in most transportation businesses, connectivity and bandwidth become commodities and ISPs find their profit margin getting increasingly diminished [11]. At the same time, content providers generate revenue by utilizing existing connectivity to deliver content to ISPs' customers, who are also consumers of the transported content. This motivates ISPs to host and distribute content to their customers using their own infrastructure. Content can be enterprise-oriented like web-based services, or residential-based like triple play as in AT&T's U-Verse [1] and Verizon's FiOS [16] deployments. Provisioning connectivity and content services like IPTV in a provider network is becoming a trend and presents new challenges to the architectural design of an ISP's network.

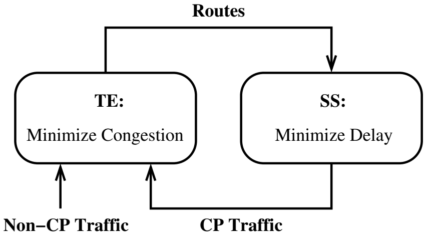


Figure 1: The interaction between traffic engineering (TE) and server selection (SS).

If an ISP were to take advantage of its own network and provide content to their customers, it needs to build a content distribution infrastructure. In practice, many content providers replicate content over a number of strategically placed servers, and direct requests to different servers to balance load and to decrease response time. Typical examples include YouTube, and content distribution networks like Akamai. Such an architecture offloads both the central server which generates the content and the network if, for example, the connections use short and lightly-loaded paths. Because of these merits and popularity, it turns out to be a promising architecture should an ISP distribute content over its network. By looking into these legacy systems, it also allows us to discover tussles between content distribution and network management today.

Nowadays, both the ISPs and the CPs try to optimize their performance. An ISP solves the *traffic engineering* (TE) problem, i.e., picking routes for the offered traffic, often to minimize the chance of having congestion in the network, so that the traffic experiences low packet drops and low latency, and that the network can gracefully absorb flash crowds. The CP solves the *server selection* (SS) problem, i.e., determining which servers direct traffic to each client. Usually there is enough aggregate server capacity to meet user requests, so the goal of SS is often to minimize network delay, so as to reduce user waiting time and to increase throughput. 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. This interaction is illustrated by Figure 1.

Note that the two optimization problems happen on different timescales. The ISP runs traffic engineering at the timescale of hours, while the CP runs dynamic server selection at the timescale of content delivery decision, usually seconds or minutes. One can

NetEcon'08, August 22, 2008, Seattle, Washington, USA.
Copyright 2008 ACM 978-1-60558-179-8/08/08 ...$5.00.

assume that SS has reached its steady state before TE recomputes routing, and that the TE routing change is instantaneous. The degrees of freedom are also "mirror-image" of each other: the ISP controls route selection, which is constant in the SS problem, while the CP controls server selection and therefore the CP's traffic, which is the constant parameter in the TE problem.

The goals of TE and SS are similar, because low link congestion usually means low end-to-end delay, and vice versa. But they are not the same, because (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. So there could be a tradeoff between the traffic-engineering objective and the server-selection objective. When the TE problem and the SS problem are solved separately, they can be modeled as playing a game where they will likely settle in a Nash equilibrium, which may not be optimal. When an ISP runs a content distribution service, it has the option of doing a joint design of TE and SS so that a global optimum can be achieved.

In this paper we study how an ISP should change the way it manages traffic to accommodate the CP, regardless of whether they are the same business entity or not. We consider three scenarios with increasing amount of cooperation between traffic engineering and server selection:

1. **Model I, current practice:** TE measures traffic and SS measures path delay. TE ignores the fact that traffic is variable and can be affected by its routing decisions, and CP has no information on the topology or routing, and therefore has no means of predicting the effect of its own actions.

2. **Model II, sharing information (network providing more information to the CP):** There are in general four types of information that can be shared: (i) physical topology information, e.g., P4PWG [18], (ii) connectivity information, e.g., routing in the ISP network, (iii) dynamic properties of links, e.g., link weight, background traffic, congestion level, (iv) dynamic properties of nodes, e.g., bandwidth and processing power that can be shared by the node. Our work focuses on type (ii) information, and studies whether it helps improve the performance of both parties.

3. **Model III, sharing control: (joint design of SS and TE):** A joint design can guarantee to provide a Pareto-optimal solution for the whole system. In particular, we study the *Nash Bargaining Solution* [10], which is Pareto optimal and can adjust to varying network conditions automatically.

The rest of the paper is organized as follows. Section 2 reviews the standard traffic-engineering model. Section 3 gives two models for the CP's server selection. The first is modeled by selfish routing and achieves Wardrop equilibrium; the second is modeled by optimal routing and is an improvement over the first. Section 4 studies the interaction between TE and SS by allowing them to play a game and reach a Nash equilibrium. We show the performance of Nash equilibria and compare them to the Pareto optimal curve. Section 5 discusses how to jointly optimize traffic engineering and server selection and we propose that the system implement the Nash Bargaining Solution. Section 6 presents related work. Finally, Section 7 concludes the paper and discusses our future work.

2. NETWORK MODEL AND TE

In this section we describe the network model and formulate the optimization problem that TE solves. We also start introducing the notation used in this paper. Note that the models in this section simply follow well-established formulations, and hence are not novel.

Consider a network represented by graph $G = (N, E)$, where N 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. A flow is from any node to any node: $v = (a, b)$ where $a, b \in N$. Let x_v or $x_{a,b}$ denote the rate of flow v. Note that we use both notations interchangeably in the remainder of this paper. Flows are carried by 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, and $w_{pl} = 0$ otherwise. 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 some traffic demand, traffic engineering changes routing in order to minimize network congestion. In practice, network operators control routing either by changing OSPF link weights [4] or by setting up MPLS paths [2]. In this paper we use the multicommodity flow solution to route traffic, because a) it is optimal, i.e., it gives the routing with minimum congestion and can provide a benchmark for all TE schemes, and b) it can be realized by routing protocols that use MPLS tunneling, or as recently shown, distributedly by a link-state routing protocol with hop-by-hop packet forwarding [19]. Formally, let $f_l^v \in [0, 1]$ denote the proportion of traffic of flow v that traverses link l. To realize the multi-commodity flow solution in a network, it can be interpreted as having a number of paths for each flow and splitting the flow among the paths. Let $F = \{f_l^v\}$ be the flow matrix, let the path matrix W include all the active paths, and let $H = \{h_{vp}\}$ be the splitting matrix, i.e., h_{vp} is the fraction of flow v that traverses path p. Then we have $F = HW$.

Let y_l denote the total traffic traversing link l, and we have $y_l = \sum_v x_v \cdot f_l^v$. A standard model for congestion cost is to use a convex increasing function of the link load, represented by $g(y, c)$. The exact shape of the function $g(\cdot)$ is not important in this work, and we use the same piecewise linear cost function as in [4].

Now traffic engineering can be formulated as the following optimization problem:

TE(x_v, c_l):

$$\begin{array}{lll} \text{minimize} & TE = \sum_l g(y_l, c_l) & (1) \\ \text{subject to} & \sum_{i:l=(i,j)} f_l^v - \sum_{i:l=(j,i)} f_l^v = I_{j=b},\ \forall v = (a,b), j \neq a & \\ & y_l = \sum_v x_v \cdot f_l^v,\ \forall l & \\ & 0 \leq f_l^v \leq 1,\ \forall v, l & \\ \text{variables} & f_l^v & \end{array}$$

where $I_{j=b}$ is an indicator function which equals 1 if $j = b$ and 0 otherwise.

Since the cost function $g(\cdot)$ is convex and the constraints are linear, 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.

Even though traffic engineering measures the total traffic and assumes that it is fixed, the two different types of traffic, CP's and background, are modeled differently. Let $S \subset N$ denote the set of CP's servers which store the same content and let $T \subset N$ denote the set of users who want to receive content from the servers. A user $t \in T$ wants to download content (such as streaming video) at rate M_t. Since every server has a copy of this content, the user can download it from any server. In fact, it can even download from multiple servers at the same time. With coordination, the user can receive different substreams from different servers so that its total receiving rate is satisfied. Let $x_{s,t}$ denote the traffic rate from server s to user t, then we must have

$$\sum_{s\in S} x_{s,t} \geq M_t.$$

We assume background traffic is normal unicast between the users and use (i,j), $i,j \notin S$, to denote the source and destination. Let $x_{i,j}$ denote the rate of the background flow from node i to node j.

3. CONTENT PROVIDER MODEL AND SS

Once the users' demand rates are satisfied, one of the major goals in server selection is to minimize delay. Delay is an important metric in applications like live video streaming, which is becoming prevalent on the Internet today. We model server selection such that the average user-perceived delay is minimized, and this model can be directly extended to other objectives such as minimizing maximum user delay.

Let d_p denote the delay of path p, which includes propagation and queueing delay. The queueing delay of a link is an increasing function of the total traffic on the link, which includes both the CP's traffic and background traffic. The CP wants to minimize the content traffic delay experienced by its users. We can model this by minimizing the average user delay, i.e., the sum of end-to-end user delay weighted by the proportion of its traffic.

To optimize user experience, the CP solves the following optimization problem
SS(M_t, d_p)**:**

$$\begin{array}{ll} \text{minimize} & SS = \sum_{v=(s,t)} x_v \sum_p H_{vp} d_p / \sum_t M_t \qquad (2) \\ \text{subject to} & \sum_{s\in S} x_{s,t} \geq M_t, \ \forall t \\ & x_{s,t} \geq 0, \ \forall s,t \\ \text{variables} & x_{s,t} \end{array}$$

We consider two different cases: when CP has no explicit information from the network and optimizes based on measured delay, and when the CP has complete network information so it can achieve optimal server selection. They correspond to the SS model in our model I and model II, respectively. Note that Model I doesn't give the optimal solution in general, while model II achieves the optimum.

3.1 SS Based on Measured Delay

In current practice, the CP has no access to the ISP's information, such as routing matrix, topology, link latency, etc. Hence, it cannot solve (2) directly, and must infer network conditions through measurements. One piece of information directly available to the CP is the end-to-end delay from a server to an end user. Examples of server selection based on end-to-end information include content distribution networks like Akamai, which selects servers mainly based on delay [15]. In practice, the CP usually assigns a user to the nearest server, where the notion of closeness refers to network delay. We can think of dynamic server selection as a variant of selfish routing [13] [12]. For instance, the CP measures the delay from all available servers to a user, and updates the traffic rate $x_{s,t}$ accordingly. Intuitively, the rate is increased if a server shows better performance, i.e., lower delay than the expectation over all servers, and decreased otherwise. Note that this is actually a greedy algorithm, which is not optimal in general. One can view this as a baseline of how well a CP can do without extra effort to infer more about the network. The benefits of such inferences will be bounded by the optimal solution (2).

In fact, server selection based on measured delay as above will reach a *Wardrop equilibrium* [17]. Intuitively, at the equilibrium point, any server should have the same delay to an end user, if the service rate is non-zero, and such delay should be smaller than that of servers with zero rate. It turns out that the equilibrium point can be viewed as the solution to a global convex optimization problem, as studied in [13].

In this work, we leverage reinforcement learning, i.e., Q-learning [8], to simulate how the CP selects servers adaptively. Basically, it is a distributed solution that drives the decision to the Wardrop equilibrium. Though it is not directly optimizing the objective function of (2), it is a distributed algorithm that is easily implementable and resembles the solution of many content providers today [15]. Readers can refer to the technical report [7] for more details.

3.2 Optimal SS with Complete Information

Now envision that CPs are able to obtain information from the ISP, for instance, by performing more accurate measurement and applying better inference algorithms, to improve the end user experience. In the best case, the CP is able to obtain the complete information about the network, i.e., routing matrix and link latency. Such situation is characterized by problem (2), which is the optimal performance the CP can achieve without further cooperation with the ISP.

Recall that the objective in (2) is the average user perceived delay. Let $\hat{y}_l$ denote the aggregate CP's traffic on link l, then we can rewrite the objective by summing over all links

$$SS = \sum_l \hat{y}_l \cdot d_l = \sum_{v=(s,t)} x_v \sum_l f_l^v \cdot d_l$$

We have $d_l = p_l + q_l$, where p_l is the constant propagation delay, and q_l is the queuing delay. Queueing delay q_l is a function of the load y_l on link l, which includes both CP's and background traffic. One common approximation is to use the M/M/1 queueing delay:

$$q_l = \frac{1}{c_l - y_l}, \quad y_l < c_l$$

Because this function has a singularity point at $y_l = c_l$, which does not work well with standard optimization solvers, we relax the condition $y_l < c_l$ and replace the segment $y_l > 0.99c_l$ with a linear function matching the slope at $y_l = 0.99c_l$.

While the optimal SS allows a distributed solution, we solve (2) centrally in our numerical simulation, since we are more interested in the performance improvement brought by cooperation. A practical server selection protocol is therefore not within the scope of our discussion here.

4. TE-SS INTERACTIONS

In this section, we explore the interaction between the ISP and the CP when they are operated independently without a coordinated design. We study the interplay in a game-theoretic framework, and evaluate its behavior and efficiency via simulation.

4.1 TE-SS game

To model the interplay between TE and SS, we start with a two-player non-cooperative Nash game. The CP and the ISP are the two players. The ISP's decision variable is the routing variable f, and the CP's decision variable is server-user traffic rates $\{x_{s,t}\}_{s\in S, t\in T}$.

Their utility functions can be viewed as the negative of the objectives in (2) and (1), respectively. Consider the CP and the ISP take turns to optimize their own networks, given the decision variable of the other player. More specifically,

$$f^{(i+1)} = \underset{f}{\text{argmin}}\ TE(x_{s,t}^{(i)}) \tag{3}$$

$$x_{s,t}^{(i+1)} = \underset{x_{s,t}}{\text{argmin}}\ SS(f^{(i+1)}) \tag{4}$$

Next we explore the interaction under the two SS models discussed earlier, and evaluate the issue of stability and efficiency. In particular, we seek to answer the following questions. First, does there exist a Nash equilibrium? Second, does the trajectory of iterative optimization of TE and SS lead to Nash equilibrium? Third, what is the stable operating region of the system and how is the performance tradeoff reflected on the Pareto curve? Last, how much efficiency does the system lose due to lack of coordination?

4.2 Pareto optimality

To measure efficiency in a system with multiple objectives, one needs to explore the operating region of the system. In particular, the Pareto surface characterizes the tradeoffs of conflicting goals. One way to trace the tradeoff curve is to optimize a weighted sum of the two objectives:

$$\begin{array}{ll} \text{minimize} & TE + \alpha \cdot SS \\ \text{variables} & f \in \mathscr{F},\ \ x_{s,t} \in \mathscr{X}_{cp} \end{array} \tag{5}$$

Here $\alpha \geq 0$ is a scalar representing the relative weight of the two objectives. $\mathscr{F}$ and $\mathscr{X}_{cp}$ are the domains for the variables:

$$\begin{aligned} \mathscr{F} &= \{0 \leq f_l^v \leq 1,\ \forall v, l : \sum_{i:l=(i,j)} f_l^v - \sum_{i:l=(j,i)} f_l^v = I_{j=b}, \\ &\qquad \forall v = (a,b), j \neq a\} \\ \mathscr{X}_{cp} &= \{x_{s,t} \geq 0,\ \forall s \in S, t \in T : \sum_{s \in S(t)} x_{s,t} \geq M_t, \text{and } \forall t\} \end{aligned}$$

If we vary α and solve each problem (5), we obtain the *Pareto curve*, i.e., the points on this curve are such that we cannot decrease one objective further without increasing the other objective. Later, we will discuss how to weigh the tradeoffs between two objectives and choose a good system operating point.

4.3 Simulation

In this section, we use a simple example to demonstrate the performance of different strategies when operating a content delivery network and a physical network together. First, we show the performance tradeoff of competing objectives in the system, observing the efficiency loss due to the lack of coordination. We then show the performance gains of cooperation between the ISP and the CP. By presenting this toy example, we hope to convey some engineering implications to the system designer who is running a network with a substantial amount of self-adaptive content traffic.

Simulation setup: The topology of the toy network is depicted in Figure 2. The network consists of four nodes, in which $S = \{1,2\}, T = \{3\}$, and eight directed physical links connecting these nodes in a ring. These links have uniform capacities and propagation delays. Two server nodes, node 1 and node 2, can both serve user node 3. There is a background traffic flow from node 4 to node 3. The CP decides how to split load between server node 1 and 2. Traffic engineering decides routing in the network. In this example, every flow has two routes, either clockwise or counter-clockwise on the ring. The background traffic demand is $x_{4,3}$, and the CP's user demand is M_3. We vary these parameters in our simulations.

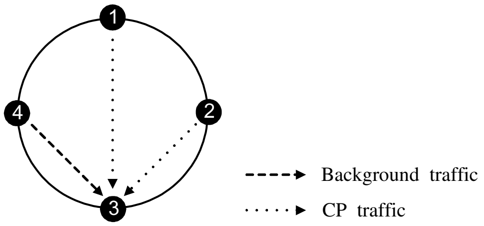
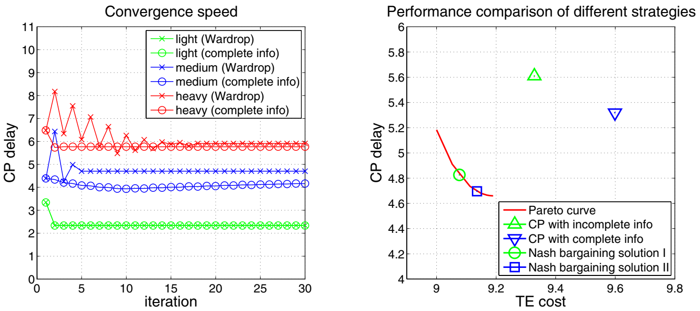


Figure 2: Topology of a toy network to show performance comparisons.



(a) convergence of interplay

(b) comparison of strategies

Figure 3: Interplay of TE and SS: trajectory of convergence.

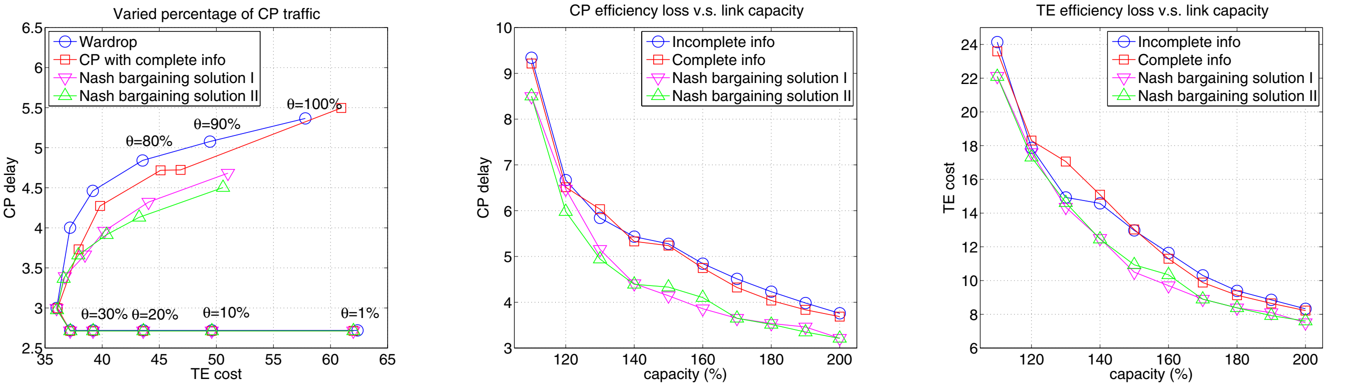
First, we explore the convergence property of the interaction between TE and SS. We show the *SS* objective, i.e., the average delay, as SS and TE are optimized iteratively. Users' traffic demand is set as $M_3 = 3$, and the background traffic is $x_{4,3} = 1$. We vary the link capacity such that the network is under low, medium, high load, i.e., $c_l = 6, 3, 2.5$, so the bottleneck link utilization is $33\%, 66\%, 80\%$, respectively. We start with random routing configuration f and random CP decision variable $x_{s,t}$. We show the Nash equilibrium when the CP has incomplete information, i.e., Wardrop equilibrium, and when the CP has complete information. Results are shown in Figure 3(a).

All curves become flat as the number of iterations increases, which indicates the reach of Nash equilibrium. But the convergence speed may be different. When the network is under low utilization, the interplay quickly reaches equilibrium, since there is less conflict between minimizing congestion and delay. When the network is operating at high load, there is more oscillation, which means that SS and TE's goals are conflicting with each other.

The Pareto curve is computed using methods introduced in (5), and illustrated in two-dimensional space (TE, SS), as shown in Figure 3(b). Two Nash equilibria, when the CP optimizes with incomplete and complete information, are also indicated on the figure. Note that when the CP leverages the complete information to optimize (2), it is able to achieve lower delay, but at the expense of higher TE cost. Though it is not obvious which operating point is better, both equilibria are away from the Pareto curve, which shows there is room for performance improvement for both parties.

5. A JOINT DESIGN

In this section, motivated by the need for a joint TE and SS design, we propose the Nash bargaining solution to reduce the performance gap observed above.



(a) varying the percentage of CP's traffic θ (b) varying the network capacity provisioning (c) varying the network capacity provisioning

Figure 4: Measure of efficiency loss: varying parameters under the same total traffic demand

5.1 Nash bargaining solution

An ISP that provides content services in its own network can jointly optimize traffic engineering goals (minimum congestion) and CP's goals (minimum user delay). However, solving (5) for each α and adaptively tuning α in a *trial-and-error* fashion may be problematic. First, it is hard to weigh the tradeoffs between one objective and the other, and tell which one is more important. Secondly, one needs to repeatedly tune the parameter α and solve an optimization problem every time, to see whether the resulting performance is satisfactory. The reconfiguration cost may be prohibitively high for some live content applications. Third, tuning α to explore a broad region of system operating points is computationally expensive. Usually, exploring a large set of α only produces a small operating region.

From the perspective of economics, a joint design paradigm is also helpful to network providers and content providers who wish to cooperate, which leads to a win-win situation. But they may prefer to keep their functionalities independent, without revealing too much information to each other. Since both want to receive as much benefit as possible, they need to resolve the conflicting goals. Intuitively, one who makes a greater contribution to the collaboration should be able to receive more benefits. Otherwise, he may choose not to cooperate at all. Hence, we borrow the notion of *Nash bargaining solution* [10] [3] in cooperative game theory. The solution concept guides the system designer in choosing an *efficient* and *fair* operating point without much effort to explore α inefficiently.

Let (TE_0, SS_0) be a constant, which we call the *disagreement point*. One can view the disagreement point as the baseline to cooperate, namely, without a joint design, it is the operating point they would end up with. The Nash bargaining solution optimizes the product of performance improvements of the two players:

$$\begin{array}{ll} \text{maximize} & (TE_0 - TE)(SS_0 - SS) \\ \text{variable} & f \in \mathscr{F}, x_{cp} \in \mathscr{X}_{cp} \end{array} \qquad (6)$$

One can view the Nash equilibrium of interplay without coordination as the disagreement point, since it is the status quo before cooperation is suggested.

The choice of Nash bargaining solution is not accidental. It has the following properties that are essential to a system designer's consideration.

- *Pareto optimality*. A NBS is pareto optimal, therefore ensuring efficiency.
- *Symmetry*. The two players should get equal share of the gains by cooperation, if two players have symmetric problem definition, i.e., disagreement point, feasible objective region.
- *Expected utility axiom*. The Nash bargaining solution is invariant under affine transformations. For instance, suppose (TE^*, SS^*) is the Nash bargaining solution when a feasible point is (TE, SS), with (TE_0, SS_0) as the disagreement point. If the disagreement point is shifted and scaled to $(\alpha_1 TE_0 + \beta_1, \alpha_2 SS_0 + \beta_2)$, and the feasible point is transformed to $(\alpha_1 TE + \beta_1, \alpha_2 SS + \beta_2)$, the new Nash bargaining solution becomes $(\alpha_1 TE^* + \beta_1, \alpha_2 SS^* + \beta_2)$. This axiom suggests that the expected performance under all network conditions is still a Nash bargaining solution.
- *Independence of irrelevant alternatives*. This means that adding 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 [10].

In practice, one can also propose other disagreement points as the starting point, which can be thought of as the minimum performance requirement. Here, we use the Nash equilibrium point as the disagreement, since it is known to both the ISP and the CP based on their empirical observation. In addition, we use it as a benchmark to evaluate how much performance improvement can be gained compared to the legacy systems.

5.2 Performance evaluation

We use simulation on the same topology in Figure 2 to demonstrate the performance gains of the Nash bargaining solution and its engineering implications. We evaluate two Nash bargaining solutions. Nash bargaining solution I is the optimal solution of (6) using the Nash equilibrium when CP operates with incomplete information as the disagreement point, and Nash bargaining solution II when CP operates with complete information.

We evaluate performance improvement through two sets of simulations. In the first case, we fix the total amount of traffic, and vary the percentage of CP's traffic θ from 1% to 100%. Four operating points, namely, two Nash equilibria and two Nash bargaining solutions are depicted. The results are shown in Figure 4(a). We make a few observations. When $\theta \leq 40\%$, the four operating points are

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

Table 1: To cooperate or not: possible strategies for content provider (CP) and network provider (ISP)

close to each other, which suggests that the legacy system is doing fine under low load. However, when $\theta \geq 50\%$, the efficiency gap begins to grow. Current practice, i.e., when CP operates with incomplete information, results in the worst performance, as indicated by the top curve. The Nash bargaining solution II, as indicated by the bottom curve, produces the best outcome for both TE and SS. The gap between these two curves shows the performance improvement for the CP.

We have shown that the efficiency loss is nontrivial when the network is highly loaded with CP's traffic. One way for ISP to handle the increasing amount of adaptive traffic is to upgrade the network capacities. To demonstrate this, we fix the amount of background traffic and CP's traffic, i.e., $x_{4,3} = 0.5, M_3 = 3.5$ where the content traffic is dominating, and vary the link capacity provisioning, i.e., the ratio of capacity and traffic rate on the bottleneck link. In Figure 4 (b)(c), we show the efficiency loss under different levels of network utilization. Note that even if we double the capacity provisioning, a constant performance gap still exists, independent of how good the overall performance is.

6. RELATED WORK

In [12], the authors show that selfish routing is close to optimal in Internet-like environments, while our work explores how strategic content distribution interacts with traffic engineering. Recently, Nash bargaining solution is used to solve an inter-domain ISP peering problem in [14]. [6] studies the problem of load balancing through overlay routing, and how to alleviate race conditions among multiple co-existing overlays. Resource allocation at inter-AS level is explored in [9], in which the economics of ISPs' revenue maximization are formulated as a Nash game. These pieces of work studied the interaction between ISPs or CPs themselves, but did not look into the intrinsic tussle between two parties.

The need for cooperation between content providers and network providers is raising much discussion in both the research community and the industry. [5] leverages price theory to reconcile the tussle between peer-assisted content distribution and ISP's resource management. [18] 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 possibilities as we discussed in Section 1. The possibility of sharing control is unfortunately neglected.

7. CONCLUSION AND FUTURE WORK

We study the interplay between content distribution and traffic engineering. Though the problem has long existed, the dramatically increasing amount of content-centric traffic, i.e., CDNs and P2P traffic, makes it more significant than ever. With the strong motivation for ISPs to provide content services, they are faced with the problem of 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 evolution of ISPs and CPs. Traditionally, ISPs provide and operate the pipes, while content providers distribute contents over the pipes, e.g., through CDN or P2P. In terms of both what control can be jointly designed and what information can be shared between ISPs and CPs, there are four general categories as summarized in Table 1. In the top left corner is the current practice, which may be an undesirable Nash equilibrium. In the bottom right corner is the benchmark where the two parties work together, which is one of the subjects studied in this paper. In the top right corner is the case where CPs share information or adapt control with ISPs, and in the bottom left corner is the case of content-aware networking. To move along either direction when the two parties remain separate business entities would require unilaterally-actionable, backward-compatible, and incrementally-deployable migration paths that are yet to be discovered.

8. REFERENCES

[1] AT&T. U-verse. http://uverse.att.com/.

[2] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, and J.McManus. RFC 2702: Requirements for traffic engineering over MPLS, 1999.

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

[4] B. Fortz and M. Thorup. Internet traffic engineering by optimizing OSPF weights. In *Proceedings of IEEE INFOCOM*, pages 519–528, 2000.

[5] M. J. Freedman, C. Aperjis, and R. Johari. Prices are right: Managing resources and incentives in peer-assisted content distribution. In *Proc. 7th International Workshop on Peer-to-Peer Systems (IPTPS08)*, Tampa Bay, FL, Feb. 2008.

[6] W. Jiang, D.-M. Chiu, and J. C. S. Lui. On the interaction of multiple overlay routing. *Perform. Eval.*, 62(1-4):229–246, 2005.

[7] W. Jiang, R. Zhang-Shen, J. Rexford, and M. Chiang. On the interplay between content distribution and traffic engineering. Technical report, Princeton University, 2008.

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

[9] S. C. Lee, W. Jiang, D.-M. Chiu, and J. C. Lui. Interaction of ISPs: Distributed resource allocation and revenue maximization. *IEEE Trans. on Parallel and Distributed Systems*, 19(2):204–218, 2008.

[10] J. F. Nash. The bargaining problem. *Econometrica*, 1950.

[11] W. B. Norton. Video internet: The next wave of massive disruption to the U.S. peering ecosystem. NANOG white paper.

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

[13] T. Roughgarden and E. Tardos. How bad is selfish routing? *J. ACM*, 49(2), 2002.

[14] G. Shrimali, A. Akella, and A. Mutapcic. Cooperative interdomain traffic engineering using Nash bargaining and decomposition. In *Proceedings of IEEE INFOCOM*, Anchorage, AK, 2007.

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

[16] Verizon. FiOS. http://www.Verizon.com/fios/.

[17] J. Wardrop. Some theoretical aspects of road traffic research. *the Institute of Civil Engineers*, 1(2):325–378, 1952.

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

[19] D. Xu, M. Chiang, and J. Rexford. Link-state routing with hop-by-hop forwarding can achieve optimal traffic engineering. In *INFOCOM*, 2008.