On Minimizing End-to-End Delay With Optimal Traffic Partitioning

Shiwen Mao, *Member, IEEE*, Shivendra S. Panwar, *Senior Member, IEEE*, and Y. Thomas Hou, *Senior Member, IEEE*

***Abstract*—Multipath transport provides higher usable bandwidth for a session. It has also been shown to provide load balancing and error resilience for end-to-end multimedia sessions. Two key issues in the use of multiple paths are 1) how to minimize the end-to-end delay, which now includes the delay along the paths and the resequencing delay at the receiver, and 2) how to select paths. This paper presents an analytical framework for the optimal partitioning of real-time multimedia traffic that minimizes the total end-to-end delay. Specifically, it formulates optimal traffic partitioning as a constrained optimization problem using deterministic network calculus and derives its closed-form solution. Compared with previous work, the proposed scheme is simpler to implement and enforce. This analysis also greatly simplifies the solution to the path selection problem as compared to previous efforts. Analytical results show that for a given flow and a set of paths, a minimal subset can be chosen to achieve the minimum end-to-end delay with $O(N)$ time, where N is the number of available paths. The selected path set is optimal in the sense that adding any rejected path to the set will only increase the end-to-end delay.**

***Index Terms*—Multimedia communications, multipath transport, network calculus, quality of service (QoS), real-time.**

I. Introduction

THE IDEA of using multiple paths for an end-to-end session, called multipath transport throughout this paper, was first proposed in [2]. Multipath transport has been applied in various settings for achieving, e.g., load balancing, a higher aggregate capacity, or exploring path redundancy for failure recovery [3]. Recently, due to the availability of a variety of network access technologies, as well as the reduction in their costs, there has been an increasing interest in taking advantage of multihomed hosts to get a larger throughput and higher reliability [4]–[6]. In addition, there has been substantial recent work on using multipath transport for real-time multimedia applications [7]–[14]. For example, multipath transport has been combined with multiple description coding (MDC) [7]–[9], [11]–[13] and forward error correction (FEC) [14] for video transport. It has been shown that when combined with source/channel coding and error control schemes, multipath transport can significantly improve the quality of the multimedia service as compared with traditional shortest path routing-based schemes. This has also inspired recent standardization efforts for multipath transport protocols [15], [16].

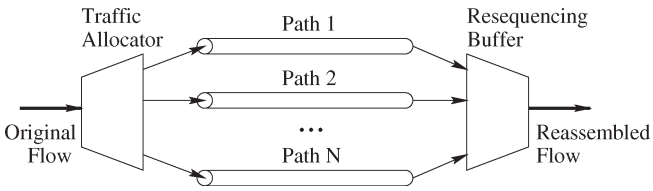

Fig. 1. General architecture of multipath transport.

The general architecture of multipath transport is illustrated in Fig. 1. We assume an underlying multipath routing protocol that maintains multiple disjoint paths between source and destination nodes. There is a rich literature on multipath routing (see, e.g., [9], [17]–[19], and the references therein). After multiple paths are found, typically source routing is used for packet forwarding [20]. On the sender side, the traffic allocator is responsible for partitioning application data, i.e., dispatching each data packet onto a specific path. The traffic partitioning strategy is affected by a number of factors, such as quality of service (QoS) requirements and the autocorrelation structure of the application data flow, the number of available paths, and path characteristics (e.g., bandwidth, delay, and loss behavior). Usually, the path parameters can be inferred from local information [21] or from receiver feedback [22] so that the traffic allocator can adjust its strategy to adapt to changes in the network. On the receiver side, received packets are put into a resequencing buffer in order to restore their original order. Packets may be out-of-order due to variations in path delays or non-first-come first-serve (FCFS) service discipline at an intermediate node.

In real-time multimedia applications, the resequencing buffer is also used to absorb jitter in arriving packets. Since the receiver displays the received media continuously, each packet is associated with a decoding deadline D_l, which is the time when it is extracted from the resequencing buffer to be decoded. In such applications, a packet will only stay in the resequencing buffer for at most D_l seconds. A packet may be lost because of transmission errors or dropped because it is overdue. Both types of packet losses are undesirable in terms of application QoS requirements. A larger resequencing buffer can reduce the

Manuscript received December 4, 2004; revised May 16, 2005 and September 21, 2005. This work was supported in part by the National Science Foundation under Grants ANI-0081375, ANI-0312655, and CNS-0347390, the Office of Naval Research under Grant N00014-03-1-0521, and the New York State Office of Science, Technology, and Academic Research (NYSTAR) through the Center for Advanced Technology in Telecommunications (CATT) at Polytechnic University, Brooklyn, NY. This paper was presented in part at IEEE INFOCOM, Miami, FL, March 2005. The review of this paper was coordinated by Prof. D. O. Wu.

S. Mao and Y. T. Hou are with the Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA 24061 USA (e-mail: smao@ieee.org; thou@vt.edu).

S. S. Panwar is with the Department of Electrical and Computer Engineering, Polytechnic University, Brooklyn, NY 11201 USA (e-mail: panwar@catt.poly.edu).

Digital Object Identifier 10.1109/TVT.2005.863360

overdue packet ratio but may result in a larger end-to-end delay. A major concern of multipath transport is how to minimize end-to-end delay, including delay on the paths as well as the additional resequencing delay at the receiver. The other key concern in using multipath transport is how to choose the set of paths to use. Routing overhead, computational complexity, and delay may prohibit the use of a large number of paths. Consequently, it is desirable to use a minimum number of paths while achieving the best QoS. In addition, the path selection algorithm should have a low computation complexity since network conditions may change quickly.

In this paper, we investigate the optimal traffic partitioning problem for real-time applications using network calculus in a deterministic setting. More specifically, we model the bottleneck link of each path as a queue with a deterministic service rate. The contribution of all other links and the propagation delay are lumped into a fixed delay element. We assume that the source flow is regulated by a $\{\sigma, \rho\}$ leaky bucket (or a token bucket, which is implemented in most commercial routers) and use deterministic traffic partitioning to split traffic into multiple flows, each conforming to a $\{\sigma_i, \rho_i\}$ regulator. Within such a setting, we formulate a constrained optimization problem on minimizing total end-to-end delay. We derive a closed-form solution and provide simple guidelines on minimizing end-to-end delay and path selection. We show that the path set chosen with our approach is optimal in the sense that adding any other paths to the chosen set will only increase the total end-to-end delay. This path selection scheme is useful since although it is always desirable to use a path with a higher bandwidth and a lower fixed delay, it is impossible to order the paths consistently, either according to their bandwidth or fixed delay, in many cases. A brute force optimization evaluating all feasible combinations of the paths would have exponential complexity [19]. Using our approach, path selection has only $O(N)$ complexity, where N is the number of available paths.

We also present an implementation to enforce an optimal partition using a number of cascaded leaky buckets: one for each path. This algorithm is suitable for the cases where the paths are dynamic. The exact optimal partition, rather than a heuristic, can be quickly computed and applied for a sequence of snapshots of the time-varying network.

The rest of this paper is organized as follows. For ease of presentation, we start with a two-path system in Section II and then extend it to the case of multiple paths in Section III. In Section IV, we discuss implementation-related issues. Related work is discussed in Section V, and Section VI concludes this paper.

II. Optimal Partition With Two Paths

We will first consider a real-time multimedia session using two paths. The two-path optimal partitioning problem is formulated in Section II-A. Making no assumption on the service discipline, we derive the corresponding optimal partition in Section II-B and then derive a tighter end-to-end delay bound assuming FCFS service discipline in Section II-C. The notation used in this paper is given in Table I.

TABLE I
Notation

Symbol	Definition
$A(t)$:	accumulative traffic of the data flow.
$\hat{A}(t)$:	envelope process of the data flow.
N:	total number of available paths.
σ:	burst factor of the envelope process.
ρ:	rate factor of the envelope process.
σ_i:	burst assigned to path i.
ρ_i:	rate assigned to path i.
c_i:	capacity of the path i bottleneck queue.
c:	aggregate capacity of all the paths.
f_i:	fixed delay on path i.
d_i:	queueing delay on the bottleneck link of path i.
D_i:	total delay on path i.
$\tilde{D}_i$:	total delay of path i obtained using (14).
D_l:	deadline, or the total end-to-end delay.
B:	resequencing buffer size.
c_d:	service rate of the resequencing buffer.
σ_i^*:	optimal burst assignment for path i.
ρ_i^*:	optimal rate assignment for path i.
D_l^*:	minimum end-to-end delay.
$\tilde{D}_l^*$:	minimum end-to-end delay obtained using (14).
σ_{th}^j:	the jth threshold that partitions σ.
ρ_{th}^j:	the jth threshold that partitions ρ.

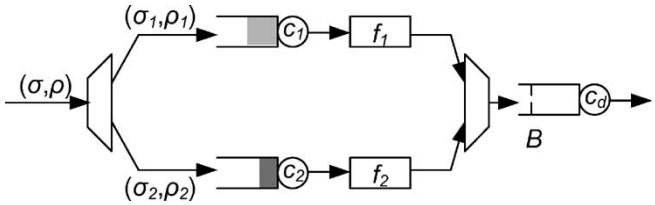

Fig. 2. Traffic partitioning model with two paths.

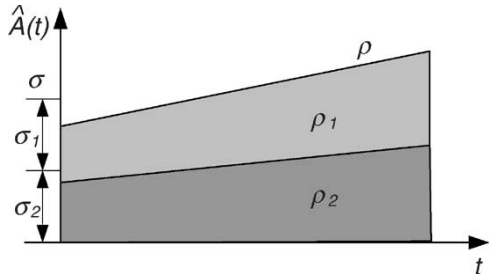

Fig. 3. Deterministic traffic partitioning scheme.

A. Problem Formulation

The corresponding two-path traffic partitioning model is shown in Fig. 2. Let the cumulative real-time traffic in $[0, t)$ be $A(t)$, which is regulated by a $\{\sigma, \rho\}$ leaky bucket, i.e., $A(t)$ conforms to a deterministic envelope process [23]

$$\hat{A}(t) = \rho t + \sigma \tag{1}$$

where ρ is the long-term average rate of the process (the rate factor), and σ is the maximum burst size (the burst factor) of $\hat{A}(t)$. The source traffic stream is then partitioned using a deterministic scheme, as illustrated in Fig. 3. With this scheme, the source flow is divided into two substreams deterministically, each of which conforms to an envelope process

$$\hat{A}_i(t) = \rho_i t + \sigma_i, \qquad i = 1, 2. \tag{2}$$

We have a further constraint $\hat{A}_1(t) + \hat{A}_2(t) = \hat{A}(t)$, which gives $\rho_1 + \rho_2 = \rho$ and $\sigma_1 + \sigma_2 = \sigma$. Therefore, the traffic partitioning operation will not cause any additional loss or delay of application data. We will discuss the implementation of such a deterministic partitioning in Section IV.

We model the bottleneck link of each path as a work conserving queueing system with a constant service rate c_i, $i = 1, 2$. This approximation is quite accurate if the queueing delay at the bottleneck link dominates all other queueing delay components [24]–[26]. To have a stable system, the aggregate service rate $c = c_1 + c_2$ should be larger than the mean rate of the data flow, i.e., $c > \rho$, if $\sigma > 0$. We also assume that $\sigma > 0$ and $\rho \geq c_i$, $i = 1, 2$, in order to exclude the trivial case where flow can be accommodated by one of the paths. In order to have a stable queue in each path, the partitioned streams should satisfy $\rho_i < c_i$ if $\sigma_i > 0, i = 1, 2$.[1] The queueing delay at the bottleneck link of path i is denoted as d_i, $i = 1, 2$. The contribution of all other links along the path, including propagation delay, is represented by a fixed delay element $f_i, i = 1, 2$. Thus, the delay along path i, D_i is the sum of the queueing delay and the fixed delay, i.e.,

$$D_i = d_i(\sigma_i, \rho_i) + f_i, \qquad i = 1, 2. \tag{3}$$

The parameters of the paths may not be constant because of the time-varying background traffic and congestion. Moreover, when a path is broken, a replacement path k may have a different c_k or f_k. We assume that c_i and f_i, $i = 1, 2$, change on a relatively large time scale. Therefore, we can compute the optimal partition for each snapshot of the network and continuously update the optimal partition as network conditions change over time. Note that c_i is similar to the notion of "available bandwidth," which captures the variation of background traffic (and network congestion) over a relatively large time scale.

At the receiver side, the two substreams are reassembled in a resequencing buffer. Then, the restored stream is extracted from the buffer and sent to the application for decoding. Note that the server of the resequencing buffer is not work conserving. It polls the queue at a fixed rate (e.g., frame rate) for packets belonging to the next frame. If packets are found in the buffer, they are served at a rate of $c_d = \text{frame\_rate} \times \text{frame\_size}$; otherwise, $c_d = 0$. The total end-to-end delay D_l is jointly determined by the traffic partitioning strategy and the path parameters. Our objective is to derive the optimal partition, i.e., the optimal values $\{\sigma_i^*\}_{i=1,2}$ and $\{\rho_i^*\}_{i=1,2}$ such that the overall end-to-end delay is minimized.

We should note that the analysis is based on a deterministic approach for "hard" QoS guarantees, and the derived bounds are thus conservative as compared with probabilistic analysis. However, such a "hard" QoS guarantee is necessary for many distributed computing or real-time multimedia applications where strict QoS guarantees are required [27], such as distributed simulations, real-time visualization of complex scientific simulation results in multiple remote locations, stock exchange transactions, and remote surgery and telemedicine [28].

[1] If $\sigma_i = 0$, it is possible to set $\rho_i = c_i, i = 1, 2$, resulting in a zero queueing delay on path i.

B. Optimal Partition With the Busy Period Bound

In this subsection, we do not impose any restrictions on the packet scheduling discipline. Consider a work conserving queue with capacity c. Its input conforms to an envelope process $\hat{A}(t)$. If the queue is stable, then the queueing delay is upper bounded by the maximum busy period of the system [29]

$$d \stackrel{\text{def}}{=} \inf\left\{t \geq 0 \ : \ \hat{A}(t) - ct \leq 0\right\}. \tag{4}$$

Substituting (1) into (4), we have

$$d = \frac{\sigma}{c - \rho}. \tag{5}$$

The delay on path i is upper bounded by $D_i = d_i + f_i = \sigma_i/(c_i - \rho_i) + f_i$, $i = 1, 2$. Consider two back-to-back tagged bits b_1 and b_2 belonging to the same multimedia frame. If b_1 is transmitted on path 1 and b_2 on path 2 at time t, then b_1 will arrive at the resequencing buffer during the time interval $(t, t + D_1]$, and b_2 will arrive at the resequencing buffer during the time interval $(t, t + D_2]$. When both bits arrive (as well as all other bits in the same frame), they can be extracted from the buffer for decoding and display. Thus, $D_l = \max\{D_1, D_2\}$ upper bounds the end-to-end delay.

Fact 1: End-to-end delay D_l, including the queueing delay at the bottleneck, fixed delay, and resequencing delay, is bounded by

$$D_l = \max\{D_1, D_2\}. \tag{6}$$

Fact 2: A partition achieving $D_l = \max\{f_1, f_2\}$ is optimal.

Proof: From (6), $D_l^* = \min_{\sigma_i, \rho_i}\{\max\{D_1, D_2\}\} \geq \max\{f_1, f_2\}$, since $D_1 \geq f_1$ and $D_2 \geq f_2$. ■

Intuitively, a delay equal to the fixed delay cannot be further improved by traffic partitioning. From (6), we can formulate the constrained optimization problem on minimizing the end-to-end delay (denoted as OPT1) as

$$\text{Minimize} : D_l = \max\{D_1, D_2\} \tag{7}$$

subject to :

$$\begin{cases} \sigma_1 + \sigma_2 = \sigma \\ \rho_1 + \rho_2 = \rho \\ 0 < \rho_i \leq c_i, & i = 1, 2 \\ \sigma_i \geq 0, & i = 1, 2 \\ \sigma_i = 0, & \text{if } \rho_i = c_i, \ i = 1, 2. \end{cases} \tag{8}$$

OPT1 is a nonlinear optimization problem with linear constraints. The feasible region is a polytope (i.e., a solid bounded by polygons), since the constraints are linear equations or inequalities. Within this feasible region, we have

$$\begin{cases} \nabla d_1 = \left[\frac{\partial d_1}{\partial \rho_1}, \frac{\partial d_1}{\partial \sigma_1}\right] = \left[\frac{\sigma_1}{(c_1 - \rho_1)^2}, \frac{1}{c_1 - \rho_1}\right] \neq \mathbf{0} \\ \nabla d_2 = \left[\frac{\partial d_2}{\partial \rho_2}, \frac{\partial d_2}{\partial \sigma_2}\right] = \left[\frac{\sigma_2}{(c_2 - \rho_2)^2}, \frac{1}{c_2 - \rho_2}\right] \neq \mathbf{0}. \end{cases} \tag{9}$$

Thus, the minimum delay must occur at one of the boundaries or vertices of the feasible region [30].

This problem can be solved using results in the game theory literature. In particular, solving this problem is equivalent to computing the Wardrop Equilibrium (WE) of the system by using convex programming [31]. In this paper, we present an alternative approach that explores the special structure of the delay bound D_i. Our approach has the advantage of producing a simple solution without the complexity associated with the Beckmann transformation [32]. The solution to OPT1 is summarized in the following theorem. The proof of Theorem 1 is omitted for brevity and can be found in [33]. Without loss of generality, we will assume that $f_1 \leq f_2$.

Theorem 1: Using the busy period bound (5), the optimal traffic partition and the minimum end-to-end delay are as follows.

1) If $\sigma > (c-\rho)(f_2 - f_1)$, then $D_l^* = \sigma/(c-\rho) + \min\{f_1, f_2\}$, and the optimal partition is

$$\begin{cases} \{\sigma_1^*, \sigma_2^*\} = \{\sigma, 0\} \\ \{\rho_1^*, \rho_2^*\} = \{\rho - c_2, c_2\}. \end{cases} \tag{10}$$

2) If $\sigma \leq (c-\rho)(f_2 - f_1)$, then $D_l^* = \max\{f_1, f_2\}$, and the optimal partition is

$$\begin{cases} \{\sigma_1^*, \sigma_2^*\} = \{\sigma, 0\} \\ \rho_1^* = \rho - \rho_2^* \\ \rho_2^* \in \left[\rho - c_1 + \frac{\sigma}{f_2 - f_1}, c_2\right]. \end{cases} \tag{11}$$

3) If $f_1 = f_2 = f$, then $D_l^* = \sigma/(c-\rho) + f$, and the optimal partition is

$$\begin{cases} \{\sigma_1^*, \sigma_2^*\} = \left\{\frac{c_1 - \rho_1}{c-\rho}\sigma, \frac{c_2 - \rho_2}{c-\rho}\sigma\right\} \\ \rho - c_2 < \rho_1^* < c_1 \\ \rho_2^* = \rho - \rho_1. \end{cases} \tag{12}$$

Note that when the paths have different fixed delays, the optimal partitioning strategy is to assign all the burst to the path with a smaller fixed delay and assigning a rate that saturates the path with the larger fixed delay. When the two paths have the same fixed delay, the two paths behave like a single path with combined capacity: the achieved minimum delay is identical to that obtained from a single path session with the same $\{\sigma, \rho\}$ flow, fixed delay f, and service rate $c = c_1 + c_2$. Another interesting observation is that any feasible $\{\rho_1^*, \rho_2^*\}$ can achieve the minimum delay when $f_1 = f_2$.

C. Optimal Partition With FCFS Queues

Theorem 1 is obtained using the system busy period bound (4) from [29]. If the service discipline is FCFS, the queueing delay can be further improved as in [34], i.e.,

$$\tilde{d} = \sup_{t \geq 0}\left\{\inf\left\{\tau \geq 0 : \hat{A}(t) \leq c(t+\tau)\right\}\right\}. \tag{13}$$

Substituting (1) into (13), we have

$$\tilde{d} = \frac{\sigma}{c} \tag{14}$$

and the end-to-end delay of path i is $\tilde{D}_i = \sigma/c_i + f_i$, $i = 1, 2$. This bound is tighter than the system busy period bound (5). In addition, it is only a function of the burst factor σ. This fact can be exploited to simplify the analysis and to improve the minimum delay given in Theorem 1.

Consider the same two-path model in Fig. 2. From (6) and (14), we can formulate the constrained optimization problem (denoted as OPT2)

$$\text{Minimize} : \tilde{D}_l = \max\{\tilde{D}_1, \tilde{D}_2\} \tag{15}$$

subject to :

$$\begin{cases} \sigma_1 + \sigma_2 = \sigma \\ \rho_1 + \rho_2 = \rho \\ 0 < \rho_i \leq c_i, & i = 1, 2 \\ \sigma_i \geq 0, & i = 1, 2 \\ \sigma_i = 0, & \text{if } \rho_i = c_i,\ i = 1, 2. \end{cases} \tag{16}$$

The solution to OPT2 is summarized in the following theorem. The proof is omitted for brevity and can be found in [33].

Theorem 2: Assuming FCFS queues, the optimal traffic partition and the minimum end-to-end delay $\tilde{D}_l^*$ are as follows.

1) If $\sigma > c_1(f_2 - f_1)$, then $\tilde{D}_l^* = (1/c)(\sigma + c_1 f_1 + c_2 f_2)$, and the optimal partition is

$$\sigma_i^* = \frac{c_i}{c}\left[\sigma + c_{3-i}(f_{3-i} - f_i)\right], \qquad i = 1, 2. \tag{17}$$

2) If $\sigma \leq c_1(f_2 - f_1)$, then $\tilde{D}_l^* = \max\{f_1, f_2\}$, and the optimal partition is

$$\{\sigma_1^*, \sigma_2^*\} = \{\sigma, 0\}. \tag{18}$$

3) If $f_1 = f_2 = f$, then $\tilde{D}_l^* = \sigma/c + f$, and the optimal partition is

$$\sigma_i^* = \sigma\frac{c_i}{c}, \qquad i = 1, 2. \tag{19}$$

4) Any feasible partition of ρ can be used to achieve the above minimum end-to-end delay.

Clearly, the partition strategy is different from Theorem 1 due to the different delay bounds. However, when the paths have equal fixed delays, the achieved minimum delay is still identical to that obtained from a single path session with the same $\{\sigma, \rho\}$ source, a fixed delay f, and a bandwidth $c = c_1 + c_2$.

It would be interesting to compare the delay bounds from Theorems 1 and 2. Define two threshold values $\sigma_{th}^L = (c-\rho)(f_2 - f_1)$ and $\sigma_{th}^H = c_1(f_2 - f_1)$ such that $\sigma_{th}^H - \sigma_{th}^L = (\rho - c_2)(f_2 - f_1) > 0$. From the theorems, we have that

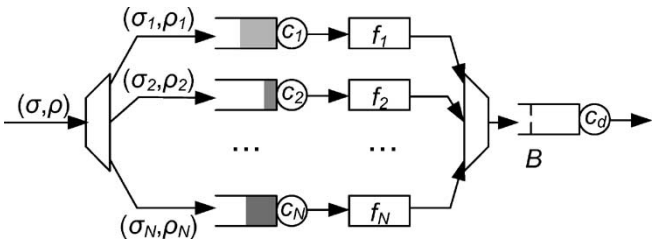

Fig. 4. Traffic partitioning model with N paths.

$D_l^* = \tilde{D}_l^* = \max\{f_1, f_2\}$ when $\sigma \le \sigma_{th}^L$. When $\sigma_{th}^L < \sigma \le \sigma_{th}^H$, we have that $\tilde{D}_l^* = \max\{f_1, f_2\} < D_l^* = \sigma/(c-\rho) + \min\{f_1, f_2\}$. When $\sigma > \sigma_{th}^H$, we have that

$$D_l^* - \tilde{D}_l^* = \frac{c_2}{c}(f_1 + f_2) + \frac{\rho\sigma}{c(c-\rho)} > 0.$$

III. Extension to Multiple Paths

In this section, we extend the optimal partition analysis to the case of multiple paths using the delay bound (14) for FCFS queues. For a given set of paths, we first combine and reorder the paths according to their fixed delays. Then, we formulate the optimal partitioning problem for multiple paths and derive its closed-form solution.

A. Multipath Extension

For any set of paths P_i' with parameters $\{c_i', f_i'\}$, $i = 1, \cdots, M$, we first do the following.

1) Sort and relabel the paths according to their fixed delays f_i' in nondecreasing order.
2) If paths $P_i', P_{i+1}', \ldots, P_{i+k-1}'$ have the same fixed delay, i.e., $f_i' = f_{i+1}' = \cdots = f_{i+k-1}'$, we can lump these paths into a new path i with $f_i = f_i'$ and $c_i = c_i' + c_{i+1}' + \cdots + c_{i+k-1}'$ according to Theorem 2, item 3.
3) Relabel the paths again. Then, we get a new set of paths P_i with parameters $\{c_i, f_i\}$, $i = 1, \ldots, N$, and $f_1 < f_2 < \cdots < f_N$.

In the following, we first determine the optimal partitioning scheme for the paths P_i, $i = 1, \ldots, N$. Then, we can further partition the assignments σ_i^* and ρ_i^* to k original paths $P_i', P_{i+1}', \cdots, P_{i+k-1}'$ with the same fixed delay f_i' using Theorem 2, item 3.

The N-path traffic partitioning model is depicted in Fig. 4, with parameters $\{c_i, f_i\}$, $i = 1, \ldots, N$ and $f_1 < f_2 < \cdots < f_N$. From (6) and (14), we can formulate the linearly constrained optimization problem for the N-path session with FCFS queues [denoted as $\mathcal{P}(N, \sigma)$] as

$$\text{Minimize}: \tilde{D}_l = \max\{\tilde{D}_1, \tilde{D}_2, \ldots, \tilde{D}_N\} \tag{20}$$

subject to:

$$\begin{cases} \sigma_1 + \sigma_2 + \cdots + \sigma_N = \sigma \\ \rho_1 + \rho_2 + \cdots + \rho_N = \rho \\ 0 \le \rho_i \le c_i, & i = 1, 2, \ldots, N \\ \sigma_i \ge 0, & i = 1, 2, \ldots, N \\ \sigma_i = 0, & \text{if } \rho_i = c_i,\ i = 1, 2, \ldots, N. \end{cases} \tag{21}$$

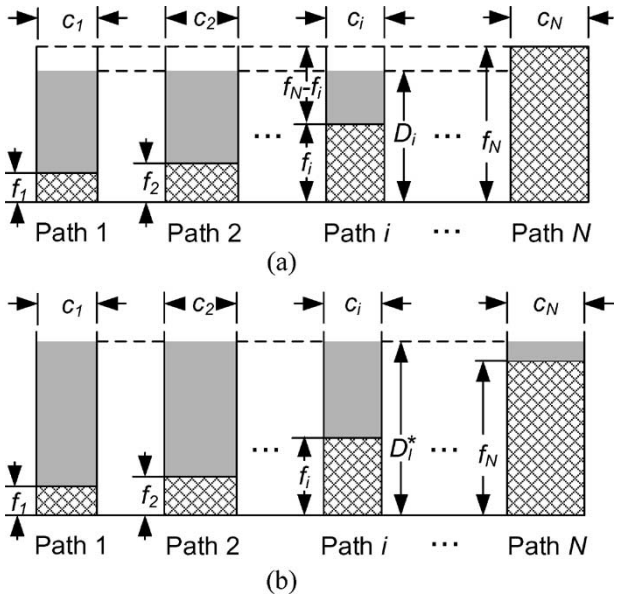

Fig. 5. Problem $\mathcal{P}(N, \sigma)$. (a) Case of $\sigma \le \sigma_{th}^N$. (b) Case of $\sigma > \sigma_{th}^N$.

The solution to $\mathcal{P}(N, \sigma)$ is summarized in the following theorem.

Theorem 3: Define $\sigma_{th}^N = \sum_{i=1}^N c_i(f_N - f_i)$. The solution to $\mathcal{P}(N, \sigma)$ is as follows.

- *Case I:* If $\sigma \le \sigma_{th}^N$, we have $\tilde{D}_l^* \le f_N$ and the optimal assignment for path N is $\sigma_N^* = 0$. $\tilde{D}_l^*$ and the optimal assignment for the remaining paths can be determined by applying this theorem recursively on $\mathcal{P}(N-1, \sigma)$, i.e., a reduced problem corresponding to (20) and (21) for the remaining $N-1$ paths and a burst σ.
- *Case II:* If $\sigma > \sigma_{th}^N$, we have $\tilde{D}_l^* = (1/c)[\sum_{i=1}^N c_i f_i + \sigma]$, and the optimal partition that achieves the minimum end-to-end delay is $\sigma_i^* = (c_i/c)[\sigma + \sum_{j=1}^N c_j(f_j - f_i)]$, $i = 1, 2, \ldots, N$.

Proof: This theorem can be proved by extending the proof for Theorem 2 (given in [33]) to the $N > 2$ case. However, we can use an intuitive "water-filling" model to solve $\mathcal{P}(N, \sigma)$ directly (which also applies to OPT2).

In Fig. 5(a), we model each path i as a bucket with a cross-section of area c_i. In addition, each bucket i is preloaded with content $c_i f_i$ to a level f_i. If path i is assigned with a burst σ_i, this is equivalent to filling σ_i units of fluid into bucket i, resulting in a higher level of $\sigma_i/c_i + f_i$. Thus, the fluid level of bucket i represents the delay on path i. With this model, the optimization problem $\mathcal{P}(N, \sigma)$ is equivalent to filling σ units of fluid into the N buckets while keeping the highest level among all the buckets as low as possible.

Consider Fig. 5(a). Assume that each bucket has a finite depth f_N, which is the highest preloaded level of the N buckets. Then, the N buckets can hold at most $\sigma_{th}^N = \sum_{i=1}^N c_i(f_N - f_i)$ units of fluid without overflow. Note that bucket N cannot hold any fluid since its level is already f_N. Thus, if the burst of data flow, or the amount of fluid, σ is less than σ_{th}^N, all the σ units of fluid can be distributed to the $N-1$ buckets, with none of the bucket having a level exceeding f_N. Thus, the optimal

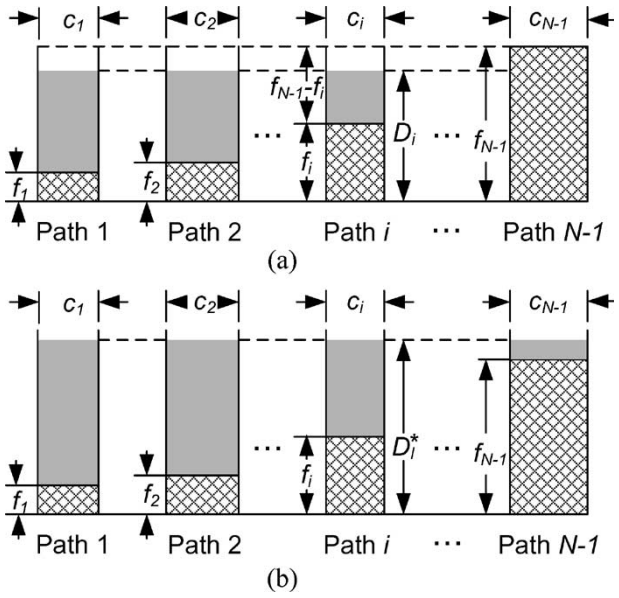

Fig. 6. Problem $\mathcal{P}(N-1,\sigma)$. (a) Case of $\sigma \le \sigma_{th}^{N-1}$. (b) Case of $\sigma > \sigma_{th}^{N-1}$.

assignment for path N is $\sigma_N^* = 0$ and $\tilde{D}_l^* \le f_N$. This corresponds to Case I of Theorem 3.

On the other hand, if $\sigma > \sigma_{th}^N$, σ units of fluid cannot be accommodated by the buckets as in Fig. 5(a). In this case, let each bucket have an infinite depth such that an arbitrarily large σ can be held in these buckets as shown in Fig. 5(b). However, in order to minimize the highest level, σ units of fluid should be distributed to the N buckets in such a manner that all the buckets should have the same fluid level. If the common fluid level is $\tilde{D}_l^*$, the amount of fluid that bucket i holds is $\sigma_i^* = c_i(\tilde{D}_l^* - f_i)$. Since the total amount of fluid is σ, we have

$$c_1(\tilde{D}_l^* - f_1) + c_2(\tilde{D}_l^* - f_2) + \cdots + c_N(\tilde{D}_l^* - f_N) = \sigma. \tag{22}$$

The minimum end-to-end delay $\tilde{D}_l^*$ can be solved from (22) as

$$\tilde{D}_l^* = \frac{1}{c}(\sigma + c_1 f_1 + c_2 f_2 + \cdots + c_N f_N). \tag{23}$$

The volume filled into bucket i, or the optimal burst assignment σ_i^*, is

$$\sigma_i^* = c_i(\tilde{D}_l^* - f_i) = \frac{c_i}{c}\left[\sigma + \sum_{j=1}^{N} c_j(f_j - f_i)\right]. \tag{24}$$

This corresponds to Case II of Theorem 3.

So far for Case I we have derived $\sigma_N^* = 0$. In order to determine the optimal partition for the remaining $N-1$ paths, we remove path N from (20) and (21). Since $\sigma_N^* = 0$, removing path N does not affect the constraints in (21) and the objective value in (20). Consequently, we obtain an $(N-1)$-path problem with a burst σ, i.e., $\mathcal{P}(N-1,\sigma)$. Define a new threshold value $\sigma_{th}^{N-1} = \sum_{i=1}^{N-1} c_i(f_{N-1} - f_i)$. We next examine the two cases of $\mathcal{P}(N-1,\sigma)$ using the same "water-filling" model as illustrated in Fig. 6(a) and (b), but with the remaining $N-1$ paths. Repeat the above steps until a Case-II-type solution is obtained. If the number of paths is reduced to two, the two-path results in Section II-C can be applied. Thus, $\tilde{D}_l^*$ and the optimal partition for all the paths can be determined. ■

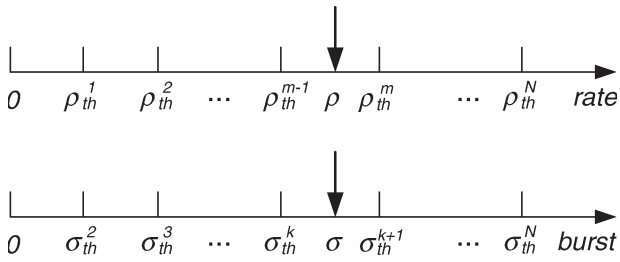

Fig. 7. Computing the optimal partition.

Note that, according to (9), the minimum delay must occur either at a boundary of the search space or at one of the vertices. Indeed, each delay term in (20) ($\tilde{D}_i = \sigma_i/c_i + f_i$) is a plane in the N-dimensional search space. In Case I of Theorem 3, we remove a plane that always dominates all other planes; using such a plane will only increase the objective function. In Case II, the minimum occurs at a boundary where all the planes intersect at a single point.

The minimum end-to-end delay is jointly determined by burst assignments and rate assignments. We first define the quantities

$$\begin{cases} \rho_{th}^j = \sum_{i=1}^{j} c_i, & j = 1,2,3,\ldots,N \\ \sigma_{th}^j = \sum_{i=1}^{j} c_i(f_j - f_i), & j = 1,2,3,\ldots,N. \end{cases} \tag{25}$$

Note that $\sigma_{th}^1 = 0$. Clearly, $\rho_{th}^i > \rho_{th}^j$ and $\sigma_{th}^i > \sigma_{th}^j$ if $i > j$. These quantities partition the rate line and the burst line, respectively, as illustrated in Fig. 7. Let m be the index such that $\rho_{th}^{m-1} \le \rho < \rho_{th}^m$ and k be the index such that $\sigma_{th}^k < \sigma \le \sigma_{th}^{k+1}$. Then, m is the highest index of the minimum set of paths required to accommodate ρ in order to satisfy the stability condition, and k is the highest index of the minimum set of paths required to accommodate σ. If $m > k$, then the minimum delay is the fixed delay on path m. Otherwise, the minimum delay is a solution to $\mathcal{P}(k,\sigma)$ [see (23)].

Corollary 3.1: For the indices m and k as defined in Fig. 7, we have the following.

1) If $m > k$, then $\tilde{D}_l^* = f_m$.
2) If $m \le k$, then $\tilde{D}_l^* = (1/\rho_{th}^k)(\sigma + \sum_{i=1}^{k} c_i f_i)$.
3) The optimal burst assignments are

$$\sigma_i = \begin{cases} \left(\frac{c_i}{\rho_{th}^k}\right)\left[\sigma + \sum_{j=1}^{k} c_j(f_j - f_i)\right], & \text{if } i \le k \\ 0, & \text{otherwise.} \end{cases} \tag{26}$$

4) The optimal rate assignments could be

$$\rho_i = \begin{cases} \left(\frac{c_i}{\rho_{th}^m}\right)\rho, & \text{if } i \le m \\ 0, & \text{otherwise.} \end{cases} \tag{27}$$

We also have the following corollary for optimal path selection.

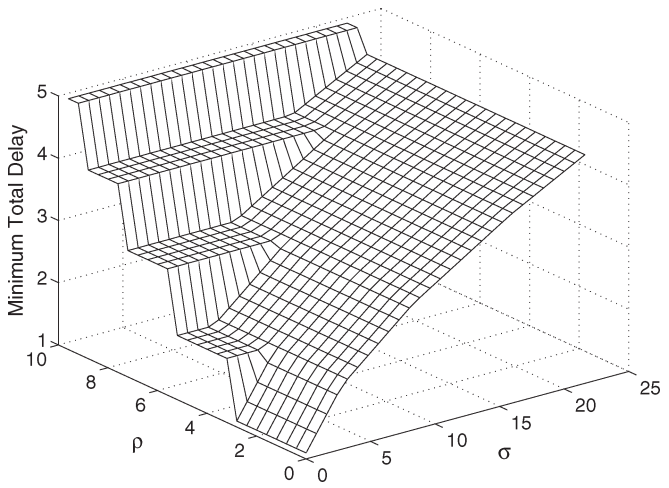

Fig. 8. Achieved minimum end-to-end delay $\tilde{D}_l^*$ for a five-path system with $\vec{\mathbf{f}} = \{1, 2, 3, 4, 5\}$ and $\vec{\mathbf{c}} = \{1, 1.5, 2, 2.5, 3\}$.

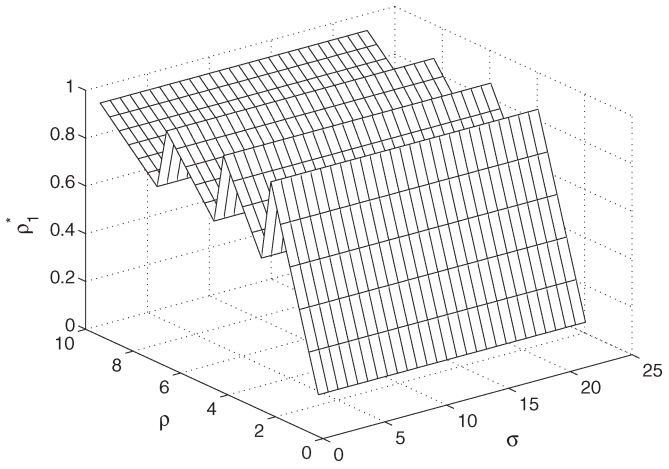

Fig. 10. Optimal Path 1 rate assignment ρ_1^* for a five-path system with $\vec{\mathbf{f}} = \{1, 2, 3, 4, 5\}$ and $\vec{\mathbf{c}} = \{1, 1.5, 2, 2.5, 3\}$.

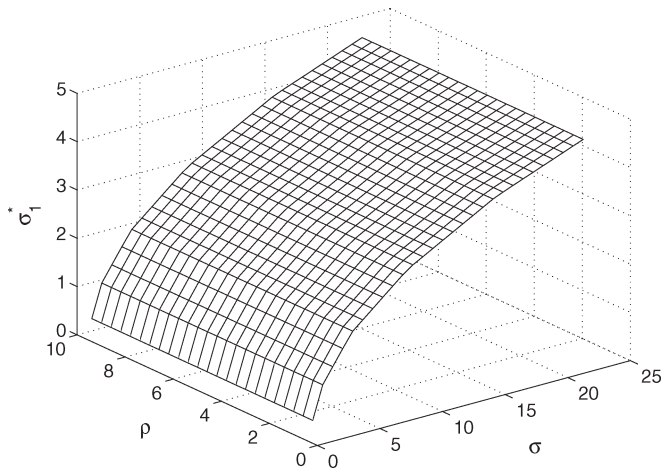

Fig. 9. Optimal Path 1 burst assignment σ_1^* for a five-path system with $\vec{\mathbf{f}} = \{1, 2, 3, 4, 5\}$ and $\vec{\mathbf{c}} = \{1, 1.5, 2, 2.5, 3\}$.

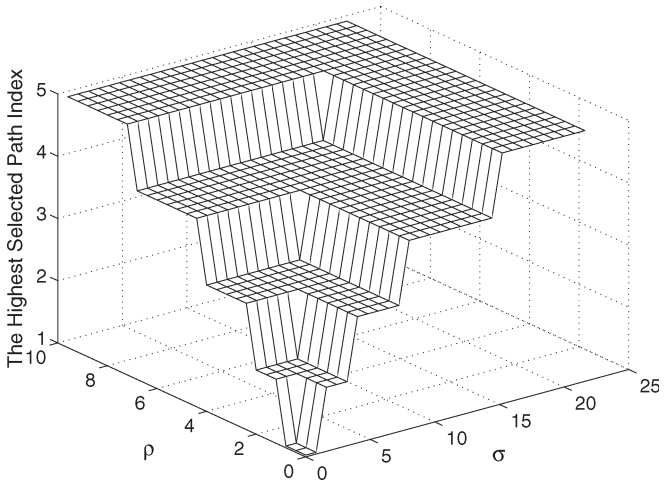

Fig. 11. Highest index of the chosen paths set for a five-path system with $\vec{\mathbf{f}} = \{1, 2, 3, 4, 5\}$ and $\vec{\mathbf{c}} = \{1, 1.5, 2, 2.5, 3\}$.

Corollary 3.2: For the indices m and k as defined in Fig. 7, the first $h = \max\{m, k\}$ paths are selected for the session such that the minimum end-to-end delay $\tilde{D}_l^*$ is achieved.

B. Example

Consider a five-path session. The fixed delays of the paths are $\vec{\mathbf{f}} = \{1, 2, 3, 4, 5\}$, while the capacities of the paths are $\vec{\mathbf{c}} = \{1, 1.5, 2, 2.5, 3\}$. The minimum end-to-end delays for various $\{\sigma, \rho\}$ pairs are plotted in Fig. 8 for increasing σ and ρ. The minimum delays are step functions along the direction of increasing ρ, while the height of the steps are f_2, f_3, f_4, and f_5, respectively. That is, the minimum end-to-end delay increases when a new path with a larger fixed delay is added to the selected path set in order to accommodate an increased rate factor ρ. Along the direction of increasing σ, however, the minimum end-to-end delay increases in a piecewise linear manner. That is, in each interval $\sigma_{th}^i < \sigma \leq \sigma_{th}^{i+1}$, $\tilde{D}_l^*$ is a linearly increasing function of σ, while the slope of $\tilde{D}_l^*$ decreases as i gets larger.

The optimal burst assignments for path 1 are plotted in Fig. 9. We find that the burst assignments are piecewise linear and concave. The optimal rate assignments for path 1 ρ_1^* are plotted in Fig. 10, which has a sawtooth form as ρ increases. This is because ρ_1^* first increases linearly with ρ but decreases when there is a new path with a higher index being used.

With Corollary 3.2, path selection is based on end-to-end delay only and is quite simple because we only use the first $\max\{m, k\}$ paths. Fig. 11 plots the highest index of the paths in use (i.e., $\max\{m, k\}$) for the given system, which has the form of a step function along both σ and ρ dimensions.

IV. Practical Considerations

In this section, we discuss some important practical considerations and present an implementation to enforce the optimal partition for an end-to-end application. This implementation uses a set of leaky buckets, which are available in most commercial routers [35].

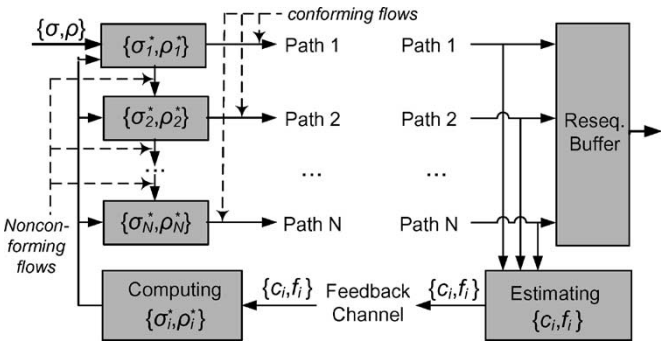

Fig. 12. Implementation of the optimal traffic partitioning scheme.

A. Optimal Path Selection

In many routing protocols, a path may be associated with more than one performance metric (e.g., each path has a fixed delay and a capacity, as in the case we have studied). When multiple paths are used, it would be nice to sort the paths according to their "quality" and use them starting with the best ones. However, we may get inconsistent orderings if we sort the paths according to different performance metrics. For example, a path may have a higher bandwidth but a higher delay, while another path may have a smaller bandwidth but a lower delay. Such inconsistency makes it very difficult to decide which paths to use. A brute force approach can examine every feasible combination of paths but at the cost of higher computational complexity. Some heuristics give preference to one performance metric over the other and use the secondary performance metric to break the tie if necessary [21]. Although such heuristics work well in some cases, it is not clear if they work in all the cases since there is no supporting analysis.

Corollary 3.1 shows that we can sort the paths consistently according to end-to-end delay, which then determines the minimum set of paths to be used. The computational complexity is $O(N)$. Path selection is optimal since adding any rejected path to this chosen set will only increase the end-to-end delay.

B. Enforcing the Optimal Partition

After the optimal partition parameters, i.e., $\{\sigma_i^*, \rho_i^*\}$, $i = 1, 2, \ldots, N$, are computed, the next question is how to enforce them on traffic flows. In the following, we show that the optimal partition can be enforced by using a set of leaky bucket regulators: one on each path.

For a point-to-point application (see Fig. 1), the sender is responsible for partitioning the traffic flow. The leaky buckets and the module that computes the optimal partition should be implemented at the sender side, as illustrated in Fig. 12. Multiple leaky buckets are cascaded in a chain, while a source flow is fed into the first leaky bucket having parameters $\{\sigma_1^*, \rho_1^*\}$. When a flow is regulated by a leaky bucket, usually, the conforming traffic is transmitted, while the nonconforming traffic (i.e., the portion that exceeds the constraint of the envelope process) is either marked or dropped. In our implementation, we simply redirect the nonconforming traffic to the next leaky bucket, rather than dropping it. The conforming traffic flow from leaky bucket i, having parameters $\{\sigma_i^*, \rho_i^*\}$, is then transmitted on path i. If $h = \max\{m, k\} < N$, then the hth leaky bucket produces no nonconforming traffic. Consequently, the remaining $(h+1)$th, $(h+2)$th, $\ldots$, Nth leaky bucket (and path) will not be used.

It is worth noting that since $\sum_{i=1}^{N} \sigma_i = \sigma$ and $\sum_{i=1}^{N} \rho_i = \rho$, there are always tokens for incoming traffic. Consequently, the above deterministic partitioning scheme does not introduce additional loss or delay to the application data.

C. Path Parameter Estimation

The proposed scheme works best when some QoS support is available in the network. For example, if the resource reservation protocol (RSVP) [36] is supported, a source can reserve the required bandwidth along each path, and a router or a switch can use the generalized processor sharing (GPS) scheduling to guarantee the reserved bandwidth [37]. If such QoS provisioning mechanisms are not available, the receiver could estimate the path parameters, i.e., c_i and f_i, $i = 1, 2, \ldots, N$, for a snapshot of the network and send the estimates back to the source, if the path conditions vary at a relatively large time scale.

Estimating path parameters based on end-to-end measurements has been an active research area for years. There exist many effective techniques that can be applied to estimate the path parameters used in our approach [38]–[42]. For example, the bprobe and cprobe schemes in [38], the self-loading periodic streams (SLoPS) scheme in [39], or the recent work CapProbe [40] can be used to estimate the end-to-end available bandwidth (or bottleneck bandwidth) of a path. If the source and the receiver are synchronized, the minimum one-way packet delay measured in the last time window would be a good approximation of the fixed delay f_i on that path. Otherwise, the approach presented in [41] can be used to estimate one-way delays from cyclic-path delay measurements that do not require any kind of synchronization among the nodes of the network.

After estimating the path parameters, the real-time transport protocol (RTP) [22] and its multiflow extension [16] can be used for delivering the parameters to the sender [via receiver reports (RR)]. The senders then compute the optimal partition and update the parameters of the leaky buckets periodically. Note that path conditions could change because of path failure, rerouting, etc. Further, variations in background traffic load using the same paths also cause variations in the estimated path parameters and trigger updates of the leaky bucket parameters. Therefore, if congestion occurs at a relative large time scale, the proposed traffic partitioning scheme can adapt to congestion as well, and the leaky bucket parameters can be updated using a TCP-like algorithm.

V. Related Work

Since the early work [2], traffic dispersion has been studied for different network service models. A survey on traffic dispersion was presented in [3]. In [43], the authors showed that

for data traffic a packet-level dispersion granularity gives better performance than a flow-level granularity in terms of delay and network resource utilization. In recent work [10], [44], [45], the authors showed that data partitioning techniques, such as striping and thinning, can effectively reduce the short-term correlations in real-time traffic and thus improve the queueing performance in the underlying network.

The problem of elastic data traffic partitioning for an end-to-end session was investigated in [21], [24], and [25] using different traffic and path models. In [24], a two-path resequencing model was presented where each path was assumed to be the combination of an M/M/1 queue and a fixed delay line. The authors showed that the optimal splitting probability may be highly dependent on the difference between the two fixed delays. However, the M/M/1 queueing model may not be suitable for real-time multimedia traffic, which usually has a more complex autocorrelation structure than the Poisson model. Furthermore, it is not clear how to extend the analysis in [24] to more than two paths.

Nelakuditi and Zhang introduced a proportional routing heuristic for routing traffic over multiple paths in [21]. The proposed path selection heuristics give near-optimal performance in terms of throughput for elastic data. In a recent paper [25], each path i was assigned with a weight ω_i such that $\sum_i \omega_i = 1$. An opportunistic scheduling-based scheduler was proposed to send packets to multiple paths while keeping the fraction of bytes transmitted on each path i at ω_i. The authors showed that the large time-scale traffic correlation could be exploited by opportunistic scheduling to reduce the queueing delays on the paths. However, fixed delays, which may have significant impact on traffic partitioning [24], were not considered in this paper. Moreover, it is not clear how to set or derive $\{\omega_i\}$ for a data flow and a set of paths.

Multipath transport was extended to many-to-one type of applications in [46]. An analytical model of parallel data downloading from multiple servers was presented to minimize the resequencing buffer size and total download time. Although this paper has similar objectives as our paper, the analysis was for elastic data transport and is not applicable to real-time applications, where packets are consumed at a certain rate at the receiver end.

In [26], Alasti *et al.* investigated the effect of probabilistic traffic partitioning on MDC and single description coding using M/M/1 and M/D/1 queues to model the paths. It was shown that different splitting probabilities result in different distortion in received data. Although the results provided some useful insights, the assumptions made in [26] limit its applicability. Furthermore, propagation delay, which could be the dominant part of end-to-end delay in high-speed networks, is not considered.

VI. Summary

In this paper, we examined two important issues on the use of multipath transport, namely, minimizing end-to-end delay and path selection. We showed that by optimal traffic partitioning, we can use a minimum set of paths while achieving the minimum delay in $O(N)$ time. The selected path set is optimal in the sense that adding any rejected path to this set will only increase the end-to-end delay. We also discussed the important implications of this work in practice, and provided a practical implementation to enforce the optimal partition on each path. The proposed scheme provides a simple, yet powerful solution to the path selection problem in multipath transport design.

References

[1] S. Mao, S. S. Panwar, and Y. T. Hou, "On optimal traffic partitioning for multipath transport," in *Proc. IEEE INFOCOM*, Miami, FL, Mar. 2005, pp. 2325–2336.

[2] N. F. Maxemchuk, "Diversity routing," in *Proc. IEEE Int. Conf. Communications (ICC)*, San Francisco, CA, Jun. 1975, pp. 10–41.

[3] E. Gustafsson and G. Karlsson, "A literature survey on traffic dispersion," *IEEE Network*, vol. 11, no. 2, pp. 28–36, Mar. 1997.

[4] D. S. Phatak and T. Goff, "A novel mechanism for data streaming across multiple IP links for improving throughput and reliability in mobile environments," in *Proc. IEEE Int. Conf. Computer Communications (INFOCOM)*, New York, Jun. 2002, pp. 773–782.

[5] H.-Y. Hsieh and R. Sivakumar, "A transport layer approach for achieving aggregate bandwidths on multi-homed mobile hosts," in *Proc. ACM Mobile Computing and Networking (MobiCom)*, Atlanta, GA, Sep. 2002, pp. 83–95.

[6] H.-Y. Hsieh, K.-H. Kim, Y. Zhu, and R. Sivakumar, "A receiver-centric transport protocol for mobile hosts with heterogeneous wireless interfaces," in *Proc. ACM Mobile Computing and Networking (MobiCom)*, San Diego, CA, Sep. 2003, pp. 1–15.

[7] N. Gogate, D. Chung, S. S. Panwar, and Y. Wang, "Supporting image/video applications in a multihop radio environment using route diversity and multiple description coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 9, pp. 777–792, Sep. 2002.

[8] S. Mao, S. Lin, S. S. Panwar, Y. Wang, and E. Celebi, "Video transport over *ad hoc* networks: Multistream coding with multipath transport," *IEEE J. Sel. Areas Commun.*, vol. 12, no. 10, pp. 1721–1737, Dec. 2003.

[9] S. Mao, Y. T. Hou, X. Cheng, H. D. Sherali, and S. F. Midkiff, "Multi-path routing for multiple description video over wireless *ad hoc* networks," in *Proc. IEEE Int. Conf. Computer Communications (INFOCOM)*, Miami, FL, Mar. 2005, pp. 740–750.

[10] S. Mao, D. Bushmitch, S. Narayanan, and S. S. Panwar, "MRTP: A multi-flow real-time transport protocol for *ad hoc* networks," *IEEE Trans. Multimedia*, vol. 8, no. 2, Apr. 2006.

[11] J. G. Apostolopoulos, T. Wong, W. Tan, and S. Wee, "On multiple description streaming in content delivery networks," in *Proc. IEEE Int. Conf. Computer Communications (INFOCOM)*, New York, Jun. 2002, pp. 1736–1745.

[12] A. C. Begen, Y. Altunbasak, O. Ergun, and M. H. Ammar, "Multi-path selection for multiple description encoded video streaming," *EURASIP Signal Process. Image Commun.*, vol. 20, no. 1, pp. 39–60, Jan. 2005.

[13] J. Chakareski, S. Han, and B. Girod, "Layered coding vs. multiple descriptions for video streaming over multiple paths," in *Proc. ACM Multimedia*, Berkeley, CA, Nov. 2003, pp. 422–431.

[14] T. Nguyen and A. Zakhor, "Path diversity with forward error correction (PDF) system for packet switched networks," in *Proc. IEEE Int. Conf. Computer Communications (INFOCOM)*, San Francisco, CA, Apr. 2003, pp. 663–672.

[15] R. Stewart, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and V. Paxson, *Stream Control Transmission Protocol*, Oct. 2000. IETF RFC 2960.

[16] S. Narayanan, D. Bushmitch, S. Mao, and S. S. Panwar, *The Multi-Flow Real-Time Transport Protocol*, Aug. 2004. IETF Internet Draft: draft-narayanan-mrtp-00.txt.

[17] R. G. Ogier, V. Rutenburg, and N. Chacham, "Distributed algorithm for computing shortest pairs of disjoint paths," *IEEE Trans. Inf. Theory*, vol. 39, no. 2, pp. 443–455, Mar. 1993.

[18] D. Eppstein, "Finding the k shortest paths," *SIAM J. Comput.*, vol. 28, no. 2, pp. 652–673, Aug. 1999.

[19] A. Tsirigos and Z. J. Haas, "Analysis of multipath routing—Part I: The effect on the packet delivery ratio," *IEEE Trans. Wireless Commun.*, vol. 3, no. 1, pp. 138–146, Jan. 2004.

[20] D. B. Johnson, D. A. Maltz, and Y.-C. Hu, *The Dynamic Source Routing Protocol for Mobile ad hoc Networks (DSR)*, Apr. 2003. IETF Internet Draft: draft-ietf-manet-dsr-09.txt.

[21] S. Nelakuditi and Z. Zhang, "On selection of paths for multipath routing," in *Proc. IEEE Int. Workshop Quality Service (IWQoS)*, Karlsruhe, Germany, Jun. 2001, pp. 170–186.
[22] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, *RTP: A Transport Protocol for Realtime Applications*, Jul. 2003. IETF Request For Comments 3550.
[23] R. L. Cruz, "A calculus for network delay—Part I: Network elements in isolation," *IEEE Trans. Inf. Theory*, vol. 37, no. 1, pp. 114–131, Jan. 1991.
[24] N. Gogate and S. S. Panwar, "On a resequencing model for high speed networks," in *Proc. IEEE Int. Conf. Computer Communications (INFOCOM)*, Toronto, ON, Canada, Jun. 1994, pp. 40–47.
[25] C. Cetinkaya and E. W. Knightly, "Opportunistic traffic scheduling over multiple network paths," in *Proc. IEEE Int. Conf. Computer Communications (INFOCOM)*, Hong Kong, Mar. 2004, pp. 1928–1937.
[26] M. Alasti, K. Sayrafian-Pour, A. Ephremides, and N. Farvardin, "Multiple description coding in networks with congestion problem," *IEEE Trans. Inf. Theory*, vol. 47, no. 3, pp. 891–902, Mar. 2001.
[27] Office of Science, Department of Energy, *The High-Performance Networks (HPN) Program*. [Online]. Available: http://www.sc.doe.gov/ascr/mics/hpn
[28] T. L. Huston and J. L. Huston, "Is telemedicine a practical reality," *Commun. ACM*, vol. 43, no. 6, pp. 91–95, Jun. 2000.
[29] C.-S. Chang, "Stability, queue length, and delay of deterministic and stochastic queueing networks," *IEEE Trans. Autom. Control*, vol. 39, no. 5, pp. 913–931, May 1994.
[30] D. A. Pierre, *Optimization Theory With Applications*. New York: Dover, 1986.
[31] E. Altman, R. El Azouzi, and V. Abramov, "Non-cooperative routing in loss networks," *Perform. Eval.*, vol. 49, no. 1–4, pp. 257–272, Sep. 2002.
[32] M. Beckmann, C. B. McGuire, and C. B. Winsten, *Studies in the Economics of Transportation*. New Haven, CT: Yale Univ. Press, 1956.
[33] S. Mao, "Realtime multimedia transport using multiple paths," Ph.D. dissertation, Dept. Elect. Comput. Eng., Polytechnic Univ., Brooklyn, NY, Jan. 2004.
[34] J.-Y. Le Boudec and P. Thiran, *Network Calculus: Theory of Deterministic Queueing Systems for the Internet*. New York: Springer-Verlag, 2002.
[35] Cisco Internet Operating System (IOS) Documentation, *Cisco IOS Configuration Fundamentals Configuration Guide—Release 12.2*. [Online]. Available: http://www.cisco.com
[36] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala, "RSVP: A new resource reservation protocol," *IEEE Network*, vol. 7, no. 5, pp. 8–18, Sep. 1993.
[37] Z.-L. Zhang, "End-to-end support for statistical quality-of-service guarantees in multimedia networks," Ph.D. dissertation, Dept. Comput. Sci., Univ. Massachusetts, Amherst, Feb. 1997.
[38] R. L. Carter and M. E. Crovella, "Measuring bottleneck link speed in packet-switched networks," *Perform. Eval.*, vol. 27–28, no. 4, pp. 297–318, Oct. 1996.
[39] M. Jain and C. Dovrolis, "End-to-end available bandwidth: Measurement methodology, dynamics, and relation with TCP throughput," *IEEE/ACM Trans. Netw.*, vol. 11, no. 4, pp. 537–549, Aug. 2003.
[40] R. Kapoor, L.-J. Chen, L. Lao, M. Gerla, and M. Y. Sanadidi, "Capprobe: A simple and accurate capacity estimation technique," in *Proc. ACM SIGCOMM*, Portland, OR, Oct. 2004, pp. 67–78.
[41] O. Gurewitz and M. Sidi, "Estimating one-way delays from cyclic-path delay measurements," in *Proc. IEEE Int. Conf. Computer Communications (INFOCOM)*, Anchorage, AK, Apr. 2001, pp. 1038–1044.
[42] F. Lo Presti, N. G. Duffield, J. Horowitz, and D. Towsley, "Multicast-based inference of network-internal delay distributions," *IEEE/ACM Trans. Netw.*, vol. 10, no. 6, pp. 761–775, Dec. 2002.
[43] R. Krishnan and J. A. Silvester, "Choice of allocation granularity in multipath source routing schemes," in *Proc. IEEE/ACM Int. Conf. Computer Communications (INFOCOM)*, San Francisco, CA, Mar. 1993, pp. 322–329.
[44] D. Bushmitch, R. Izmailov, S. Panwar, and A. Pal, "Thinning, striping and shuffling: Traffic shaping and transport techniques for variable bit rate video," in *Proc. IEEE Global Telecommunications Conf. (GLOBECOM)*, Taipei, Taiwan, R.O.C., Nov. 2002, pp. 1495–1501.
[45] D. Bushmitch, "Thinning, striping and shuffling: Traffic shaping and transport techniques for VBR video," Ph.D. dissertation, Dept. Elect. Comput. Eng., Polytechnic Univ., Brooklyn, NY, Jan. 2004.
[46] Y. Nebat and M. Sidi, "Resequencing considerations in parallel downloads," in *Proc. IEEE Int. Conf. Computer Communications (INFOCOM)*, New York, Jun. 2002, pp. 1326–1335.

Shiwen Mao (S'99–M'04) received the B.S. and M.S. degrees in electrical engineering from Tsinghua University, Beijing, China, in 1994 and 1997, respectively, and the M.S. degree in system engineering and the Ph.D. degree in electrical and computer engineering from Polytechnic University, Brooklyn, NY, in 2000 and 2004, respectively.

From 1997 to 1998, he was a Research Member at IBM China Research Lab, Beijing, China. In the summer of 2001, he was a Research Intern at Avaya Labs Research, Holmdel, NJ. He is currently a Research Scientist at the Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg. He is the coauthor of the textbook *TCP/IP Essentials: A Lab-Based Approach* (Cambridge University Press, 2004). His research interests include multimedia and wireless networking.

Dr. Mao was the corecipient of the 2004 IEEE Communications Society Leonard G. Abraham Prize in the Field of Communications Systems.

Shivendra S. Panwar (S'82–M'85–SM'00) received the B.Tech. degree in electrical engineering from the Indian Institute of Technology, Kanpur, India, in 1981 and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Massachusetts, Amherst, in 1983 and 1986, respectively.

He was a Visiting Scientist at the IBM T.J. Watson Research Center, Yorktown Heights, NY, in the summer of 1987, and was a Consultant at AT&T Bell Laboratories, Holmdel, NJ. He joined the Department of Electrical Engineering, Polytechnic Institute of New York, Brooklyn (now Polytechnic University). He is currently the Director of the New York State Center for Advanced Technology in Telecommunications, New York, and a Professor at the Electrical and Computer Engineering Department, Polytechnic University. He is the coeditor of two books: *Network Management and Control, Vol. II* and *Multimedia Communications and Video Coding* (Plenum, 1994 and 1996) and coauthored a textbook *TCP/IP Essentials: A Lab-Based Approach* (Cambridge University Press, 2004). His research interests include the performance analysis and design of networks. Current work includes video systems over peer-to-peer networks, switch performance, and wireless networks.

Dr. Panwar is a member of the Technical Committee on Computer Communications. He was the Secretary of the Technical Affairs Council of the IEEE Communications Society from 1992 to 1993. He was the corecipient of the 2004 IEEE Communications Society Leonard G. Abraham Prize in the Field of Communications Systems.

Y. Thomas Hou (S'91–M'98–SM'04) received the B.E. degree from the City College of New York in 1991, the M.S. degree from Columbia University, New York, NY, in 1993, and the Ph.D. degree from Polytechnic University, Brooklyn, NY, in 1998, all in electrical engineering.

From 1997 to 2002, he was a Research Scientist and Project Leader at the IP Networking Research Department, Fujitsu Laboratories of America, Sunnyvale, CA (Silicon Valley). Since Fall 2002, he has been an Assistant Professor at the Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg. In recent years, he has worked on scalable architectures, protocols, and implementations for differentiated services Internet; service overlay networking; multimedia streaming over the Internet; and network bandwidth allocation policies and distributed flow control algorithms. He has published extensively in the above areas. His research interests are algorithmic design and optimization for network systems. His current research focuses on wireless sensor networks and multimedia over wireless *ad hoc* networks.

Dr. Hou was a corecipient of the 2004 IEEE Communications Society Multimedia Communications Best Paper Award, the 2002 IEEE International Conference on Network Protocols Best Paper Award, and the 2001 IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY Best Paper Award.