An Approach to Path-Splitting in Multipath Networks

Ram Krishnan and John A. Silvester

CENG Technical Report 92-10

Department of Electrical Engineering - Systems
University of Southern California
Los Angeles, California 90089-2562
(213)740-4579

An Approach to Path-Splitting in Multipath Networks

Ram Krishnan and **John A. Silvester**
Department of Electrical Engineering - Systems
University of Southern California
Los Angeles, CA 90089-2562
Phone : (213)-740-4579
Fax : (213)-740-4449
e-mail : ram@girtab.usc.edu, silveste@usc.edu

COMSOC Technical Committee : Computer Communications

Abstract

Path splitting in networks is an area that has received little attention. In this paper, we propose that a single connection at the transport layer be implemented as multiple source routes in the network layer resulting in a balanced loading of network resources. The paths are not, necessarily, of the same length. The problem of traffic bifurcation at the source, that achieves path splitting, is solved by computing the flows on all the links in the network to minimize a given objective function, such as average delay or packet loss probability. From a knowledge of the flows, the proportion of the input traffic that has to be routed over the different source routes is determined. A variety of schemes, such as probabilistic and deterministic switching exist that achieve the desired bifurcation. In this paper, we propose the use of Join-Biased Queue (JBQ) rule to effect the required traffic splitting. The superiority of the JBQ rule over the other schemes is demonstrated. It is also shown that the values obtained for the flows depend on the objective function being optimized. Finally, a bound on the size of the resequencing buffer necessary at the destination is computed for packets that are received out of order.

1 Introduction

It is an accepted fact that network resources are more effectively utilized if multiple connections between a specific source-destination pair use different paths for their transmission instead of carrying all of them on a single path. Similarly, it may be advantageous to implement a single connection at the transport layer as multiple source routes in the network layer. As an example of another application, multiple transport level connections can be mapped into a single network level connection which provides service using multiple routes selected from the available paths between the corresponding source-destination pair. At the destination, the network layer returns the packets to the appropriate connection. Although this implies additional processing at the end-to-end hosts, the use of multiple paths (not necessarily of the same length) gives additional flexibility in balancing link utilizations. Multiple paths between each source and destination also make the network more reliable. Multiple paths also make it possible to avoid heavily used segments of the network to equalize the load.

In networks with regular topologies like ShuffleNet and Manhattan Street Net, there exist multiple minimum hop paths between pairs of nodes. If fixed (i.e., single path) routing is employed, only one of these paths would be used, with the throughput saturating for relatively low values of the offered input traffic. This effect is more marked for nonuniform traffic patterns or when there are network failures. Originally, multipath routing (also called dispersity routing in [5]) was proposed for store-and forward networks with multiple equal distance paths between node pairs. In this paper, we limit our discussion to multipath source routing as applied to connection-oriented networks and describe a centralized scheme that facilitates usage of alternate paths, all of which are not equivalent in terms of number of hops[1]. The flows on all the links in the network are determined by optimizing a specific performance measure such as average delay, blocking probability, etc. A local adaptive rule, called the Join-Biased Queue (JBQ) rule is used to achieve the computed flows. The proportion to which such longer hop paths are to be used in conjunction with the minimum hop paths is an issue that is addressed by this scheme. In high speed networks, where processing is to be kept to a minimum, it is desirable to set up multiple routes between source-destination pairs as an alternative

[1]Two paths are also said to be non-equivalent if they differ in their available capacities.

to adaptive datagram transmission. As the entire route is already set up between the source and the destination, the intermediate switches and gateways have less processing to perform. It also eliminates exchange of information between the switches and the associated computation to determine the best routes [4].

The organization of the paper is as follows: In the next section, we describe the need to employ path-splitting using multiple routes between source-destination pairs. Our main interest in this paper is to study the improvement obtained by using multiple source routes at the network layer for a single transport level connection. In Section 3, we show how to determine the amount of traffic bifurcation at the source. As will be demonstrated, the amount of traffic bifurcation depends on the performance measure being minimized. Section 3 also describes the various techniques used to achieve the desired traffic bifurcation between the multiple routes. An example has been worked out in detail in section 4 that highlights the advantages offered by path-splitting techniques. A brief discussion follows in the next section where a bound on the size of the resequencing buffer necessary for receiving out of order packets is computed. Section 6 concludes the paper.

2 Why path splitting ?

In this paper we address networks whose topologies are such that there exist multiple paths between node pairs. In such networks, the throughput is determined by the edge with the maximum loading. Thus if there are multiple connections at the transport layer, it makes sense to establish the corresponding virtual circuits over each of the available paths so that the load is distributed in a more uniform fashion. Consequently, packets belonging to a particular connection are routed over the corresponding virtual circuit[2]. In situations where the number of transport-level connections is limited to, say, one or two between source-destination pairs, traffic belonging to a particular connection can be adaptively routed at the source over these multiple paths to achieve a similar effect. We address the second situation in this paper, viz., the problem of splitting the traffic belonging to a single connection over multiple source routes to achieve balancing of load, and hence higher throughputs. An adaptive algorithm based on datagram transmission

[2]The choice of allocation granularity is arguable. We address that issue in [1].

that routes packets over less-busy channels whenever there exist multiple minimum hop paths to the packet's destination has been suggested in [2]. It is a distributed algorithm that uses only local state information available at the users. The queue sizes directly convey some information about the load and the algorithm uses this information to distribute the non-uniform traffic and make the channel loads more balanced.

Quoting Jain [3],

> Path splitting among long paths of differing capacities is not well understood. In most networks today, all traffic from a given source to a given destination either passes through the same path or is split equally among different paths of equal capacities. Thus, if the optimal path is congested and a slower path is available, the slower path is not used. Designing a scheme that allows slower paths to be used depending upon the load levels on all paths is a topic for further research.

The centralized scheme proposed in this paper, attempts to achieve this objective.

Consequently, instead of setting up a single path for a connection, which in most instances is the shortest path between the source and the destination, we propose establishing multiple paths for the same connection, utilizing all the available routes for the corresponding pair of nodes. It is possible that one path[3] is longer than the other, i.e., they are not necessarily of the same length.

In the SNA architecture [4], the source selects a *virtual route* for a session, when the session is established. If multiple routes are available, the decision to choose an appropriate route is based on the current load assigned to the various routes, and the service class desired for the session. Multiple routes, in the SNA architecture, are established only for reliability purposes. In [5], Maxemchuk describes distributing the data between a source and destination over several paths through the network instead of concentrating it on a single path. However, the paper addresses only those networks having equal length paths between source-destination pairs. In a scenario where the network traffic exhibits topological locality, i.e., nearby nodes are more likely to communicate than distant nodes, this

[3]In this paper, we use the terms paths and routes synonymously.

constraint is not desirable. Source-destination pairs that are in close proximity might not possess multiple equal length paths and traffic has to be necessarily routed over longer paths if balanced loading is to be achieved. Further, he suggests splitting the input traffic equally between all paths, without taking into account the relative loading on these multiple paths.

The problem is to determine, given the traffic pattern and the logical topology of the network, the proportion of input traffic at the source that has to be sent through these multiple routes. If a measure such as propagation delay is minimized, it is intuitively clear that only a small proportion of the incoming traffic would be routed over the longer paths.

3 The Algorithm

3.1 Flow patterns

In this section, we address the problem of determining the proportion of the traffic at the source that has to be routed over the available multiple paths. One approach would be to determine the flows on all the links in the network that minimize an objective function, for a given traffic requirement and network topology. If the objective is to minimize the overall average delay, which is the sum of the queueing delays and the propagation delays, the flow deviation algorithm can be invoked to compute the set of flows.

The flow deviation method [6] is a special case of the Frank-Wolfe method for solving general, nonlinear programming problems with convex constraint sets. The flow pattern obtained in this manner is the optimum pattern for stochastic fixed rules. The best stochastic rule is globally optimum in the sense that it gives a minimum overall average time delay, averaged over all nodes, given that traffic is bifurcated (or routed to different outgoing links) by fixed probability assignments at each node. Though the above algorithm is not optimal when adaptive rules are employed, it nevertheless serves as a satisfactory heuristic for computing the flows in the network. In ATM networks, queueing delay is only a small component of the overall delay at low offered loads. As the offered load increases, the queueing delay component increases and at high enough loads, it becomes much larger than the propagation delay. In the high-speed network environment, since the propagation delay is usually much larger than the packet transmission time, this dominance occurs only at heavy loads. As a result, a more relevant measure such as packet loss probability should be minimized. The problem then shifts to computing the flows so as to minimize the maximum packet loss probability over all the nodes in the network. In [7], a minimax algorithm is described which determines a global flow assignment that achieves this objective.

From a knowledge of the link flows, each source can determine the proportion of the incoming traffic that has to be routed through each of its multiple routes. Once the packet is routed to an output channel at the source, the path to the destination remains fixed.

3.2 Techniques to achieve the flow pattern

We next address the issue of achieving the obtained flow pattern in the network by using a variety of routing schemes. For example, the incoming traffic can be deterministically switched between the different outgoing links in a round robin like fashion. The order in which the packets are to be deterministically routed can be computed from the knowledge of the flows. This method, though simple to implement, might exhibit degraded performance under heavy loads, where a technique that uses local queue size information for making routing decisions would perform better. However, deterministic switching is superior to stochastic or probabilistic switching as the former reduces the variance of the arrival process compared to the latter [8].

In this section, we study an adaptive way of bifurcating the traffic flow that uses local queue size information. The approach is similar to the join-the-shortest-queue (JSQ) rule [9], except that a bias term is introduced in comparing the queue lengths. By adjusting the bias term in the Join-Biased Queue (JBQ) rule, the proportions of traffic bifurcation can be regulated at will. If probabilistic switching is used, a Poisson arrival stream retains its distribution because random bifurcation of Poisson processes remains Poisson; and $M/M/1$ results can be used for queue length distribution. For the JBQ adaptive rule on the other hand, the message arrivals are state dependent because traffic

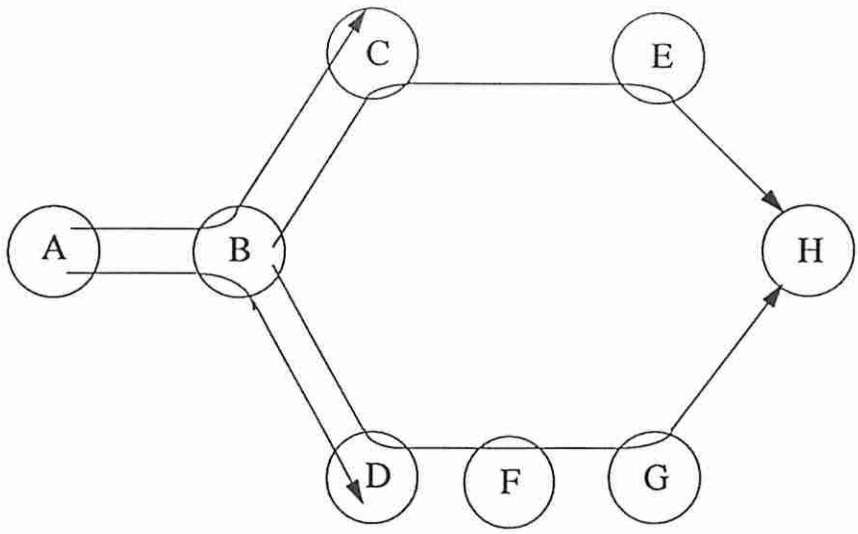

Figure 1: An eight-node network with sessions $A-C, A-D$ and $B-H$.

---

bifurcation is based on the instantaneous queue lengths. The use of the JBQ rule is illustrated by an example.

Consider the eight node network, with unidirectional flow, as shown in Figure ??. There are three sessions, viz., $A-C, A-D$ and $B-H$. The traffic for session $B-H$ is split between the shorter hop path $BCEH$ and the longer one $BDFGH$. On the other hand, traffic for sessions $A-C$ and $A-D$ have to be routed through links BC and BD respectively. As described earlier, the problem is (i) to determine the proportion of the traffic for session $B-H$ that has to be routed through links BC and BD, and (ii) to achieve the desired bifurcation. As a first step, the flows on all the links in the network that minimize the desired performance metric are computed. The fraction of the input traffic at B that has to be routed through each of its output links, and the traffic from sessions $A-C$ and $A-D$ that has to use links BC and BD respectively are therefore determined. As a result, we can identify three different types of traffic at node B, i.e, the arrival rates from sessions $A-C$ and $A-D$ and the traffic that originates at B and can be routed through either of its output links.

Let us consider node B in isolation, with the three different arrival rates and two outgoing links, as shown in Figure ??. For simplicity, Poisson arrival rates of messages

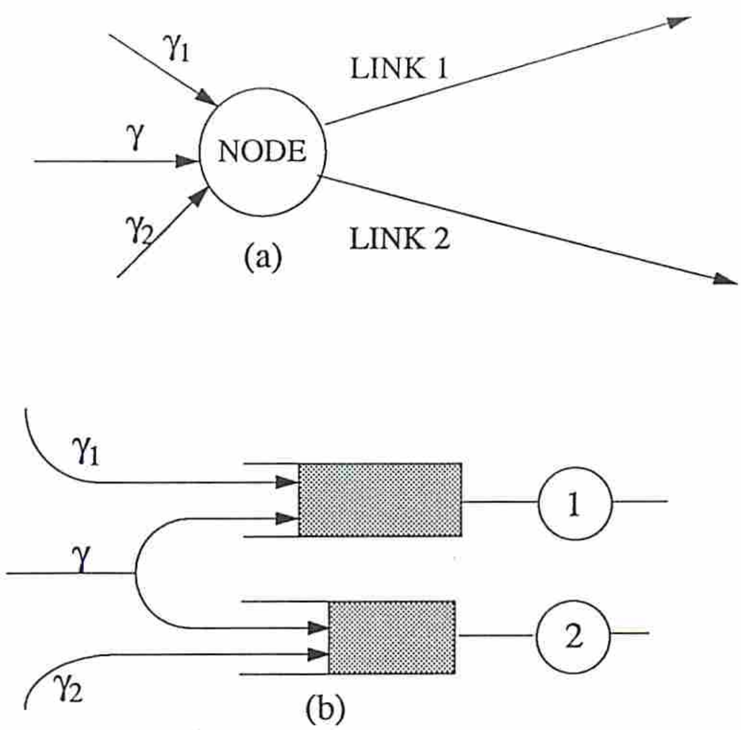

Figure 2: (a) An isolated network node and (b) its queueing model.

and exponential service rates are assumed. Let γ_1, γ_2 represent Poisson arrival rates of messages that are constrained to be sent out via link BC and link BD, respectively. These are called the fixed arrival rates. Let q_1, q_2 be the lengths of queue 1 (Q_1) and queue 2 (Q_2); and let λ_1, λ_2 be the actual input rates to Q_1 and Q_2. Let all queues be formed in finite buffers of size M and let γ represent the adaptive Poisson input arrival rate of messages to node B that can be switched to either Q_1 and Q_2.

In words, the JBQ rule says: route a message in the adaptive category to Q_1 if the length of Q_1 is less than the length of Q_2 plus Δ; route it to Q_2 if greater. If the length of Q_1 equals the length of Q_2 plus Δ, then route it to Q_1 and Q_2 with probability β and $(1-\beta)$ respectively. The special case $(\Delta, \beta) = (0,\ 0.5)$, $\gamma_1 = \gamma_2 = 0$, reduces to the Join-Shortest Queue (JSQ) rule. The presence of fixed-path messages (i.e., $\gamma_1, \gamma_2 \neq 0$) in our way of modeling makes the JBQ rule particularly suitable for extension to the network case. The following is the JBQ rule to route the γ messages:

$$\left.\begin{aligned} \lambda_1 &= \gamma_1 + \gamma \\ \lambda_2 &= \gamma_2 \end{aligned}\right\} \qquad \text{when } q_1 < q_2 + \Delta$$

$$\left.\begin{array}{l}\lambda_1 = \gamma_1 \\ \lambda_2 = \gamma_2 + \gamma\end{array}\right\} \quad \text{when } q_1 > q_2 + \triangle$$

$$\left.\begin{array}{l}\lambda_1 = \gamma_1 + \beta\gamma \\ \lambda_2 = \gamma_2 + (1-\beta)\gamma\end{array}\right\} \text{when } q_1 = q_2 + \triangle$$

Here, $\triangle$ in $\{0,\pm 1, \pm 2, \cdots\}$ is an integer representing the bias level and β in [0,1) is the *a priori* probability for routing the γ messages to Q_1 when $q_1 = q_2 + \triangle$. Thus, β is a fine tuning parameter.

For a particular set of input parameters $\{\gamma, \gamma_1, \gamma_2, \triangle, \beta, M\}$, the JBQ rule can be represented by a two-dimensional Markov chain with the transition rates dependent on the state (q_1, q_2). This is possible because Poisson arrival rates and exponential service rates are assumed. Let $P_{i,j}$ = Prob $[q_1 = i, q_2 = j]$. The complete set of states for $\triangle$= 2 is shown in Figure ??. Groups of states with the same inward-outward transition rates can be identified from the sketch, with the state equations for these states differing from each other only by the indices i and j. A typical set, for example, the one corresponding to region 3, may be written as follows (the average message length is normalized to unity):

$$P_{0,j} = \frac{\gamma_2 P_{0,j-1} + P_{1,j} + P_{0,j+1}}{1 + \gamma_1 + \gamma_2 + \gamma} \quad j = 1, 2, \cdots, M-1. \tag{1}$$

These equations are then numerically solved to find $P_{i,j}$. Using Little's formula, the average delay T is calculated to be

$$\begin{aligned} T &= \frac{E(q_1) + E(q_2)}{(\gamma_1 + \gamma_2 + \gamma)(1 - P_B)} \\ &= \frac{\sum_{i=1}^{M} i \sum_{j=0}^{M} P_{ij} + \sum_{j=1}^{M} j \sum_{i=0}^{M} P_{ij}}{\gamma_1 + \gamma_2 + \gamma} \end{aligned}$$

We can now use the JBQ rule to split the source traffic at B such that the desired queue utilizations are achieved. The $(\triangle, \beta)$ needed is obtained by quantizing β to a desired value. Then the Markov chain of the JBQ rule is solved repeatedly with different $(\triangle, \beta)$. Each set of solutions gives a set of link flows. The needed set of $(\triangle, \beta)$ is that which yields the desired set of flows. For local routing rules with exponentially distributed messages, the departure processes can be assumed to be Poisson [9]. This approximation allows us to decouple queues at different nodes and apply the JBQ rule separately. The

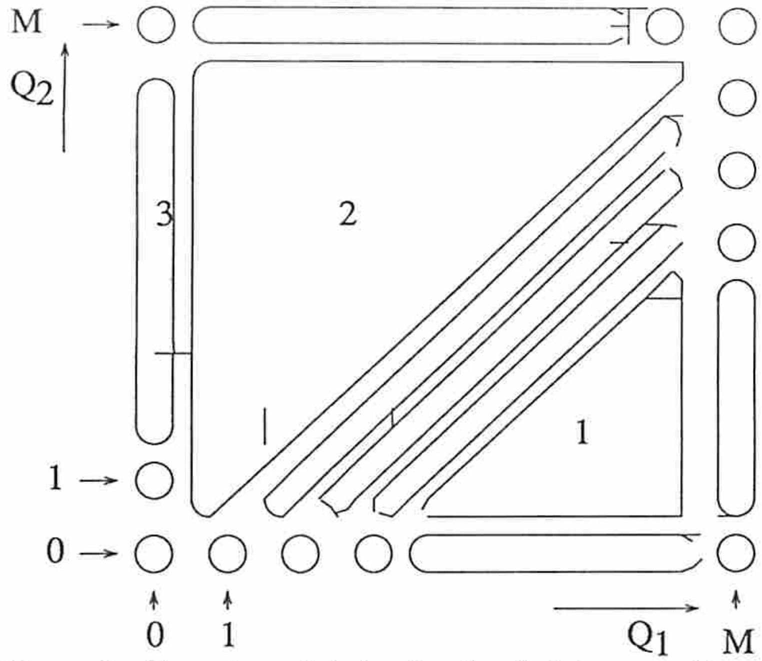

Figure 3: Grouping of states for the defining equation, JBQ rule, $\triangle = 2$.

centralized portion of the rule is fixed; it collects the traffic rate information about the entire network (global information) and determines the amount of flow on each link. The local (or distributed) portion of the rule is adaptive, making use of the locally available queue length information and determines the instantaneous flow of messages in their local environment [10]. Thus, the effect of the local routing is to cause the desired flows on the links.

The process of computing $(\triangle, \beta)$'s for all the nodes where the incoming traffic has to be bifurcated is computationally intensive. For a node with two outgoing links a simple search procedure can be used. A storage table can be used to retrieve the $(\triangle, \beta)$ pair for each triplet $(\gamma_1, \gamma_2, \gamma)$. Exhaustive search becomes complex for a node with three or more outgoing links. For the three link case, we have to trifurcate the source traffic into three streams of rates λ_1, λ_2 and λ_3. The problem is how to set the bias level on each queue so as to achieve the specified flow distribution. A three dimensional Markov chain has to be solved for the three queue system and the states have to be partitioned into three regions for the three routing decisions; the incoming message should join Q_1, Q_2 or Q_3. We have developed some simple approximation methods based on decoupled iterative techniques in [11].

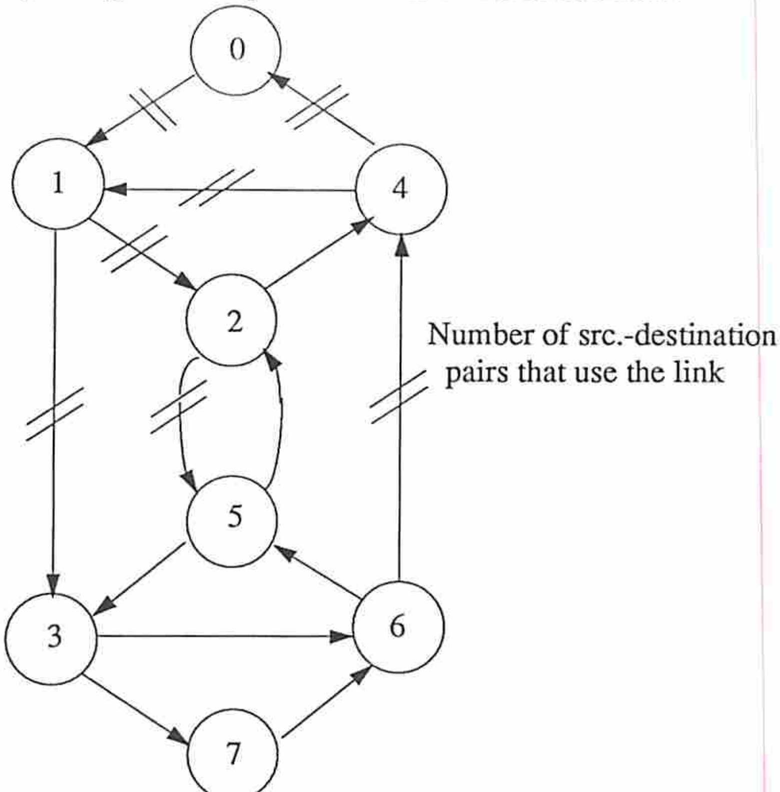

Figure 4: An eight node network with three src.-destination pairs.

4 Numerical Example

In this section, we present an example that illustrates the advantages of using the JBQ rule. We consider a network topology that possesses alternate paths between pairs of nodes, though not necessarily having the same number of hops. The use of shortest path routing in these topologies causes the loading on some edges to be significantly higher than others. The longer path routes are used in order to lower the maximum edge loading at the cost of increasing the average number of hops.

Most typical traffic patterns are non-uniform in nature. The communication between nodes is localized to a large extent with the result that if simple fixed routing rules are employed, only a fraction of the network resources are put to use. An eight node network is shown in Figure ??. A typical non-uniform traffic scenario between three source-destination pairs in the graph is shown in Table 1. Poisson input traffic rates are assumed. The available multiple paths and the corresponding number of hops are also identified.

Table 1: Traffic pattern

Commodity k	Source-Destination pair	Possible virtual routes	Number of hops	Rate
1	$4 \rightarrow 1$	$4 \rightarrow 1$ $4 \rightarrow 0 \rightarrow 1$	1 2	λ
2	$1 \rightarrow 3$	$1 \rightarrow 3$ $1 \rightarrow 2 \rightarrow 5 \rightarrow 3$	1 3	λ
3	$4 \rightarrow 3$	$4 \rightarrow 1 \rightarrow 3$ $4 \rightarrow 0 \rightarrow 1 \rightarrow 2 \rightarrow 5 \rightarrow 3$	2 5	λ

In order to determine the flows on all the links, we use a delay objective based on an infinite buffer assumption.

$$\overline{T} = \frac{1}{3\lambda} \sum_{i=1}^{7} [\frac{f_i}{1 - f_i} + f_i T_i] \tag{2}$$

T_i is the propagation delay (normalized with respect to the service time) and is assumed to be equal for all the links. To calculate T_i, let's consider the following parameter set, a standard for optical transmission.
transmission rate : 1.7 Gbps
packet size : 4 Kbytes
If the distance between neighboring nodes is, say, 20 km, then T_i is approximately 5.

For small values of λ, the propagation delay dominates the queueing delay with the result that there is no bifurcation of the input traffic of source-destination pairs $1 \rightarrow 3$ and $4 \rightarrow 3$. This is intuitively clear as the alternate routes for these node pairs, viz., $1 \rightarrow 2 \rightarrow 5 \rightarrow 3$ and $4 \rightarrow 0 \rightarrow 1 \rightarrow 2 \rightarrow 5 \rightarrow 3$ traverse two and three additional hops respectively. Since our intention is to minimize the overall delay, which amounts to minimizing the propagation delay for low load traffic, it follows that no traffic gets routed over these longer hop paths. As a result, the maximum edge loading is equal to that of a scheme using fixed single-path routing scheme. However, as the load increases, queueing delay becomes comparable to propagation delay with the result that more and more traffic gets routed over the longer distance paths. This causes a more balanced distribution of the load and the maximum edge loading is significantly lower than that of the fixed-path routing scheme. In this example, $1 \rightarrow 3$ is the most heavily loaded edge. Table 2 compares the link flows obtained with fixed single path routing with that obtained by using multiple routes for $\lambda = 0.45$. Clearly, multiple routes permit a more

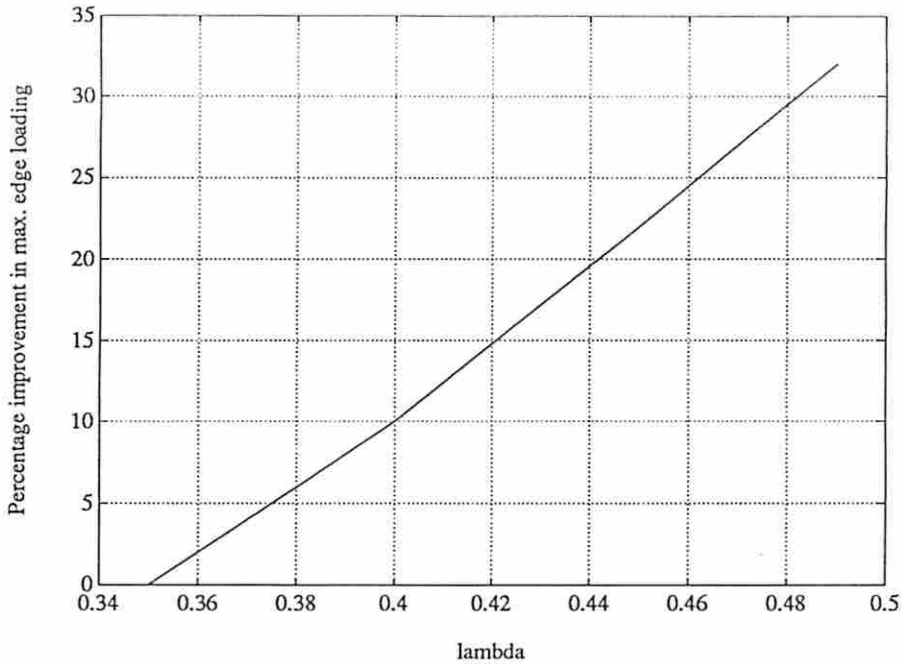

Figure 5: Percentage improvement in maximum edge loading, obtained by using multiple routes, as a function of λ.

---

balanced distribution of the traffic in the network demonstrated by a reduction in the average delay.

Table 2: Link flows on the network

Routing	$f_{4\to1}$	$f_{4\to0}$	$f_{0\to1}$	$f_{1\to3}$	$f_{1\to2}$	$f_{2\to5}$	$f_{5\to3}$
Single path routing	0.9	0	0	0.9	0	0	0
Multipath routing	0.66	0.24	0.24	0.73	0.17	0.17	0.17

Figure ?? shows the percentage improvement in the maximum edge loading, obtained by using multiple routes as opposed to a single route, as a function of the parameter λ. As illustrated in the figure, traffic bifurcation becomes more pronounced as the load λ increases, whereas for low values of λ, the flow pattern is similar to single path routing. This suggests that a single source route can be set up between source-destination pairs for low traffic rates. As the load builds up, more traffic bifurcation is possible and multiple source routes can be established. It should be pointed out that comments apply only for network topologies which do not possess equal length minimum hop paths between node pairs. In topologies like the ShuffleNet, where a majority of the node pairs have alternate minimum hop paths, multiple source routes can be established for all traffic intensities.

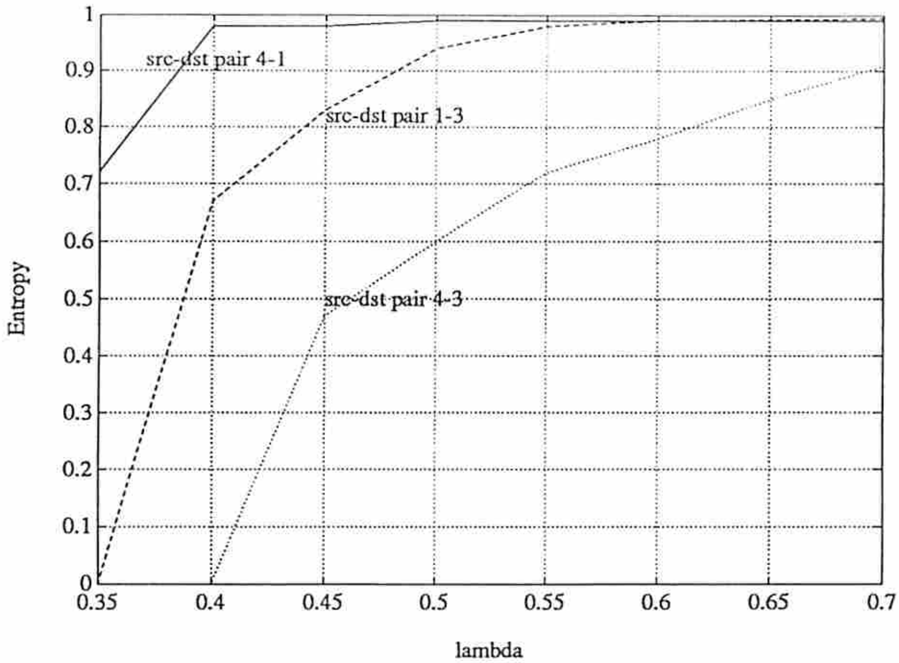

Figure 6: Entropy of the traffic bifurcation, as a function of λ, for all the three src.-destination pairs.

When we have to choose only between two paths it is easy to view the load splitting that is achieved by looking at, say, the fraction routed to path 1. For a larger number of choices, it is not so simple to view the degree of splitting that has been achieved. One approach is to look at the entropy measure [8]. Entropy is maximum, viz., unity, when the traffic is bifurcated equally and zero when there is no bifurcation. Entropy for the set $p_1, p_2, \ldots, p_n$ where $0 \leq p_i \leq 1$ and $\sum_{i=1}^{n} p_i = 1$, is defined as :

$$E(p_1, p_2, \ldots, p_n) = \sum_{i=1}^{n} -p_i \log p_i \tag{3}$$

If a_1 and $1 - a_1$ are percentages of bifurcation achieved in the two path case the entropy $E(a_1, 1 - a_1)$ is written as:

$$E(a_1, 1 - a_1) = -a_1 \log a_1 - (1 - a_1) \log(1 - a_1) \tag{4}$$

Figure ?? shows the entropy of the traffic bifurcation achieved, as a function of the load, for all the three source-destination pairs. As stated earlier, for low loads, there is no splitting of traffic for source-destination pair $4 \rightarrow 3$ as illustrated by the low values of entropy. However, as the load λ increases, the amount of traffic bifurcation also increases, with the entropy reaching values close to unity. The same effect is demonstrated to a lesser extent by the source-destination pair $1 \rightarrow 3$.

The first row of Table 3 shows the flows obtained by using the flow deviation algorithm and the corresponding mean queue sizes that probabilistic bifurcation achieves, for an input load of $\lambda = 0.45$. For stochastic splitting the mean queue sizes are computed using standard $M/M/1$ analysis [6]. The second row lists the values for the same parameters obtained using JBQ splitting. The mean queue sizes, in this case, are computed by explicitly solving a two-dimensional Markov chain as outlined in section 3. The last row lists the $(\triangle, \beta)$ values employed at the source nodes to achieve the flows. As expected, using a rule based on local queue size information to split traffic at the source yields smaller mean queue sizes, and hence smaller delays than probabilistic bifurcation.

Table 3: Stochastic vs JBQ splitting

Rules	$f_{4\to1}$	$f_{4\to0}$	$f_{1\to3}$	$f_{1\to2}$	$E(q_{4\to1})$	$E(q_{4\to0})$	$E(q_{1\to3})$	$E(q_{1\to2})$
Using Stochastic splitting	0.67	0.31	0.74	0.24	2.03	0.45	2.85	0.32
Using JBQ splitting	0.68	0.30	0.74	0.24	1.41	0.45	1.81	0.32
$(\triangle, \beta)$ that achieves this bifurcation	(2,0.83)		(2,0.5)					

Let us now optimize a different objective function other than average delay, viz., largest packet loss probability for equal buffer sizes. This problem reduces to minimizing the maximum flow on any link. Applying the minimax algorithm to the same example, we clearly observe that one optimal solution is to bifurcate the incoming traffic equally among the two available paths for each source-destination pair. This immediately implies that each link carries traffic corresponding to two source-destination pairs. For $\lambda = 0.45$, this means that each link carries a flow equal to 0.45. We see that the flows obtained using the minimax algorithm are quite different from those obtained using a delay objective (Table 2). As traffic is to be equally split between both paths, the parameter set $\triangle = 0, \beta = 0.5$, used at each node achieves the desired traffic bifurcation. The blocking probability, in the single node case, is given by $P[\text{Queue size} = M]$ where M is the buffer size. This is not valid, however, in a network of queues. A measure such as maximum $P[\text{Queue size} = M]$ over all nodes in the network is necessary to characterize the blocking probability, given that all buffers are of size M. Figure 7 plots this measure for the two cases when probabilistic and JBQ splitting is used, against different values of λ. As clearly illustrated, bifurcation using the JBQ rule experiences smaller loss probabilities than the stochastic case. This is to be expected as the JBQ rule uses local queue size information to route the packets.

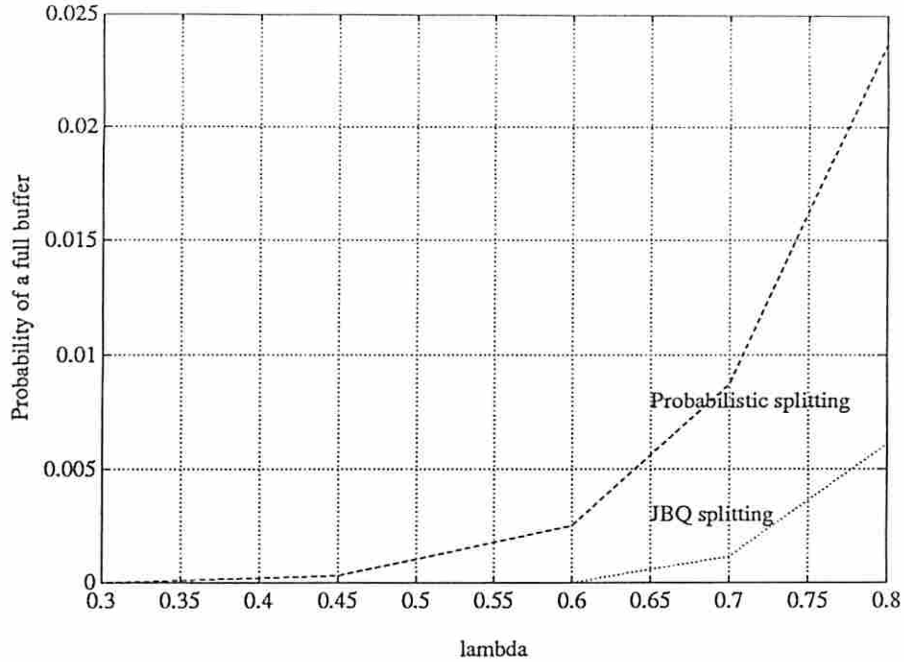

Figure 7: A comparison of blocking probabilities for probabilistic and JBQ splitting.

5 Discussion

Since there are multiple paths for packets flowing between a given source-destination pair, some of them longer than others, it is possible that packets may get out of sequence and may require resequencing buffers at the destination hosts. In this section, we compute a bound on the required number of resequencing buffers. Packets that arrive out of sequence at their destination are stored in a resequencing buffer until either the missing packets are received or until the time-out interval expires, at which point the packets are passed to the user. The required buffer size can be computed using the statistics of the source-destination delay distribution. A packet may reach its destination out of sequence if it traverses a longer hop path while other packets of the same source-destination stream happen to take shorter paths. If the path has k hops more than the shortest hop path, then the out of sequence packet takes at most k extra hops while the resequencing buffer receives no more than $k.t_p.\lambda$ of packets, where t_p is the maximum propagation delay between neighboring nodes and λ is the arrival rate (in packets per second). With a speed of light in fiber of 2.10^8 m/s, $t_p = 100\mu$sec over 20 km of fiber. We should point out that the queueing delays have been ignored in the above computation.

We now consider the queueing delays encountered at the intermediate nodes. As previously mentioned, the JBQ scheme helps maintain a small variance in the queue size, and therefore a small variance in the queueing delay. Suppose the out of sequence packet takes the longest path available, say, h, to reach its destination, and encounters the worst possible queueing delay of M packets at each node where M is the buffer size. During that time, the resequencing buffer receives no more than $h.M.t_m.\lambda$ packets, where t_m is the transmission time (in seconds) of a packet. For the parameter set given earlier, $t_m = 20\mu$secs. Combining both the propagation delay and the queueing delay, a bound on the resequencing buffer size is given by $(h.M.t_m + k.t_p).\lambda$ packets. The use of the JBQ rule reduces the output buffer size required for a specified overflow probability. Applying the above expression to the example worked out in the last section, the resequencing buffer size necessary for the source-destination pair $4 \rightarrow 3$ is only 13 packets for the following set of parameters: $h = 5$, $M = 10$, $t_p = 100\mu$secs, $t_m = 20\mu$secs and λ= 10,000 packets/sec (i.e., 10 Mb/s if there are 1000 bits per packet). If λ= 64 packets/sec (i.e., 64Kb/s voice with 1000 bits per packet), then no resequencing buffer is required because the maximum source-destination delay is smaller than the time interval between packet transmission from the source.

6 Conclusions

This paper addresses the efficient usage of network resources by utilizing all available paths between a source-destination pair for a specific connection at the transport layer. Previous work on path splitting have focused on the usage of multiple minimum hop paths only. Many networks do not possess alternate minimum-hop paths. This implies that a new approach to path splitting is necessary. In this paper, we have extended our work to include longer hop paths as well.

The primary drawback of the scheme suggested in this paper is that the load on the network be known *a priori*. It is also assumed that the traffic rates remain steady and do not change rapidly with time. The algorithm that is used to compute the flows and the associated $\triangle, \beta$'s has to be invoked every time a new connection has to be set up or when the traffic pattern changes.

References

[1] R. Krishnan and J.A. Silvester. "Choice of Allocation Granularity in Multipath Source Routing Schemes". Submitted to *INFOCOM '93*, July 1992.

[2] M.J. Karol and S.Z. Shaikh. "A Simple Adaptive Routing Scheme for ShuffleNet Multihop Lightwave Networks". In *Proc. of GLOBECOM' 88*, pages 1640–1647, November 1988.

[3] R. Jain. "Congestion Control in Computer Networks: Issues and Trends". *IEEE Network*, pages 24–30, May 1990.

[4] A. Tanenbaum. *"Computer Networks"*. Prentice-Hall International Inc., Englewood Cliffs, NJ, 1987.

[5] N.F. Maxemchuk. "Dispersity Routing". In *Proc. of ICC'75*, pages 41.10–13, San Francisco, CA, June 1975.

[6] D. Bertsekas and R.G. Gallagher. *"Data Networks"*. Prentice-Hall International Inc., Englewood Cliffs, NJ, 1989.

[7] M-J. Lee and J.R. Yee. "An Algorithm for Optimal Minimax Routing in ATM Networks". Submitted to *IEEE Journal on Selected Areas on Communications*, January 1991.

[8] T.S.P. Yum. "The Design and Analysis of a Semi-dynamic Deterministic Routing Rule". *IEEE Transactions on Communications*, COM-29(4):498–504, April 1981.

[9] T.S.P. Yum and M. Schwartz. "The Join-Biased Queue Rule and its Application to Routing in Computer Communication Networks". *IEEE Transactions on Communications*, COM-29(4):505–511, April 1981.

[10] R. Krishnan and J.A. Silvester. "Adaptive Routing for Congestion Control in Multipath Regular Networks". In *Proc. of 28th Annual Allerton Conf. on Communication, Control and Computing*, October 1990.

[11] R. Krishnan. "On the Use of the Join-Biased Queue Rule for Adaptive Routing in Multipath Regular Networks". Master's thesis, Dept. of EE-Systems, University of Southern California, Los Angeles, CA, December 1990.