Flow Assignment and Packet Scheduling for Multipath Routing

Ka-Cheong Leung and Victor O. K. Li

Abstract: **In this paper, we propose a framework to study how to route packets efficiently in multipath communication networks. Two traffic congestion control techniques, namely, flow assignment and packet scheduling, have been investigated. The flow assignment mechanism defines an optimal splitting of data traffic on multiple disjoint paths. The resequencing delay and the usage of the resequencing buffer can be reduced significantly by properly scheduling the sending order of all packets, say, according to their expected arrival times at the destination. To illustrate our model, and without loss of generality, Gaussian distributed end-to-end path delays are used. Our analytical results show that the techniques are very effective in reducing the average end-to-end path delay, the average packet resequencing delay, and the average resequencing buffer occupancy for various path configurations. These promising results can form a basis for designing future adaptive multipath protocols.**

Index Terms: **Computer communications, dispersity routing, flow assignment, high speed networks, multipath routing, packet scheduling, performance modeling, resequencing, traffic dispersion, traffic engineering.**

I. INTRODUCTION

Multipath routing or spatial traffic dispersion [1]–[6] is a load balancing technique in which the total load from a source to a destination is spatially distributed over several paths. It is useful for relieving congestion and delivering quality of service (QOS) guarantees in communication networks. Generally speaking, congestion describes the network state where the number of packets transmitted through a network approaches or exceeds the network packet-handling capacity. The idea of congestion control is to keep the number of packets within the network below the level at which network performance degrades dramatically [7].

The convergence of the computer, communications, entertainment, and consumer electronics industry is driving an explosive growth in multimedia applications [8]. Recent studies show that multimedia traffic exhibits variability or correlation on various time scales [9], [10]. Such long-range dependence property has a considerably unpleasant impact on queueing performance, and is a dominant characteristic for a number of packet traffic engineering problems [11].

Nowadays, the standard routing approach in communication networks consists of finding a single shortest path from a source to a destination based on some heuristic link cost metrics, which are updated periodically. Although such unipath routing protocols can adapt very quickly to changing network conditions, they become unstable under heavy loads or bursty traffic as the link cost metrics used in the routing algorithms are related to delays or congestion experienced over the network links [12], [14].

Multipath routing has recently been found to be an effective method to alleviate the adverse effects of traffic bursts [3]. In addition, multipath routing protocols help to spread out congestion and thus minimize network delays [14]. The key to multipath routing is how to allocate a proper portion of traffic to each participating path so as to satisfy the desired objective, such as the minimization of the average end-to-end path delay. This is known as *flow assignment* or *optimal routing* [12]–[14]. Most of the existing work considered the traffic arrival rate at every link in the network, and found an optimal routing which directs traffic exclusively on least-cost paths with respect to some link costs that depend on the flows carried by the links. It is computationally expensive to find an optimal flow assignment for each source-destination pair. The solution is generally not scalable in terms of the size of the network considered.

Moreover, packets which travel along different paths may arrive out of order. Those packets arriving out of order may have to be resequenced, i.e., stored in a buffer, called a resequencing buffer, until they can be delivered to the end process in the proper order. Broadly speaking, there are two different approaches to recover the order at the receiver. The first and the most popular way to do so is to make use of or insert various variants of sequence numbers into packets [3], [15]. Another method is to utilize synchronization markers and some properties of routing protocols, like the use of the deficit counters in [16]. The latter approach can avoid the use of explicit sequence numbers, but is very sensitive to packet losses as a packet loss can cause resequencing protocols to lose synchronization, which can be recovered only when a marker arrives. Nevertheless, resequencing should be minimized [17].

Cyclic traffic dispersion is a method to evenly distribute packets over all active paths. Many existing traffic dispersion strategies, such as dispersity routing [1], [5], [6], the string-mode protocol [2], and the vector routing algorithm [18], utilize this idea. The major advantage of this approach is that it is quite simple to implement. However, this method does not take the path heterogeneity into account. With heterogeneous paths, the best way to spread traffic along multiple paths may not be by cyclic dis-

Manuscript received January 11, 2002; approved for publication by Jaiyong Lee, Division III Editor, July 16, 2003.

K.-C. Leung is with the Department of Computer Science, Texas Tech University, Lubbock, TX 79409-3104, USA, email: kcleung@cs.ttu.edu.

V. O. K. Li is with the Department of Electrical and Electronic Engineering, the University of Hong Kong, Pokfulam Road, Hong Kong, China, email: vli@eee.hku.hk.

This research was supported in part by the Areas of Excellence Scheme established under the University Grants Committee of the Hong Kong Special Administrative Region, China (Project No. AoE/E-01/99). An earlier version of this paper appeared in the Proceedings of the IEEE GLOBECOM 1999.

persion, since it may not achieve the desired objective, such as minimizing the end-to-end path delay. In addition, it may induce substantial packet resequencing delay when the end-to-end delays are quite different among these paths.

Adaptive routing schemes [19], [20] have been proposed to spread packets dynamically over multiple paths according to the network load. These procedures require parameters to determine the load distribution. Yet, the calculations of such parameters are either computationally intensive or done in an ad hoc manner. Thus, there is a need for new multipath routing schemes that allow a rapid computation of the optimal load distribution parameters.

Since resequencing is due to packets arriving out of order at the destination, instead of sending packets from the source according to their sequence numbers, the packet sending order can be *scheduled* anew to minimize or to reduce the necessity for resequencing.

A. Our Contributions

The objective of this work is to propose a framework to study how to route packets efficiently in multipath communication networks. Two traffic congestion control techniques, namely, flow assignment and packet scheduling, have been investigated. The flow assignment technique works for any routing strategies, such as deterministic routing (with a pre-determined routing sequence to route packets to a set of paths) and probabilistic routing (by sending a packet on a path at random), to distribute traffic on multiple disjoint paths. The packet scheduling technique operates for any routing strategies with pre-determined routing sequences. These two techniques can also be applied to both connectionless and connection-oriented communication networks.

The flow assignment mechanism defines an optimal splitting of data traffic on multiple disjoint paths based on pre-defined objectives, such as the minimization of the average end-to-end path delay. Instead of formulating the problem based on link costs, we formulate it as an end-to-end optimal routing problem. This allows the problem to be solved in real time.

The resequencing delay and the usage of the resequencing buffer can be reduced significantly by properly scheduling the sending order of all packets, say, according to their expected arrival times at the destination. The difference in the average end-to-end path delays are compensated, and hence the need for resequencing can be diminished.

Here, the words "source" and "destination" are quite general. A source can mean a source host, which generates traffic. It can also mean a source router, from which network traffic departs. A destination can be defined similarly. It can mean a destination host, which takes and absorbs traffic. It can also mean a destination router, to which network traffic arrives. Though our proposed techniques are end-to-end based for source-based routing, they can be applied to perform traffic engineering under various scenarios, ranging from inter-router to inter-host traffic.

We will investigate the effectiveness of these control mechanisms for multipath routing by examining three basic questions:

- What is the optimal split of traffic to achieve the best performance?
- How effective is the packet scheduling technique in improving performance?
- Under what circumstances should we employ the optimal split of traffic instead of cyclic traffic dispersion, and the packet scheduling technique, in order to achieve a significant performance improvement?

B. Organization of the Paper

This paper is organized as follows. Section II gives a flow assignment model to compute an optimal splitting of data traffic on multiple paths. Section III presents a packet scheduling model to find a proper sending schedule so as to reduce the packet resequencing delay and the resequencing buffer occupancy. Section IV examines the analytical results derived from the framework and studies the effectiveness of the flow assignment and packet scheduling techniques on multipath routing. Section V concludes and discusses some possible extensions to our work.

II. FLOW ASSIGNMENT MODEL

Our flow model consists of a disordering network connecting the source to the destination, and flows of packets. The disordering network consists of a set of N disjoint paths, namely Path 1, Path 2, $\cdots$, Path N, connecting the source to the destination, such that packets may arrive at the destination in a different order as they are sent. Paths are *disjoint* if and only if they do not share any directed links.

We assume the source has an unlimited supply of packets, which are delivered to paths according to deterministic or probabilistic routing. The routing weight for Path i, p_i, is defined as the portion of dispersed traffic to be routed to Path i, where $\sum_{i=1}^{N} p_i = 1$. Denote the flow configuration by $\boldsymbol{p} = \begin{pmatrix} p_1 & p_2 & \cdots & p_N \end{pmatrix}$. Due to limited path capacity, it may not be possible to route all packets to some of these paths. Therefore, the load distribution p_i is feasible if and only if $0 \le p_i < M_i \le 1$, where M_i is the upper bound of p_i, for all $i = 1, 2, \cdots, N$. A flow configuration is said to be a *feasible flow configuration* if and only if $0 \le p_i < M_i \le 1$ for all $i = 1, 2, \cdots, N$. The following lemma states that the set of all feasible flow configurations is convex.

Lemma 1: The set of all feasible flow configurations, $\mathcal{S}$, is convex.

Proof: Suppose $\boldsymbol{f}$ and $\boldsymbol{g}$ are two feasible flow configurations. Since f_i and g_i are non-negative for all $i = 1, 2, \cdots, N$,

$$\alpha\, f_i + (1 - \alpha) \cdot g_i \ge 0, \tag{1}$$

where $0 \le \alpha \le 1$.

Since f_i and g_i are both upper-bounded by M_i for all $i = 1, 2, \cdots, N$,

$$\alpha\, f_i + (1 - \alpha) \cdot g_i < \alpha\, M_i + (1 - \alpha) \cdot M_i = M_i. \tag{2}$$

Combining, $\alpha\, \boldsymbol{f} + (1 - \alpha) \cdot \boldsymbol{g}$ is a feasible flow configuration. Thus, the set $\mathcal{S}$ is convex. □

Let $C_i(p_i)$ denote the non-negative monotonically increasing cost function of transmitting packets on Path i with the routing weight p_i, where $i = 1, 2, \cdots, N$. $C_i(p_i)$ is well defined for all feasible flow configurations. The notion of 'cost' here is quite general. It can represent any system and user costs, including the average end-to-end path delay, the average packet loss probability, the average total buffer requirement for all nodes along the path, and so on. Hence, the expected cost for multipath routing can be written as:

$$\overline{C}(\boldsymbol{p}) = \sum_{i=1}^{N} p_i \, C_i(p_i). \tag{3}$$

Let $C_i(p_i)$ be a non-negative monotonically increasing convex function in p_i. The subsequent two lemmas show that $p_i \, C_i(p_i)$ and $\overline{C}(\boldsymbol{p})$ are also convex functions.

Lemma 2: For every $i = 1, 2, \cdots, N$, $p_i \, C_i(p_i)$ is a convex function in p_i, where p_i is a feasible load distribution.

Proof: Without loss of generality, suppose f_i and g_i are two feasible load distributions such that $f_i \leq g_i$. In other words, $0 \leq C_i(f_i) \leq C_i(g_i)$, which implies that

$$\alpha \cdot (1-\alpha) \cdot (g_i - f_i) \cdot [C_i(f_i) - C_i(g_i)] \leq 0, \tag{4}$$

where $0 \leq \alpha \leq 1$.

Since the set of all feasible flow configurations is convex, the set of all feasible load distributions is convex and thus $h_i = \alpha f_i + (1-\alpha) \cdot g_i$ is also a feasible load distribution. Using the above expression,

$$\begin{aligned} & h_i \, C_i(h_i) \\ & \leq [\alpha \, f_i + (1-\alpha) \cdot g_i] \cdot [\alpha \, C_i(f_i) + (1-\alpha) \cdot C_i(g_i)] \\ & = \alpha f_i \, C_i(f_i) + (1-\alpha) \cdot g_i \, C_i(g_i) \\ & \quad + \alpha \cdot (1-\alpha) \cdot (g_i - f_i) \cdot [C_i(f_i) - C_i(g_i)] \\ & \leq \alpha \, f_i \, C_i(f_i) + (1-\alpha) \cdot g_i \, C_i(g_i). \end{aligned} \tag{5}$$

This means that $p_i \, C_i(p_i)$ is a convex function in p_i, where $i = 1, 2, \cdots, N$. □

Lemma 3: The expected cost $\overline{C}(\boldsymbol{p})$ is a convex function in $\boldsymbol{p}$, where $\boldsymbol{p}$ is a feasible flow configuration.

Proof: Suppose $\boldsymbol{f}$ and $\boldsymbol{g}$ are two feasible flow configurations. By applying Lemma 2,

$$\begin{aligned} & \overline{C}(\alpha \, \boldsymbol{f} + (1-\alpha) \cdot \boldsymbol{g}) \\ & = \sum_{i=1}^{N} [\alpha \, f_i + (1-\alpha) \cdot g_i] \cdot C_i(\alpha \, f_i + (1-\alpha) \cdot g_i) \\ & \leq \sum_{i=1}^{N} [\alpha \, f_i \, C_i(f_i) + (1-\alpha) \cdot g_i \, C_i(g_i)] \\ & = \alpha \sum_{i=1}^{N} f_i \, C_i(f_i) + (1-\alpha) \cdot \sum_{j=1}^{N} g_j \, C_j(g_j) \\ & = \alpha \, \overline{C}(\boldsymbol{f}) + (1-\alpha) \cdot \overline{C}(\boldsymbol{g}). \end{aligned} \tag{6}$$

Thus, the convexity of the cost function $\overline{C}(\boldsymbol{p})$ follows. □

Since the expected cost $\overline{C}(\boldsymbol{p})$ is a convex function in $\boldsymbol{p}$ and all feasible flow configurations $\boldsymbol{p}$ composes the convex set $\mathcal{S}$, $\overline{C}(\cdot)$ has an unique global minimum on $\mathcal{S}$ if $\mathcal{S}$ is non-empty [21]. Iterative feasible direction algorithms like the constrained gradient projection method [12] can be adopted to compute the minimum expected cost and the corresponding feasible flow configuration.

If the path cost function is not known in advance and it actually represents the average end-to-end path delay, we can apply the technique described in [22] to perform on-line marginal delay estimation for each participating path. The method is based on perturbation analysis that assumes no knowledge of network parameters, such as arrival rates and link capacities. We also believe that it is possible to apply the technique to other cost functions as well.

To complete our flow assignment model, the cost function $C_i(p_i)$ is formulated as the average end-to-end delay on Path i, $\overline{D_i}(p_i)$, for our succeeding analytical studies. That is,

$$C_i(p_i) = \overline{D_i}(p_i), \tag{7}$$

where $i = 1, 2, \cdots, N$.

A. Error-Correcting Codes

Error correcting codes may be used to detect and correct a number of missing or erroneous bits of data. Suppose the number of additional bits required for the error correcting codes is proportional to the amount of user data. Given the routing weight for Path i is p_i, the cost of transmitting packets on Path i, including error correcting codes, is:

$$C_i(v_i) = C_i(p_i \cdot (1 + \eta_i)), \tag{8}$$

where η_i is a constant and path specific, for all $i = 1, 2, \cdots, N$.

Provided that $\boldsymbol{v}$ is a feasible flow configuration, all lemmas derived above are still valid. It is thus always possible to apply the constrained gradient projection method [12] to find the optimal split of traffic.

B. Non-Disjoint Paths

A pair of paths are said to be *non-disjoint* if and only if they share at least one directed link. A set of non-disjoint paths is one that has at least one pair of non-disjoint paths. Although the aforementioned problem formulation requires that all participating paths have to be disjoint, the authors believe that the method can be extended to handle cases where some paths are not disjoint with each other. The source can still easily check whether it is feasible to distribute traffic over a set of paths under a certain configuration, as the set of all feasible flow configurations over a set of paths (either disjoint or non-disjoint) can be shown to be convex. Given a convex objective cost function, it is always possible to apply the constrained gradient projection method [12] to find the optimal split of traffic.

III. PACKET SCHEDULING MODEL

In Section II, we have proposed to employ the flow assignment method to minimize the expected cost for multipath routing. However, this method merely distributes the load to N paths. Given a sequence of W ordered packets to be transmitted, there are many ways to order the sending sequence for these packets from the source. Each order of sending sequence for a set of packets is called a *sending schedule*. A sending schedule can have very different packet resequencing delay and resequencing buffer distributions when compared with another schedule. This means that care must be taken to determine the best schedule. Given a pre-determined routing sequence to route these packets to a set of paths, we propose a packet scheduling mechanism to minimize or to reduce the consumption of the resequencing buffer and the delay on resequencing.

Suppose p_i is the routing weight for Path i, where $\sum_{i=1}^{N} p_i = 1$. Denote $\boldsymbol{p} = \begin{pmatrix} p_1 & p_2 & \cdots & p_N \end{pmatrix}$. To achieve the best system performance, packets are distributed to each path in a cyclical fashion so that the arrival instances of any two packets to each path is as uniform as possible. Algorithms for constructing such routing sequences can be found in [23]. These routing sequences contain subsequences that can be used to re-construct the whole sequence by the repeated applications of the same subsequence. Nevertheless, our packet scheduling model does not assume any specific routing sequences are used.

A *round* is defined as the minimal subsequence. It is possible to construct a round of length Q such that q_i packets are sent on Path i, $i = 1, 2, \cdots, N$, in each round, where $p_i \approx \frac{q_i}{Q}$ and $Q = \sum_{j=1}^{N} q_j$. Thus, we can denote $\mathcal{P}_i(j) = k$ as the position, in every round, of the j^{th} packet to be sent on Path i, and define $\mathcal{R}(k) = i$ to be the identity of the path on which the packet in Position k of every round will be transmitted. For example, consider the periodic routing sequence $121312\cdots$ such that six consecutive packets are transmitted on Paths 1, 2, 1, 3, 1, 2, respectively, and so on. Therefore, $\boldsymbol{p} = \begin{pmatrix} \frac{1}{2} & \frac{1}{3} & \frac{1}{6} \end{pmatrix}$, $Q = \sum_{j=1}^{3} q_j = 3 + 2 + 1 = 6$, $\mathcal{P}_1(2) = 3$, and $\mathcal{R}(3) = 1$.

The sequence number of the packet sent at Round r, $r \geq 1$, and Position k is designated as $\mathcal{O}(r, k)$, where $\mathcal{O}(r, k) \geq 1$. Denote by t_m, d_m, and a_m, respectively, the sending time of the first bit of the packet with the sequence number m from the source, the end-to-end path delay experienced by the packet, and the arrival time of the last bit of the packet at the destination. The relationship among these three quantities is:

$$a_m = t_m + d_m. \tag{9}$$

Let $R(\boldsymbol{s})$ denote the resequencing cost function for multipath routing under the sending schedule $\boldsymbol{s}$. Our objective is to find an optimal sending schedule $\boldsymbol{s}^*$ such that $R(\boldsymbol{s})$ is minimized at $\boldsymbol{s} = \boldsymbol{s}^*$.

To complete our packet scheduling model, the resequencing cost function $R(\boldsymbol{s})$ is expressed as:

$$R(\boldsymbol{s}) = \sum_{k=2}^{W} \max_{l=1}^{k-1}(0, \overline{a_l} - \overline{a_k}), \tag{10}$$

where $\overline{a_k} = E[a_k]$.

$R(\boldsymbol{s})$ can be minimized to take on value 0 if it is possible to obtain the optimal sending schedule such that the expected arrival times for all packets are in ascending order of their sequence numbers. Ties are resolved by sending the packet with the smaller sequence number earlier. If all packets arrive at their expected arrival times, resequencing is not needed. The task is reduced to searching for such a schedule.

Denote by $\overline{S}(i, r, j)$ and $\overline{A}(i, r, j)$, respectively, the expected sending time of the first bit of the packet at Round r and Position $\mathcal{P}_i(j)$ from the source, and the expected arrival time of the last bit of the packet at the destination. Without loss of generality, the first packet, i.e., at Round 1 and Position 1, is expected to be sent at Time 0. That is,

$$\overline{S}(\mathcal{R}(1), 1, 1) = 0. \tag{11}$$

Thus, the packet at Round r and Position $\mathcal{P}_i(j)$ is expected to be transmitted at:

$$\overline{S}(i, r, j) = [(r-1) \cdot Q + \mathcal{P}_i(j) - 1] \cdot \overline{\Delta}, \tag{12}$$

where $\overline{\Delta}$ denotes the average inter-packet time, i.e., the mean time between two successive packets to be transmitted from the source. This information can be inferred directly from the estimated average bandwidth consumed by the traffic source.

Suppose the average end-to-end delay for Path i is $\overline{D_i}$. This information may be obtained by inferring the network statistics from the underlying network protocols, if they find feasible paths based on the cost metrics, which involve the use of the average path delays. In addition, the source can estimate the average path delays by exchanging control information periodically with the destination. The expected packet arrival time can be expressed as:

$$\overline{A}(i, r, j) = \overline{S}(i, r, j) + \overline{D_i}. \tag{13}$$

Packets are scheduled such that a packet with a smaller sequence number $\mathcal{O}(r_1, \mathcal{P}_{i_1}(j_1))$ is expected to arrive at the destination no later than one with a larger sequence number $\mathcal{O}(r_2, \mathcal{P}_{i_2}(j_2))$. This schedule order relationship can be written as:

$$\mathcal{O}(r_1, \mathcal{P}_{i_1}(j_1)) < \mathcal{O}(r_2, \mathcal{P}_{i_2}(j_2)), \tag{14}$$

if and only if

$$\overline{A}(i_1, r_1, j_1) < \overline{A}(i_2, r_2, j_2) \tag{15}$$

or

$$\begin{cases} \overline{A}(i_1, r_1, j_1) = \overline{A}(i_2, r_2, j_2) \\ \overline{S}(i_1, r_1, j_1) < \overline{S}(i_2, r_2, j_2). \end{cases} \tag{16}$$

It is time to introduce the theorem that guides the construction of the optimal schedule.

Theorem 1: By rearranging the packet sending order, the packet sent at Round r, $r = 1, 2, \cdots$, and Position $\mathcal{P}_i(j)$, $i = 1, 2, \cdots, N$, $j = 1, 2, \cdots, q_i$, is the one with the sequence number:

$$\begin{aligned} \mathcal{O}(r, \mathcal{P}_i(j)) = & \sum_{\substack{u=1 \\ u \neq i}}^{N} \sum_{v=1}^{q_u} \max(0, r - \xi(i, j, u, v)) \\ & + (r-1) \cdot q_i + j, \end{aligned} \tag{17}$$

where

$$\begin{aligned} & \xi(i, j, u, v) \\ & = \begin{cases} \frac{(\overline{D_u} - \overline{D_i}) + [\mathcal{P}_u(v) - \mathcal{P}_i(j)] \cdot \overline{\Delta}}{Q \cdot \overline{\Delta}} + 1 \\ \qquad \text{if } \begin{cases} \frac{\overline{D_u} - \overline{D_i}}{\overline{\Delta}} \equiv \mathcal{P}_i(j) - \mathcal{P}_u(v) \pmod{Q} \\ \overline{D_i} > \overline{D_u} \end{cases} \\ \left\lceil \frac{(\overline{D_u} - \overline{D_i}) + [\mathcal{P}_u(v) - \mathcal{P}_i(j)] \cdot \overline{\Delta}}{Q \cdot \overline{\Delta}} \right\rceil \qquad \text{otherwise.} \end{cases} \end{aligned} ; \tag{18}$$

Proof: Denote $\gamma(i,j,u,v)$ as the number of packets sent on Path u at Position $\mathcal{P}_u(v)$ with sequence number less than $\mathcal{O}(r,\mathcal{P}_i(j))$, where $u \neq i$. There are three different cases to consider.

Case 1:

$$\begin{cases} \frac{\overline{D_u}-\overline{D_i}}{\overline{\Delta}} \equiv \mathcal{P}_i(j) - \mathcal{P}_u(v) \pmod{Q} \\ \overline{D_i} > \overline{D_u}, \end{cases} \tag{19}$$

$$\begin{aligned} \mathcal{O}(\gamma(i,j,u,v),\mathcal{P}_u(v)) &< \mathcal{O}(r,\mathcal{P}_i(j)) \\ &< \mathcal{O}(\gamma(i,j,u,v)+1,\mathcal{P}_u(v)), \end{aligned} \tag{20}$$

if and only if

$$\overline{A}(i,r,j) = \overline{A}(u,\gamma(i,j,u,v)+1,v), \tag{21}$$

if and only if

$$\gamma(i,j,u,v) = \frac{(\overline{D_i}-\overline{D_u}) + [\mathcal{P}_i(j)-\mathcal{P}_u(v)]\cdot\overline{\Delta}}{Q\cdot\overline{\Delta}} + r - 1. \tag{22}$$

When $\frac{(\overline{D_i}-\overline{D_u})+[\mathcal{P}_i(j)-\mathcal{P}_u(v)]\cdot\overline{\Delta}}{Q\cdot\overline{\Delta}} \leq 1-r$, $\mathcal{O}(r,\mathcal{P}_i(j)) < \mathcal{O}(1,\mathcal{P}_u(v))$ and hence $\gamma(i,j,u,v) = 0$. Combining, $\gamma(i,j,u,v) = \max(0, \frac{(\overline{D_i}-\overline{D_u})+[\mathcal{P}_i(j)-\mathcal{P}_u(v)]\cdot\overline{\Delta}}{Q\cdot\overline{\Delta}} + r - 1)$.

Case 2:

$$\begin{cases} \frac{\overline{D_u}-\overline{D_i}}{\overline{\Delta}} \equiv \mathcal{P}_i(j) - \mathcal{P}_u(v) \pmod{Q} \\ \overline{D_i} < \overline{D_u}, \end{cases} \tag{23}$$

$$\begin{aligned} \mathcal{O}(\gamma(i,j,u,v),\mathcal{P}_u(v)) &< \mathcal{O}(r,\mathcal{P}_i(j)) \\ &< \mathcal{O}(\gamma(i,j,u,v)+1,\mathcal{P}_u(v)), \end{aligned} \tag{24}$$

if and only if

$$\overline{A}(i,r,j) = \overline{A}(u,\gamma(i,j,u,v),v), \tag{25}$$

if and only if

$$\gamma(i,j,u,v) = \frac{(\overline{D_i}-\overline{D_u}) + [\mathcal{P}_i(j)-\mathcal{P}_u(v)]\cdot\overline{\Delta}}{Q\cdot\overline{\Delta}} + r. \tag{26}$$

When $\frac{(\overline{D_i}-\overline{D_u})+[\mathcal{P}_i(j)-\mathcal{P}_u(v)]\cdot\overline{\Delta}}{Q\cdot\overline{\Delta}} \leq -r$, $\mathcal{O}(r,\mathcal{P}_i(j)) < \mathcal{O}(1,\mathcal{P}_u(v))$ and hence $\gamma(i,j,u,v) = 0$. Combining, $\gamma(i,j,u,v) = \max(0, \frac{(\overline{D_i}-\overline{D_u})+[\mathcal{P}_i(j)-\mathcal{P}_u(v)]\cdot\overline{\Delta}}{Q\cdot\overline{\Delta}} + r)$.

Case 3:

$$\frac{\overline{D_u}-\overline{D_i}}{\overline{\Delta}} \not\equiv \mathcal{P}_i(j) - \mathcal{P}_u(v) \pmod{Q}, \tag{27}$$

$$\begin{aligned} \mathcal{O}(\gamma(i,j,u,v),\mathcal{P}_u(v)) &< \mathcal{O}(r,\mathcal{P}_i(j)) \\ &< \mathcal{O}(\gamma(i,j,u,v)+1,\mathcal{P}_u(v)), \end{aligned} \tag{28}$$

if and only if

$$\overline{A}(u,\gamma(i,j,u,v),v) < \overline{A}(i,r,j) < \overline{A}(u,\gamma(i,j,u,v)+1,v), \tag{29}$$

if and only if

$$\gamma(i,j,u,v) = r - \lceil \frac{(\overline{D_u}-\overline{D_i}) + [\mathcal{P}_u(v)-\mathcal{P}_i(j)]\cdot\overline{\Delta}}{Q\cdot\overline{\Delta}} \rceil. \tag{30}$$

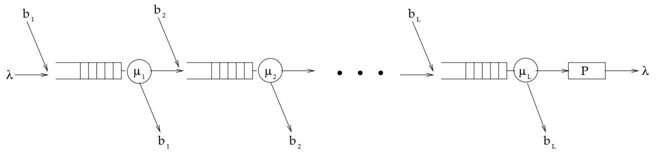


Fig. 1. The L-hop path example.

When $\lceil \frac{(\overline{D_u}-\overline{D_i})+[\mathcal{P}_u(v)-\mathcal{P}_i(j)]\cdot\overline{\Delta}}{Q\cdot\overline{\Delta}} \rceil \geq r$, $\mathcal{O}(r,\mathcal{P}_i(j)) < \mathcal{O}(1,\mathcal{P}_u(v))$ and hence $\gamma(i,j,u,v) = 0$. Combining, $\gamma(i,j,u,v) = \max(0, r - \lceil \frac{(\overline{D_u}-\overline{D_i})+[\mathcal{P}_u(v)-\mathcal{P}_i(j)]\cdot\overline{\Delta}}{Q\cdot\overline{\Delta}} \rceil)$.

Combining the above three cases, $\gamma(i,j,u,v) = \max(0, r - \xi(i,j,u,v))$.

Since $\mathcal{O}(r,\mathcal{P}_i(j))$ is equal to the total number of packets that are expected to arrive at the destination before the tagged packet,

$$\mathcal{O}(r,\mathcal{P}_i(j)) = \sum_{\substack{u=1 \\ u\neq i}}^{N} \sum_{v=1}^{q_u} \gamma(i,j,u,v) + (r-1)\cdot q_i + j. \tag{31}$$

Thus, the theorem is proved. □

Take $\delta(i,j) = \xi(\mathcal{R}(1),1,i,j)$. By using Theorem 1, it can be shown that:

$$\mathcal{O}(r+1,\mathcal{P}_i(j)) = \mathcal{O}(r,\mathcal{P}_i(j)) + Q, \tag{32}$$

where $r \geq \max_{\substack{u,v \\ u\neq\mathcal{R}(1)}} (1,\delta(u,v)) - \min_{\substack{x,y \\ x\neq\mathcal{R}(1)}} (1,\delta(x,y)) + 1$.

The determination for the set of $\delta(i,j)$ characterizes the steady state behavior of the sending schedule, and hence the packet resequencing delay and resequencing buffer occupancy distributions.

IV. ANALYTICAL INVESTIGATION

This section discusses the numerical results based on the analytical expressions obtained in Sections II and III. To evaluate the effectiveness of flow assignment and packet scheduling, we need to calculate the resequencing delay and the resequencing buffer occupancy. Those expressions can be found from our proposed resequencing model in [24]. With the help of some numerical examples, we can illustrate the effectiveness of these control mechanisms for multipath routing by answering the three basic questions posed in Section I.

The expressions we have obtained are applicable to general path delay distributions. To illustrate our multipath routing scheme, and without loss of generality, the following tandem queue path delay model is used.

A path is modelled as an L-node $M/M/1$ tandem network with a fixed delay line, as exhibited in Fig. 1. L $M/M/1$ queues and a delay line are connected in tandem. The i^{th} queue receives input from two traffic sources: The tagged dispersed traffic of rate λ, and the interfering or background traffic of rate b_i. The service rate of the i^{th} server is μ_i. The delay line, which generally represents the total propagation delay of the path, is P. Denote by $\overline{D}$ and σ_D^2 the mean and the variance of the end-to-end path delay. It can be shown [25] that:

$$\overline{D} = \sum_{i=1}^{L} \frac{1}{\mu_i(1-\rho_i)} + P, \tag{33}$$

$$\sigma_D^2 = \sum_{i=1}^{L} \frac{1}{\mu_i^2(1-\rho_i)^2}, \tag{34}$$

where the utilization of the i^{th} server, $\rho_i = \frac{\lambda+b_i}{\mu_i}$. L is set to 5 for our subsequent studies.

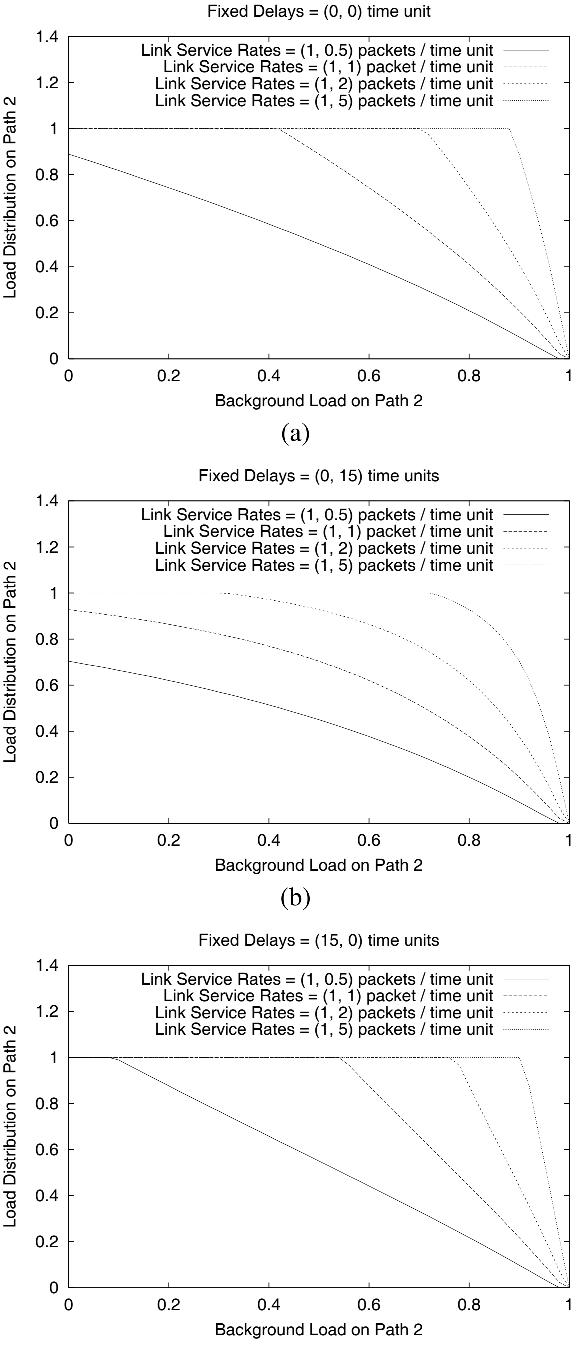
The central limit theorem [26] suggests that the end-to-end path delay, which is the sum of a large number of hop delays, is approximately normally distributed. The mean and the variance of the end-to-end path delay provide sufficient information to generate an approximate distribution, which can then be utilized to compute the resequencing delay distribution. This approach, which was used to solve the end-to-end percentile-type delay objective allocation problem for networks supporting Switched Multi-megabit Data Service (SMDS), has been shown to provide the best approximation to the reference values [27]. Thus, the end-to-end path delay is assumed to be Gaussian or normally distributed with mean $\overline{D}$ and variance σ_D^2.

Since $\frac{\partial \overline{D}}{\partial \lambda} > 0$ and $\frac{\partial^2 \overline{D}}{\partial \lambda^2} > 0$ when λ is evaluated at a value between 0 and $\mu_i - b_i$, $\overline{D}$ is a non-negative monotonically increasing convex function, which can be employed as the cost function to find the unique optimal flow configuration illustrated in Section II.

The inter-sending time between any two consecutive packets from the tagged source, denoted as the inter-packet time, is five time units. There are two disjoint paths, namely, Path 1 and Path 2, connecting the source to the destination. Packets are routed in a cyclical fashion so that the arrival instances of any two packets to each path is as uniform as possible. Each server on Path i, $i = 1, 2$, serves a packet with an average service time of ϕ_i time units, where $\phi_1 = 1$, and $\phi_2 = 0.5, 1, 2, 5$. This captures the cases where capacities of participating paths are different from each other. The set of fixed delays for Path 1 and Path 2 consists of three different combinations, namely (0, 0), (0, 15), and (15, 0) time units, where the first and second entries of each tuple are the fixed delay on Path 1 and Path 2, respectively. This captures the cases where the average end-to-end delays of participating paths differ significantly. The background load on Path 1 is fixed to be 0.75 and that on Path 2 varies between 0 and 1, at an increment of 0.02. This set of experiments can then fully demonstrate why we need to find an optimal traffic split instead of cyclic traffic dispersion for multipath routing, and to apply our proposed packet scheduling technique to reduce the necessity for resequencing.

The results are provided in two sets. The first set studies the effectiveness of the optimal path-splitting of traffic for multipath routing. It provides the load distributions on Path 2, p_2 (where $p_1 = 1 - p_2$), and the average end-to-end path delays. The second set studies the effectiveness of the flow assignment and packet scheduling techniques on resequencing. It includes the average resequencing delays for various settings.

We examine the first set of results. Fig. 2 shows the load distributions on Path 2 when the background load on Path 2 varies between 0 and 1. The link service rate tuple has two entries, corresponding to the link service rate on Paths 1 and 2, respectively. The load distribution drops from 1 to 0 as the background load increases from 0 to 1. Initially, the background load on Path 1 is much higher than that on Path 2. It is beneficial to transmit a majority of packets on Path 2 instead of cyclic dispersion in order to take advantage of a much lower average end-to-end delay on Path 2. Its superiority fades as the background load on Path 2 increases. At a certain background load on Path 2, cyclic



(a)

(b)

(c)

Fig. 2. Load distribution plots: (a) When fixed delays are (0, 0) time unit, (b) when fixed delays (0, 15) time unit, and (c) when fixed delays when fixed delays (15, 0) time unit.

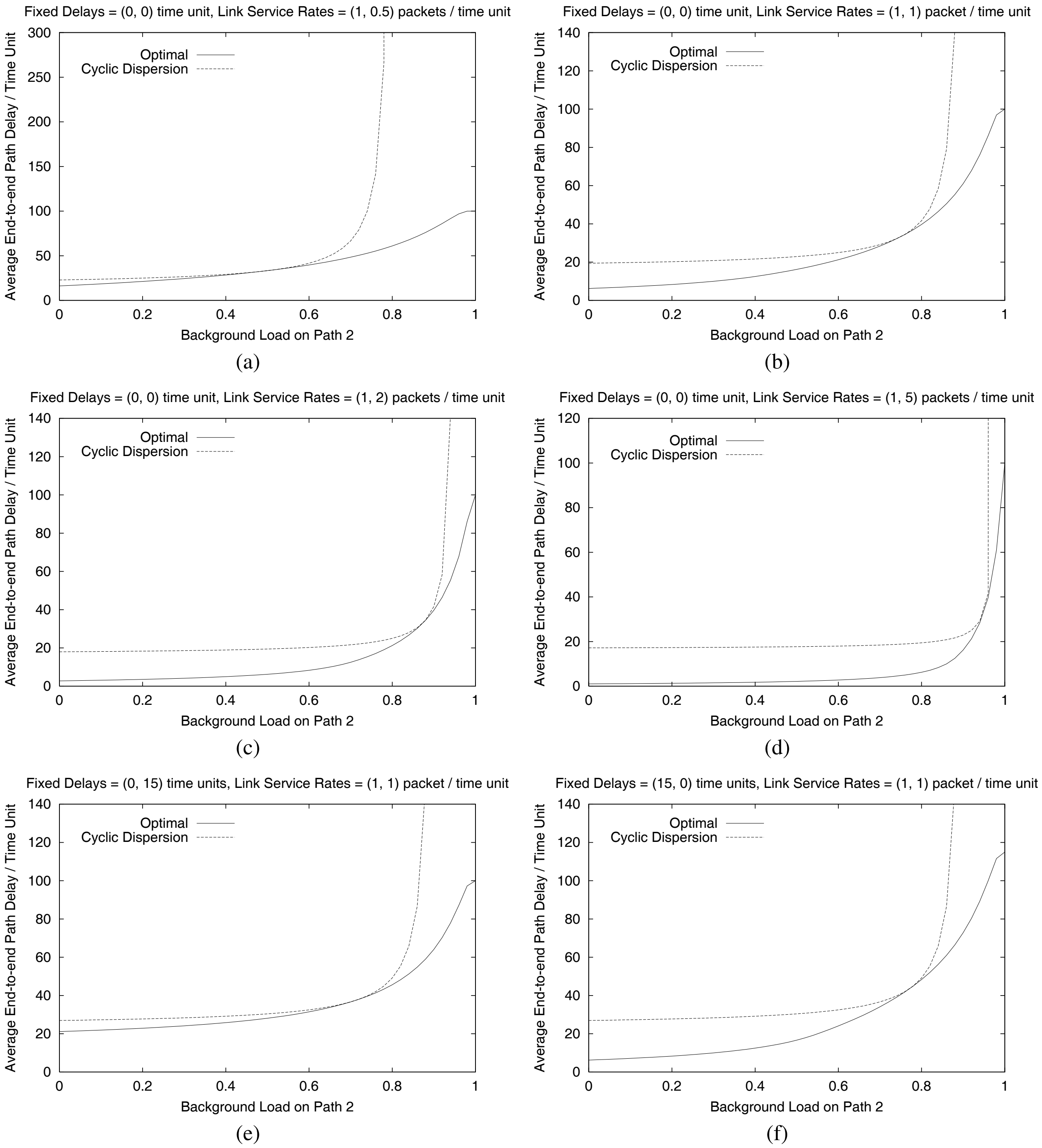
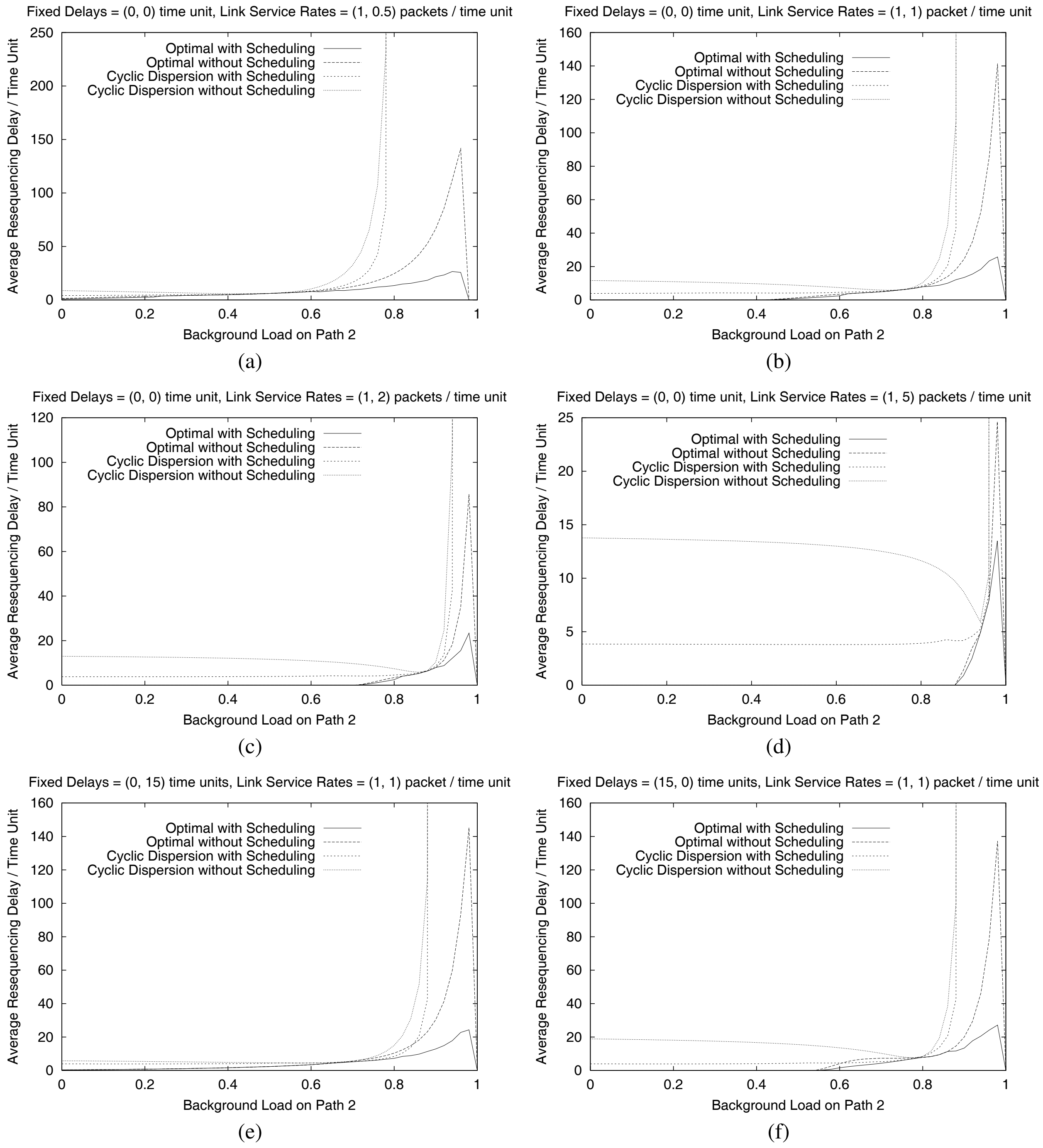


Fig. 3. Average end-to-end path delay plots: (a) When fixed delays are (0, 0) time unit and link service rates are (1, 0.5) packets per time unit, (b) when fixed delays are (0, 0) time unit and link service rates are (1, 1) packets per time unit, (c) when fixed delays are (0, 0) time unit and link service rates are (1, 2) packets per time unit, (d) when fixed delays are (0, 0) time unit and link service rates are (1, 5) packets per time unit, (e) when fixed delays are (0, 15) time unit and link service rates are (1, 1) packets per time unit, and (f) when fixed delays are (15, 0) time unit and link service rates are (1, 1) packets per time unit.

dispersion becomes the best strategy for traffic splitting. As the background load on Path 2 continues to rise, it is advantageous to distribute a higher load on Path 1 rather than Path 2. Eventually, all packets should be sent on Path 1.

Besides, it is always possible to achieve optimal splitting with a higher load distribution on Path 2 with faster transmission links. A faster transmission link means that it can take a higher portion of traffic from the source to get the same offered load, given the total work feeding into the network from the source is fixed. Moreover, a higher propagation delay on Path 2 causes a greater end-to-end path delay, resulting in a reduction in the optimal load distribution on Path 2.

Fig. 3 exhibits the average end-to-end path delays for both the optimal split of traffic and cyclic dispersion when the back-



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 4. Average resequencing delay plots: (a) When fixed delays are (0, 0) time unit and link service rates are (1, 0.5) packets per time unit, (b) when fixed delays are (0, 0) time unit and link service rates are (1, 1) packets per time unit, (c) when fixed delays are (0, 0) time unit and link service rates are (1, 2) packets per time unit, (d) when fixed delays are (0, 0) time unit and link service rates are (1, 5) packets per time unit, (e) when fixed delays are (0, 15) time unit and link service rates are (1, 1) packets per time unit, and (f) when fixed delays are (15, 0) time unit and link service rates are (1, 1) packets per time unit.

ground load on Path 2 varies between 0 and 1. At low and moderate background load on Path 2, we see that the optimal split of traffic yields a greater reduction on the average delay when the link service rate for Path 2 becomes higher, since an increase in the link service rate reduces the overall offered load and thus the average end-to-end delay on Path 2. In addition, the improvement decreases when the propagation delay on Path 2 is higher, and vice versa.

The second set of results demonstrates the effectiveness of the flow assignment and packet scheduling techniques on resequencing. Our results in Fig. 4 show that the optimal flow assignment always yields a lower average resequencing delay and hence a lower average resequencing buffer occupancy[1] than cyclic dispersion of traffic. A further improvement can be ob-

[1] The average resequencing buffer occupancy is proportional to the average resequencing delay by applying Little's Theorem [25].

tained by applying the proposed packet scheduling technique. When the background load on Path 2 rises from 0 to 1, the end-to-end delay on Path 2 increases. This reduces the difference in path delays and hence the average packet resequencing delay when cyclic dispersion is used. At a certain background load on Path 2, the average difference in path delays and also the average resequencing delay attain their minimum values. As the background load on Path 2 continues to rise, the difference in path delays and hence the average resequencing delay increase till Path 2 becomes overloaded. By employing the packet scheduling technique together with cyclic dispersion, the average resequencing delay is kept below the inter-packet time until the offered load on Path 2 becomes very heavy. We see that a substantial performance improvement can be achieved for cases where there is a large difference in path delays only when the packet scheduling technique is applied. This means that the packet scheduling mechanism is very effective in providing performance improvement when there is a large difference in path fixed delays.

V. CONCLUSIONS

In this paper, we have proposed a framework to study how to route packets efficiently in multipath communication networks. Two traffic congestion control techniques, namely, flow assignment and packet scheduling, have been investigated. The flow assignment mechanism defines an optimal splitting of data traffic on multiple disjoint paths, so as to minimize the average cost for multipath routing, such as the average end-to-end path delay experienced by a packet. Yet, packets may still experience a substantial difference in average path delays if they are sent on different paths. The packet scheduling mechanism can be utilized to minimize or to reduce the consumption of the resequencing buffer and delay on resequencing by reducing the chance for out of order packet arrivals at the destination, say, by scheduling them according to their expected arrival times at the destination. This technique is particularly useful for transmitting archival information, such as stored videos, since all archival packets are available and they can be sent in any desired order.

To illustrate our model, and without loss of generality, Gaussian distributed end-to-end path delays are used. Our analytical results show that the techniques are very effective in reducing the average end-to-end path delay, the average packet resequencing delay, and the average resequencing buffer occupancy for various path configurations.

Now we re-visit the three questions posed in Section 1. As expected, the optimal flow assignment aims to minimize the specified objective cost, and shifts the offered load from one path to other paths when the background load of the path continues to rise. It is always possible to achieve optimal traffic splitting by sending more traffic on a path with faster transmission links.

Compared with cyclic traffic dispersion, the optimal split of traffic yields a greater reduction on the average delay when the background loads of participating paths differ substantially. However, a large difference in the end-to-end propagation delays generally causes a great difference in the average end-to-end path delays, resulting in a reduction in the performance improvement. This means that the optimal split of traffic can yield a significantly better network performance whenever the background loads of participating paths differ substantially or there exist large differences in end-to-end propagation delays. Our results also show that the optimal flow assignment always yields a lower average resequencing delay and hence a lower average resequencing buffer occupancy than cyclic dispersion of traffic.

A further performance improvement can be achieved by applying the proposed packet scheduling technique, which helps to effectively compensate for the difference in path delays and hence reduce the necessity for resequencing even when cyclic dispersion is used. By applying the packet scheduling technique together with cyclic dispersion, the average resequencing delay is kept below the inter-packet time until some participating paths become heavily loaded. Since the optimal split of traffic may not reduce the differences in end-to-end path delays, the proposed packet scheduling technique can complement the flow assignment technique by reducing the necessity for resequencing, thereby improving the effectiveness in using the resequencing buffer. When transmitting archival and some delayed real time information, it is always possible to reduce the necessity for resequencing by applying the packet scheduling technique. These promising results can form a basis for designing and deploying future adaptive multipath protocols.

There are several possible extensions to our work, some of which are listed below:

- Devise an adaptive multipath protocols for packet-switching networks, such as IP-based and ATM networks;
- Incorporate quality of service routing [17] with multipath routing;
- Extend the framework to consider reliability and fault tolerant issues.

ACKNOWLEDGEMENTS

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions which assisted us in improving the quality of the paper.

REFERENCES

[1] S. N. Chiou and V. O. K. Li, "Diversity transmissions in a communication network with unreliable components," in *Proc. IEEE ICC '87*, vol. 2, Seattle, WA, USA, 7-10 June 1987, pp. 968–973.

[2] J. H. Déjean, L. Dittmann, and C. N. Lorenzen, "String mode – A new concept for performance improvement of ATM networks," *IEEE J. Select. Areas Commun.*, vol. 9, no. 9, pp. 1452–1460, Dec. 1991.

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

[4] K.-C. Leung and V. O. K. Li, "Generalized load sharing for packet-switching networks," in *Proc. ICNP 2000*, Osaka, Japan, 14-17 Nov. 2000, pp. 305–314.

[5] N. F. Maxemchuk, "Dispersity routing," in *Proc. IEEE ICC '75*, San Francisco, CA, USA, June 1975, pp. 41-10 – 41-13.

[6] N. F. Maxemchuk, "Dispersity routing in high-speed networks," *Computer Networks and ISDN Systems*, vol. 25, no. 6, pp. 645–661, Jan. 1993.

[7] W. Stallings, *High-Speed Networks: TCP/IP and ATM Design Principles*, Prentice-Hall International, Inc., 1998.

[8] V. O. K. Li and W. Liao, "Distributed multimedia systems," *Proc. IEEE*, vol. 85, no. 7, July 1997, pp. 1063–1108.

[9] J. Beran *et. al.*, "Long-range dependence in variable-bit-rate video traffic," *IEEE Trans. Commun.*, vol. 43, no. 2–4, pp. 1566–1579, Feb.–Apr. 1995.

[10] W. E. Leland, *et. al.*, "On the self-similar nature of ethernet traffic (extended version)," *IEEE/ACM Trans. Networking*, vol. 2, no. 1, pp. 1–15, Feb. 1994.

[11] A. Erramilli, O. Narayan, and W. Willinger, "Experimental queueing analysis with long-range dependent packet traffic," *IEEE/ACM Trans. Networking*, vol. 4, no. 2, pp. 209–223, Apr. 1996.

[12] D. Bertsekas and R. Gallager, *Data Networks*, Second Edition. Prentice Hall, 1992.

[13] K.-C. Leung and V. O. K. Li, "Flow assignment and packet scheduling for multipath networks," in *Proc. IEEE GLOBECOM '99*, vol. 1, Rio de Janeiro, RJ, Brazil, 5-9 Dec. 1999, pp. 246–250.

[14] S. Vutukury and J. J. Garcia-Luna-Aceves, "A simple approximation to minimum-delay routing," *Computer Communication Review*, vol. 29, no. 4, pp. 227–238, Oct. 1999.

[15] D. A. Khotimsky, "A packet resequencing protocol for fault-tolerant multipath transmission with non-uniform traffic splitting," in *Proc. IEEE GLOBECOM '99*, vol. 2, Rio de Janeiro, RJ, Brazil, 5-9 Dec. 1999, pp. 1283–1289.

[16] H. Adiseshu, G. Varghese, and G. Parulkar., "An architecture for packet-striping protocols," *ACM Trans. Computer Systems*, vol. 17, no. 4, pp. 249–287, Nov. 1999.

[17] S. Chen and K. Nahrstedt, "An overview of quality of service routing for next-generation high-speed networks: Problems and solutions," *IEEE Network*, vol. 12, no. 6, pp. 64–79, Nov.-Dec. 1998.

[18] T. T. Lee, S. C. Liew, and Q.-L. Ding, "Parallel communications for ATM network control and management," *Performance Evaluation*, vol. 30, no. 4, pp. 243–264, Oct. 1997.

[19] S. Bahk and M. E. Zarki, "Preventive congestion control based routing in ATM networks," in *Proc. IEEE ICC '94*, vol. 3, New Orleans, LA, USA, 1-5 May 1994, pp. 1592–1599.

[20] R. Krishnan and J. A. Silvester, "An approach to path-splitting in multipath networks," in *Proc. IEEE ICC '93*, vol. 3, Geneva, Switzerland, 23-26 May 1993, pp. 1353–1357.

[21] D. G. Luenberger, *Linear and Nonlinear Programming*, Second Edition, Addison-Wesley, 1984.

[22] C. G. Cassandras, M. V. Abidi, and D. Towsley, "Distributed routing with on-line marginal delay estimation," *IEEE Trans. Commun.*, vol. 38, no. 3, pp. 348–359, Mar. 1990.

[23] Y. Arian and Y. Levy, "Algorithms for generalized round robin routing," *Operations Research Letters*, vol. 12, no. 5, pp. 313–319, Nov. 1992.

[24] K.-C. Leung and V. O. K. Li, "A resequencing model for high speed networks," in *Proc. IEEE ICC '99*, vol. 2, Vancouver, BC, Canada, 6-10 June 1999, pp. 1239–1243,.

[25] L. Kleinrock, *Queueing Systems (Volume I: Theory)*, John Wiley & Sons, 1975.

[26] W. W. Hines and D. C. Montgomery, *Probability and Statistics in Engineering and Management Science*, Third Edition, John Wiley & Sons, 1990.

[27] F. Y. S. Lin, "Allocation of end-to-end delay objectives for networks supporting SMDS," in *Proc. IEEE GLOBECOM '93*, vol. 3, Houston, TX, USA, Nov. 29–Dec. 2, 1993, pp. 1346–1350.

Ka-Cheong Leung was born in Hong Kong in 1972. He received the B.Eng. degree in Computer Science from the Hong Kong University of Science and Technology, Hong Kong, in 1994, the M.Sc. degree in Electrical Engineering (Computer Networks) and the Ph.D. degree in Computer Engineering from the University of Southern California, Los Angeles, California, USA, in 1997 and 2000, respectively. He worked as a senior research engineer at Nokia Research Center, Nokia Inc., Irving, Texas, USA from 2001 to 2002. Since August 2002 he has been with the Texas Tech University, Lubbock, Texas, USA, where he is an Assistant Professor at the Department of Computer Science. His research interests include routing, congestion control, and quality of service guarantees in high-speed communication networks, high-performance computing, and parallel applications.

Victor O. K. Li was born in Hong Kong in 1954. He received SB, SM, EE, and ScD degrees in Electrical Engineering and Computer Science from the Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, in 1977, 1979, 1980, and 1981, respectively. He joined the University of Southern California (USC), Los Angeles, California, USA in February 1981, and became Professor of Electrical Engineering and Director of the USC Communication Sciences Institute. Since September 1997 he has been with the University of Hong Kong, Hong Kong, where he is Chair Professor of Information Engineering at the Department of Electrical and Electronic Engineering, and Managing Director of Versitech Ltd., the technology transfer and commercial arm of the University. He also serves on various corporate boards. His research is in information technology, including high-speed communication networks, wireless networks, and Internet technologies and applications. He is a Principal Investigator of the Area of Excellence in Information Technology funded by the Hong Kong Government. Sought by government, industry, and academic organizations, he has lectured and consulted extensively around the world. Prof. Li chaired the Computer Communications Technical Committee of the IEEE Communications Society 1987-1989, and the Los Angeles Chapter of the IEEE Information Theory Group 1983-1985. He co-founded the International Conference on Computer Communications and Networks (IC3N), and chaired its Steering Committee 1992-1997. He also chaired various international workshops and conferences, and is most recently appointed General Chair of IEEE INFOCOM 2004. Prof. Li has served as an editor of IEEE Network, IEEE JSAC Wireless Communications Series, and Telecommunication Systems. He also guest edited special issues of IEEE JSAC, Computer Networks and ISDN Systems, and KICS/IEEE Journal of Communications and Networks. He is now serving as an editor of ACM/Kluwer Wireless Networks and IEEE Communications Surveys and Tutorials. Prof. Li was recently appointed to the Hong Kong Information Infrastructure Advisory Committee by the Chief Executive of the Hong Kong Special Administrative Region. He also serves on the Innovation and Technology Fund (Electronics) Vetting Committee, the Small Entrepreneur Research Assistance Programme Committee, the Engineering Panel of the Research Grants Council, and the Task Force for the Hong Kong Academic and Research Network (HARNET) Development Fund of the University Grants Committee. He was a Distinguished Lecturer at the University of California at San Diego, at the National Science Council of Taiwan, and at the California Polytechnic Institute. Prof. Li has also delivered keynote speeches at many international conferences. He has received numerous awards, including, most recently, the Croucher Foundation Senior Research Fellowship, in March 2002, and the Bronze Bauhinia Star, Government of the Hong Kong Special Administrative Region, China, July 1 2002. He was elected an IEEE Fellow in 1992.