Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER

Performance Evaluation 62 (2005) 229–246

PERFORMANCE EVALUATION
An International Journal

www.elsevier.com/locate/peva

On the interaction of multiple overlay routing

Wenjie Jiang[a,*], Dah-Ming Chiu[b], John C.S. Lui[a]

[a] *Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong*
[b] *Department of Information Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong*

Available online 10 August 2005

Abstract

In the past few years, overlay networks have received much attention but there has been little study on the "*interaction*" of multiple, co-existing overlays on top of a physical network. In addition to previously introduced concept of overlay routing strategy such as selfish routing, we introduce a new strategy called "*overlay optimal routing*". Under this routing policy, the overlay seeks to minimize its weighted average delay by splitting its traffic onto multiple paths. We establish that (i) the overlay optimal routing can achieve *better* delay compared to selfish routing and (ii) there exists a Nash equilibrium when multiple overlays adopt this strategy. Although an equilibrium point exists for overlay optimal routing and possibly for selfish routing, we show that the interaction of multiple overlay routing may not be Pareto optimal and that some fairness anomalies of resource allocation may occur. This is worthy of attention since overlay may not know the existence of other overlays and they will continue to operate at this sub-optimal point. We explore two pricing schemes to resolve the above issues. We show that by incorporating a proper pricing scheme, the overlay routing game can be led to the desired equilibrium and avoid the problems mentioned above. Extensive fluid-based simulations are performed to support the theoretical claims.

Keywords: Interaction; Co-existing; Overlay optimal routing; Anomaly; Pricing; Nash equilibrium

[*] Corresponding author.
E-mail addresses: wjjiang@cse.cuhk.edu.hk (W. Jiang), cslui@cse.cuhk.edu.hk (J.C.S. Lui), dmchiu@ie.cuhk.edu.hk (D.-M. Chiu).

0166-5316/$ – see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.peva.2005.07.005

1. Introduction

The fundamental design philosophy of the Internet is to build a robust, highly scalable and evolvable network. To achieve scalability, a hierarchical form of routing is adopted in which routing policies are left to network operators. Static routing, meaning that link costs are kept constant, is used to keep things simple. When traffic demand changes, congestion control at the source regulates the traffic load rather than tries to find an alternate path which has a higher available network bandwidth or better performance. Although such design choices might not achieve the optimal performance, they are considered as the contributing factors to the scalability and robustness, hence the success of Internet.

This form of simple but sub-optimal network design allows a more intelligent traffic routing policy at the application layer. For the past few years, there has been tremendous interest on the routing and deployment of overlay or peer-to-peer networks [2,15]. These networks provide features like resiliency [2] and services such as content delivery and large scale file distribution [1]. In particular, application-layer routing schemes are shown to effectively address the problems of traditional IP routing. Measurements from Refs. [2,15,16] indicate that in the current Internet, not only a large percentage of flows can find a better route by relaying packets with the assistance of overlay nodes, but also able to ensure a higher QoS guarantee. From a theoretic point of view, application layer routing is an optimization problem in which an overlay maximizes its utility based on the available network resources.

Although the concept of overlay networks has received much attention in recent years, there has been little focus on the "*interaction*" of "*co-existence*" of overlay networks. In this work, we consider scenarios in which multiple overlays are constructed on top of a physical network. These overlays may share some physical links or nodes, but they may not realize the existence of other overlays. Each overlay is "*selfish*" by nature in that it performs overlay routing so as to optimize its own performance without considering the impact on other overlays. In particular, we explore this form of interaction and how the interaction can affect the network stability, performance and fairness in resource allocation.

For the traditional overlay routing, the strategy is to select the *best* path for a given flow so that the performance, say end-to-end delay, can be minimized. Note that this form of overlay routing does *not* split the flow among all available paths, but rather selects a path which has the minimum delay. Because the existence of this flow will increase the congestion level of the traversed links, the average delay of this flow is generally not minimized [14]. In Ref. [11], authors present an architectural framework of shared routing underlay so as to offer a query service for all overlays above the physical network. For instance, an overlay can find out the congestion level or delay of a physical link. In Refs. [6,7], authors show that when overlays have the ability to assign traffic among its available paths, it can minimize the average delay. These results suggest the feasibility of implementing an optimal routing scheme for an overlay network.

In this paper, we seek to understand the interaction of multiple overlays. We first propose the "*overlay optimal routing*" policy and show that it has a better performance than classical overlay selfish routing [13,14,18]. Secondly, we model the interaction of overlays as a non-cooperative strategic game and show that when multiple overlays use the overlay optimal routing policy, there always exists a *Nash equilibrium*, under very mild assumptions about the delay functions of the overlay's links and general network topology. Thirdly, we point out that in general, when multiple overlays use "*any*" form of application layer overlay routing policy, the equilibrium point is not Pareto (or social) optimal. More importantly, the interaction may lead to some resource fairness anomaly. That is, at the equilibrium point, it is possible for some overlays to obtain a higher percentage of the common resource (e.g., link bandwidth) as compared to other overlays, and cause these overlays to experience a significant performance degradation. To overcome the performance and fairness issues, we propose two pricing schemes to resolve these issues.

The outline of the paper is as follows. In Section 2, we present the mathematical models for selfish routing and overlay optimal routing. In Section 3, we study the interaction between co-existing overlays as a non-cooperative strategic game, and show that under a general network topology, the overlay optimal routing has a Nash equilibrium. In Section 4, we illustrate that when multiple overlays interact with each other, there exist some potential performance and fairness problems. In Section 5, we present two pricing schemes to solve the above mentioned problems. Related work is given in Section 6 and finally, Section 7 concludes.

2. Mathematical models for overlay routing

We begin this section by formally defining the physical network and various co-existing overlays. We then present the "*overlay optimal routing*" policy and formulate it as a constrained convex optimization problem. We also illustrate the relationship between classical selfish routing and overlay optimal routing.

2.1. Formulation of overlay network routing problem

Consider a *physical network* with a set $\mathcal{J}$ of *resources*, which denotes a set of physical links. For each link $j \in \mathcal{J}$, let C_j represent its finite capacity (unit is bps). The number of links in the network is finite such that $|\mathcal{J}| = m$. Let a *route* r be a non-empty subset of $\mathcal{J}$, and denote $\mathcal{R}$ as the set of all possible routes of the physical network. Since the set $\mathcal{J}$ is finite, the number of possible routes is also finite, say, $|\mathcal{R}| = q$. We use A to represent an $m \times q$ matrix with $A_{jr} = 1$ if $j \in r$, which indicates that link $j \in r$, and $A_{jr} = 0$ otherwise. Thus, the matrix A defines a 0–1 link-route indicator matrix $(A_{jr}, j \in \mathcal{J}, r \in \mathcal{R})$.

An overlay network is a connected subgraph of the underlying physical network which consists of a set of logical nodes and logical links. A logical path is interpreted as a set of logical links, each of which may consist of one or more physical links. The logical topology of the overlay network heavily depends on how this overlay is organized. With proper translation, we can map every logical path to a set of corresponding physical links. Thus, the routing matrix for overlay s can be similarly defined as $A^{(s)}$, which is a partial matrix of A. We allow multiple overlays to co-exist on top of a physical network and these overlays may share some of the physical resources in $\mathcal{J}$. Within an overlay, there can be "*multiple*" source–sink pairs and each source–sink is associated with a traffic flow f, which has a constant traffic demand of x_f (units is bps). Though the overlay's traffic demand may be variant, we seek to simulate a static underlay environment and non-variant overlay traffic, to explore the essential properties and implications due to overlays' interaction in a less dynamic environment. However, we will show in Section 4 that even when each overlay has one single source–sink, there will be undesirable interaction between overlays and the situation can get worse when there are more source–sink pairs within an overlay. For the ease of presentation, we use the term *source–sink* or *flow* interchangeably to denote a particular traffic transmission within an overlay.

Denote $\mathcal{N}$ as a finite set (with $|\mathcal{N}| = n$) which represents all overlays on top of a physical network. Suppose that for each overlay $s \in \mathcal{N}$, there is a finite set $\mathcal{F}_s$ of source–sink pairs. For each flow $f \in \mathcal{F}_s$, there is a set $\mathcal{R}_f$ of different paths (may not be disjoint) that can be used by the flow f to deliver information from its source to its sink, where $\mathcal{R}_f$ is a non-empty subset of all possible paths $\mathcal{R}$ in the physical network. Hence, $\mathcal{R}_f$ contains all possible paths for the flow f from its source to its sink. Let H be an $\sum_{s\in\mathcal{N}} |\mathcal{F}_s| \times q$

matrix and we have $H_{fr} = 1$ if $r \in \mathcal{R}_f$ ($f \in \mathcal{F}_s$, $s \in \mathcal{N}$, $r \in \mathcal{R}$) such that route r serves flow f in overlay s, and we set $H_{fr} = 0$ otherwise. Again, H defines a 0–1 indicator matrix $H = (H_{fr}, f \in \mathcal{F}_s, r \in \mathcal{R})$, specifying all possible routes that can be used by flow f in the overlay s.

Each overlay has the ability to control the routing of its traffic *within* its overlay network. Therefore, source nodes of an overlay network may choose to split and assign their traffic onto different paths so that the weighted average delay of the *whole* overlay network can be minimized. Note that this is different from the traditional overlay routing, because normally a source node in an overlay merely chooses a currently best path from a set of available paths, i.e., minimum end-to-end delay, assigning *all* its traffic along this path. For each flow f in the overlay s, there is a traffic demand x_f (in terms of bps) assigned to the corresponding source–sink pair. The overlay needs to decide, for all its flows, how to assign traffic to every possible path $r \in \mathcal{R}_f$ so as to optimize its desired performance. Thus, each flow f in the overlay s has a routing decision vector $y^{(s,f)} = (y_1^{(s,f)}, y_2^{(s,f)}, \ldots, y_{|\mathcal{R}_f|}^{(s,f)})^{\mathcal{T}}$, where $y_k^{(s,f)}$ is the amount of traffic along path k for flow f in overlay s, and $\sum_{k=1}^{|\mathcal{R}_f|} y_k^{(s,f)} = x_f$ where $|\mathcal{R}_f|$ is the total number of paths available for the flow f. For the compactness of presentation, we rewrite the routing decision for overlay s as a concatenation of the flow vectors of all its source–sink pairs: $y^{(s)} = (y^{(s,f_1)}, y^{(s,f_2)}, \ldots, y^{(s,f_{|\mathcal{F}_s|})})$.

We say that a flow vector $y = (y^{(s)}, s \in \mathcal{N}) = (y^{(1)}, y^{(2)}, \ldots, y^{(n)})$ *supports* traffic rate $x = (x_f, f \in \mathcal{F}_s, s \in \mathcal{N})^{\mathcal{T}}$ if $Hy = x$. In other words, summing the rate y_r on the route r serving the flow f for all possible routes is equal to the overall traffic demand x_f. We call a flow pattern *feasible* if for $y = (y_r, r \in \mathcal{R})$, $y \geq 0$ and $Ay \leq C$, where $C = (C_j, j \in \mathcal{J})$. In other words, the aggregate rate of traffic that traverses link j is no more than the capacity C_j of link j.

2.2. Overlay optimal routing

Let $d_j(l_j)$ denote the delay function for the physical link $j \in \mathcal{J}$, where l_j is the *aggregate* rate of traffic that traverses link j. In this work, we only assume that the delay function is continuous, non-decreasing, and convex. Note that this is a reasonable assumption since this applies to a link with a fixed propagation delay, or a link whose delay is represented by general queueing delay models. The end-to-end delay of a route is the sum of delay for each physical link that comprises this route. For a particular overlay $s \in \mathcal{N}$, the weighted average delay for this overlay s is

$$\text{delay}^{(s)} = \frac{1}{\sum_{f \in \mathcal{F}_s} x_f} \sum_{f \in \mathcal{F}_s} \sum_{k \in \mathcal{R}_f} y_k^{(s,f)} \cdot D_k^{(f)}, \tag{1}$$

where $D_k^{(f)} = \sum_{j \in k} d_j(l_j)$ is the end-to-end delay of path k for flow f. Note that x_f is a given traffic demand, which can be treated as a constant and will not affect the optimization procedure that we will carry out in later sub-section.

Let $L = (l_1, l_2, \ldots, l_m)^{\mathcal{T}}$ denote a traffic rate vector for all physical links in the network. We define a delay function $\mathcal{D}(L) : \mathrm{R}^m \to \mathrm{R}^m$, where for each rate vector L, this function returns a delay vector for all physical links:

$$\mathcal{D}(L) = (d_1(l_1), \ldots, d_j(l_j), \ldots, d_m(l_m))^{\mathcal{T}}, \quad j \in \mathcal{J} \tag{2}$$

where $L = (l_1, l_2, \ldots, l_j, \ldots, l_m)^{\mathcal{T}}, \quad j \in \mathcal{J}$.

Let $A^{(s,f)}$ be an $m \times |\mathcal{R}_f|$ routing matrix for the flow f in the overlay s, of which the definition is similar to that of A. Therefore, $A^{(s,f)}$ is the partial matrix of $A^{(s)}$. For the ease of presentation, we rewrite $A^{(s)} = (A^{(s,f_1)}, A^{(s,f_2)}, \ldots, A^{(s,f_{|\mathcal{F}_s|})})$, and $A = (A^{(1)}, A^{(2)}, \ldots, A^{(n)})$. With these notations, one immediately obtains the following result of representing the traffic rate vector in the physical network:

$$L = \sum_{s\in\mathcal{N}} \sum_{f\in\mathcal{F}_s} A^{(s,f)} y^{(s,f)} = \sum_{s\in\mathcal{N}} A^{(s)} y^{(s)} = Ay. \tag{3}$$

The weighted average delay for an overlay s can be expressed in a compact form as:

$$\text{delay}^{(s)}(y^{(s)}; y^{(-s)}) = \frac{1}{\sum_{f\in\mathcal{F}_s} x_f} \cdot y^{(s)\mathcal{T}} \left[A^{(s)\mathcal{T}} \mathcal{D}(L)\right] = \frac{1}{\sum_{f\in\mathcal{F}_s} x_f} \cdot y^{(s)\mathcal{T}} \left[A^{(s)\mathcal{T}} \mathcal{D}\left(\sum_i A^{(i)} y^{(i)}\right)\right] \tag{4}$$

where y is a feasible flow pattern of vector $(y_r, r \in \mathcal{R}) = (y^{(1)}, y^{(2)}, \ldots y^{(s)}, \ldots, y^{(n)})$ and $y^{(-s)}$ denotes the vector of traffic flows in other overlays except the overlay s.

Finally, the overlay optimal routing for an overlay s can be expressed as a constrained optimization problem that represents the interaction with other overlays in the network. Mathematically, we have:

$$\textbf{OVERLAY}^{(s)} \quad [y^{(s)}; A, H, C, x, y^{(-s)}]: \tag{5}$$

$$\text{Minimize} \quad y^{(s)\mathcal{T}} \left[A^{(s)\mathcal{T}} \mathcal{D}\left(\sum_i A^{(i)} y^{(i)}\right)\right] \tag{6}$$

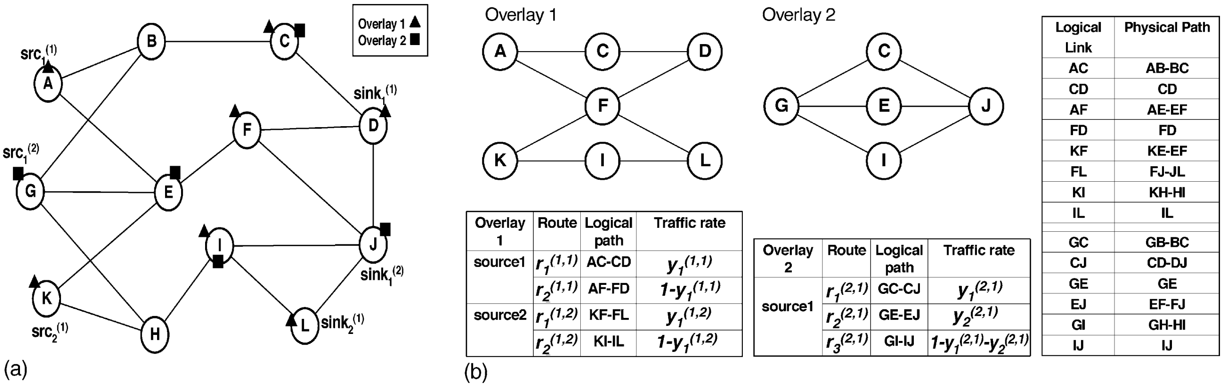
$$\text{s.t.} \qquad \text{for} \qquad \forall f \in \mathcal{F}_s, \qquad \sum_{k=1}^{|\mathcal{R}_f|} y_k^{(s,f)} = x_f, \tag{7}$$

$$Ay \le C, \quad y^{(s)} \ge 0 \tag{8}$$

For this optimization problem, overlay s considers other overlays' routing decisions as *fixed* when it makes the routing decision by solving the above optimization problem. Note that the objective function of this optimization problem is continuous, differentiable and convex. Since the feasible region defined by constraints in the optimization problem of Eq. (5) is convex and compact, the optimal value and the minimizer can be found by the Lagrangian method. Furthermore, an alternative solution is to apply the marginal cost flow approach [14] discussed shortly. During the interaction process among these overlays, each overlay s computes its current best strategy $y^{(s)}$ given other overlays' routing strategies $y^{(-s)}$, which is assumed to be invariant in the computation.

2.2.1. Illustration of notations and overlay networks

To illustrate the concept and various notations, let us consider the physical (or underlying) network depicted in Fig. 1(a). The physical network consists of 12 physical nodes and a set of physical links. There are two co-existing overlay networks above the physical network, in which overlay 1 has two source–sinks while overlay 2 has one single source–sink pair. Each overlay is a logical network using a set of physical nodes and logical links. In general, the topology of an overlay network depends on how the overlay is organized and Fig. 1(b) shows one possibility of the logical topologies. A physical node can also belong to multiple overlays, i.e., node C, I belong to both overlays. Every logical link is interpreted as a physical route between two neighbor overlay nodes, which is determined by the underlying IP level



Overlay 1	Route	Logical path	Traffic rate
source1	$r_1^{(1,1)}$	AC-CD	$y_1^{(1,1)}$
	$r_2^{(1,1)}$	AF-FD	$1-y_1^{(1,1)}$
source2	$r_1^{(1,2)}$	KF-FL	$y_1^{(1,2)}$
	$r_2^{(1,2)}$	KI-IL	$1-y_1^{(1,2)}$

Overlay 2	Route	Logical path	Traffic rate
source1	$r_1^{(2,1)}$	GC-CJ	$y_1^{(2,1)}$
	$r_2^{(2,1)}$	GE-EJ	$y_2^{(2,1)}$
	$r_3^{(2,1)}$	GI-IJ	$1-y_1^{(2,1)}-y_2^{(2,1)}$

Logical Link	Physical Path
AC	AB-BC
CD	CD
AF	AE-EF
FD	FD
KF	KE-EF
FL	FJ-JL
KI	KH-HI
IL	IL
GC	GB-BC
CJ	CD-DJ
GE	GE
EJ	EF-FJ
GI	GH-HI
IJ	IJ

Fig. 1. (a) A physical network with two co-existing overlays. (b) Two overlays and their routing decisions. Overlay 1 has two flows while overlay 2 has one flow. Each flow has one unit of traffic.

routing, i.e., the shortest path routing algorithm. The logical link to physical path mappings are depicted in Fig. 1(b).

For each source node in both overlays, there is a set of available logical paths to the corresponding sink. Tables in Fig. 1(b) list the set of available paths for each source–sink pair in each overlay. The routing decision vector $y^{(s)}$, $s = 1, 2$, assigns a non-negative amount of traffic to each of these available paths such that the sum of the amount of flow over all paths is equal to the traffic demand of the flow. In this example, the traffic demand of *each* flow is 1 unit. These overlays interact with each other due to the fact that the logical paths in different overlays have overlapping physical links. For example, the logical paths *AF–FD*, *KF–FL* in overlay 1 and *GE–EJ* in overlay 2 has the common physical link *EF*. Similar situation occurs for links *BC*, *CD*, *HI*, etc. As we will discuss later, since one overlay's routing decision depends on the routing decision of other overlays, there will be inevitable interaction between the routing behaviors of different overlays.

2.3. Relationship between global optimal routing, selfish routing and overlay optimal routing

In this section, we present the relationship between global optimal routing [3], selfish routing and overlay optimal routing. These routing schemes are solving three different types of system optimization problems, however, there is a fundamental distinction between their objectives and this provides users a spectrum of efficiency and fairness for operating an overlay network. Let us use the same network illustrated in Fig. 1 to show how these three routing strategies work. Note that the delay of each link is a function of the aggregate rate of traffic that traverses this link. In Table 1, we list out the delay function of each physical link. We denote $d_j(l_j)$ as the delay function of link j and l_j as the aggregate traffic rate of link j. For the simplicity of presentation, the first set of links in Table 1 has a "*linear*" delay function (i.e., delay proportional to the aggregate traffic), while the rest of the links in the network have a constant delay of zero or one unit of time.

Let us consider the overlay routing decision vector: $y^{(1,1)} = (y_1^{(1,1)}, 1 - y_1^{(1,1)})^{\mathcal{T}}$ denotes the traffic rate through routes *AC–CD* and *AF–FD* for the first flow in overlay 1, respectively, and $y^{(1,2)} = (y_1^{(1,2)}, 1 - y_1^{(1,2)})^{\mathcal{T}}$ denotes traffic rate through routes *KF–FL* and *KI–IL* for the second flow in overlay 1, respectively. Similarly, $y^{(2,1)} = (y_1^{(2,1)}, y_2^{(2,1)}, 1 - y_1^{(2,1)} - y_2^{(2,1)})$ denotes the traffic rate through routes *GC–CJ*, *GE–EJ*

Table 1
Delay function of physical links

Delay function	Physical links
$d_j(l_j) = l_j$	AB, BC, EF, FJ, HI
$d_j(l_j) = 0$	FD, DJ, KE, GE, JL
$d_j(l_j) = 1$	AE, GB, GH, KH, CD, IJ, IL

and *GI–IJ* for the flow in overlay 2. We assume the traffic demand for all the flows in both overlays is one unit, and there is no underlying traffic in the physical network. For the simplicity of presentation, we assume the capacity for each link is sufficiently large to support the given traffic flow demand.

The classical global optimal routing [3] is a centralized routing policy and the objective of it is to minimize the weighted average delay for "*all*" traffic in the physical network. Due to the lack of space, we will not repeat its theoretical formulation here (readers can refer to Ref. [3]) but just give the optimal solution for the example illustrated in Fig. 1: $y^{(1,1)} = \left(\frac{3}{8}, \frac{5}{8}\right)$, $y^{(1,2)} = \left(\frac{1}{8}, \frac{7}{8}\right)$ and $y^{(2,1)} = \left(\frac{1}{2}, \frac{1}{2}, 0\right)$. Note that in general, the solution to the global optimal routing may not be unique.

Selfish routing, on the other hand, is actually a greedy strategy which maximizes local benefit for a single user. From the theoretical aspect of selfish routing, each user controls an *infinitesimally small portion* of traffic and selects a path with the shortest delay. In real networks, the smallest unit is a packet. So the closest approximation of implementing the selfish routing is to assign each packet to the path which currently has the shortest delay. To allow end users to choose routes by themselves, either source routing (e.g., Nimrod [4]) or overlay routing (e.g., Detour [15] or RON [2]) can be used. In overlay routing, end users can find multiple overlay paths, with the assistance of other nodes in the same overlay relaying packets to the sink. Based on the result in Ref. [14], this is equivalent to solving the following constrained convex optimization problem:

$$\begin{aligned} &\textbf{SELFISH}\ [y; A, H, C, x]: \\ &\text{Minimize} \quad \sum_{j\in\mathcal{J}} \int_0^{l_j} d_j(t)\,\mathrm{d}t, \\ &\text{s.t.} \qquad Hy = x; \quad Ay \le C; \quad L = Ay; \quad y \ge 0. \end{aligned}$$

The Karush–Kuhn–Tucker (KKT) condition for the solution of selfish routing is that every route with non-zero traffic serving the same source–sink pair has the same end-to-end delay, moreover, it should be the minimum among all available routes. The equilibrium routing decisions for two overlays are $y^{(1,1)} = \left(\frac{3}{4}, \frac{1}{4}\right)$, $y^{(1,2)} = \left(\frac{3}{4}, \frac{1}{4}\right)$ and $y^{(2,1)} = \left(0, \frac{1}{2}, \frac{1}{2}\right)$. The detailed derivation is omitted here, and interesting readers can refer to Ref. [8]. In general, the equilibrium point for a selfish routing scheme may not be unique.

While selfish routing represents egoism from an end users' perspective, overlay optimal routing represents such selfishness from a larger entity's point of view. A special case is that when there are only two overlays, the interaction can be modeled similarly as a duopoly game [5]. For an individual overlay, overlay optimal routing achieves optimum within one overlay's range, similar to the way global optimal routing achieves global optimality within the *entire* underlying physical network through *traffic engineering*. Overlays can obtain from a common routing underlay the routing information of the underlying network, e.g., topology of physical network, delay function of physical links, current traffic rate of

Table 2
End-to-end delays of different routing strategies

Weighted delay, $\text{delay}^{(s)} = \frac{1}{\sum_{f\in\mathcal{F}_s} x_f} \sum_{f\in\mathcal{F}_s} \sum_{k\in\mathcal{R}_f} y_k^{(s,f)} \cdot D_k^{(f)}$	Overlay 1	Overlay 2	Overall delay
Centralized global optimal routing	2.50	2.38	2.46
Selfish routing	2.63	2.75	2.67
Overlay optimal routing	2.46	2.53	2.48

different links [11]. With these information, overlays adaptively regulate traffic within the overlay so that the average delay of the overall traffic is minimized. In the case that there are multiple overlays co-existing above the same physical network, it is very likely that there will be partially overlapping paths. Moreover, one physical node may belong to several overlays. Thus, there will be interaction if multiple overlays optimize their performance simultaneously.

To illustrate the routing decisions and the existence of Nash equilibrium of the overlay routing game, let us use the same example illustrated in Fig. 1. To find the Nash equilibrium, we need to first calculate the best response of each overlay to the routing decisions of other overlays. Given the fixed routing strategies of other overlays during the period of the optimization process, this is the solution of the convex optimization problem in Eq. (5). That is, for every path with non-zero traffic serving the same source–sink pair, the "*marginal length*" of these paths should be the same. In here, the length of a link is defined as the *first derivative of the weighted delay* for this link. Furthermore, the length should be minimum among all available routes serving this source–sink pair. One can rewrite the first derivative of the weighted delay for each link j as $(l_j \cdot d_j(l_j + l_j^*))' = d_j(l_j + l_j^*) + l_j \cdot d_j'(l_j + l_j^*)$, where l_j is the aggregate traffic of its own overlay that traverses link j, and l_j^* denotes other traffic traversing link j. Similarly, in here we only give the equilibrium routing decisions for two overlays, when both of them take the *overlay optimal routing* strategy: $y^{(1,1)} = \left(\frac{35}{72}, \frac{37}{72}\right)$, $y^{(1,2)} = \left(\frac{83}{216}, \frac{133}{216}\right)$ and $y^{(2,1)} = \left(\frac{11}{36}, \frac{49}{108}, \frac{13}{54}\right)$. Detailed derivation can also be referred to Ref. [8].

Finally, Table 2 depicts the performance measures in terms of weighted average delay for the three routing strategies. From the table, one can observe that three routing strategies achieve different overall performances, as well as the performance for individual overlays, representing a spectrum of efficiency and fairness for different routing strategies. For global performance, global optimal routing achieves the best while selfish routing achieves the worst. For individual overlays, selfish routing is still the worst among three of them, which again verifies the efficiency loss stated in previous studies [14]. When one considers the performance of the overlay optimal routing, we observe that the weighted delay of overlay 1 decreases while the weighted delay of overlay 2 increases, when compared to their respective performance achieved using the global optimal routing. It is intriguing to find one overlay achieves a better performance at the expense of the other overlay. In later sections, we will explore the fairness and resource allocation issues of this phenomenon.

3. Nash equilibrium for the overlay optimal routing

In this section, we present the game theoretic analysis of the interaction between co-existing overlays which are using the overlay optimal routing policy. We also present results of the fluid simulations to demonstrate the network stability.

Based on the mathematical formulation of the overlay optimal routing we presented in the previous section, one can model the interaction between multiple co-existing overlays as a Nash routing game. First, a finite set of players $\mathcal{N}$ consists of all overlays, where $\mathcal{N} = \{1, 2, \ldots, n\}$. Second, the strategy that an overlay can take is a vector of flow on all available paths, wherein each component of the vector is non-negative and satisfies the transmission demand. Further, the rate vector should satisfy the capacity constraint of each link. Formally, the set of action profiles for overlay s is:

$$\Gamma_s = \{y^{(s)} | y^{(s)} \in \mathbb{R}_+^{r_s}, (Hy)_s = x_s, Ay \leq C\} \tag{9}$$

where $\mathbb{R}_+$ denotes the set of non-negative real numbers and $r_s = \sum_{f \in \mathcal{F}_S} |\mathcal{R}_f|$. If $y = (y^{(1)}, y^{(2)}, \ldots, y^{(s)}, \ldots, y^{(n)})^{\mathcal{T}}$ satisfying the above constraint, then it is a feasible strategy profile. Furthermore, we define a preference relation $\succsim_s$ for player s. For any two feasible strategy profiles y and y', we say $y \succsim_s y'$ (player s prefers strategy profile y to y') if $\text{delay}^{(s)}(y) \leq \text{delay}^{(s)}(y')$, where $\text{delay}^{(s)}(y)$ is the average delay for overlay s as defined by Eq. (1). Thus, it is equivalent to defining the payoff function for player s as the negative of $\text{delay}^{(s)}(y)$.

We formulate the interaction between co-existing overlays as an n-player non-cooperative strategic game, which we call the *overlay optimal routing game*: $G_{\text{overlay}}\langle \mathcal{N}, (\Gamma_s), (\succsim_s)\rangle$. We have the following definition of *Nash equilibrium*.

Definition 1. A feasible strategy profile $y^* \in \Gamma_1 \times \cdots \times \Gamma_n$, $y^* = (y^{*(1)}, \ldots, y^{*(s)}, \ldots, y^{*(n)})^{\mathcal{T}}$ is called a Nash equilibrium if for every player $s \in \mathcal{N}$, $\text{delay}^{(s)}(y^{*(1)}, \ldots, y^{*(s)}, \ldots, y^{*(n)})$ is less than or equal to $\text{delay}^{(s)}(y^{*(1)}, \ldots, y^{'(s)}, \ldots, y^{*(n)})$ for any other feasible strategy profile $y'^{(s)}$.

The analysis of this strategic routing game is based on a discrete time model, that is, each overlay calculates its optimal strategy at every routing update. Note that the routing update period can be different for different overlays. We assume one overlay has sufficient time to complete this optimization before other overlays begin their optimization process. In the following sections, we prove the existence of Nash equilibrium in general networks, then we use extensive fluid simulations to illustrate our result.

In the following, we show that the overlay routing game $G_{\text{overlay}}\langle \mathcal{N}, (\Gamma_s), (\succsim_s)\rangle$ possesses a Nash equilibrium for a general network setting as long as the delay function for each physical link is continuous, non-decreasing and convex.

Lemma 1. *A strategic game $\langle \mathcal{N}, (\Gamma_s), (\succsim_s)\rangle$ has a Nash equilibrium if for all players $s \in \mathcal{N}$ that (1) the set Γ_s of action profiles for player s is a non-empty compact convex subset of a Euclidean space and (2) the preference relation $\succsim_s$ is (a) continuous and (b) quasi-concave on Γ_s.*

Proof. Note that Lemma 1 provides a standard approach to prove the existence of Nash equilibrium in a strategic game. For detailed proof of Lemma 1, readers can refer to Ref. [12]. □

Lemma 2. *In the overlay optimal routing game $G_{overlay}\langle \mathcal{N}, (\Gamma_s), (\succsim_s)\rangle$, the set Γ_s of action profiles for overlay s is a non-empty compact convex subset of a Euclidean space.*

Proof. The set of action profiles for overlay s is defined as $\Gamma_s = \{y^{(s)} | y^{(s)} \in \mathbb{R}_+^{r_s}, (Hy)_s = x_s, Ay \leq C\}$. Since the feasible region is closed and bounded, Γ_s is compact, and because all constraints are affine functions, the feasible region which is the intersection of half-spaces and hyperplanes, is also convex. □

Lemma 3. *The preference relation $\succsim_s$ in an overlay optimal routing game $G_{overlay}\langle \mathcal{N}, (\Gamma_s), (\succsim_s)\rangle$ is continuous and quasi-concave on Γ_s.*

Proof. for the sake of limited space, please refer to a detailed proof in Ref. [8]. □

Theorem 1. *In the overlay optimal routing game* $G_{overlay}\langle\mathcal{N}, (\Gamma_s), (\succsim_s)\rangle$, *there exists a Nash equilibrium if the delay function* $\text{delay}^{(s)}(y^{(s)}; y^{(-s)})$ *is continuous, non-decreasing and convex.*

Proof. By Lemmas 1–3, we can immediately show this result. □

We use matlab simulink to perform a fluid-based simulation on a network with six co-existing overlays. We show the existence of Nash equilibrium and the variation of routing decisions for different overlays. Readers can refer to Ref. [8] for a detailed description of simulation results.

4. Anomalies due to overlay interaction

In this section, we discuss some intrinsic problems of overlays interaction. These problems include sub-optimality in performance and certain fairness anomaly in resource allocation. It is important to point out that these problems are not unique to overlay optimal routing policy, but rather, common to all forms of application layer routing that have interaction among overlays. Since many overlays are now appearing in the Internet, unregulated application routing may degrade the performance of all users. Worse yet, because overlay may not realize the existence of other overlays, these problems will persist due to the convergence to the equilibrium point.

Let us use an example to illustrate these issues. A physical network consisting of six nodes is depicted in Fig. 2. There are two overlays in the network: overlay 1 consists of nodes A, C and E while overlay 2 consists of nodes B, D and F. For overlay 1, all logical links map to the corresponding physical links except for the logical link between nodes C and E, which corresponds to the physical links C–D–E. For overlay 2, all logical links map to the corresponding physical links except for the logical link between nodes B and D, which corresponds to the physical links B–C–D. Thus, the physical link C–D is the common link which is *shared* by these two overlays.

4.1. Sub-optimality of Nash equilibrium

Assume that both overlays have one source–sink pair and one unit of traffic demand: $x_1 = x_2 = 1.0$. We define the following delay functions for various physical links in the physical network: $d_{A,E}(y) =$

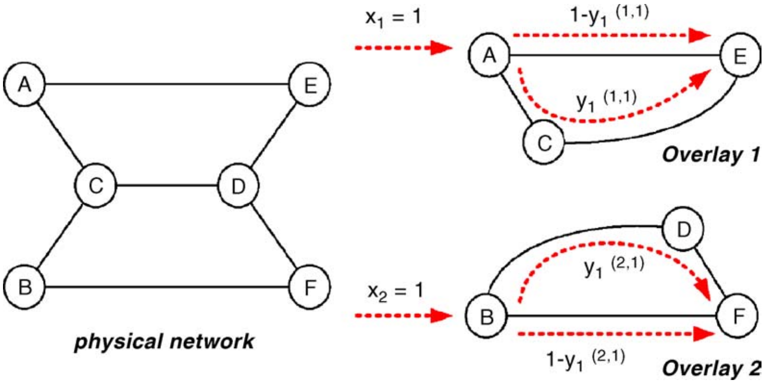


Fig. 2. A simple network with two overlays to illustrate potential problems.

$a + y; d_{C,D}(y) = by^{\alpha}; d_{B,F}(y) = c + y$, while other links have zero delay. Here, y represents the aggregate traffic traversing a link, and a, b, c and α are some non-negative parameters of the delay functions.

Let us consider the routing decisions of these two overlays. For overlay 1, it routes $y_1^{(1,1)}$ *fractional* unit of traffic through the logical path *A–C–E* and $(1 - y_1^{(1,1)})$ fractional unit of traffic via the logical path *A–E*. On the other hand, overlay 2 routes $y_1^{(2,1)}$ fractional unit of traffic through the logical path *B–D–F* and $(1 - y_1^{(2,1)})$ fractional unit of traffic via *B–F*. Similarly, the KKT conditions for overlay 1 is

$$a + 2[1 - y_1^{(1,1)}] = b[y_1^{(1,1)} + y_1^{(2,1)}]^{\alpha} + y_1^{(1,1)} \cdot b\alpha[y_1^{(1,1)} + y_1^{(2,1)}]^{\alpha-1} \tag{10}$$

while the Karush–Kuhn–Tucker conditions for overlay 2 is

$$c + 2[1 - y_1^{(2,1)}] = b[y_1^{(1,1)} + y_1^{(2,1)}]^{\alpha} + y_1^{(2,1)} \cdot b\alpha[y_1^{(1,1)} + y_1^{(2,1)}]^{\alpha-1}. \tag{11}$$

where $y_1^{(1,1)}, y_1^{(2,1)} \in [0, 1]$.

One can easily show that in the overlay optimal routing game described above, the Nash equilibrium point is *not* Pareto optimal. A Pareto optimal point is defined as a strategy profile for all overlays such that no overlay can use another routing strategy that can decrease its own weighted average delay *without* increasing other overlays' weighted average delay. Although all overlays perform an individual optimization at every routing update and that the system will finally reach a Nash equilibrium point, the equilibrium may be inefficient since there exists another routing strategy at which *all* overlays can achieve a better performance than at the Nash equilibrium.

To show the sub-optimality of the Nash equilibrium in the optimal routing game for the network depicted in Fig. 2, we consider the KKT conditions specified by Eqs. (10) and (11). Assume we have the following parameters for the delay functions: $\alpha = 1$, $a = 1$, $b = 1$ and $c = 2.5$, one can simply verify that the Nash equilibrium in this example is $\{y_1^{(1,1)} = 0.5, y_1^{(2,1)} = 1\}$, that is, overlay 1 uses both paths while overlay 2 uses a single path, which consists of the shared link. The weighted average delay for overlays 1 and 2 is both 1.5. However, if we consider another routing strategy profile of $\{y_1^{(1,1)} = 0.4, y_1^{(2,1)} = 0.9\}$, one can find that the weighted average delay for overlays 1 and 2 are 1.48 and 1.43, respectively, which are *lower* than the delay achieved at the Nash equilibrium.

4.2. Fairness paradox

Another more severe problem is the notion of fairness in resource allocation. We use the same network in Fig. 2 to illustrate the problem. Note that these two overlays are symmetric, each having two paths: a shared path and a private path. Although overlay 2 is "*worse off*" by having a private path (link *B–F*) with higher delays than that of overlay 1's private path (link *A–E*), as in the previous example, it is able to achieve the same average delay as overlay 1 in the Nash equilibrium. This is because overlay 2 is able to fully take advantage of the lower delay of the shared path, whereas it only makes sense for overlay 1 to send part of its traffic over the shared link due to its superior private path. In fact, one can find delay functions such that the situation is *arbitrarily worse*.

To illustrate, note that the delay function for the shared link *C–D* is $d_{C,D}(y) = by^{\alpha}$. One can ask for what values of a and c, which are the parameters of the delay functions for the private link of overlays 1 and 2, so that the Nash equilibrium solution remains at $\{y_1^{(1,1)} = 0.5, y_1^{(2,1)} = 1\}$? The values for a and c

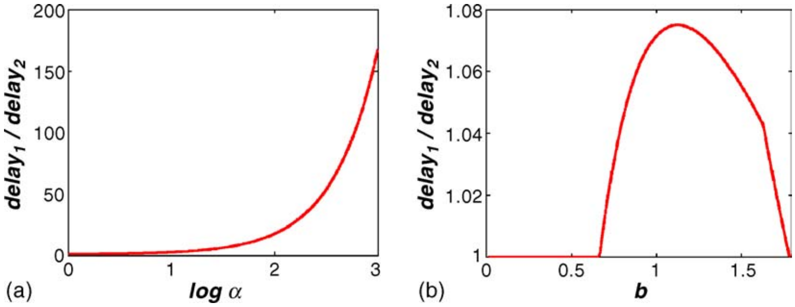


Fig. 3. $\text{Delay}_1/\text{delay}_2$ ratio vs. (a) $\log(\alpha)$: unfairness becomes unbounded and (b) parameter b with $a = 2, c = 4$ and $\alpha = 1$: bounded unfairness.

do exist, in particular, when $b = 1$ and $a < c$, we have:

$$a = \left(\frac{3}{2}\right)^{\alpha} + \frac{\alpha}{2}\left(\frac{3}{2}\right)^{\alpha-1} - 1; \qquad c = \left(\frac{3}{2}\right)^{\alpha} + \alpha\left(\frac{3}{2}\right)^{\alpha-1}.$$

In Section 4.1, we showed when $a < c$ and $\alpha = 1$, $\text{delay}_1 = \text{delay}_2$. When $\alpha > 1$:

$$\text{delay}_1 = \left(\frac{3}{2}\right)^{\alpha} + \frac{\alpha}{4}\left(\frac{3}{2}\right)^{\alpha-1} - \frac{1}{4}; \qquad \text{delay}_2 = \left(\frac{3}{2}\right)^{\alpha},$$

and observe now that delay_1 becomes *greater* than delay_2. This implies that the overlay 2 is able to achieve *better* performance despite starting with a *worse* private link in an otherwise symmetric situation with overlay 1. Furthermore, as we increase α, this unfairness can be *unbounded*, that is:

$$\left.\frac{\text{delay}_1}{\text{delay}_2}\right|_{\alpha\to\infty} = \infty$$

and this is depicted in Fig. 3(a). This type of anomaly also exists in other operating range, for example, when $a = 2$, $c = 4$, $\alpha = 1$ and we vary the values of b, one can observe that overlay 1 will have a worse performance compared to overlay 2 (though the unfairness is bounded in this case). In summary, we illustrate that there exist delay functions for the links such that although overlay 1 seemingly has better paths than overlay 2, it is destined to lose the routing game to overlay 2 by an arbitrary margin: a rather "*paradoxical*" situation.

5. Pricing

In the previous section, we observed some undesirable properties achieved at the equilibrium due to unregulated competition between overlays for the common resources. In this section, we explore some pricing schemes to address these potential problems, namely (a) sub-optimality at the Nash equilibrium and (b) fairness anomaly issues. We show the efficiencies of our pricing schemes and give their economic interpretations.

5.1. Pricing mechanism to improve end-to-end delay

In the previous sections, we show that the Nash equilibrium achieved at the overlay optimal routing game has a performance gap compared to the global optimal routing, in terms of the weighted average delay for the overall traffic in the network. Furthermore, the Nash equilibrium is not Pareto optimal. In this section we explore a pricing scheme such that the overlay routing game can be led to an efficient equilibrium, i.e., the operating point achieved by the global optimal routing.

The basic idea of the pricing scheme is to introduce a price (cost per unit traffic) for every physical link in the network. When overlays make their routing decisions, not only should they consider the end-to-end delays, but also the total price they have to pay to the network operators for consuming the bandwidth. Formally, for each link j, we set a price $p_j^{(s)}$ for each overlay s in the network, which is the per unit charge for the traffic in overlay s by the link j. Note that in here we adopt a price-discriminating strategy: different prices for users in different overlays. Under this pricing scheme, each overlay tries to optimize the following objective function:

$$\text{Minimize} \quad \sum_{j\in\mathcal{J}} \frac{1}{\sum_{f\in\mathcal{F}_s} x_f} \cdot [l_j^{(s)} \cdot d_j(l_j^{(s)} + l_j^{(-s)}) + l_j^{(s)} \cdot p_j^{(s)}] \tag{12}$$

wherein $l_j^{(s)}$ denotes the traffic of overlay s on link j, and $l_j^{(-s)}$ denotes traffic on link j other than overlay s. Note that the constraints to this optimization problem is the same as the overlay optimization given in Eq. (5). Such a pricing scheme has the following interpretations: an overlay aims to minimize the average delay and the cost it has to pay to the network operators simultaneously. If the price $p_j^{(s)}$ is set properly to reflect the economic relationship with the delay metric, i.e., to the same order of delay on link j, one can gain a balance between overlays' performance and the cost it has to pay. Let us state how the price of each link should be set so as to achieve global optimality.

Theorem 2. *In the overlay optimal routing game $G_{overlay}$ if (a) the objective function of each overlay is set as in Eq.* (12) *and (b) $p_j^{(s)} = l_j^{(-s)} \cdot d_j'(l_j)$, where $l_j = l_j^{(s)} + l_j^{(-s)}$, then the Nash equilibrium achieves the same performance as the global optimal routing.*

Proof. The objective of the global optimal routing is to minimize the weighted average delay for the overall traffic in the network, which is $\sum_{j\in\mathcal{J}} l_j \cdot d_j(l_j)$, where l_j is the total traffic on link j. Thus, the KKT condition for the optimal point achieved at the global optimal routing can be written as: for every route r with non-zero traffic, and route r' with zero-traffic, serving the same source–sink pair/flow f

$$\sum_{j\in r}[l_j \cdot d_j(l_j)]' = u_f \leq \sum_{j\in r'}[l_j \cdot d_j(l_j)]'.$$

On the other hand, the KKT condition for the optimization problem defined in Eq. (12) requires that for every route r with non-zero traffic, and route r' with zero traffic, serving the same flow f in overlay s, must satisfy:

$$\sum_{j\in r}[l_j^{(s)} \cdot d_j(l_j^{(s)} + l_j^{(-s)}) + l_j^{(s)} \cdot p_j^{(s)}]' = u_f \leq \sum_{j\in r'}[l_j^{(s)} \cdot d_j(l_j^{(s)} + l_j^{(-s)}) + l_j^{(s)} \cdot p_j^{(s)}]'.$$

We expand the expression and substitute $p_j^{(s)} = l_j^{(-s)} \cdot d'_j(l_j)$, and we have

$$\begin{aligned}
&\sum_{j\in r}[d_j(l_j^{(s)} + l_j^{(-s)}) + l_j^{(s)} \cdot d'_j(l_j^{(s)} + l_j^{(-s)}) + l_j^{(-s)} \cdot d'_j(l_j^{(s)} + l_j^{(-s)})] = u_f \\
&\quad \Leftrightarrow \sum_{j\in r}[d_j(l_j) + l_j \cdot d'_j(l_j)] = u_f \\
&\quad \Leftrightarrow \sum_{j\in r}[l_j \cdot d_j(l_j)]' = u_f
\end{aligned}$$

which proves that the Nash equilibrium of the overlay routing game under this pricing scheme has the same KKT condition as that of global optimal routing. □

5.2. Pricing mechanism to improve fairness

One important incentive for the network operators (ISPs) to perform network upgrade is to increase profit. Furthermore, it is desirable if the pricing and revenue sharing schemes correctly represent the contribution of different ISPs, while provide fair resource allocation to the users in the network. In the previous section, we have observed some fairness anomalies due to unregulated competition between overlays. Worse yet, such unfairness can even be unbounded for general situations. In here, we explore another pricing scheme that can achieve a spectrum of fairness balanced between the performance and the cost, and reflects the economic benefit for the network operators.

Our pricing scheme is based on the following two natural arguments. Firstly, pricing should reflect users' willingness to pay, meaning that those users who pay more have more opportunity to receive better performance. Secondly, the network operators set prices to maximize their own economic interests. Before proceeding to the formal pricing model, we make some additional assumptions, for both simplicity and ease of presentation. In here we assume that those "common" physical links in the network represent network operators and have their own economic interests. The argument is that an overlay network may reside over many ISPs and these physical links constitute the bottleneck links within one ISP. This is because one provider's egress traffic is often charged by its downstream providers and these bottleneck links often get saturated before other links do.

Under this assumption, the pricing scheme can be decomposed into two components, one for the overlay users and the other for the network operators. For each physical link, it is interpreted as a logical ISP which tries to maximize its own economic benefit. Each link j sets a price p_j, which is the charge for per unit traffic traversing this link, to maximize its own profit. The profit of a link j is defined as $P_j(l_j) = p_j \cdot l_j - c_j(l_j)$, where the first part accounts for the revenue received by the link, while the latter part $c_j(l_j)$ is interpreted as the operating cost at current congestion level. Note that a link sets the same price for *all* overlays and there is no price discriminate. Formally, each link j solves the following optimization problem:

$$\text{Maximize} \quad P_j(l_j) = l_j \cdot p_j - c_j(l_j) \tag{13}$$

To maximize the profit, link j would not set the price arbitrarily large, otherwise no overlay would choose to traverse this link. Given a fixed price p_j, the optimal aggregate traffic on a link j to maximize the profit can be obtained by solving $\mathrm{d}P_j/\mathrm{d}l_j = p_j - c'_j(l_j) = 0$, thus having $p_j = c'_j(l_j)$. To match the actual

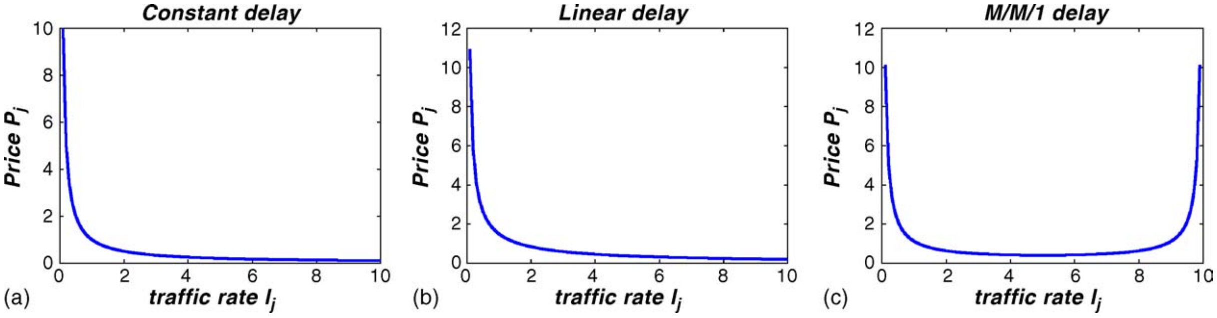


Fig. 4. Price–traffic curves for links with different delay functions, with $C_j = 10$ units.

bandwidth consumption demand, link j will set its price by $p_j = c'_j(l_j)$. Note that the operating cost $c_j(l_j)$ is a non-decreasing function of the aggregate traffic on the link. We argue that this is an approach for links to estimate the traffic demand on them. This equivalently determines the supply curve for the link resources.

For the overlays, each of them is associated with a load function $U_s(y^{(s)}; y^{(-s)})$ that it wants to minimize. With the prices set by the links, one overlay s determines its routing by solving the following optimization problem:

$$\text{Minimize} \quad U_s(p^{(s)}; p^{(-s)}) = \alpha_s \cdot \text{delay}^{(s)} + \sum_{j \in \mathcal{J}} l_j^{(s)} \cdot p_j \tag{14}$$

where delay$^{(s)}$ is the average delay and the latter part accounts for the total payment for consuming the link bandwidth. α_s is interpreted as the *sensitivity factor*, which is determined by the overlay to reflect the overlay's own preference balanced between the performance and the cost. Namely, the larger α_s is, the more overlay s claims for a better performance. Also note that the constraints to this optimization problem is the same as the overlay optimization given in Eq. (5). By writing down the KKT conditions to this optimization problem, it is easy to show that the overlay routing game under objective function in Eq. (14) has a Nash equilibrium when p_j is considered as a fixed parameter by overlays. The routing decisions made by the overlays actually determine the demand curve of the link resources (bandwidth).

To see how this pricing scheme works, the cost function of link j can be set in the following way: $c_j(l_j) = \log[l_j \cdot d_j(l_j)]$, such that the cost of operating a link increases logarithmically with the weighted delay on that link. In other words, $l_j \cdot d_j(l_j)$ can be interpreted as the total "*bits*" carried by this link. Therefore, the price of a link is determined:

$$p_j = c'_j(l_j) = \frac{1}{l_j} + \frac{d'_j(l_j)}{d_j(l_j)} \tag{15}$$

wherein the second part can be interpreted as the *elasticity* of the delay function of link j. In Fig. 4, we plot the price–traffic curve for different delay functions. In here some observation is made. Firstly, the price function is a decreasing function of the traffic traversing it. The justification is that when *less* traffic traverses a link, the link bears a *lower* delay, which implies a *higher* price. Therefore, for overlays that choose links with lower delay to ensure a better performance, they have to pay more to buy the service. Note that in Fig. 4(c), though the price tends to be an increasing function when the traffic rate exceeds half of the capacity, the high delay when the rate approaches the capacity prevents overlays from using this link.

6. Related Work

With the increasing popularity of overlay networks, there exists an opportunity for end users of overlay network to choose the routing path and this is called the *overlay routing*. Usually, the objective of these end users is to choose routes with the minimum delay (or smallest loss rate, maximal available bandwidth). This behavior is considered selfish by nature and is also called the selfish routing. In Ref. [8], we consider interaction of multiple overlays and the associated anomalies when each overlay has a single flow only.

Network routing scheme which achieves the global optimality was studied by Bertsekas and Gallager [3]. Such a global optimal routing achieves the minimum average latency for *all* the traffic in the network by solving a centralized optimization problem. Based on this global optimal performance, Roughgarden and Tardos [14] showed that the global performance measure will be reduced if each end user performs selfish routing. Under the theoretical framework that each infinitesimal bit of information is sent in a selfish manner and that the link's delay function is linear, authors [14] provided the theoretical upper bound for this efficiency loss. Instead of making a selfish decision on an infinitesimal basis, a more practical framework [13,18] is to make a selfish routing decision on a per flow basis. Unfortunately, this type of selfish routing will cause severe routing oscillations [17] due to simultaneous routing update. Recent measurements in real network environment [13] show that selfish routing achieves a close-to-optimal performance in terms of average delay, but at the cost of overloading some links and it is difficult to achieve an equilibrium [10]. In Ref. [18], authors proposed a probabilistic routing scheme to implement the selfish routing in dynamic environments so as to ensure that the selfish behaviors of end users will converge to an *equilibrium* point, i.e., the routing decision stabilizes but it may *not* optimize the average delay for the flow.

Our work differs from the selfish routing listed above in that we consider *multi-path* routing which aims at *minimizing* the weighted average delay for an overlay and at the same time, guarantees the system will reach an equilibrium point. We use a non-cooperative game theoretic framework to show the existence of the Nash equilibrium. The seminal work to ours is by Zhang and co-workers [9], wherein authors use a two-player game to model the interaction between one overlay and the underlying ISP. Different from their work, we focus on the interaction between multiple co-existing overlays. Not only do we prove the existence of Nash equilibrium under general network settings, but we also present some undesirable properties of the equilibrium point and we propose two pricing strategies to alleviate the problems.

7. Conclusion

In this paper, we study the interaction of multiple overlays on top of a physical network. In particular, each overlay will perform application layer routing so as to optimize its individual performance. We first propose the concept of overlay optimal routing and illustrate that it can achieve better performance than the classical selfish routing policy. We use the constrained convex optimization framework to characterize the selfish routing and the overlay optimal routing. To understand the interaction, we use a non-cooperative game theoretic framework and analyze the property of the Nash equilibrium point under general network settings. Through illustration, we demonstrate some inherent anomalies of this form of individual routing optimization, namely the performance measure is sub-optimal and the fairness paradox in terms of performance and allocation of common resource. These two issues are important since overlay may not be aware of the existence of other overlays in the network. The implication is that they will continually operate at the sub-optimal point and that some overlays may experience poor performance due to the unregulated application layer routing. To resolve the above issues, we present two pricing schemes in which one can have a finer control about the performance and resource distribution.

Acknowledgements

We wish to thank the anonymous referees for their insightful comments. The research of J.C.S. Lui was supported in part by the RGC. This work is partially supported by a grant from the CUHK (Project no. 2050314).

References

[1] Akamai, http://www.akamai.com.

[2] D. Anderson, H. Balakrishnan, M.F. Kaashoek, R. Morris, Resillient overlay networks, Proceeding of the ACM SIGCOMM, Karlsruhe, Germany, October 2001.

[3] D. Bertsekas, R. Gallager, Data Networks, Prentice-Hall, NJ, 1992.

[4] I. Castineyra, N. Chiappa, M. Steenstrup, The Nimrod Routing Architecture, RFC, 1992, IETF, August 1996.

[5] D. Fudenberg, J. Tirole, Game Theory, MIT Press, 1957.

[6] P. Ganesan, G.S. Manku, Optimal routing in chord, Proceeding of 15th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA, January 2004, pp. 169–178.

[7] T. Guven, R.J. La, M.A. Shayman, B. Bhattacharjee, Measurement based optimal routing on overlay architectures for unicast sessions, Comput. Netw. J. Netw. Model. Simul., 2004 (Special Issue).

[8] W. Jiang, D.-M. Chiu, J.C.S. Lui, Interaction of overlay networks: properties and implications, Tech. Report, 2005.

[9] Y. Liu, H. Zhang, W. Gong, D. Towsley, On the interaction between overlay routing and underlay routing, Proceeding of IEEE INFOCOM'05, Miami, FL, USA, March 2005.

[10] T. Luecking, M. Mavronicolas, B. Monien, M. Rode, P. Spirakis, I. Vrto, Which is the worst-case nash equilibrium? 28th International Symposium on Mathematical Foundations of Computer Science, MFCS 2003, 2003.

[11] A. Nakao, L. Peterson, A. Bavier, A routing underlay for overlay networks, ACM SIGCOMM, 2003.

[12] M. Osborne, A. Rubinstein, A Course in Game Theory, MIT Press, 1957.

[13] L. Qiu, Y.R. Yang, Y. Zhang, S. Shenker, On selfish routing in internet-like environments, SIGCOMM, 2003.

[14] T. Roughgarden, E. Tardos, How bad is selfish routing? J. ACM 49 (2) (2002) 236–259.

[15] S. Savage, T. Anderson, A. Aggarwal, D. Becker, N. Cardwell, A. Collins, E. Hoffman, J. Snell, G. Voelker, J. Zahorjan, Detour: a case for informed internet routing and transport, IEEE Micro 19 January (1) (1999) 50–59.

[16] S. Savage, A. Collins, E. Hoffman, J. Snell, T. Anderson, The end-to-end effects of internet path selection, Proceedings of ACM SIGCOMM '99, Cambridge, MA, August 1999, pp. 289–299.

[17] M. Seshadri, R. H. Katz. Dynamics of simultaneous overlay network routing, UC Berkeley, CSD-03-1291, 2003.

[18] H. Xie, L. Qiu, Y.R. Yang, Y. Zhang, On self adaptive routing in dynamic environments, Proceedings of International Conference on Network Protocols, Berlin, Germany, October 2004.

Wenjie Jiang was born in China. He received his BSc degree from University of Science and Technology of China in 2003. Currently, he is an MPhil candidate in the Department of Computer Science and Engineering at the Chinese University of Hong Kong. His research interests lie in the theoretic topics of data networks, performance evaluation, game theory. His personal hobbies include films and sports.

Dah-Ming Chiu received his BSc degree from Imperial College London in 1975, and PhD degree from Harvard University in 1980. He worked for Bell Labs, Digital Equipment Corporation and Sun Microsystems Laboratories. Currently, he is a professor in the Department of Information Engineering at the Chinese University of Hong Kong. He is on the editorial board of the International Journal of Communication Systems.

John C.S. Lui was born in Hong Kong. He received his PhD in computer science from UCLA. After his graduation, he joined the IBM Almaden Research Laboratory/San Jose Laboratory and participated in various R&D projects on file systems and parallel I/O architectures. He later joined the Department of Computer Science and Engineering at the Chinese University of Hong Kong. He has been a visiting professor in computer science departments at UCLA, Columbia University, University of Maryland at College Park, Purdue University, University of Massachusetts at Amherst and Universit degli Studi di Torino in Italy. His research interests span both in system and in theory/mathematics, with current research interests in theoretic/applied topics in data networks, distributed multimedia systems, network security, OS design issues and mathematical optimization and performance evaluation theory. John received various departmental teaching awards and the CUHK Vice-Chancellor's Exemplary Teaching Award. He is an associate editor in the Performance Evaluation Journal, member of ACM, a senior member of IEEE and an elected member in the IFIP WG 7.3. John was the TPC co-chair of ACM Sigmetrics 2005. His personal interests include films and general reading.