Path Diversity with Forward Error Correction (PDF) System for Packet Switched Networks

Thinh Nguyen and Avideh Zakhor
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley, CA 94720
{thinhq, avz}@eecs.berkeley.edu

Abstract*— Packet loss and end-to-end delay limit delay sensitive applications over the best effort packet switched networks such as the Internet. In our previous work, we have shown that substantial reduction in packet loss can be achieved by sending packets at appropriate sending rates to a receiver from multiple senders, using disjoint paths, and by protecting packets with forward error correction. In this paper, we propose a Path Diversity with Forward error correction (PDF) system for delay sensitive applications over the Internet in which, disjoint paths from a sender to a receiver are created using a collection of relay nodes. We propose a scalable, heuristic scheme for selecting a redundant path between a sender and a receiver, and show that substantial reduction in packet loss can be achieved by dividing packets between the default path and the redundant path. NS simulations are used to verify the effectiveness of PDF system.

I. Introduction

Delay sensitive applications such as video streaming and conferencing are challenging to deploy over the Internet due to a number of factors such as, high bit rates, delay, and loss sensitivity. Transport protocols such as TCP are not suitable for delay sensitive applications since they retransmit lost packets, resulting in long delay. To this end, many solutions have been proposed from different perspectives. From source coding perspective, layered and error-resilient video codecs have been proposed. A layered video codec deals with heterogeneity and time-varying nature of the Internet by adapting its bit rate to the available bandwidth [1]. An error-resilient codec modifies the bit stream in such a way that the decoded video degrades more gracefully in lossy environments[1][2][3]. From channel coding perspective, Forward Error Correction (FEC) techniques have been proposed to reduce delay due to retransmission, at the expense of bandwidth expansion [4][5]. Another commonly used technique in lossy environments is retransmission. While retransmission results in the least amount of bandwidth overhead, it does introduce additional delay of roughly a round trip time between the sender and receiver. Hence, the overall delay using retransmissions often exceeds 150 *milliseconds*, the tolerable delay limit for many interactive applications such as video conferencing [6]. From protocol perspective, TCP-friendly protocols [1][7] use equation based rate control to compete fairly with other TCP traffic for bandwidth, while stabilizing the throughput, and reducing jitter for multimedia streaming [7]. From network perspective, content delivery network (CDN) companies such as Akamai use edge architecture as shown in Figure 1 to achieve better load balancing, lower latency, and higher throughput. Edge architecture reduces latency by moving content to the edge of the network in order to reduce round-trip time and to avoid congestion in the Internet.

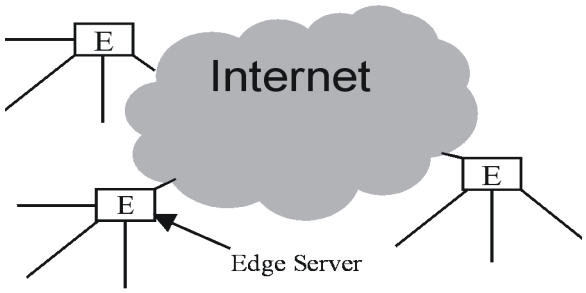

Fig. 1. *Edge server architecture.*

Most of the above schemes assume a single fixed path between the receiver and the sender throughout the session. If the network is congested along that path, video streaming suffers from high loss rate and jitter. Furthermore, authors in [8][9] have revealed the ill-behaved systematic properties in Internet routing, which can lead to sub-optimal routing paths. Based on these, it is conceivable to send packets simultaneously over multiple paths as a diversification scheme to combat the unpredictability and congestion in the Internet. If the path between a particular sender and a receiver experiences packet loss due to congestion, packets traversed through other paths can be used to recover the lost packets. In previous work [10], we have shown that by sending packets at appropriate rates on the disjoint paths from multiple senders to the receiver, and by employing redundancy through FEC, the effective packet loss rate can be significantly reduced as compared to sending all packets on a single path with the same level of redundancy. In essence we have shown that path diversity transforms a single path with bursty loss behavior into multiple paths with uniform loss for which FEC is quite effective.

In this paper, we extend our previous work to propose a single sender, single receiver, Path Diversification system with Forward error correction (PDF), over packet switched networks such as the Internet. Our proposed PDF system is similar to Resilient Overlay Network (RON) [11] in that it

consists of a set of participating nodes that receive and forward packets to other nodes. Unlike RON however, our PDF system forwards packets simultaneously over multiple redundant paths rather than selecting an optimal path to send all the packets on.

As we have shown previously [10], the performance of FEC in multi-sender and multi-path scenario depends heavily on the correlation of packet loss between paths [10]. If the packet loss between multiple paths are correlated, path diversity and FEC are not enough to bring packet loss rate down to acceptable levels. Hence, the central question in one sender one receiver scenario with path diversity is whether there exists sufficiently disjoint paths between a pair of senders and receivers on the Internet to result in uncorrelated loss patterns between paths. If the paths are not entirely disjoint, then the probability of congestion on the shared links between the paths must be small in order to minimize the overall end-to-end loss. Recent work [11] using RON, has shown how to route packets around all the observed outages between any pair of senders and receivers in an experimental network. This suggests the existence of path redundancy between nodes on the Internet. Many Internet topological models such as *Albert-Barabasi* [12] also exhibit a high level of disjointness between the paths connecting two nodes. In this paper, we propose a heuristic scheme to create a redundant path using the overlay framework [13][11][14]. We further characterize the disjointness between the redundant and default paths for various Internet topologies, and show significant reduction in packet loss using PDF system over uni-path scheme.

The remainder of our paper is organized as follows. In Section II, we present relevant work on path diversity. In Section III, we motivate our proposed approach by showing theoretical results on packet loss reduction by coupling path diversity with FEC. In Section IV, we describe the proposed PDF system. In Section V, we propose a heuristic scheme for selecting the redundant path based on *traceroute* [15] tool. In Section VI, we present the simulation results showing the average path length of the redundant path and disjointness between itself and the default Internet path. We also provide NS [16] simulation results, demonstrating the reduction in packet loss using the proposed PDF system. Finally, we conclude in Section VII.

II. Related Work

Many diversity schemes have been proposed in wireless literature, ranging from frequency and time, to spatial diversity [17]. In wired networks, path diversity was first proposed in [18] and the theoretical work on information dispersion for security and load balancing was proposed in [19]. There have been other works dealing with simultaneous downloading of data from multiple mirror sites to accomplish path diversity. If the data is not delay sensitive, it is possible to use multiple TCP connections to different sites, with each TCP connection downloading a different part of the data. For example, Digital Fountain has used an advanced class of linear-time codes to enable the receivers to receive any N linear-time coded packets from different senders, so as to recover N original data packets [20].

More recently, path diversity using overlay networks has been proposed in [13][11][14]. To create a redundant path in the overlay framework, the sender sends packets to a relay node, and the relay node then forwards packets to the receiver [13][11]. By selecting the relay node appropriately, the packets traveling through the relay node take a different underlying physical path than that of the Internet default path between the sender and receiver. Hence, path diversity is accomplished by sending packets on both the default path and the redundant path. From traffic engineering point of view, RON [11] also provides alternate paths between sender and receiver using relay nodes. However, RON actively probes for the current "best" path based on delay and loss, and sends all packets through that path rather than simultaneously sending packets on multiple paths. Note that path diversity can also be created using source based routing supported at the routers. Using this approach, one can explicitly specify a set of nodes for each packet to transverse to the destination. Currently, source based routing is available only to a few autonomous systems (AS).

III. Motivation

In this section, we present the intuition and theoretical results showing that significant reduction in packet loss rate can be achieved by using path diversity together with FEC. We begin with the network model.

An accurate model for packet loss over the Internet is quite complex. Instead, we model our network as a simple two-state continuous-time Markov chain, which has been shown to approximate the packet loss behavior over the Internet fairly accurately [21][22][23]. A two-state continuous-time Markov chain is characterized by the rates at which the chain changes from "good" to "bad" state and vice versa. The probability of packet lost is small when the network is in the "good" state, and is large, otherwise. It is well known that lost packets in the Internet often happen in bursts. This is due to the way successive packets are dropped at the network routers during congestion. Most routers employ First-In-First-Out (FIFO) policy in which, the successive arrived packets are dropped if the network buffer becomes full. Hence, a Markov chain which models a bursty loss environment, approximates the Internet traffic reasonably well. Also the average congestion period during which, the amount of aggregate traffic exceeds the link's capacity, can be thought of as the average time that the Markov chain is in the bad state. Clearly, sending more packets during congestion results in larger number of lost packets.

Let us now consider the following path diversification scheme. Rather than sending packets on the default path at 800kbps, suppose we send 400kbps on a default path, and the remaining 400kbps on a disjoint, redundant path. If congestion happens on either path, but not simultaneously on both, then we would expect the number of successive lost packets to be smaller in the 2-path scenario than in the single path one.

This is because sending more packets while the path is in congestion results in larger number of successive packet loss.

The effect of sending packets at a lower rate on multiple independent paths in effect transforms the bursty loss into a uniform loss, thus increasing the efficiency of FEC techniques. Naturally, given a number of independent paths each with a different loss behavior, the source bit rate, and the total amount of FEC protection, there should be an optimum partition of sending rates for each path in order to minimize the irrecoverable loss probability. The irrecoverable loss probability is the probability that FEC cannot recover the lost packets in a FEC block. For a Reed-Solomon code $RS(N, K)$ which contains K data packets and $N - K$ redundant packets, the irrecoverable loss probability is the probability that more than $N - K$ are lost per N packets. In our previous work [10], we have provided a procedure for computing the optimal sending rates for different paths from multiple senders to one receiver in order to minimize the irrecoverable loss probability. We now present the numerical results based on our previous work to motivate the problem and approach for this paper.

Let us consider sending packets on two disjoint paths A and B. Packets are protected using RS(30, 23), packet size is 500 bytes, and total sending rate is 800 *kbps*. We refer to the "average bad time" as the average duration that the path is in "bad" state or in congestion, and the "average good time" as the average duration that the path is in "good" state or without congestion. The longer average bad time indicates longer burst loss. The average good time for both paths is 1 *second*. The average bad time for path A is 10 *milliseconds*, while that of path B varies from 10 *milliseconds* to 50 *milliseconds* as shown in Figure 2. Figure 2 shows N, the optimal number of packets per FEC block that should be sent on path A, or equivalently, sending rate on path A as a function of the average bad time of path B. As seen, when the average good and bad times of the two paths are identical at 1 *second* and 10 *milliseconds* respectively, packets are divided equally between them at 15 packets each. When the average bad time of path B increases, more packets are sent on path A. This is intuitively plausible since path A is a better path. Figure 3 shows the ratio of irrecoverable loss probability by sending all packets on path A, the better path, over that of optimally dividing packets between paths A and B. As seen, the irrecoverable loss probability using path diversity can be reduced as much as 15 times over that of using uni-path scheme. An important observation to be made is that even though the average loss rate of path B is roughly 5%, i.e. five times than that of path A, protecting packets with $RS(30, 23)$ and sending packets simultaneously over both paths, can still reduce the irrecoverable loss probability. This suggests that if there exists a redundant path which is not "nearly" as good as the default path between sender and receiver, then it is still advantageous to deploy the path diversity scheme. Further quantitative results on amount of FEC and burstiness of the network can be found in [24].

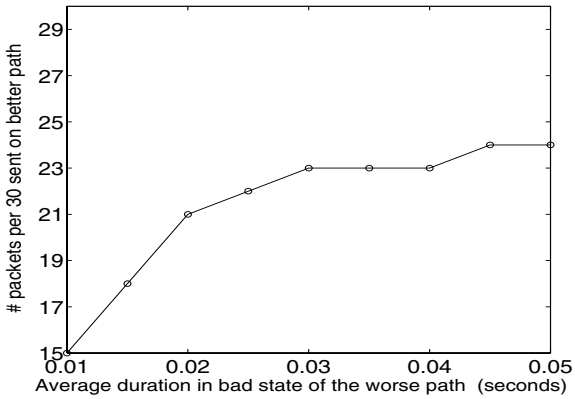

Fig. 2. *Optimal rate partition using two paths.*

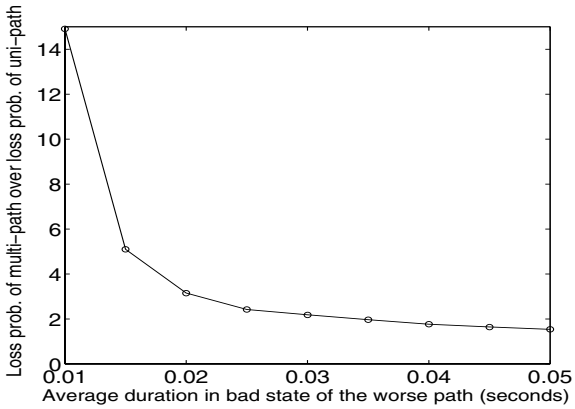

Fig. 3. *Ratio of irrecoverable loss probabilities of the uni-path scheme to multi-path scheme.*

IV. System Description

In this section, we describe the system architecture for path diversity based on overlay framework [11][14]. As seen in Figure 4, the system consists of a set of participating nodes. Circles represent participating overlay nodes, squares denote routers, and the bold solid and dashed lines represent the underlying physical and virtual paths between the nodes, respectively. The thin vertical solid lines connecting circles and squares represent the correspondence between virtual nodes and physical routers. At any instance, a node can act simultaneously as a receiver, a sender, or a relay node. A sender can send video packets to the receiver using the default Internet path or via a relay node which then forwards the video packets to the receiver. By choosing an appropriate relay node, the packets traverse an underlying physical path that is different from the one used by default Internet path. Nodes can be deployed at various locations on the Internet, although redundant paths via nodes in the same AS may have larger number of shared links. Hence, nodes are preferably located in different AS to reduce correlated congestion and outages so as to improve the performance of the PDF system.

A participating node which is neither a receiver nor a

sender, may still receive and forward packets on behalf of other senders and receivers. For two way applications such as video conferencing, they act simultaneously as both senders and receivers. In addition, the senders and receivers do not necessarily have to be the participating nodes. We envision the participating nodes as an overlay network to be deployed by companies or organizations that are interested in providing low delay communication services such as video streaming or conferencing over the Internet between their geographically diverse sites. Companies and organizations typically have multiple gateways from their ASes to other ASes that potentially enable large number of independent paths [25].

To set up a communication channel between the sender and receiver, the sender first executes *traceroute* [15] from itself to all the participating nodes and the receiver. The information returned from the *traceroute* includes the link latencies, and the names of the routers along the default path from the sender to the receiver and all paths between the sender and the participating nodes. The sender also sends a setup packet to all participating nodes instructing them to execute *traceroute* from themselves to the receiver. Next, all the participating nodes send the path information between themselves and the receiver obtained from *traceroute* to the sender. The sender now has the names of the routers and their associated link latencies for the default path, the paths between itself to all participating nodes, and the paths between participating nodes to the receiver. This information is used to compute the optimal redundant path as described later in Section V.

After the redundant path via a chosen relay node is selected, the sender sends a setup packet to the selected relay node, instructing it to forward packets to the receiver on behalf of the sender from then onward. The setup packet contains a flow ID, IP address, and the port number of the receiver. Upon receiving the setup packet, the relay node stores an entry containing a flow ID, an IP address, and a port number of the receiver in a table. This table is used to forward packets on behalf of different senders to their receivers. Each packet sent from a different sender to the relay node contains a different flow ID in its header. The relay node forwards a packet to the right IP address and port number of the receiver based on its flow ID. Note that all the setup messages and executions of *traceroute* are done only at the start of the session.

In delay sensitive applications of the PDF system, we choose to only use one relay node to forward packets between a pair of sender and receiver. The reason is that the delay of a redundant path via multiple participating nodes in series is likely to be larger than that of the redundant path via only one node. However, our proposed PDF system can be easily extended to the case where there are multiple redundant paths via multiple relay nodes.

We use UDP to send all packets between participating nodes since TCP is not appropriate for real-time data transmission due to its variable throughput, and lack of precise rate control. In addition, one of the main concerns with using multiple paths to send packets is that packets likely arrive at the destination out of order. For UDP traffic, this is not a concern since a reordering buffer at the receiver can be used. For TCP traffic, however, out-of-order packets are treated as packet loss, and TCP *fast retransmit* algorithm continually resends packets which are merely in transit, reducing the connection bandwidth.

One major difference between our PDF system and RON [11] is that RON dynamically determines the "best" path to send all packets on, while our PDF system sends packets simultaneously on multiple paths, similar to Detour [14]. Detour, however, uses the multi-path at the router level whereas we accomplish multi-paths at the application level, allowing the flexibility to send packets at different rates on different paths to the destination.

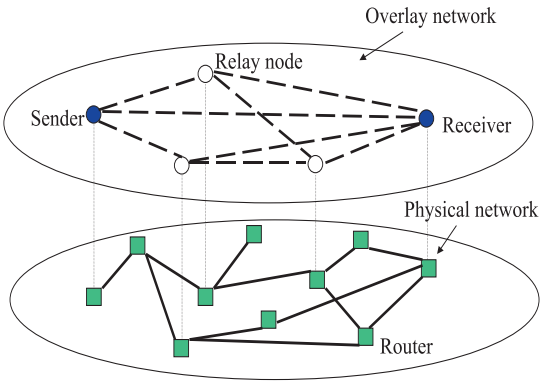

Fig. 4. *System architecture; bold dashed and solid lines denote virtual and physical paths; The thin vertical solid lines connecting circles and squares represent the correspondence between virtual nodes and physical routers.*

V. Redundant Path Selection

In this section, we propose a scalable, heuristic method for finding a redundant path via a participating relay node. Selecting an optimal path between a pair of nodes on the Internet at any one instant is difficult and complex. If the traffic conditions of the path vary rapidly, the problem becomes almost infeasible. For scalability reasons, the Internet domain routing is handled primarily by the Border Gate Protocol (BGP) [26]. BGP only exchanges and updates summarized information between ASes, ignoring the link usages and topologies within ASes. Hence, accurate path information such as number of links along the path and their associated latencies can not be obtained through BGP. Other link state routing protocols such as Open Shortest Path First (OSPF) [27] periodically probe the links between the routers for updated information such as latency, bandwidth, and link failures. OSPF can provide these information reasonably accurately, however, it is not scalable and therefore, is only used within ASes.

To measure end-to-end latency, bandwidth, and packet loss between nodes at various locations on the Internet, probing tools [28][29][30] can be used at the expense of bandwidth expansion due to sending probed packets.

Another approach is to use passive probing in which the application packets themselves are used as the probing packets, to determine the packet loss rate, latency, and bandwidth. The advantage of this approach is lack of bandwidth expansion; however, its drawback is that the measurement process depends on the application sending rates. If an application sends packets at a slow rate, the measurement resolution is low, resulting in inaccurate end-to-end estimates of packet loss rate and bandwidth.

In contrast to scalable designs such as BGP in which, the routing and link state information is kept to a minimum, systems such as RON use more complex path computation algorithms to increase path performance. A RON node aggressively probes all the paths connecting itself to other participating nodes in order to estimate detailed link quality metrics such as loss rate, bandwidth, and latency. Also, each RON node can exchange complete topologies and execute complex routing algorithms to improve the path performance, and to respond rapidly to path outages. All these benefits come at the expense of scalability, as the number of RON nodes is usually limited to fewer than 50 [11].

Our approach in this paper is to use a simple, but suboptimal technique for selecting redundant paths. In particular, we argue that finding two paths with absolute lowest loss rates for the proposed PDF system may not be needed in practice due to two reasons: (a) complexity increases due to active monitoring of probed packets and maintaining the link state information associated with all the paths i.e. scalability reasons, and (b) sending packets on two paths with the absolute lowest loss rates may not be necessary to achieve reasonable performance in a PDF system with appropriate FEC protection level. This can be justified considering the results in Section III, indicating that substantial gains can be achieved even in situations where the packet loss rate on the redundant path is many times that of the default path.

Based on the above discussion, we propose a scalable, heuristic scheme to select the redundant path via a participating node using path information from *traceroute* tool [15]. *Traceroute* can provide the names of the routers and roundtrip delays of links between the two communicating nodes on the Internet.

Let us formally denote a network topology as directed graph $G = (V, E)$ consisting of the vertices $v_i \in V$ and edges $e = (v_i, v_j) \in E$. Vertices v_i's can be thought of as routers or domains, and the path $p(v_1, v_n) = [v_1, v_2, ..., v_n]$ as the physical path from v_1 to v_n. A redundant path $p'(v_1, v_k, v_n)$ from v_1 to v_n, via node v_k is then $p(v_1, v_k) \cup p(v_k, v_n)$. Associated with every pair of vertices (v_i, v_j) is a weight $w(v_i, v_j)$. This weight can be thought of as delay, bandwidth, or loss rate associated with the physical link between v_i and v_j. Hence, the weight $w(p(v_k, v_l))$ associated with a path from v_k to v_l is the sum of the weights of the individual physical links joining v_k and v_l. In this paper, weights denote the latencies between participating nodes since they are readily available from *traceroute*. Let $O = [v_m, ..., v_n]$ be the set of participating nodes; then the relay node k' for creating the redundant path between vertices u and v is computed in a two-step procedure.

1) First, we compute a set of relay nodes O' that result in the minimum number of joint links between the default Internet path and all the redundant paths via a node in O, namely,

$$O' = \arg\min_k p'(u, k, v) \cap p^*(u, v)$$

where $k \in O$, $p'(u, k, v)$ denotes the redundant path via node k, and $p^*(u, v)$ denotes the default Internet path. Note that the set O' can have more than one element since there could potentially be two or more nodes in O that result in the same minimum number of joint links between the default and redundant paths. Since the average number of links between two nodes on the Internet is 16 [9], the operation $p'(u, k, v) \cap p^*(u, v)$ can be done fast. To find $\arg\min_k p'(u, k, v) \cap p^*(u, v)$, exhaustive search for the set of nodes O' that results in minimum number of joint links between the redundant and default paths can be done in $O(N)$ with N being the number of participating nodes.

2) Next, we choose the node k' that results in minimum weight associated with the corresponding redundant path, namely,

$$k' = \arg\min_l w(p'(u, l, v))$$

where $l \in O'$.

Note that it is the sender that runs the redundant path selection algorithm, and that the input to the algorithm is the path information, specifically, the names of routers and the associated link latencies of the default path $p(u, v)$ and the redundant paths $p'(u, l, v)$. The names of the routers are used in the first step to compute the number of shared links between the redundant and default paths while the latencies are used in the second step to find the path with minimum delay. As mentioned previously in Section IV, the link latencies and names of routers along the path $p(u, v)$ are readily available using *traceroute* from node u to v. The latency and router information for redundant path $p'(u, k, v)$ via node k can be obtained by executing *traceroute* twice, once from nodes u to k, and another time from nodes k to v. The information returned from the two *traceroute* executions is then concatenated to form the path $p'(u, k, v)$. Note that sender either runs this algorithm at the beginning of the new session, or it can use the stored paths provided from previous session since the paths are relatively stable. This will reduce the overhead of executing *traceroute* unnecessarily.

Intuitively, the path selection scheme first finds a set of redundant paths that are as disjoint as possible from the default path. Within this set of redundant paths, it then selects the one that results in minimum latency. An alternative might seem to be to select the redundant path based on traffic characteristics of each link along the path between two nodes. However, since we do not have knowledge of loss rates and bandwidths for individual links, we choose the redundant path to be maximally

disjoint with the default Internet path so as their losses are uncorrelated; this results in the PDF system to be effective in minimizing the packet loss rate.

In case the latency of the redundant path exceeds the the desired delay, that redundant path is ignored, and the same two-step procedure is applied to all the remaining relay nodes $k \in \{O \setminus O'\}$ to select a new redundant path. Note that after each iteration, the number of shared links for the redundant path increases.

Note that our PDF system does not aggressively send out probing packets to monitor the current loss rates and bandwidths between participating nodes. Instead, it simply uses the route information between participating nodes, and only invokes *traceroute* at the start of the session. Hence, our solution is scalable, even though its performance is potentially lower than that of non-scalable systems [11] with aggressive probing for network conditions.

One of the drawbacks of PDF system is that its performance depends on the information provided by *traceroute*. This information can be incomplete or inaccurate. For example, *traceroute* can only differentiate between routers and not switches. Two paths with completely different routers may share the same backbone switches. Hence, if that backbone switch is overloaded, packet loss between routes can be correlated. Although, packet loss rate in backbone switches is small, it is still a concern.

Another drawback is that some ASes do not report accurately or deliberately hide information about their networks; only certain routers are visible to outside and therefore, complete information is not available.

VI. Simulation Results

In this section, we characterize the disjointness, hop counts, and latency of the redundant path using the proposed scheme for various network topologies, and investigate the corresponding performance of the proposed PDF system using NS simulations.

A. Simulation Topologies

We use the Internet topology generator software Brite [31] to generate various *Albert-Barabasi* topologies for all the simulations. *Albert-Barabasi* model has been shown to approximate the Internet topology reasonably well [12][31]. In particular, we generate two, two-level hierarchical *Albert-Barabasi* models, and one flat *Albert-Barabasi* model. All topologies contain 1500 nodes each. The top level of the two-level hierarchical topology models the collection of ASes while bottom level models the routers within an AS. The two two-level *Albert-Barabasi* models are meant to model wide area Internet topology consisting of many ASes with various degrees of interconnectivity, and the flat *Albert-Barabasi* model represents the router interconnectivity within an AS as shown in Figures 5 and 6, respectively . The relevant information for each simulated topology is listed in Table I. *H-Albert-Barabasi* stands for hierarchial *Albert-Barabasi* model.

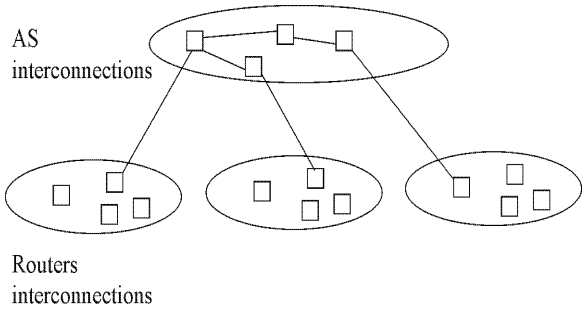

Fig. 5. *Two-level hierarchical topology*

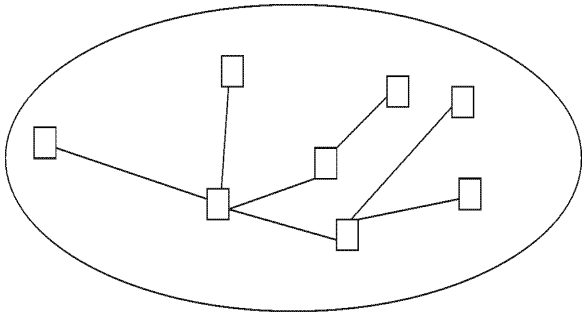

Fig. 6. *Flat topology*

To estimate the average latency, hop counts, and the degree of disjointness between the redundant path and the default Internet path, we perform the following three steps in our simulations. In step one, we randomly choose a set of participating nodes. In step two, we randomly choose a pair of receiver and sender in the set of participating nodes. In step three, we use our scheme in Section V to find the redundant and the default paths for a given configuration of sender, receiver, and participating nodes. The default path in our simulations is assumed to be the one with smallest latency or equivalently the shortest path between the sender and receiver, and hence can be computed using OSPF algorithm [32] for a given topology. This assumption is not critical to our redundant path selection scheme as we merely need a way to compute a default path.

Next, we repeat step one to three over 5000 times to obtain the average latency, hop counts, and the number of shared links between redundant and default paths. Define the jointness percentage between the default and redundant paths as the number of shared links between them over the number of links of the default path. Figure 7 shows the jointness percentage between the default and redundant paths for all three topologies as a

Models	No. Nodes	No. Edges
Flat *Albert-Barabasi*	1500	2967
H-Albert-Barabasi I	1500	2997
H-Albert-Barabasi II	1500	4377

TABLE I

Information for various topologies

function of percentage of participating nodes. As expected, the jointness percentage decreases as the number of participating nodes increases for all three topologies. This phenomenon is intuitively plausible since a redundant path is created via a participating node. Larger number of participating nodes allows more choices of redundant paths, thus producing more disjoint paths than a configuration with fewer participating nodes. As seen in Figure 7, on average, to achieve less than 10% shared links or less than one shared link between the default and redundant paths for all three topologies, only 2% of total nodes are required to be participating nodes. Another observation is that the percentage of shared links is smaller for *Albert-Barabasi* II model than that of *Albert-Barabasi* I. The reason is that *Albert-Barabasi* II has larger degree of interconnectivity between its nodes than that of *Albert-Barabasi* I; hence, more disjoint paths can be found. Note that the flat *Albert-Barabasi* model has the lowest ratio of all three topologies due to following reasons. As mentioned previously, the top level of the two-level hierarchical *Albert-Barabasi* topologies represents interconnectivity between AS while the bottom level represents the interconnectivity between routers within an AS. If two participating nodes are located in two different ASes, all the paths between them must go through a few AS nodes. As a result, the redundant paths may share larger number of links with the default path than with a flat topology.

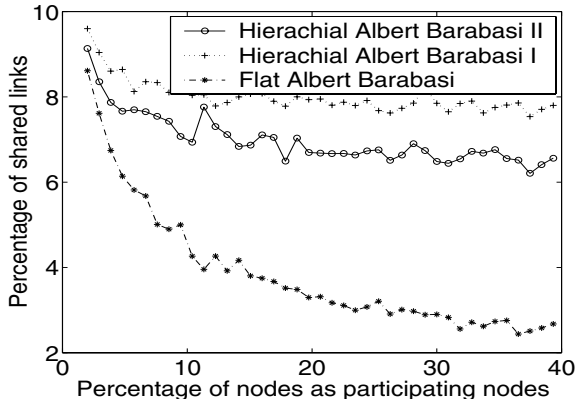

Fig. 7. *Percentage of shared links between the redundant and the default paths.*

Figure 8 shows the ratio of average latency of the redundant path to that of the default path as a function of percentage of participating nodes for all three topologies. As expected, the latency decreases as the number of participating nodes increases for all three topologies. This phenomenon is intuitively plausible since as the number of participating nodes increases, there are more choices for redundant paths, one of which is likely to have shorter latency. Another observation is that the latency ratio is smaller for *Albert-Barabasi* II model than that of *Albert-Barabasi* I. The reason is that *Albert-Barabasi* II has a larger degree of interconnectivity between its nodes than *Albert-Barabasi* I, thus resulting in redundant paths with lower latencies.

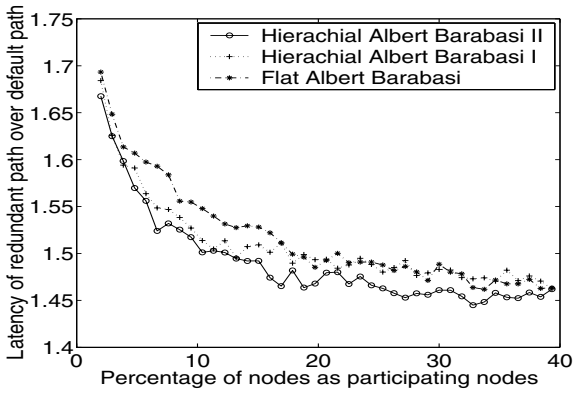

Fig. 8. *Latency of redundant path over the latency of default path.*

For small percentage of participating nodes, the latency ratio of redundant over default paths is as high as 1.7. In this case, if the round-trip time of the default path is 100 *milliseconds*, then the average round-trip time of the corresponding redundant path is 170 *milliseconds*, thus exceeding the tolerable delay of 150 *milliseconds* for two-way interactive applications. This problem can be remedied by either increasing the percentage of participating nodes to 10%, or by allowing the redundant path to share larger number of links with the default path. With the default round-trip time of 100 *milliseconds* and with 10%, or more participating nodes, the average latency ratio of redundant over default paths is 1.5, corresponding to the delay of 150 *milliseconds* for the redundant path, thus satisfying the requirements for two-way interactive applications. The fact that typical round-trip time between two sites within North America is less than 100 *milliseconds* makes the deployment of our proposed PDF system feasible from a practical point of view. For example, the average round-trip times from Georgia Tech, Purdue university, and Duke university to U.C. Berkeley are 62, 57, and 90 *milliseconds*, respectively.

An important observation to make is that for all three topologies, the latency ratios and the jointness percentages decrease rapidly when the percentage of participating nodes is less than 20%, and decrease rather gradually beyond 20%. This suggests that it may not be all that beneficial to increase the number of participating nodes beyond a certain value.

Figure 9 shows the ratio of the average number of links in the redundant path to that of default path as a function of percentage of participating nodes. For all three topologies, the ratio decreases slightly at first, then stays relatively constant. This phenomenon together with plots in Figure 8 indicate that major reduction in latency does not result from fewer links, rather from selecting links with shorter latencies. Figure 10 shows the cumulative distribution of shared links between redundant and default paths for the three topologies, with 10% of the nodes participating. As seen, the probability that the redundant and default path share few links is large for all three topologies. For example, the probability of two or

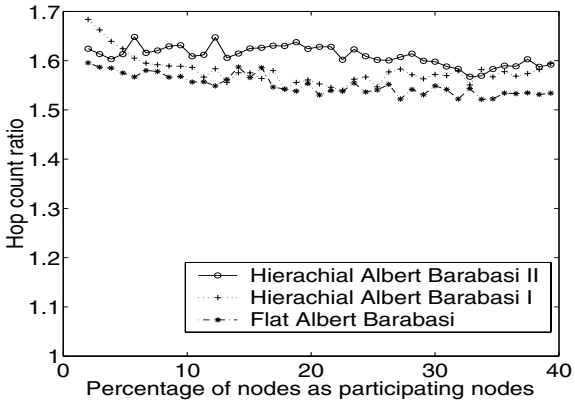

Fig. 9. *Number of hops of the redundant path over the number of hops of the default path.*

fewer shared links for *Flat Albert-Barabasi*, *Albert-Barabasi II*, and *Albert-Barabasi I* is roughly 100%, 90%, and 85%, respectively. Note that the average number links of the default paths for *Flat Albert-Barabasi*, *Albert-Barabasi II*, and *Albert-Barabasi I* are 6, 11, and 12, respectively. This indicates that a redundant path with few shared links can be found with high probability, and hence PDF system can be deployed effectively. In Section VI-B, we will characterize the packet loss reduction for various number of shared links between the default and redundant paths.

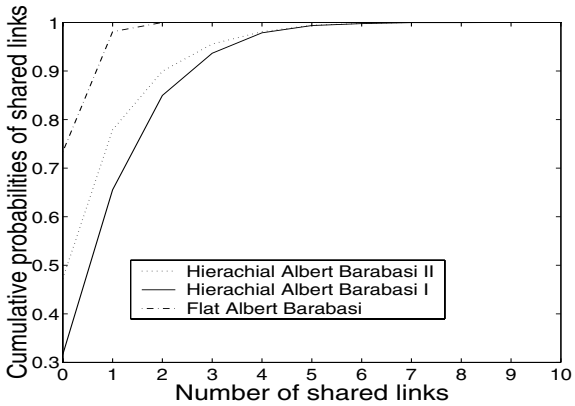

Fig. 10. *Cumulative distribution of shared links for various network topologies.*

B. NS Simulations

In this section, we characterize the packet loss reduction for various number of shared links between redundant and default paths using NS [16]. The simulation topology for the two disjoint redundant and default paths is shown in Figure 11. Based on the average number of links between two participating nodes in Section VI-A, the number of links for default and redundant paths are set to 11 and 18 respectively. Each link's capacity is 2Mbs with propagation delay of 4 *milliseconds*. To simulate bursty packet loss of the Internet, random exponential traffic is generated at each link with the peak rate of 1.8Mbs, average idle period of 8 seconds, and the burst period of 40 *milliseconds*. We compare the packet

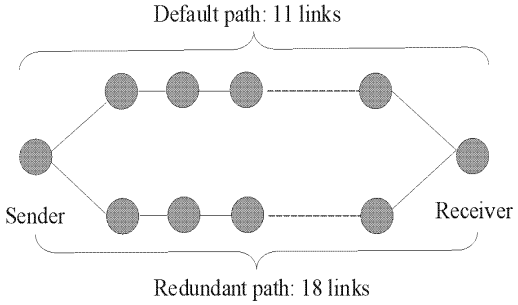

Fig. 11. *Simulation configuration for two disjoint redundant and the default paths.*

loss rate for following three scenarios: (1) sender streams the video to the receiver at 800kbps on the default path, (2) sender streams the video to the receiver on both redundant and default paths at 400kbps for each path with two paths are assumed to be completely disjoint, and (3) same as scenario 2 except there is one shared link between redundant and default paths.

In all scenarios, the video packet size is 500 bytes, and packets are protected using Reed-Solomon code RS(30, 23) with 23 data packets and 7 redundant packets for each FEC block. Hence, if there are more than 7 lost packets per FEC block, then lost packets cannot be entirely recovered. Figures 12, 13, and 14 show the number of lost packets per 30 packets versus the packet sequence number for scenarios 1, 2, and 3, respectively. The points above the horizontal line represent irrecoverable loss events. As seen, there are considerably more irrecoverable loss events for scenario 1 than for 2. This is intuitively plausible since sending packets at a lower rate on multiple independent paths transforms the bursty loss behavior of a single path into a uniform loss behavior, thus reducing the burstiness, and increasing the recoverable probability. Also, the number of irrecoverable loss events for scenario 3 is larger than that of 2. This is due to the one shared link between the redundant and default paths in scenario 3. Assuming that links are independently congested, the larger number of shared links between the redundant and default paths leads to higher chance of simultaneous bursty loss on both paths, for which FEC is ineffective. One can think of scenario 1 where all packets are sent on only default path as an extreme case of path diversity in which all the links of two paths are shared.

Clearly, the recoverable probability of FEC decreases as the number of shared links between the redundant and default paths increases. To characterize the effect of number of shared links on the loss rate, Figure 15 shows the ratio of the effective packet loss rate of uni-path scheme to that of dual path scheme as a function of number of shared links between them. The effective packet loss rate is the ratio between the number of irrecoverable lost packets and the total sent packets. As seen, the effective loss rate for the single path scheme is more than

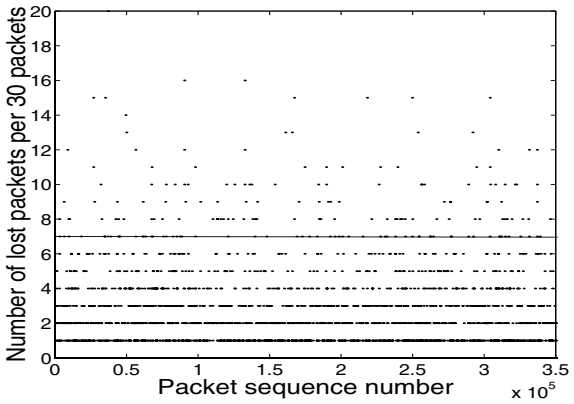

Fig. 12. *Sending packets using traditional default path.*

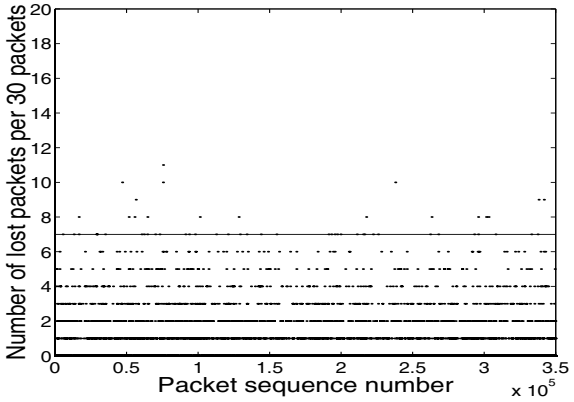

Fig. 13. *Sending packets using both redundant and default paths without shared link between them.*

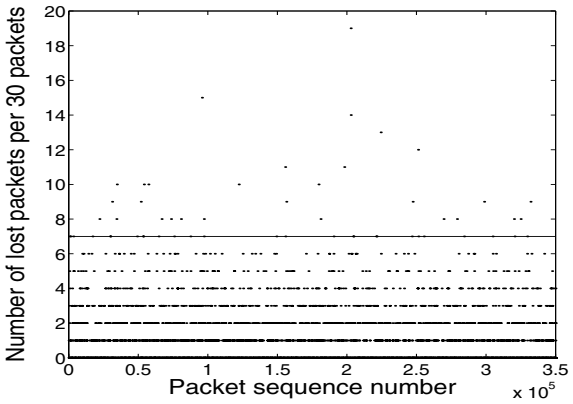

Fig. 14. *Sending packets using both redundant and default paths with one shared link between them.*

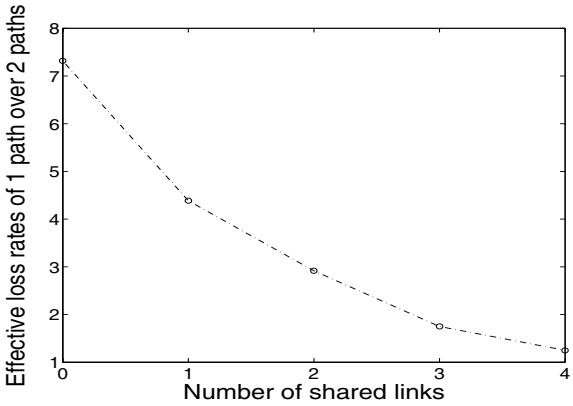

Fig. 15. *The ratio of average loss rates using one path over that of using both redundant and default paths with various number of shared links between them.*

7 times that of the path diversity scheme with completely disjoint redundant and default paths. The reduction in effective loss rate from using path diversity decreases as the number of shared links between the two paths increases. However, even when 3 of out 11 links are shared, the effective loss rate using path diversity scheme is twice smaller than that of using uni-path scheme.

VII. Conclusions

In this paper, we have proposed a PDF system for delay sensitive applications over packet switched networks in which, disjoint paths from a sender to a receiver are established using a collection of relay nodes. A scalable, heuristic scheme for selecting a redundant path has been proposed, and the resulting redundant path's lengths and disjointess for various Internet-like topologies have been characterized. Our simulations have demonstrated that, for various Internet-like topologies, only 10% of participating nodes are required for the proposed path redundant selection scheme to effectively find a redundant path sharing 2 or fewer links with the default path; this effectively results in a factor of 3 reduction in irrecoverable packet loss as compared to uni-path scheme. We are currently implementing our PDF system for delay sensitive video applications over the actual Internet.

Acknowledgment

This work was supported by NSF under grant CCR-9979442 and AFOSR contract F49620-00-1-0327.

References

[1] W. Tan and A. Zakhor, "Real-time internet video using error resilient scalable compression and tcp-friendly transport protocol," *IEEE Transactions on Multimedia*, vol. 1, pp. 172–186, june 1999.

[2] G. De Los Reyes, A. Reibman, S. Chang, and J. Chuang, "Error-resilient transcoding for video over wireless channels," *IEEE Transactions on Multimedia*, vol. 18, pp. 1063–1074, june 2000.

[3] J. Robinson and Y. Shu, "Zerotree pattern coding of motion picture residues for error-resilient transmission of video sequences," *IEEE Journal on Selected Areas in Communications*, vol. 18, pp. 1099–1110, June 2000.

[4] H. Ma and M. Zarki, "Broadcast/multicast mpeg-2 video over wireless channels using header redundancy fec strategies," in *Proceedings of The International Society for Optical Engineering*, November 1998, vol. 3528, pp. 69–80.

[5] W. Tan and A. Zakhor, "Error control for video multicast using hierarchical fec," in *Proceedings of 6th International Conference on Image Processing*, October 1999, vol. 1, pp. 401–405.

[6] IMTC, International Multimedia Telecommunications Consortium, http://www.imtc.org/h323.htm, *H.323 Standard, Packet-based multimedia communications systems*.

[7] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Equation-based congestion control for unicast application," in *Architectures and Protocols for Computer Communication*, October 2000, pp. 43–56.

[8] C. Labovitz, G. Malan, and F. Jahanian, "Internet routing instability," *IEEE/ACM Transactions on Networking*, vol. 6, pp. 515–528, October 1998.

[9] V. Paxson, "End-to-end routing behavior in the internet," *IEEE/ACM Transactions on Networking*, vol. 6, pp. 601–615, October 1997.

[10] T. Nguyen and A. Zakhor, "Distributed video streaming with forward error correction," in *Packet Video Workshop*, Pittsburg, PA, April 2002.

[11] D.G. Andersen, H. Balakrishnan, M.F. Kaashoek, and R. Morris, "The case for resilient overlay networks," in *Proceeding of HotOS VIII*, May 2001.

[12] A.L. Barabasi and R. Albert, "Emergence of scaling in random networks," *Science*, pp. 509–512, October 1999.

[13] J. Apostolopoulos, "Reliable video communication over lossy packet networks using multiple state encoding and path diversity," in *Proceeding of The International Society for Optical Engineering*, January 2001, vol. 4310, pp. 392–409.

[14] Stefan Savage, Tom Anderson, Amit Aggarwal, David Becker, Neal Cardwell, Andy Collins, Eric Hoffman, John Snell, Amin Vahdat, Geoff Voelker, and John Zahorjan, "Detour: a case for informed internet routing and transport," *IEEE Micro*, vol. 19, no. 1, pp. 50–59, january 1999.

[15] *Traceroute*, http://www.tracert.com.

[16] Information Sciences Institute, http://www.isi.edu/nsnam/ns, *Network simulator*.

[17] H.V. Poor and G.W. Wornell, *Wireless Communications: Signal Processing Perspectives*, Prentice Hall, 1998.

[18] N.F. Maxemchuk, *Dipersity Routing in Store and Forward Networks*, Ph.D. thesis, University of Pennsylvania, 1975.

[19] M.O. Rabin, "Efficient dispersal of information for security, load balancing, and fault tolerance," *Journal of the Association for Computing Machinery*, vol. 36, no. 2, pp. 335–348, April 1989.

[20] J. Byers, M. Luby, and M. Mitzenmacher, "Accessing multiple mirror sites in parallel: using tornado codes to speed up downloads," in *Proceedings of Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies*, March 1999, vol. 1, pp. 275–283.

[21] J. Bolot, S. Fosse-Parisis, and D. Towsley, "Adaptive fec-based error control for internet telephony," in *Proceedings of IEEE INFOCOM*, 1999, vol. 3, pp. 1453–60.

[22] U. Horn, K. Stuhlmuller, M. Link, and B. Girod, "Robust internet video transmission based on scalable coding and unequal error protection," *Signal Processing: Image communication*, vol. 15, pp. 77–94, 1999.

[23] M. Yajnik, S. Moon, J. Kurose, and D. Towsley, "Measurement and modelling of the temporal dependence in packet loss," in *INFOCOM*, 1999, vol. 1, pp. 345–52.

[24] T. Nguyen and A. Zakhor, "Distributed video streaming," *Submitted to IEEE/ACM Transactions on Multimedia and Networking*.

[25] netVMG, http://www.netvmg.com/index2.html, *netVMG*.

[26] Y.Rekhter and T.Li, *A Border Gateway Protocol 4 (BGP-4)*, Internet Engineering Task Force, RFC 1771, 1995.

[27] J. Walrand, *Communication Networks*, WCB/McGraw-Hill, 1998.

[28] *CAIDA Tools*, http://www.caida.org/tools/, 2001.

[29] V. Paxon, G. Almes J. Mahdavi, and M. Mathis, *Framework for IP Performance Metrics, RFC 2330*, IETF, May 1998.

[30] V. Jacobson, *pathchar - A tool to Infer Characteristics of Internet Paths*, ftp://ee.lbl.gov/pathchar/, 1997.

[31] *Internet topology generator*, http://www.cs.bu.edu/brite.

[32] A. Khanna and J. Zinky, "The revised arpanet routing metric," in *ACM SIGCOMM*, 1989, pp. 45–46.