Multipath Overlay Data Transfer

Bing Wang, Jim Kurose, Don Towsley, Wei Wei

Department of Computer Science
University of Massachusetts, Amherst, MA 01003

UMass CMPSCI Technical Report 05-45

***Abstract*— For applications involving data transmission from multiple sources, an important problem is: when the sources use multiple paths, how to maximize the aggregate sending rate of the sources using application-layer techniques via TCP? We solve this problem in the context of an overlay network by allowing a source to send data over k $(k \geq 1)$ overlay paths to its destination. Our goal is to select the overlay paths for each source and control the sending rate on each path via TCP to maximize the aggregate sending rate of the sources. We prove that optimal path selection is NP-hard and develop two practical application-level multipath rate controllers that use TCP. Our performance evaluation demonstrate that very simple path-selection and rate-control algorithms perform reasonably well in a wide range of settings. Furthermore, a small number of overlay paths for a source and a small amount of extra bandwidth in the network are sufficient to realize most of the performance gains.**

I. Introduction

A wide range of applications require data transmission from geographically distributed sources to one or multiple destinations using the Internet. For instance, in the Engineering Research Center (ERC) for Collaborative Adaptive Sensing of the Atmosphere (CASA) [1], multiple X-band radar nodes are placed at geographically distributed locations, each remotely sensing the local atmosphere. Data collected at these radar sites are transmitted to a central or multiple destinations using a state-wide public network for hazardous weather detection. In another example, high-volume astronomy data are stored at multiple geographically distributed locations (e.g., the Sloan Digital Sky Survey data). Scientists may need to retrieve and integrate data from archives at several locations for temporal and multi-spectra studies using the Internet (e.g., via Sky-Server [2]). In yet another example, an ISP places multiple data monitoring sites inside its network. Each monitoring site collects traffic data and transmits them to a central location for analysis and network diagnosis.

A crucial factor for the success of the above applications is efficient data transfer from the multiple sources to the destinations. In these applications, the sources and destinations typically have high access bandwidths while non-access links may limit the sending rate of the sources as indicated by recent measurement studies [3]. This is clearly true in CASA: the sending rates of the radar nodes are restrained by low-bandwidth links inside the state-wide public network. When the bandwidth constraints are inside the network, using multiple paths between a source and destination can provide a much higher throughput [4], [5]. The problem we address is: *when the sources use multiple paths, how to maximize the aggregate sending rate of the sources?* We seek to solve this problem using application-layer techniques via TCP due to several reasons. First, these applications require reliable data transfer which makes TCP a natural choice. Secondly, since TCP is the predominant transport protocol in the current Internet, application-layer approaches via TCP are easy to deploy. Furthermore, all applications in the Internet are expected to be TCP friendly [6] and using TCP is by definition TCP-friendly. Our focus is on scenarios where multiple paths between a source and destination are formed using an overlay network, which has been show to be an effective multipath architecture for throughput improvement over using a single path ([5] shows an improvement of 20-55%). More specifically, we consider the following problem. Consider a set of sources, a set of relays and a set of destinations forming an overlay network. A source selects k $(k \geq 1)$ overlay paths (i.e., network paths via one or multiple relays) and spreads its data among the overlay paths. We restrict the source to use no more than k overlay paths since data splitting involves overheads (e.g., meta data are required in order to reassemble data at the destination). Our goal is to select the overlay paths for each source and control the sending rate on each path via TCP in order to maximize the aggregate sending rate of the sources.

We focus on distributed algorithms for path selection and rate control because centralized algorithms are often unrealistic in practice. Joint optimization of these two problems is difficult even in a centralized setting [7]. Therefore, we address these two problems separately. Our main contributions are:

- We prove that the problem of optimal overlay path selection, even in extremely simple settings, is NP-hard for any given rate controller.
- We develop practical multipath rate controllers composed from single-path rate controllers. Specifically, we design two application-layer rate controllers that use TCP to control the sending rate on each path. Our controllers are not specific to overlay networks and can be applied to general multipath settings. They are easy to implement and are readily deployable. We analyze the fairness properties of multipath rate controllers and prove that one of our controllers maximizes the aggregate sending rate of the sources in settings with two logical-hops and a single destination (see Section V).
- We evaluate the performance of two randomized path-

selection algorithms coupled with various multipath rate controllers, and show that very simple path selection algorithms and rate controllers perform reasonably well in a wide range of settings. Furthermore, a small number of paths, i.e., k of 2 to 4, and a small amount of extra bandwidth in the network are sufficient to realize most of the performance gains.

The rest of this paper is organized as follows. Section II describes related work. Section III presents the general problem setting. Path selection and multipath rate control are studied in Sections IV and V respectively. Section VII presents a performance evaluation using numerical techniques and *ns* simulation. Finally, Section VIII concludes this paper and describes future work.

II. Related work

The studies of [8], [9], [10] consider multipath routing at the network layer, as an improvement to the single-path IP routing. We, in contrast, consider multipath data transfer at the application level, without any change to IP routing. Hence, our approach is readily deployable in the current Internet. The studies of [11] and [12] focus on data uploading and replication respectively, allowing a source to use multiple paths inside an overlay network. They develop *centralized* algorithms to minimize the transfer time. Our focus is on developing efficient *distributed* algorithms to maximize the aggregate sending rate of the sources.

Recent findings on overlay network form the basis of our performance evaluation (Section VII). The studies [13], [14], [15] have found that using a single relay on overlay paths provide performances close to those using multiple relays. Furthermore, [13] shows that the single relay can be chosen randomly from a group of relays. Motivated by the above results, in our performance evaluation, we restrict ourselves to overlay paths containing only a single relay and propose two randomized algorithms to select relays.

Our path selection and rate control problems differ significantly from upstream-ISP selection [16], [17], [18] and egress-data routing [4], [19], [20], [18] in multihoming. Optimal overlay paths selection in our study is potentially more difficult than ISP selection in multihoming since the number of overlay paths can be much larger than that of the ISPs. The rate control in our work is within individual flows, an approach not used by egress-traffic routing for multihomed sources [4], [19], [20], [18]. Our application-layer multipath controllers can be applied to general multipath settings (including those formed by multihoming).

Rate control (also interchangeably referred to as flow control or congestion control in the literature) is modeled as an optimization problem in [21], [22]. In this framework, each user is associated with a utility function. The objective of rate control is to maximize the aggregate utility. Based on this framework, a number of studies developed different approaches to rate control when each source uses a single path [23], [24], [25], [26], [27] and multiple paths [22], [28], [29], [30], [31], [32], [33], [34]. All the above algorithms require congestion price feedback from the network and are difficult to realize in practice. Our emphasis is on efficient application-level approaches that are easy to implement, rather than solving the optimization problem exactly. To this end, we design TCP-based multipath algorithms that use TCP on a per-logical-hop basis and take advantage of the congestion control and reliable data transfer embedded in TCP. Our algorithms only require simple rate regulation at the application level and therefore is readily deployable.

Notation	**Definition**
S, S_s, S_m	Set of sources, single-path and multipath sources
R	Set of relays
D	Set of destinations (receivers)
x_s	Source rate of source $s \in S$
x_{sj}	Path rate on the j-th path of source $s \in S$
m_s	Maximum source rate of source $s \in S$
$U(x_s)$	Utility function of source $s \in S$
L	Set of links in the network
c_l	Capacity of link $l \in L$
L_{sj}	Set of links on the j-th path of source $s \in S$

TABLE I

Key notation.

III. Problem setting

In this section, we formally describe the problem setting. The key notation is summarized in Table I for easy reference. Consider a set of sources S, a set of relays R and a set of destinations D forming an overlay network. Each source is associated with a destination (receiver). One type of source, referred to as a *single-path source*, transfers data using a single path (e.g., the default IP path, i.e., the path from the source to its receiver determined by IP). The other type of source, referred to as a *multipath source*, selects k ($k \geq 1$) overlay paths (i.e., network paths via one or multiple relays) and spreads its data to the overlay paths[1]. Multipath sources are illustrated in Fig. 1, where $k = 2$. We denote the set of single-path and multipath sources as S_s and S_m respectively. Then $S_s \cup S_m = S$ and $S_s \cap S_m = \emptyset$. The sources and destinations have high access bandwidth (e.g., through well-connected access networks or multihoming, an increasingly common practice [18]). The relays are placed (e.g., using techniques in [14]) such that multiple overlay paths do not share performance bottlenecks.

We denote by *path rate* the rate at which a source sends data over a path. The sum of the path rates associated with a source is the *source rate*. For ease of exposition, we index a source's path(s), starting from 1. For source s, let x_{sj} denote its path rate on the j-th path and x_s denote its source rate, $x_s \geq 0, x_{sj} \geq 0$. Then, $x_s = x_{s1}$, $\forall s \in S_s$ and $x_s = \sum_{j=1}^{k} x_{sj}$, $\forall s \in S_m$. Let m_s be the maximum source rate of source s, referred to as the *demand* of source s. This maximum source

[1] In practice, a multipath source may also send data over its default IP path. We only consider overlay paths since our path selection selects overlay paths and rate control does not differentiate the default IP path and overlay paths.

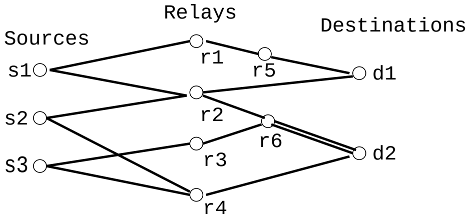


Fig. 1. Illustration of multipath sources: a multipath source spreads its data to k overlay paths to its destination. In this example, $k = 2$.

rate may come from the bandwidth limit of the source or the data generation rate at the source. A source is *satisfied* if its maximum sending rate is achieved, i.e., $x_s = m_s$. Each source is associated with a utility function, $U(x)$, which represents, for instance, the satisfaction of a source with the source rate of x. Throughout this paper, we assume that $U(x)$ is increasing and concave.

The problem we consider is, for a multipath source, how to select overlay paths and control the path rates in order to maximize the aggregate utility over all sources (including both multipath and single-path sources). We next define path selection and rate control formally. Let L denote the set of links in the network. The capacity of link l is $c_l, l \in L$. Let L_{sj} denote the set of links traversed by the j-th path of source s. The path-selection problem determines L_{sj} for $s \in S_m$, $j = 1, \ldots, k$. The path-rate control in the network can be stated as an optimization problem $\mathbf{P}$:

$$\mathbf{P}: \quad \text{maximize:} \quad \sum_{s \in S} U(x_s) \qquad (1)$$

$$\text{subject to:} \quad x_s = \sum_{j=1}^{k} x_{sj}, x_{sj} \geq 0, s \in S_m \qquad (2)$$

$$x_s = x_{s1}, x_{s1} \geq 0, s \in S_s \qquad (3)$$

$$0 \leq x_s \leq m_s, s \in S \qquad (4)$$

$$\sum_{s,j:l \in L_{sj}} x_{sj} \leq c_l, \forall l \in L \qquad (5)$$

where (5) describes the link capacity constraints. Existing multpath rate controllers typically require the utility function to be strictly concave and $\lim_{x \to 0} U(x) = -\infty$ [22], [28], [29], [30], [31], [32], [33], [34]. When the utility function is strictly concave, there exists a unique optimal solution for source rate x_s in problem $\mathbf{P}$. Otherwise, there may exist multiple rates that achieve the same maximization of (1).

IV. Overlay Paths selection

We first look at the problem of path selection. Namely, for each multipath source and a given a rate controller, how to choose the overlay paths such that the aggregate source utility is maximized. We prove that this problem is NP-hard even in an extremely simple setting, referred to as *single-receiver 2nd-hop-constrained setting*. In this setting, all sources are multipath sources and have the same receiver. Furthermore, only a single relay is allowed on each overlay path and the 2nd hop (i.e., from the relays to the destination) constrains the source rates. Last, the bandwidths from the relays to the destination are fixed and not correlated. In this setting, selecting an overlay path for a source is equivalent to selecting a relay, which is similar to load balancing (see, e.g., [35], [36]). However, existing work on load balancing chooses a single relay for a source instead of distributing the load of the source onto multiple relays simultaneously.

Let a_r denote the available bandwidth on the path from relay r ($r \in R$) to the receiver, referred to as relay r's bandwidth. The complexity results on optimal path selection are summarized in Theorem 1, Theorem 2 and Corollary 1.

Theorem 1: In a single-receiver 2nd-hop-constrained setting, when all sources have the same maximum source rate over the overlay paths (i.e., $m_s = m, \forall s \in S_m$), for any given rate controller, the problem of optimal path selection is NP-hard. In particular, when $k \geq 3$, this problem is strongly NP-hard.

Proof: Consider a setting in which $|S_m| = 2$, $|R| = 2k$, $m_s = m$, $\forall s \in S$, and $\sum_{r \in R} a_r = m|S_m|$. The optimal relay selection is that each source chooses two disjoint sets R_1 and R_2, $R_1 \subset R, R_2 \subset R$, $\sum_{r \in R_1} a_r = \sum_{r \in R_2} a_r = m$. In this case, all source demands are satisfied using any rate controller. This is the Partition problem and is NP-hard.

We next prove that, when $k = 3$, the above problem is strongly NP-hard. We prove this by reducing 3-Partition to this problem. Consider a setting in which $m_s = m, \forall s \in S_m$, $|R| = 3|S_m|$, $k = 3$, $\sum_{r \in R} a_r = m|S_m|$ and $m/4 < a_r < m/2, \forall r \in R$. In the optimal relay selection, each relay must be used and no two sources share a relay. We reduce 3-Partition to this problem. Since $\sum_{r \in R} a_r = m|S_m|$, the optimal solution is to partition R into $|S_m|$ disjoint sets, $R_1, \ldots, R_{|S_m|}$, such that, $\sum_{r \in R_i} a_r = m, 1 \leq i \leq |S_m|$. Note that each set must consist of 3 elements, which is the 3-Partition problem and is strongly NP-hard. Similarly, for $k > 3$, we prove that the above problem is strongly NP-hard by reducing k-Partition to this problem. ■

Theorem 2: In a single-receiver 2nd-hop-constrained setting, when relay bandwidths are the same (i.e., $a_r = a, \forall r \in R$), for any given rate controller, the problem of optimal path selection is NP-hard. In particular, when $k \geq 3$, this problem is strongly NP-hard.

Proof: Consider a setting in which $|R| = 2$, $k = 1$, $a_r = a, \forall r \in R$, and $\sum_{s \in S_m} m_s = 2a$. The optimal relay selection is to divide S_m into two disjoint sets S_1 and S_2, such that, $\sum_{s \in S_1} m_s = \sum_{s \in S_2} m_s = a$. In this case, all sources attain their maximum sending rate using any rate controller. This is the Partition problem and is NP-hard.

We next prove that, when $k = 3$, the above is strongly NP-hard. We prove this by reducing 3-Partition to this problem. Consider a setting in which $a_r = a, \forall r \in R$, $|S_m| = 3|R|$, $k = 3$, $\sum_{s \in S_m} m_s = a|R|$ and $a/4 < m_s < a/2, \forall s \in S_m$. We reduce 3-Partition to this problem. Since $\sum_{s \in S_m} m_s = a|R|$, the optimal solution is to partition S into $|R|$ disjoint sets, $S_1, \ldots, S_{|R|}$, such that, $\sum_{s \in S_i} m_s = a, 1 \leq i \leq |R|$. In this case, all sources attain their maximum sending rate using any rate controller. Note that since $a/4 < m_s < a/2$, S_i must contain 3 elements, $1 \leq i \leq |R|$, which is 3-Partition problem and is strongly NP-hard. Similarly, for $k > 3$, we prove the above problem to be strongly NP-hard by reducing k-Partition to this problem. ■

Corollary 1: For a given rate controller, the problem of optimal path selection is NP-hard. This problem remains NP-hard in the extremely simple single-receiver 2nd-hop-constrained setting.

In practice, the problem of optimal path selection is even more complicated since (1) multipath and single-path sources may interact with each other when sharing underlay links in the network; (2) accurate and up-to-date inference of the network (e.g., available bandwidth on an end-to-end path) is difficult to obtain. Existing work has demonstrated the benefits of randomized path selection [13]. Therefore, in our performance evaluation (Section VII), we use two randomized algorithms for path selection. Developing efficient distributed path-selection algorithms is left as future work.

V. Multipath Rate Controller

We now consider the problem of multipath rate control. Namely, after selecting paths, how to control the sending rates on the multiple paths to maximize the aggregate source utility. Ideally, the sources should coordinate with each other to maximize the aggregate utility. We refer to this type of controller as *coordinated controller*. In practice, however, it is much easier to compose a multipath rate controller from single-path rate controllers (e.g., TCP) as follows: on each path, the sending rate is determined by a single-path rate controller; in addition, the source regulates the path rates such that the source rate does not exceed the maximum value. We refer to this form of controller as *uncoordinated controller* since each source regulates its path rates independently. In the following, we first briefly review existing coordinated controllers and then develop two uncoordinated controllers that are easy to implement. At the end of this section, we describe the properties of coordinated and uncoordinated controllers.

A. Coordinated Controllers

Centralized multipath rate control requires knowledge of the entire network (i.e., topology, link capacities, routes of every source), which is clearly unrealistic. Distributed multipath rate controllers that require no global knowledge have been proposed in [22], [28], [29], [30], [31], [32], [33], [34]. The key idea of these algorithms is as follows. Network routers compute congestion prices and feed these prices back to the sources. Based on the congestion price on each path, a source adjusts its source rate and the path rates to their optimal values. Communicating the link congestion prices explicitly requires support from the network and introduces communication overheads. An alternative is to allow the sources to infer the aggregate congestion price on a path (e.g., through end-end queuing delay). However, accurate inference of the congestion price is not directly supported by current transport protocols; measurement of the congestion price at the application level is also difficult due to additional delays with the operating system and incur control overhead.

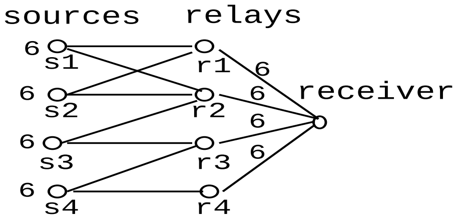


Fig. 2. An example network to illustrate the rate allocations of UC-maxmin and UC-maxflow, $k = 2$. The bandwidths from sources to relays are all 24 Mbps and do not constrain the source rates.

B. Uncoordinated Controllers

Uncoordinated controllers are much easier to implement than coordinated ones. We design two uncoordinated controllers, *UC-maxmin* and *UC-maxflow*, each running a single-path rate controller on a per-logical-hop basis. Neither controller requires explicit network knowledge (e.g., topology, available bandwidth) or any additional support from the network. Therefore, they are readily deployable. However, as we shall see, they do not necessarily maximize the aggregate utility due to the fact that they are composed from single-path controllers.

Both UC-maxmin and UC-maxflow can be applied to general network settings. In the following, for ease of understanding, we illustrate the rate allocations produced by these two controllers using a simple network depicted in Fig. 2. All sources are multipath sources with the demand of 6 Mbps. The bandwidth from a relay to the destination is 6 Mbps. Each source uses two overlay paths. The bandwidths from sources to relays are all 24 Mbps, and hence do not constrain the source rates. We next describe these two controllers and their realization using TCP (the predominant single-path rate controller in the current Internet).

1) UC-maxmin: In UC-maxmin, the sources control their path rates independently as follows. When a source has data to send, it cycles through its overlay paths in a round-robin fashion and sends a unit of data (e.g., a fixed-size packet) on a path that can send. We name this controller UC-maxmin because it is similar to the standard max-min flow control algorithm [37], [38], which allocates bandwidths to be as equal as possible subject only to the link capacities. More specifically, cycling over the paths in a round-robin fashion and sending a data unit when possible in UC-maxmin is similar to increasing the path rates linearly when the paths are not saturated (i.e., the ”filling” procedure in max-min flow control). The rate allocation using UC-maxmin reaches a steady state when all sources are satisfied or all overlay paths of unsatisfied sources are saturated.

When using UC-maxmin, in the steady state, the rate allocation for the network in Fig. 2 is as follows. The sending rates of sources s_1 and s_2 are identical: with a rate of 3 Mbps and 2 Mbps via relays r_1 and r_2 respectively. The sending

```
$P_{sj}(n) = P_{sj}(n-1)$, $j = 1, \ldots, k$
$X_{sj}(n) = X_{sj}(n-1)$, $j = 1, \ldots, k$
$h = H_s(n-1)$
**if** (path $h$ is congested) {
  $P_{sh}(n) = P_{sh}(n-1)/2$
  $X_{sh}(n) = X_{sh}(n-1) - \epsilon$
  Normalize $P_{sj}(n)$, $j = 1, \ldots, k$ s.t. $\sum_{j=1}^{k} P_{sj}(n) = 1$
  Randomly select a path (other than $h$) according to
  $P_{sj}(n)$, $j = 1, \ldots, k$
}
**else** {
  $P_{sh}(n) = \min(2P_{sh}(n-1), 1)$
  Normalize $P_{sj}(n)$, $j = 1, \ldots, k$ s.t. $\sum_{j=1}^{k} P_{sj}(n) = 1$
  Randomly select a path according to $P_{sj}(n)$, $j = 1, \ldots, k$
}
Record the chosen path as $H_s(n)$
$h = H_s(n)$
$X_{sh}(n) = \min(X_{sh}(n-1) + \epsilon, M_s)$
**if** ($\sum_{j=1}^{k} X_{sj}(n) > M_s$) {
 Normalize $X_{sj}(n)$, $j = 1, \ldots, k, j \neq h$
 s.t. $\sum_{j=1}^{k} X_{sj}(n) = M_s$
}
```

Fig. 3. UC-maxflow: an arbitrary source s determines its path rates in the n-th control interval, $s \in S_m$.

rate from source s_3 via relays r_2 and r_3 is 2 and 3 Mbps respectively. The sending rates from source s_4 via relays r_3 and r_4 are both 3 Mbps. Only source s_4 is satisfied. The aggregate sending rate over all sources is 21 Mbps. In this example, UC-maxmin does not maximize the aggregate source utility (In the optimal rate allocation, all sources attain their maximum source rates).

Implementing UC-maxmin using TCP is straightforward. For each source, a TCP connection is established on each logical hop. The TCP receiver of one logical hop is the TCP sender of its next logical hop. When one logical hop is saturated, it back-pressures its previous hop such that the throughput on an overlay path is the minimum throughput over all logical hops on the path. When the source has data to send, it cycles through the TCP sockets on the first logical hop in a round-robin fashion, finds a TCP socket that is writable, and writes a unit of data into that socket.

2) UC-maxflow: When $k = 1$, UC-maxflow is the same as UC-maxmin. When $k \geq 2$, UC-maxflow differs from UC-maxmin in that, based on an initial rate allocation, each source independently probes for overlay paths with spare bandwidths and increases its sending rates on those paths, as described below.

In UC-maxflow, each source divides time into control intervals and sends data in units (a unit can be a fixed-size packet). The lengths of the control interval for different sources need not to be the same. Let M_s represent the maximum number of units that source s sends in a control interval, $s \in S_m$. In each control interval, a source probes network bandwidth by randomly selecting a path and increasing its path rate by ϵ units (the sending rates on the other paths may need to be reduced so that the source rate does not exceed the maximum value). Fig. 3 describes how an arbitrary source $s \in S_m$ adjusts its path rates in the n-th control interval. In the n-th control interval, let $P_{sj}(n)$ represent the probability that source s chooses path j, let $H_s(n)$ denote the path that source s selects for bandwidth probing, and let $X_{sj}(n)$ denote the number of data units that source s sends on the j-th path. Initially, $X_{sj}(0)$ and $P_{sj}(0)$ can be set to any valid values. At the beginning of the n-th control interval, if source s finds that the rate increment in the previous interval, i.e., on path $H_s(n-1)$, leads to congestion on that path, the probability associated with that path is reduced by half; otherwise, the probability is doubled (not exceeding 1). In either case, the probabilities for all the paths are then normalized so that the sum of the probabilities is 1. Afterwards, a path is selected randomly for bandwidth probing based on the normalized probability.

We name this multipath controller UC-maxflow since it achieves the maximum aggregate flow rate when the sources have the same destination and each overlay path allows a single relay, as stated in the following theorem.

Theorem 3: When assuming perfect congestion detection, UC-maxflow converges to a rate allocation that maximizes the aggregate source rate when all sources have the same destination and each overlay path allows a single relay.

Proof: The detailed proof is in Appendix I. ■

When using UC-maxflow, the rate allocation for the network in Fig. 2 is as follows. Suppose that the initial rate allocation is obtained using UC-maxmin. We only describe one possible rate adjustment sequence. Source s_4 shifts its data gradually from relay r_3 to r_4 by 1 Mbps. Correspondingly, source s_3 increases its sending rate to relay r_3 by 1 Mbps and becomes satisfied. Then s_4 shifts its data gradually from relay r_3 to r_4 by 1 Mbps; s_3 shifts its data from relay r_2 to r_3 by 1 Mbps; and source s_2 increases its sending rate to relay r_2 by 1 Mbps and becomes satisfied. This process continues. Eventually, sources s_1 and s_2 have sending rates of 3 Mbps via relays r_1 and r_2; source s_3 has sending rates of 0 and 6 Mbps via relays r_2 and r_3 respectively; and source s_4 has sending rates of 0 and 6 Mbps via relays r_3 and r_4 respectively. Therefore, in this example, all sources are satisfied and UC-maxflow maximizes the aggregate source rate and utility.

One key problem to realizing UC-maxflow using TCP is how to detect whether a rate increment on a path causes congestion. One method is as follows. For source $s \in S$, let $x_{sj}(n)$ and $y_{sj}(n)$ denote respectively the sending rate and goodput on the j-th path in the n-th control interval. The goodput $y_{sj}(n)$ is measured at the receiver and transmitted back to the source at the end of the n-th control interval. Suppose that source s increases the rate on path h in the n-th control interval. Then we say that the rate increment does not cause congestion on path h iff $y_{sh}(n)/x_{sh}(n) \geq 1 - \delta$. Here δ is a small positive constant, chosen to accommodate measurement noises and network delay.

C. Properties of multipath rate controllers

Fairness Properties. We first consider fairness properties of coordinated and uncoordinated controllers. In particular, we look at the following setting. Consider an arbitrary multipath source, $s \in S_m$, using k overlay paths and either a coordinated or uncoordinated controller. The paths are indexed from 1 to k. Recall that, for source s, x_s and x_{sj} denote the source rate and the path rate on the j-th path respectively, and m_s denotes the maximum source rate. On each of the paths, there also exists a single-path source controlled by a single-path controller with no bound on its maximum source rate. For the single-path source on the j-th path, let T_j denote its source (path) rate. The fairness properties that we investigate include (i) how does the source rate of the multipath source compare to those of the single-path sources (i.e., x_s versus T_j, $j = 1, \ldots, k$)? (ii) on each path, how does the path rate of the multipath source compared to that of the single-path source on that path (i.e., x_{sj} versus T_j, $j = 1, \ldots, k$). The results are summarized as follows.

Theorem 4: In the setting described above, when the utility function is strictly concave, (i) under a coordinated controller, $x_s = \min(m_s, \max_{1 \le j \le k} T_j)$ and $x_{sj} \le T_j$, $j = 1, \ldots, k$; (ii) under an uncoordinated controller, $x_s = \min(m_s, \sum_{j=1}^{k} T_j)$ and $x_{sj} \le T_j$, $j = 1, \ldots, k$.

Proof: The detailed proof is in Appendix II. ■

The above result indicates that coordinated controllers exhibit a desired fairness property: the source rate of a multipath source is no more than the maximum rate of the single-path sources over all the paths. Under uncoordinated controllers, the fairness achieved is less ideal. However, the source rate of a multipath source is bounded by the aggregate sending rate of the single-path sources over all paths.

Aggregate-source-rate properties. We now describe the properties of coordinated and uncoordinated controllers in terms of the aggregate source rate. For coordinated controllers, we prove that they maximize the aggregate source rate In a single-receiver 2nd-hop-constrained setting (defined in Section IV).

Theorem 5: In a single-receiver 2nd-hop-constrained setting, for a path selection, a coordinated controller maximizes the aggregate source rate. Furthermore, the aggregate source rate is non-decreasing when one or multiple sources incrementally add more overlay paths.

Proof: The detailed proof is in Appendix III. ■

We have proved in Theorem 3 that UC-maxflow maximizes the aggregate source rate when all multipath sources have the same destination and each overlay path allows a single relay. We now state a theorem on the property of UC-maxflow when increasing the number of overlay paths for the sources.

Theorem 6: Under the same conditions in Theorem 3, the aggregate source rate under UC-maxflow is non-decreasing when one or multiple sources incrementally add more overlay paths.

Proof: This is directly from Theorem 3 that UC-maxflow maximizes the aggregate source rate under the given conditions. When a source adds an additional overlay path, the sending rate on this path is allowed to be non-zero, which relaxes the constraints of the maximization problem, and hence leads to a higher (or equal) aggregate source rate. ■

VI. Iterative Path Selection

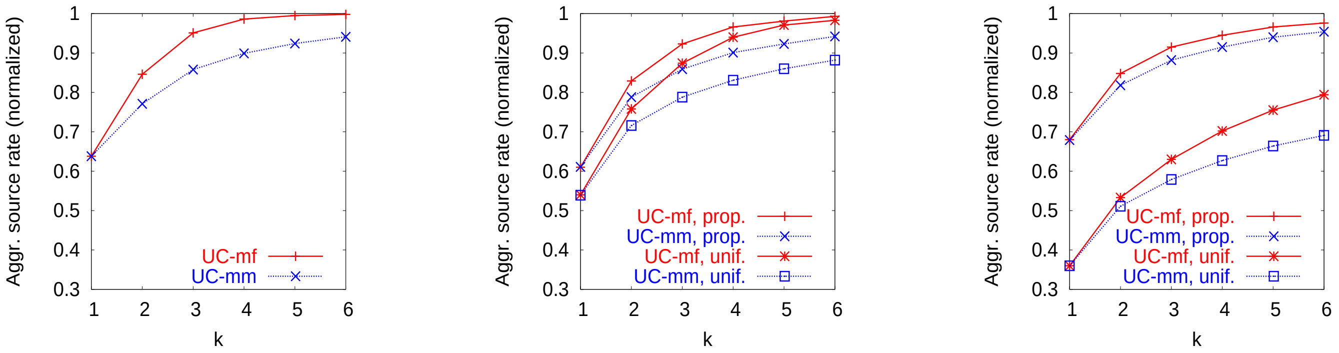
Due to the complexity of optimal path selection, we propose to use an iterative approach for path selection (when allowing path reselection) as follows. In the first iteration, all sources randomly select paths and run a multipath rate controller (either coordinated or uncoordianted) to obtain a rate allocation. In the next iteration, the sources that are not satisfied in the previous iteration randomly reselect paths. The iteration continues until all sources are satisfied. We choose to use randomized path selection based on the observations in [13]. In particular, we consider two randomized algorithms (initial path selection and path reselection use the same randomized algorithm): *uniform choice rule* and *proportional choice rule*. In uniform choice rule, a source uniformly chooses k distinct paths. In proportional choice rule, a source chooses an overlay path with a probability proportional to the bandwidth on that overlay path. Note that the uniform rule requires no knowledge of the network while the proportional choice rule requires knowing the bandwidth on each of the overlay paths (e.g. estimated using techniques in [39]).

Depending on the number of paths reselected by an unsatisfied source, we propose two iterative path selection schemes: k-*path-reselection* and 1-*path-reselection*. In the former, an unsatisfied source randomly (using uniform or proportional rule) reselects all of its k paths. In the latter, an unsatisfied source only randomly reselects one path to replace the one with the minimum rate in the previous iteration. Note that both k-path-reselection and 1-path-reselection can be easily implemented in an asynchronous manner, that is, the sources do not need to synchronize their iterations. We prove the following convergence property for k-path-reselection and conjecture the same convergence result holds for 1-path-reselection.

Theorem 7: In a general network, when all source demands are the same (i.e., $m_s = m, \forall s \in S_m$) and there exists a path selection such that all sources are satisfied, k-path-reselection converges to find such a path selection.

Proof: The proof is in Appendix IV. ■

We also compare the above two iterative schemes with a baseline scheme called *All-source-reselection*, in which all sources (including satisfied and unsatisfied) randomly choose k paths until all sources are satisfied. Proving that All-source-reselection converges (i.e., finds a path selection to satisfy all sources if such a path selection exists) is straightforward, since the probabilities of all combinations of relay selection are positive when using All-source-reselection. However, this baseline scheme has the drawback that it is difficult to realize in practice since all sources need to have a synchronized clock and a source needs to know whether the other sources are satisfied. We compare the performance of k-path-reselection, 1-path-reselection and All-source-reselection in Section VII.



(a) $\beta = 0$

(b) $\beta = 0.5$

(c) $\beta = 1$

Fig. 4. Aggregate source rate (normalized) versus k when using uniform or proportional choice rule, $\alpha = 1$. The results for coordinated controller overlap with those from UC-maxflow and therefore omitted. Confidence intervals are tight and omitted.

VII. Perfomance evaluation

In this section, we evaluate the performance of path-selection algorithms in combination with multipath rate-control controllers. Our performance evaluation is in the single-receiver 2nd-hop-constrained setting (defined in Section IV). We choose this setting for two reasons. First, existing studies have demonstrated the benefits of using a single relay on an overlay path [13], [14], [15]. Second, this setting is a non-trivial baseline: optimal path selection in this setting remains NP-hard (Section IV) and rate control is necessary (since the multipath sources share the relays and the second hop is bandwidth constrained). We therefore expect that evaluation results in this setting will provide insights on behaviors in more general settings.

Let a_r represent relay r's bandwidth to the receiver. Recall that m_s denotes the maximum source rate (demand) of source s. Let $\alpha = \sum_{r \in R} a_r / \sum_{s \in S_m} m_s$, that is, α represents the ratio of network bandwidth over the aggregate source demands. In our performance evaluation, we vary α from 0.4 to 3. We set $|S_m| = |R| = 100$ and $m_s = 1$ Mbps, $\forall s \in S_m$. The bandwidth from the j-th relay to the receiver is proportional to $1/j^{\beta}$, where $0 \leq \beta \leq 1$. We refer to β as the *skew factor*. When $\beta = 0$, all relays have the same bandwidth. As β increases, the bandwidth distribution among the relays becomes more skewed.

Our performance evaluation is through both numerical study and simulation using the *ns-2* simulator. The numerical study assumes an idealized environment (i.e., no network delay and perfect flow interaction) while the simulation takes into account of practical issues (e.g., network delay, packetized network flows, and bursty packets transmission). Unless otherwise specified, our results from numerical study and simulation are averaged over 30 and 10 runs respectively. The confidence intervals are tight and hence omitted from the plotted results. Our performance metric is the aggregate source rate normalized by the aggregate source demands, i.e., $\sum_{s \in S} x_s / \sum_{s \in S_m} m_s$. We use aggregate source rate instead of the aggregate source utility because the former is more intuitive. When all sources are satisfied (i.e., the normalized aggregate source rate is one), the maximum aggregate utility is achieved. In the single-receiver 2nd-hop-constrained setting, UC-maxflow and coordinated controller both achieve the maximum aggregate source rate for a given path selection (proved in Theorem 3 and 5 respectively).

We next describe the results from the numerical and simulation studies respectively. In both studies, we consider two scenarios: not allowing and allowing path reselection. We use uniform and proportional choice rules for path selection (defined in Section VI). When allowing path reselection, we use iterative path selection schemes, k-path-reselection, 1-path-reselection and All-source-reselection (see Section VI).

A. Numerical study

Not allowing path reselection. When not allowing path reselection, each source selects path once and then runs a multipath rate controller to determine its path rate. We first explore the effect of increasing k by incrementally adding relays to each source. Fig. 4 plots the normalized aggregate source rate versus k, under UC-maxmin and UC-maxflow. The results for coordinated controller overlap with those from UC-maxflow and therefore omitted. In the figure, the aggregate relay bandwidth equals the aggregate source demand (i.e., $\alpha = 1$) and the skew factor, β, is 0, 0.5 or 1. Results under both uniform and proportional choice rules are shown in the figure. Note that when $\beta = 0$, relay bandwidths are homogeneous and the results under uniform and proportional choice rules coincide. From Fig. 4, we note that the aggregate source rate increases with k under all controllers. This is expected for coordinated controllers and UC-maxflow (proved in Theorem 5 and Theorem 6 respectively). Under UC-maxmin, although increasing k may lead to lower aggregate source rate in a single run, we observe that, on average, the aggregate source rate increases with k. On the other hand, there is a diminishing gain from increasing k on the aggregate source rate. This indicates that small values of k (i.e., 2 to 4) can realize most of the performance gains.

From Fig. 4, we observe that when $\beta = 0.5$, proportional choice rule only leads to slight performance gains compared to uniform choice rule. The performance improvement is much

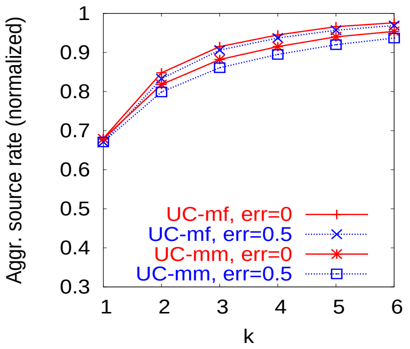


Fig. 5. Aggregate source rate (normalized) using proportional choice rule with and without estimation error, $\beta = 1$, and $\alpha = 1$.

more dramatic under a highly skewed bandwidth distribution (i.e., $\beta = 1$). This indicates that when relay bandwidths are not very skewed, it may not be worthwhile to make the relay bandwidth estimates needed by the proportional choice rule. This bandwidth estimation, however, may be very beneficial when relay bandwidths are highly skewed. Existing bandwidth estimation techniques typically exhibit estimation errors. Therefore, we also investigate the performance of proportional choice rule in the presence of bandwidth estimation errors. We assume that the relative estimation error is uniformly distributed in $[0, err]$, where err denotes the maximum relative estimation error. We found that proportional choice rule is not sensitive to estimation errors: the performance degradation is negligible when allowing up to a 20% estimation error; even an estimation error up to 50% only degrades the performance slightly. One comparison result between perfect bandwidth estimation and estimation error up to 50% is shown in Fig. 5, where $\beta = 1$ and $\alpha = 1$.

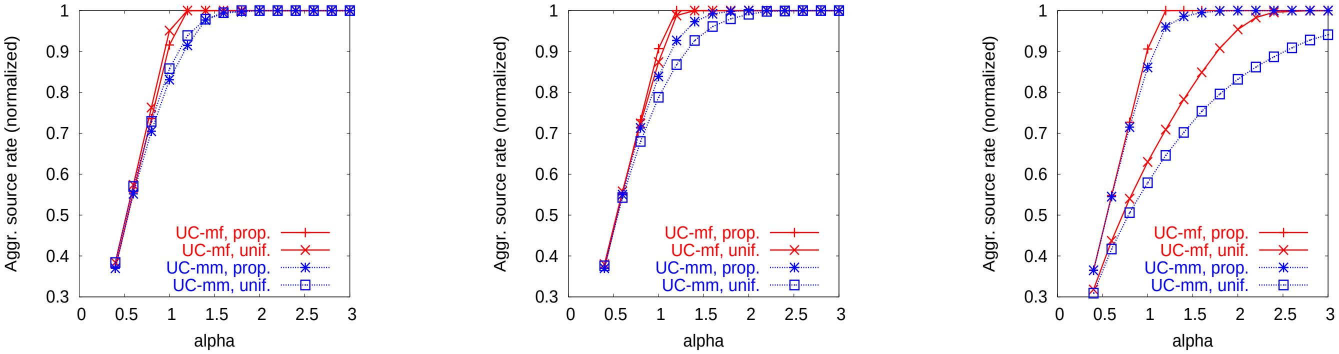
We now vary α, the ratio of aggregate relay bandwidth over aggregate source demand, from 0.4 to 3. Fig. 6 plots the normalized aggregate source rate versus α when $k = 3$ using uniform and proportional choice rules. The maximum relative bandwidth estimation error when using proportional choice rule is 50%. Again, the results for coordinated controllers coincide with those for UC-maxflow and are not plotted. We observe a diminishing gain from increasing α on performance: as α increases from 0.4 to 1.0, the performance improves dramatically and the improvement is less dramatic afterwards. Furthermore, under UC-maxflow, the aggregate source rate approaches the maximum value (i.e., the normalized value approaches 1 and all source demands are satisfied) when α is above 2 using both uniform and proportional choice rules. This is also true under UC-maxmin except for one case (i.e., when relay bandwidth highly skewed and using uniform choice rule). We also observe that, although coordinated controllers and UC-maxflow outperform UC-maxmin, the difference is significant only in a narrow range (α from 1 to 2). The simple multipath rate controller UC-maxmin performs reasonably well in a wide range of settings under a proper choice of relays. Therefore, UC-maxmin may be an attractive choice in practice.

Allowing path reselection. When allowing path reselection, we examine the performance of two practical iterative path selection schemes, k-path-reselection and 1-path-reselection, and a baseline scheme, All-source-reselection. Fig. 7 plots the number of iterations required to find a path selection that satisfies all sources versus α using UC-maxmin, uniform choice rule when $\beta = 0.5$ and $k = 2, 3, 4$. The 95% confidence interval is obtained from 30 runs. We observe that baseline scheme, All-source-reselection, converges more slowly than the two practical schemes, k-path-reselection and 1-path-reselection. The performance of 1-path-reselection is similar to that of k-path-reselection, with slightly worse performance under small values of α. We observe similar comparative performance among these three iterative schemes under proportional choice rule and other multipath rate controllers. This demonstrates that k-path-reselection and 1-path-reselection are not only easy to implement but also achieve better performance than the impractical All-source-reselection.

B. Simulation results

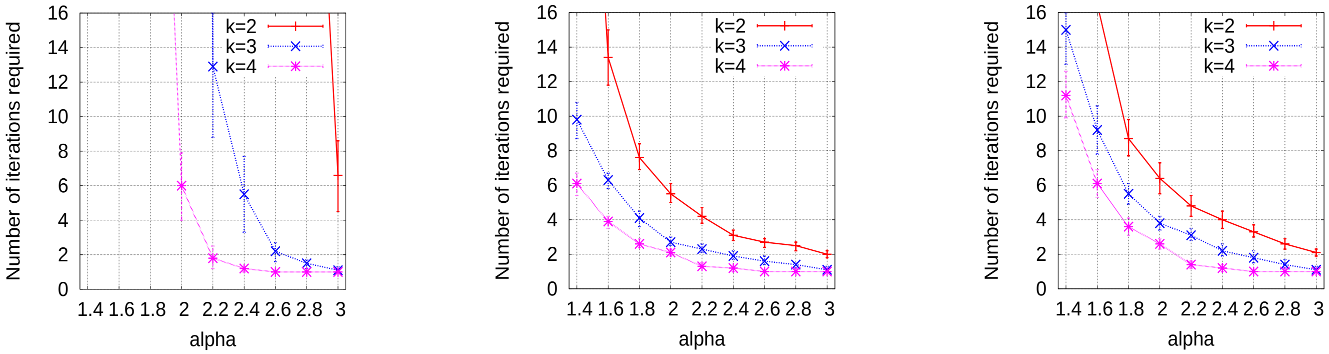
In our simulation, the round-trip propagation delays from a source to a relay and from a relay to the receiver are both set to 20 ms. We first describe how we set the various parameters for UC-maxmin and UC-maxflow. There is a clear tradeoff in choosing the size of a data unit. A data unit should be sufficiently large compared to the size of a packet header. However, when using a very large unit size, the spare bandwidth on an overlay path might not be fully utilized. In our implementation, we set unit size to 500 bytes. When implementing UC-maxflow, we set the length of the control interval for a source to 0.4 or 0.8 second. When randomly probing for bandwidth (see Section V), a small increment, ϵ, leads to slow detection of spare bandwidth, while a large ϵ may lead to congestion in the network. We set ϵ to 1 or 2 units. When detecting congestion along a path, we set the threshold δ to 0.01 or 0.03.

We observe similar results as those in the numerical study when increasing k and α. We next illustrate the processes in which UC-maxmin and UC-maxflow converge to their steady-state rate allocations. Figures 8(a) and (b) plot the normalized aggregate source rate versus time under UC-maxmin and UC-maxflow respectively for one simulation run, using uniform choice rule. In Fig. 8(a), at time 0, each source selects a single relay (i.e., $k = 1$). Then after every 60 seconds, each source adds one relay (i.e., increasing k by 1). The numerical and simulation results are both plotted in the figure. For each k, the numerical result is a horizontal line. We observe a good match between the numerical and simulation results. The throughput fluctuations in the simulation may be due to network delays and the packetized nature of flows. We also observe that, for each k, the steady state of rate allocation is reached very quickly. Fig. 8(b) plots the results for UC-maxflow. Each source adds one relay every 500 seconds. When $k = 1$, UC-maxflow is identical to UC-maxmin. For $k \geq 2$, we run UC-maxmin in the first 30 seconds to obtain an initial



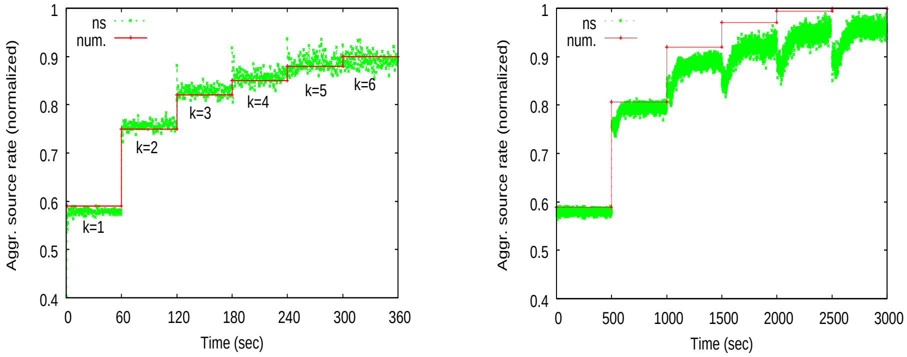
(a) $\beta = 0$ (b) $\beta = 0.5$ (c) $\beta = 1$

Fig. 6. Aggregate source rate (normalized) versus α using uniform or proportional choices of relays when $k = 3$. The results for coordinated controller overlap with those from UC-maxflow and therefore omitted. Confidence intervals are tight and omitted.



(a) All-source-reselection. (b) k-path-reselection. (c) 1-path-reselection.

Fig. 7. Number of iterations required to find a path selection that satisfies all sources versus α using UC-maxmin and uniform choice rule, $\beta = 0.5$.



(a) UC-maxmin (b) UC-maxflow

Fig. 8. The aggregate source rate as k increases over time using uniform choice rule, $\beta = 0.5$, $\alpha = 1$ for one simulation run in *ns*. For UC-maxflow, the control interval is 0.4 second, $\epsilon = 2$ units, and $\delta = 0.03$.

rate allocation for UC-maxflow. We observe a linear region in which the aggregate source rate converges to the steady-state value. As expected, the convergence rate under longer control intervals and smaller rate increments is slower. A formal study on the convergence rate is part of our future work. We also observe that the aggregate source rate from simulation is slightly lower than that predicted numerically. This discrepancy might be caused by imprecise detection of network congestion.

VIII. Conclusions and Future Work

In this paper, we have considered multipath overlay data transfer where a source sends its data over k $(k \geq 1)$ overlay paths to its destination. We studied how to select overlay paths and determine the sending rate on each path in order to maximize the aggregate source utility. We proved that the problem of optimal path selection is NP-hard for any given rate controller. For multipath rate control, we developed two practical application-level rate controllers, UC-maxmin and UC-maxflow, that use TCP as the transport protocol on a per-logical-hop basis. Our evaluation showed that the simple controller UC-maxmin combined with a proper choice of relays performed reasonably well in a wide range of settings. Furthermore, a small number of relays (2 to 4) and a small amount of extra bandwidth in the network are sufficient to realize most of the performance gains. As future work, we are pursuing in the following directions: (1) designing efficient distributed path selection algorithms; (2) performance evaluation in more general settings and in a testbed; (3) investigation of the interaction and the fairness among multiple multipath sources.

References

[1] *Engineering Research Center for Collaborative Adaptive Sensing of the Atmosphere*. http://www.casa.umass.edu.

[2] J. Gray and A. S. Szalay, "The world-wide telescope, an archetype for online science," Tech. Rep. MSR-TR-2002-75, Microsoft Research, June 2002.

[3] A. Akella, S. Seshan, and A. Shaikh, "An empirical evaluation of wide-area internet bottlenecks," in *IMC*, (Miami, Florida), 2003.

[4] A. Akella, B. Maggs, S. Seshan, A. Shaikh, and R. Sitaraman, "A measurement-based analysis of multihoming," in *Proc. ACM SIGCOMM*, August 2003.

[5] A. Akella, J. Pang, B. Maggs, S. Seshan, and A. Shaikh, "A comparison of overlay routing and multihoming route control," in *Proc. ACM SIGCOMM*, 2004.

[6] S. Floyd and K. Fall, "Promoting the use of end-to-end congestion control in the Internet," *IEEE/ACM Trans. Networking*, 1999.

[7] G. Baier, E. Kohler, and M. Skutella, "The k-splittable flow problem," in *10th Annual European Symposium on Algorithms*, 2002.

[8] J. Chen, P. Druschel, and D. Subramanian, "An efficient multipath forwarding method," in *Proc. IEEE INFOCOM*, 1998.

[9] S. Vutukury and J. Garcia-Luna-Aceves, "MPATH: a loop-free multipath routing algorithm," *Elsevier Journal of Microprocessors and Microsystems*, pp. 319–327, 2000.

[10] W. T. Zaumen and J. J. Garcia-Luna-Aceves, "Loop-free multipath routing using generalized diffusing computations," in *INFOCOM (3)*, pp. 1408–1417, 1998.

[11] B. Cheng, C. Chou, L. Golubchik, S. Khuller, and Y.-C. Wan, "Large scale data collection: a coordinated approach," in *Proc. IEEE INFOCOM*, (San Francisco, CA), March 2003.

[12] S. Ganguly, A. Saxena, S. Bhatnagar, S. Banerjee, and R. Izmailov, "Fast replication in content distribution overlays," in *Proc. IEEE INFOCOM*, (Miami, FL), March 2005.

[13] K. P. Gummadi, H. V. Madhyastha, S. D. Gribble, H. M. Levy, and D. J. Wetherall, "Improving the reliability of internet paths with one-hop source routing," in *Proceedings of the 6th Symposium on Operating Systems Design and Implementation*, (San Francisco, CA), December 2004.

[14] J. Han, D. Watson, and F. Jahanian, "Topology aware overlay networks," in *Proc. IEEE INFOCOM*, March 2005.

[15] H. Pucha and Y. C. Hu, "Overlay TCP: Ending end-to-end transport for higher throughput," in *Proc. ACM SIGCOMM*, August 2005.

[16] A. Orda and R. Rom, "Multihoming in computer networks: A topology-design approach.," *Computer Networks and ISDN Systems*, vol. 18, no. 2, pp. 133–141, 1989.

[17] H. Wang, H. Xie, L. Qiu, A. Silberschatz, and Y. R. Yang, "Optimal ISP subscription for internet multihoming: Algorithm design and implication analysis," in *Proc. IEEE INFOCOM*, March 2005.

[18] A. Dhamdhere and C. Dovrolis, "ISP and egress path selection for multihomed networks," in *Proc. IEEE INFOCOM*, (Barcelona, Spain), April 2006.

[19] F. Guo, J. Chen, W. Li, and T. cker Chiueh, "Experiences in building a multihoming load balancing system," in *Proc. IEEE INFOCOM*, 2004.

[20] D. Goldenberg, L. Qiu, H. Xie, Y. R. Yang, and Y. Zhang, "Optimizing cost and performance for multihoming," in *Proc. ACM SIGCOMM*, August 2005.

[21] F. Kelly, "Charging and rate control for elastic traffic," *European Transactions on Telecommunications*, vol. 8, pp. 33–37, January 1997.

[22] F. Kelly, A. Maulloo, and D. Tan, "Rate control in communication networks: shadow prices, proportional fairness and stability," in *Journal of the Operational Research Society*, vol. 49, 1998.

[23] S. Kunniyur and R. Srikant, "End-to-end congestion control schemes: Utility functions, random losses and ECN marks," in *INFOCOM 2000*, 2000.

[24] R. J. Gibbens and F. P. Kelly, "Resource pricing and the evolution of congestion control," *Automatica*, vol. 35, pp. 1969–1985, 1999. http://www.statslab.cam.ac.uk/∼frank/PAPERS/evol.html.

[25] S. H. Low and D. Lapsley, "Optimization flow control – I: Basic algorithm and convergence," *IEEE/ACM Transactions on Networking*, vol. 7, pp. 861–874, December 1999.

[26] K. Kar, S. Sarkar, and L. Tassiulas, "A simple rate control algorithm for maximizing total user utility," in *Proc. IEEE INFOCOM*, pp. 133–141, 2001.

[27] R. J. La and V. Anantharam, "Charge-sensitive TCP and rate control in the Internet," in *Proc. IEEE INFOCOM*, 2000.

[28] W.-H. Wang, M. Palaniswami, and S. H. Low, "Optimal flow control and routing in multi-path networks," *Performance Evaluation*, vol. 52, pp. 119–132, 2003.

[29] X. Lin and N. B. Shroff, "The multi-path utility maximization problem," in *41st Annual Allerton Conference on Communication, Control, and Computing*, (Monticello, IL), October 2003.

[30] S. H. Low, "Optimization flow control with on-line measurement," in *Proceedings of the 16th International Teletraffic Congress*, (Edinburgh, U.K.), June 1999.

[31] B. A. Movsichoff and C. M. L. H. Che, "Decentralized optimal traffic engineering in the Internet," *IEEE Journal on Selected Areas in Communications*, 2005.

[32] K. Kar, S. Sarkar, and L. Tassiulas, "Optimization based rate control for multipath sessions," in *Proceedings of Seventeenth International Teletraffic Congress (ITC)*, (Salvador da Bahia, Brazil), December 2001.

[33] R. Srikant, *The Mathematics of Internet Congestion Control*. Springer-Verlag, March 2004.

[34] H. Han, S. Shakkottai, C. Hollot, R. Srikant, and D. Towsley, "Overlay TCP for multi-path routing and congestion control," in *IMA Workshop on Measurements and Modeling of the Internet*, JAN 2004.

[35] D. L. Eager, E. D. Lazowska, and J. Zahorjan, "Adaptive load sharing in homogeneous distributed systems," *IEEE Trans. on Software Engineering*, vol. 12, pp. 662–675, May 1986.

[36] M. Mitzenmacher, "The power of two choices in randomized load balancing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 12, October 2001.

[37] D. P. Bersekas and R. Gallager, *Data Networks*. Prentice Hall, second ed., 1992.

[38] L. Massoulié and J. Roberts, "Bandwidth sharing: Objectives and algorithms," in *Proc. IEEE INFOCOM*, vol. 3, pp. 1395–1403, 1999.

[39] M. Jain and C. Dovrolis, "End-to-end available bandwidth: measurement methodology, dynamics, and relation with TCP throughput," in *Proc. ACM SIGCOMM*, 2002.

[40] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*. The MIT press, 1990.

Appendix I
Proof of Theorem 3

Proof: We prove this theorem by first transforming the rate control problem into a network flow problem [40]. We construct a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ to represent the network we consider as follows. The vertex set $\mathcal{V}$ contains the set of multipath sources S_m, the set of relays R, the destination d and an additional vertex b, referred to as the *origin*. We use (u, v) to represent a directed edge from u to v, $\forall u, v \in \mathcal{V}$. Furthermore, let c_{uv} denote the capacity on the directed edge (u, v). The origin b and each source $s \in S$ is connected by a directed edge (b, s) with the capacity as the demand of the source, that is, $c_{bs} = m_s$. If source $s \in S_m$ selects a relay $r \in R$, then s is connected to r by a directed edge (s, r). The capacity of the edge (s, r), c_{sr}, is the available bandwidth on the path from s to r. A relay $r \in R$ is connected to the

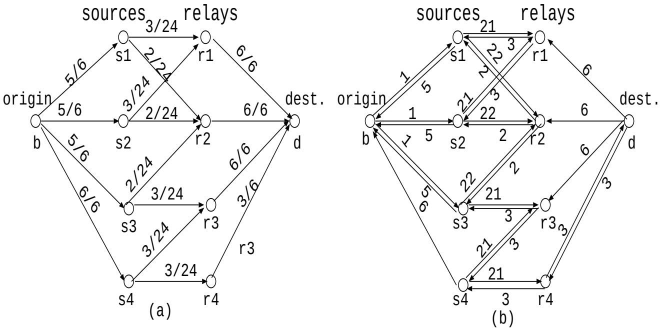


Fig. 9. Illustration of network flow representation when proving Theorem 3. (a) The network flow representation of the network in Fig. 2. On each edge, the slash is used to separate the flow and capacity of this edge. (b) The residual network induced by the network in (a). The residual capacity on each edge is marked on that edge.

destination d by a directed edge (r, d). The capacity of the edge (r, d), c_{rd}, is the available bandwidth on the path from the relay to the destination. In the directed graph $\mathcal{G}$, if two vertices u and v are not connected, i.e., $(u, v) \notin \mathcal{E}$, then $c_{uv} = 0$.

Let $f(u, v)$ be the network flow from vertix u to v. We next describe the how UC-maxflow sets the initial value of the flow between two vertices u and v, $\forall u, v \in \mathcal{V}$. Let x^0_{sj} denote the initial rate allocation on the j-th path of source s, $s \in S_m, j = 1, 2, \ldots, k$. Then $f(b, s) = \sum_{j=1}^{k} x^0_{sj}$. That is, the flow from the origin to source s is the source rate of this source. Let r_{sj} denote the relay used by the j-th overlay path of source s. Then $f(s, r_{sj}) = x^0_{sj}$. That is, the flow from a source to its selected relay is the sending rate on that overlay path. On the edge from a relay r to the destination d, the flow $f(r, d) = \sum_s \sum_{j=1}^{k} \mathbf{1}(r_{sj} = r) x^0_{sj}$, where $\mathbf{1}(\cdot)$ is the indicator function. That is, the flow from a relay to the destination is the aggregate sending rate over all sources that uses that relay to the destination. The flows of all other edges are 0.

We next use an example to illustrate the network flow representation. Fig. 9(a) shows the network flow representation for the network in Fig. 2 assuming that the initial rate allocation is by UC-maxmin. An edge (u, v) is labeled as $f(u, v)/c_{uv}$, where the slash notation is used to separate the flow and the capacity of this edge. For instance, the edge (s_1, r_1) is labeled as $3/24$ since the rate allocated by UC-maxmin is 3 Mbps, and the capacity between s_1 and r_1 is 24 Mbps.

We next show that UC-maxflow has positive probability to find *augmenting paths* in the *residual network* [40] of the network described above. For completeness, we briefly describe residual network and augmenting path. Given a flow network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and the flows between two vertices, a residual network $\mathcal{G}_f$ induced by these flows is $\mathcal{G}_f = (\mathcal{V}, \mathcal{E}_f)$, where $\mathcal{E}_f = \{(u, v \in \mathcal{V} \times \mathcal{V} : c_f(u, v) > 0\}$, where $c_f(u, v)$ is the *residual capacity* of edge (u, v), i.e., $c_f(u, v) = c_{uv} - f(u, v)$. Given the residual network $\mathcal{G}_f$, an augmenting path is a path from the origin b to the destination d in $\mathcal{G}_f$. By the definition of residual network, each edge along an augmenting path can admit positive flow without violating the capacity of this edge. For instance, Fig. 9(b) shows the residual network of the network in Fig. 9(a). The residual capacity on each edge is marked on that edge. One augmenting path is the path over b, s_3, r_3, s_4, r_4,and d.

We represent an augmenting path by the sequence of vertices along the path. Let $\mathcal{P} = (b, s_{i_1}, r_{i_1}, \ldots, s_{i_n}, r_{i_n}, d)$ be an arbitrary augmenting path in the residual network, where $n \geq 1$. Since edge (b, s_{i_1}) can admit positive flow, source s_{i_1} is not satisfied. Let $c_f(\mathcal{P})$ be the minimum residual capacity along this path. That is, $c_f(\mathcal{P}) = \min\{c_f(u, v), (u, v) \in \mathcal{P}\}$. Under perfect detection of network congestion, a source increases its sending rate on a path iff there is spare bandwidth on that path. We next prove that, with perfect congestion detection, there is a positive probability for UC-maxflow to find this augmenting path and increase the flow on the path by $c_f(\mathcal{P})$. We prove this by induction on n.

- Case 1 ($n = 1$). In this case, since source s_{i_1} is not satisfied, there is clearly a positive probability that source s_{i_1} increases its sending rate on the path from s_{i_1} to the destination via relay r_{i_1} by the amount of $c_f(\mathcal{P})$.
- Case 2 ($n > 1$). We first show that it is sufficient to consider augmenting paths in which sources $s_{i_n}, s_{i_{n-1}}, \ldots, s_{i_2}$ are all satisfied. Suppose that source s_{i_n} is not satisfied. Then there is an augmenting path of (b, s_{i_n}, r_{i_n}, d). From Case 1, when using UC-maxflow, there is a positive probability for s_{i_n} to increase its path rate on (s_{i_n}, r_{i_n}, d) until it is satisfied or the path rate cannot be increased any more (i.e., either path (s_{i_n}, r_{i_n}) or path (r_{i_n}, d) is saturated). The former case is desired. In the latter case, path $\mathcal{P}$ is not an augmenting path any more (so we do not need to consider path $\mathcal{P}$ any more). Similarly, we only need to consider augmenting paths in which sources $s_{i_{n-1}}, \ldots, s_{i_2}$ are all satisfied. When sources $s_{i_n}, s_{i_{n-1}}, \ldots, s_{i_2}$ are all satisfied, the aggregate source rate can be increased by $c_f(\mathcal{P})$ when UC-maxflow adjusts the sending rates in the following manner: source s_{i_n} gradually shifts its data from the path $(s_{i_n}, r_{i_{n-1}}, d)$ to path (s_{i_n}, r_{i_n}, d), thus leaving spare bandwidth on the path of $(r_{i_{n-1}}, d)$ and allowing source $s_{i_{n-1}}$ to shift its data from $(s_{i_{n-1}}, r_{i_{n-2}}, d)$ to $(s_{i_{n-1}}, r_{i_{n-1}}, d)$, ..., and allowing source s_{i_1} to increases its sending rate on the path of (s_{i_1}, r_{i_1}, d). This sequence of rate adjustment leads to a rate increment of $c_f(\mathcal{P})$ in the aggregate source rate.

Since path $\mathcal{P}$ is arbitrary, we have proved that UC-maxflow can find any augmenting path in the residual network. UC-maxmin continues the process of finding an augmenting path and adjusting rate along that augmenting path until no augmenting path can be found. This is equivalent to the Ford-Fulkerson algorithm in maximum network flow [40]. Suppose at time T, no augmenting path can be found. Then the maximum aggregate source rate is reached [40]. We prove that later rate changes of UC-maxflow does not lower the aggregate source rate (hence the rate allocation converges) by considering the following two cases:

- Case 1: all relay bandwidths are fully utilized at time

T. The rate allocation does not change in this case, and hence UC-maxflow converges.

- Case 2: not all relay bandwidths are fully utilized at time T. If a relay is not selected by any source, it can be removed without affecting the rate allocation. Therefore, without loss of generality, we assume each relay is selected by at least one source. Consider an arbitrary relay r with spare bandwidth and an arbitrary source s that selects relay r. If source s is satisfied, it may shift its data from other paths to path (s, r, d). However, by the assumption, the shifting occurs iff there is still spare bandwidth on path (s, r, d), which does not affect the sending rate of any other source, and hence does not reduce the aggregate source rate. If source s is not satisfied, then there is no spare bandwidth on the path of (s, r). Otherwise, the sending rate of source s can be increased, which contradicts with that the maximum aggregate source rate has been reached. Under the assumption of perfect bandwidth detection, source s does not increase the rate on path (s, r, d) and hence does not affect the aggregate source rate.

■

Appendix II
Proof of Theorem 4

Before proving Theorem 4, we first state two lemmas on the optimal solution of $\mathbf{P}$ (defined in Section III) since a coordinated controller obtains the optimal solution to this problem. We assume the utility function is strictly concave.

Lemma 1: Let the Lagrangian of $\mathbf{P}$ be:

$$\begin{aligned} L(x, \lambda, p, \mu) &= \sum_{s \in S} (U(x_s) + \lambda_s (m_s - x_s)) \\ &+ \sum_{l \in L} p_l (c_l - \sum_{s,j:l \in L_{sj}} x_{sj}) \\ &+ \sum_{s \in S_m} \sum_{j=1}^{k} \mu_{sj} x_{sj} + \sum_{s \in S_s} \mu_{s1} x_{s1} \end{aligned}$$

where $\lambda_s \geq 0, \forall s \in S$, $p_l \geq 0, \forall l \in L$, and $\mu_{sj} \geq 0$. In particular, p_l which can be interpreted as the *congestion price on link l*. Let $p^{sj} = \sum_{l:l \in L_{sj}} p_l$. That is, p^{sj} is the sum of the link prices on the j-th path of source s, referred to as the *congestion price on the j-th path*. Then the optimal solution of problem $\mathbf{P}$ must satisfy the following:

$$p^{sj} = U'(x_s) - \lambda_s + \mu_{sj}, \forall s \in S_m, j = 1, \ldots, k \quad (6)$$

$$p^{s1} = U'(x_s) - \lambda_s + \mu_{s1}, \forall s \in S_s \quad (7)$$

$$\mu_{sj} x_{sj} = 0, \forall s \in S, j = 1, \ldots, k \quad (8)$$

$$\lambda_s (m_s - x_s) = 0, \forall s \in S \quad (9)$$

$$p_l (c_l - \sum_{s,j:l \in L_{sj}} x_{sj}) = 0, \forall l \in L \quad (10)$$

Proof: This follows directly from the Karush-Kuhn-Tucker Theorem. ■

Lemma 2: For an arbitrary multipath source s using a coordinated controller, the congestion prices on the paths with positive rates must be the same, equal to the minimum congestion price among all paths. Let p_s^* denote this minimum congestion price. Then

$$x_s = \min(U^{'-1}(p_s^*), m_s)$$

Proof: Without loss of generality, assume that at optimality the j-th path has positive sending rate, i.e., $x_{sj} > 0$. Then from (6) to (8), its congestion price is $U'(x_s) - \lambda_s$ and is the minimum congestion price among all paths. The above is true for any path with positive rate. Since p_s^* is the minimum congestion price, $p_s^* = U'(x_s) - \lambda_s$ and $x_s = U^{'-1}(p_s^* + \lambda_s)$. Using (9), this can be written in a more compact form as $x_s = \min(U^{'-1}(p_s^*), m_s)$. ■

We now present the proof of Theorem 4.

Proof: We first prove the result for a coordinated controller. For the multipath source s, by Lemma 2, $x_s = \min(U^{'-1}(p_s^*), m_s)$, where p_s^* denotes the minimum congestion price among all paths. Suppose that source s has n paths with positive path rates, $1 \leq n \leq k$. Without loss of generality, assume that these are the first n paths, that is, path $1, \ldots, n$. That is, $x_{sj} > 0, j = 1, \ldots, n$ and $x_{sj} = 0, j = n + 1, \ldots, k$. Let p_j denotes the congestion price on the j-th path. For the single-path source on the j-th path, since its maximum source rate is not bounded, we have $T_j = U^{'-1}(p_j)$ (derived in a similar manner as in Lemma 2 by looking at (7) to (9)). Since the congestion prices on the first n paths are the minimum and equal to p_s^*, we have $p_j = p_s^*, j = 1, \ldots, n$ and $p_j > p_s^*, j = n + 1, \ldots, k$. Then, $T_j = U^{'-1}(p_s^*), j = 1, \ldots, n$ and $T_j < U^{'-1}(p_s^*), j = n+1, \ldots, k$. In other words, $U^{'-1}(p_s^*) = \max_{0 \leq j \leq k} T_j$. Therefore, $x_s = \min(\max_{0 \leq j \leq k} T_j, m_s)$. For $j = 1, \ldots, n$, we have $x_{sj} < T_j$ since $x_s = \sum_{j=1}^{n} x_{sj} \leq T_j$ and $x_{sj} > 0$. For $j = n + 1, \ldots, k$, we have $x_{sj} \leq T_j$ since $x_{sj} = 0$ and $T_j \geq 0$.

We now prove the result for an uncoordinated controller. When the maximum source rate of source s, m_s, is not bounded, source s obtains a fair share with single-path sources on each path since $U(x)$ is strictly concave. That is, $x_{sj} = T_j$ and $x_s = \sum_{j=1}^{k} x_{sj} = \sum_{j=1}^{k} T_j$. When m_s is bounded, we have $x_{sj} \leq T_j$ and $x_s = \min(m_s, \sum_{j=1}^{k} T_j)$. ■

Appendix III
Proof of Theorem 5

Proof: For ease of description, we define another optimization problem $\mathbf{P'}$, which differs from problem $\mathbf{P}$ (see Section III) only in that the objective is to maximize the aggregate source rate, $\sum_{s \in \mathcal{S}} x_s$. Note that, unlike problem $\mathbf{P}$, the optimal solution of problem $\mathbf{P'}$ may not be unique. We first prove that, in a single-receiver 2nd-hop-constrained setting, for a given selection of relays, the optimal solution for problem $\mathbf{P}$ is also an optimal solution for problem $\mathbf{P'}$. Since a coordinated controller solves problem $\mathbf{P}$, it also solves $\mathbf{P'}$ (i.e., maximizes the aggregate source rate) in a single-receiver 2nd-hop-constrained setting.

Let $\{x_s^*\}$ denote the optimal solution of problem $\mathbf{P}$, $s \in S, 0 \leq x_s^* \leq m_s$. (Note that $S = S_m$ in a single-receiver 2nd-hop-constrained setting. If a relay is not selected by any source, it can be removed without affecting the solution to problem $\mathbf{P}$ and $\mathbf{P}'$. Therefore, we assume that a relay is selected by at least one source. We prove the above claim by considering the following three cases:

- Case 1: All sources are satisfied, i.e., $x_s^* = m_s$, $\forall s \in S$. Then $\{x_s^*\}$ is clearly an optimal solution of problem $\mathbf{P}'$.
- Case 2: No source is satisfied, i.e., $x_s^* < m_s$, $\forall s \in S$. In this case, all relay bandwidths are fully utilized, since $U(x)$ is an increasing function of x. This is clearly an optimal solution for problem $\mathbf{P}'$.
- Case 3: A subset of the sources are unsatisfied, i.e., $x_s^* < m_s, \forall s \in S_1$, $x_s^* = m_s, \forall s \in S_2$ and $S_1 \cup S_2 = S$. If no relay has spare bandwidth, then the result holds as in Case 2. We now consider an arbitrary relay, $r \in R$, with spare bandwidth. Consider an arbitrary source $s \in S$ that selects relay r. Then source s must be satisfied. This is proved by contradiction as follows. Suppose source s is not satisfied. Since $U(x)$ is an increasing function, we can increase the aggregate utility by increasing x_s^*. This contradicts with the assumption that $\{x_s^*\}$ is the optimal solution. By Theorem 4, for any another relay that source s selects, source s either has zero sending rate on that relay or the relay has spare bandwidth. We therefore can remove all satisfied sources along with the relays on which they have non-zero sending rate. We are then left with a subset of sources not satisfied with their selected relays. All these relay bandwidths are fully utilized following a similar argument as in Case 2. We therefore have the desired result.

The proof of the second part of the theorem follows directly from the result that a coordinated controller solves problem $\mathbf{P}'$ in a single-receiver 2nd-hop-constrained setting. When a source adds an additional overlay path, the sending rate on this path is allowed to be non-zero, which relaxes the constraints of problem $\mathbf{P}'$, and therefore leads to a higher (or equal) aggregate source rate. ■

APPENDIX IV
PROOF OF THEOREM 7

Before proving Theorem 7, we first present a lemma. We assume strictly concave utility functions.

Lemma 3: Suppose that two multipath sources, s and s', share the same set of k paths. Furthermore, suppose their demands are the same, i.e., $m_s = m_{s'} = m$. Then $x_s = x_{s'}$ for both coordinated and uncoordinated controllers.

Proof: We first prove the above result under coordinated controllers. Let p^* denote this minimum price over the k paths. Then from Lemma 2, we have $x_s = x_{s'} = \min(m, U'^{-1}(p^*))$. Under uncoordinated controllers, this is true because of the fairness properties of single-path controllers on the same path. ■

We now prove Theorem 7.

Proof: Suppose after the i-th round, n sources are satisfied, $0 \leq n \leq |S_m|$. We prove the theorem by induction on n. If $n = 0$, we are done since by Random, all sources needs to reselect paths, and there is a positive probability that this reselection leads to a solution. We now suppose that the result holds for n, $0 < n < |S_m|$ and prove that the result holds for $n+1$, that is, when there are $n+1$ satisfied sources. If $n+1 = |S_m|$, we are done. Otherwise, we pick a satisfied source arbitrarily, denoted as source s. Under Random, there is a positive probability that all of the unsatisfied sources choose the k paths used by source s in the $(i+1)$-th round. By Lemma 3, under coordinated and uncoordinated controllers, sources with the same demands using the same paths obtain the same rate. Therefore, in the $(i+1)$-th round, either all of the unsatisfied sources become satisfied (and hence all sources are satisfied), or source s becomes unsatisfied, leading to n satisfied sources, and the result holds by the inductive hypothesis. ■