On the Effectiveness of Proactive Path-Diversity Based Routing for Robustness to Path Failures

Chansook Lim[1], Stephan Bohacek[2], João P. Hespanha[3], and Katia Obraczka[4]

[1] Dept. Computer Information & Communication, Hongik University, 339-701, Republic of Korea, chansooklim@hongik.ac.kr
[2] Dept. Electrical & Computer Engineering, University of Delaware, Newark, DE 19716, USA, bohacek@eecis.udel.edu
[3] Dept. Electrical & Computer Engineering, University of California, Santa Barbara, CA 93106-9560, USA, hespanha@ece.ucsb.edu
[4] Computer Engineering Department, University of California, Santa Cruz, CA 95064, USA, katia@cse.ucsc.edu

Abstract. Path disruptions are frequent occurrences on today's Internet. They may be due to congestion or failures, which in turn may be attributed to unintentional factors (e.g., hardware failures) or caused by malicious activity. Several efforts to-date have focused on enhancing robustness from the end-to-end viewpoint by using path diversity. Most of these studies are limited to single- or two-path approaches. This paper is the first to address the question of what degree of path diversity is needed to effectively mitigate the effect of path failures. We seek to answer this question through extensive experiments in PlanetLab. To evaluate the effect of path diversity on routing robustness in regards to a wide spectrum of applications, we introduce a new performance metric we named *outage duration*. Experimental results show that proactively forwarding packets using a high degree of path diversity is more effective in overcoming path failures in comparison with single-path or two-path approaches. In addition, for applications in which low packet loss probability is as important as uninterrupted connectivity, we suggest a packet forwarding scheme based on *link gains* and discuss the trade-offs between robustness and packet delivery probability.

Keywords: path failure, robustness, routing, path diversity

1 Introduction

Currently, Internet routing adopts a reactive approach for handling path disruptions. In other words, it waits until failures are detected to take any action. While this strategy has been satisfactory thus far, it certainly will not provide the levels of robustness required by more recent as well as emerging (e.g., real-time, interactive, etc.) applications.

Several efforts over the last few years have attempted to mask path failures and improve end-to-end path availability. Early seminal work includes overlay

network routing schemes such as Detour and RON [1, 2]. These schemes monitor paths and upon detecting a path failure, forward packets indirectly via intermediate nodes. More recently, Gummadi et al. [3] suggested a low-overhead on-demand scheme that uses probes to find a new working path, only when a path failure is detected. A common feature among these schemes is that they are purely reactive. Thus they cannot avoid packet losses at least for tens of seconds, which is the time required to detect a path failure and switch to another path. In the case of RON, each node has information about all available paths, but forwards packets using a single path. Thus, a failure will result in a burst of losses until the path is updated. Depending on the application and the length of time to detect and react to the route failure, the resulting burst of packet losses might not be tolerated.

One way to avoid routing disruptions and thus prevent packet loss bursts is through proactive approaches that use path diversity, e.g., multipath routing, which forward packets over multiple paths. Proactive multipath routing reduces bursty packet losses by enabling some of the packets to still be delivered to the destination as long as at least one of the paths from source to destination remains viable. Reliability enhancement through path diversity is not a new idea; indeed, it has been extensively studied [4–7]. One question that comes up though is what degree of multipath diversity is necessary/sufficient to overcome path failures on the current Internet so that application-specific performance requirements are satisfied. The answer to this question depends, among other things, on how often simultaneous link failures occur. However, there has been little work on characterizing simultaneous path failures on the Internet. Although there have been some useful studies on failure characterization within one ISP (e.g., [8]), it is hard to predict how often an arbitrary connection passing through usually more than one AS may experience simultaneous path failures.

Nevertheless, most studies to-date try to cope with a single point of failure. A notable example is the work on intra-domain path diversity reported in [5]. This study focuses on proactive routing over two paths assuming that the chance of network components within an AS located physically apart to fail at exactly the same moment is extremely slim. Whether this assumption can hold even beyond intra-domain is not clear because, as pointed out before, there is little statistics on simultaneous path failures on the Internet. This is one of the main reasons why we are interested in improving routing robustness for arbitrary Internet source-destination pairs, whose routes may cross several ASs. To our knowledge, our work is the first attempt to address the question of what degree of multipath is effective in mitigating the impact of path failures on the Internet.

Our approach is based on implementing a proactive multipath routing scheme, namely the Game Theoretic Stochastic Routing (GTSR) mechanism [7], as an application-layer overlay routing protocol and conducting extensive experiments using PlanetLab [9]. To maximize the degree of path diversity, we formed many small PlanetLab groups and performed experiments where packets were sent from source to destination over multiple paths for each group. To evaluate the effect of the degree of path diversity on robustness for a wide spectrum of ap-

plications, we introduce a performance metric we call *outage duration*. Our experimental results show that proactively forwarding packets using high degree of path diversity is more effective in mitigating the effects of path failures when compared to single-path or two-path approaches.

Additionally, we also consider the case of applications that require low end-to-end loss rate and propose a packet distribution scheme using heterogeneous *link gains* in a max-flow optimization. Under this *link gain* mechanism, lossy paths may be "penalized" so that more packets are sent over paths with low loss rates. We discuss how to balance the trade-off between end-to-end loss probability and maximum outage duration by adjusting the fraction of packets to be sent on each path.

The rest of this paper is organized as follows. In the next section, we present related work as motivation to our work. Our empirical study of the impact of proactive multipath on routing robustness, including our experimental methodology and results, is presented in Section 3. In Section 4, we describe the *link gain* packet distribution scheme. Lastly, Section 5 concludes the paper and presents some directions of future work.

2 Related Work

Most existing schemes to enhance Internet robustness to path failures use reactive, single-path routing approaches. However, single-path approaches cannot avoid bursty packet losses which may happen while detecting a failure and trying to switch to an alternate path. In contrast, proactive multipath approaches which forward packets over multiple paths can reduce bursty packet losses by enabling some of the packets to still make it through, as long as at least one of the paths from source to destination remains viable. Preventing long outages will enable applications to provide uninterrupted service even while new routes are being computed/discovered. Schemes taking such approaches for the purpose of robustness to path failures include our own GTSR [7] and the work presented in [5].

In the past, an important deterrent to the use of multipath routing was the fact that TCP is known to exhibit significant performance degradation when subject to persistent packet reordering, which will likely happen under multipath routing. A number of TCP variants have recently been proposed to obviate this problem [10–12].

The benefit of proactive multipath routing becomes even more evident when considering that, from the end-host's viewpoint, identifying failures is not easy. Indeed, prior studies use different definitions of failure, some of which quite arbitrary. For example, in [3], a failure is defined as a sequence of packet losses that would have a significant or noticeable application impact. Considering TCP reaction to losses, the authors consider as failure a loss incident that began with 3 consecutive probe losses and the failure of the initial traceroute. On the other hand, in RON [2], a path outage is defined to be an incident where the observed packet loss rate over a short time interval is larger than some threshold.

One drawback with any technique to identify failures is that is it not easy to differentiate between packet losses due to severe congestion and packet losses due to true path failures. Furthermore, a failure should depend on the application (e.g., non-interactive vs. interactive applications).

Another noteworthy problem with identifying failures is that not trivial to conduct meaningful evaluation studies of different routing approaches in terms of their robustness. More specifically, existing performance metrics such as the percentage of recovered failures and the time until the failure is recovered focus on reactive routing approaches for recovering relatively long-term failures. However, these metrics are inadequate for evaluating proactive techniques aimed at providing uninterrupted connectivity and reducing bursty packet losses.

To fulfill the need for adequate ways of evaluating the robustness of proactive routing mechanisms in regards to a wide spectrum of applications, we propose a new metric named *outage duration*, which measures the duration of failures experienced by end hosts. This means that routing schemes will be judged based on how frequent longer outages are observed. In other words, if one routing scheme results in failures of longer durations more frequently compared to an other routing scheme, the latter is considered more robust than the former.

As previously pointed out, while many studies try to cope with a single point of failure, little is known on how frequently simultaneous path failures occur. In particular, to the best of our knowledge, what degree of multipath diversity is effective in overcoming failures on arbitrary Internet source-destination pairs has not been studied. We are interested in investigating whether more than 2 paths is more effective in mitigating the effect of path failures. It seems intuitive that bursty packet losses decrease as the number of paths between the source and destination increases. However, if simultaneous multiple path failures hardly occur, forwarding packets through more than two paths would not result in improved robustness.

We seek the answer to the above question through extensive measurement experiments using a proactive multipath routing protocol implemented on PlanetLab [9]. In the next section, we describe our implementation of the proactive multipath routing protocol, Game Theoretic Stochastic Routing (GTSR) [7], on PlanetLab.

3 Proactive Multipath for Routing Robustness

Game Theoretic Stochastic Routing (GTSR) uses a game-theoretic approach to multipath routing [7]. It finds all paths between a source-destination pair and computes *next-hop probabilities*, i.e., the probabilities that a packet takes a particular next-hop. This contrasts with single-path algorithms that simply determine the next-hop or even with deterministic multipath routing approaches.

GTSR determines the next-hop probabilities using a max-flow computation. This computation has a game theoretical interpretation because GTSR's routing policies are saddle solutions to a zero-sum game, in which we regard routing as one player that attempts to defeat worst-case link/node faults. In this game, one

associates to each link ℓ a probability $p_{fail,\ell}$ of fault occurrence. It was shown in [13] that optimal saddle routing policies for this game can be computed by solving a max-flow optimization [14] over a graph with link capacities given by $1/p_{fail,\ell}$. When GTSR is utilized to improve robustness, the designer typically selects low values for $p_{fail,\ell}$ in links that are perceived to be more robust. This favors sending more traffic through these links. We thus call $1/p_{fail,\ell}$ the *level-of-robustness (LOR)* of link ℓ. As mentioned above, GTSR solves max-flow optimizations with link capacities given by the LOR. Once the flow x_ℓ for each link ℓ is determined, the next-hop probability r_ℓ for link ℓ is obtained from $r_\ell := \frac{x_\ell}{\sum_{\ell' \in L[\ell]} x_{\ell'}}$, where the summation is over the set $L[\ell]$ of links that exit from the same node as ℓ.

When one wants to favor shorter paths in terms of the number of hops, this can be done by introducing a *link gain* ϵ. In a generalized max-flow optimizations with a link gain, the *flow-conservation law* can be interpreted that the incoming flow to a node is equal to the outgoing flow from the same node, possibly amplified by ϵ when $\epsilon \geq 1$. Therefore, for $\epsilon > 1$, longer paths are penalized since the cost incurred increases as the number of hops increases. In fact, as $\epsilon \to \infty$ the potential burden of an extra hop is so large that the optimal solution will only consider paths for which the number of hops is minimal, leading to shortest path routing but not necessarily single-path.

3.1 Experimental Methodology

We implemented GTSR as an application-layer overlay routing protocol in Linux. Although GTSR reacts to path failures by monitoring link status, its main feature emphasized in our experiments is its proactive multipath routing ability. Implemented as a link-state protocol, GTSR computes routing tables by solving max-flow optimizations on the entire network graph.

The main objective of our experiment is to investigate the impact of multipath diversity degree on robustness to path failures. The experiment was performed using PlanetLab [9]. To probe as many paths as possible with the assigned number of hosts, we partitioned them into several groups so that each group consists of 5 hosts. A simple overlay network is then configured for each group. 4 paths connecting the source– and destination nodes - one direct path and three indirect overlay paths through three intermediary nodes. By doing this, for each group, we can examine four single-paths, $\binom{4}{2}$ two-paths, $\binom{4}{3}$ three-paths, and one 4-path. Since we associated to each link ℓ the same $p_{fail,\ell}$, the routing policy that GTSR generates for this simple topology is simply forward packets over the 4 paths with equal probabilities. To diagnose the network easily, we used static routing and forwarded packets over the four paths in a round-robin fashion rather than in a stochastic manner. Every 50ms, a source node sends a packet which includes a sequence number. Forwarding sequentially-numbered packets in a round-robin manner enables us to measure the duration of path outages.

We started the first experiment with 10 groups consisting of 50 hosts. In this experiment conducted from May 14 to May 16, 2006, we selected PlanetLab hosts so that the hosts in the same group were geographically distributed

(similarly to [3]). The second experiment was performed using 245 hosts from November 15 to November 26, 2006. In this experiment, we did not explicitly try to choose geographically distributed PlanetLab hosts. That is, we formed each group by randomly selecting hosts; we avoided cases where any group would contain more than one host from the same site. Also in each group, the roles of source, destination, and intermediary nodes were randomly assigned. Out of 49 groups, we could collect data from 28 groups (totaling 140 hosts) successfully.

Introducing a New Performance Metric

To compare robustness over different multipath degrees easily, we compute the distribution of outage durations experienced by the receiver. More specifically, we calculate the fraction of time that the connection is in a failure for duration of at least X seconds. As a simple example, suppose that we measured path outages over a path during 24 hours and the observed outage durations in seconds were {30, 7, 10, 100, 600}. Now let's calculate the fraction of time that a connection over this path is in failure for duration of at least 60 seconds. Since there were two outages which lasted for at least 60 seconds, the fraction of time is $(100 + 600)/(24 * 60 * 60) \approx 0.0081$.

3.2 Experimental Results

Figure 1 shows the resulting distribution of outage durations for one example group. In this group, the source node is in Singapore, the destination node is in Michigan, and the three intermediary nodes are located in Japan and USA. What we can observe from the graph is that the single-path suffered outage duration of up to 57 seconds and the fraction of time that the connection is in failure with duration of at least 10 seconds is 0.1%. When one more path is added, the fraction of time that the connection is in failure of duration at least 10 seconds is almost 0.01%, which means that 2-paths was significantly helpful in overcoming path failures that the single-path suffered. For 3-paths, no failure duration of 10 seconds is observed. Also, we can see that the fraction of time that the connection is in failure duration ≥ 7 seconds is about 2×10^{-4} and 5×10^{-5}, for the 2-paths and 3-paths, respectively. However for failure duration ≥ 1 second, the curves for 3-paths and 4-paths are very close to each other. This means that adding the 4th path was not very helpful. A possible explanation for this is the fact that the underlying network does not have enough physical path redundancy; for these experiments, the physical paths for some two paths are not independent. This dependence may be avoided by choosing "better" intermediary nodes. However, if failures occur at the end hosts they cannot be avoided.

In order to summarize results over all groups, we perform the following computation. For each degree of multipath, we collect all possible distributions (curves) regardless of the group. And then, for every failure duration data point, the median value is obtained. As a result, we obtain one curve for each degree of

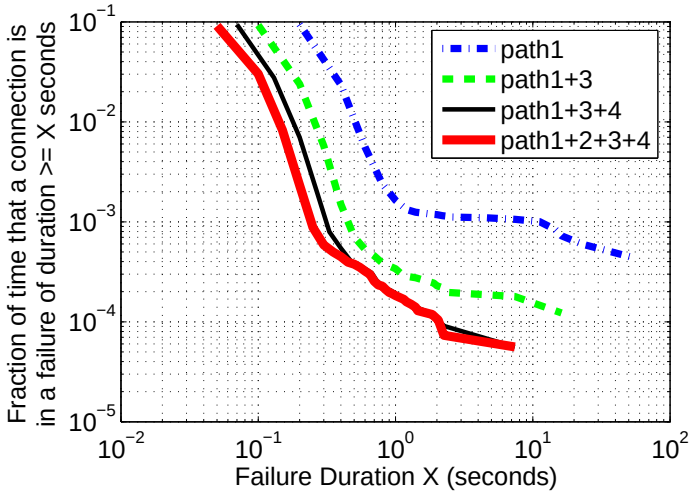


Fig. 1. An example result for a group. The Y-axis represents the fraction of time that the connection is in a failure of the duration $\geq X$ seconds.

multipath, i.e., four curves total. If a curve does not have any value for some failure duration value, the value of the curve is assumed to be 0. Figure 2 (a) shows the distribution of the median values for the first experiment. We can see the gaps between the curves corresponding to each degree of multipath. This implies that as the number of paths increases, robustness to path failures is improved.

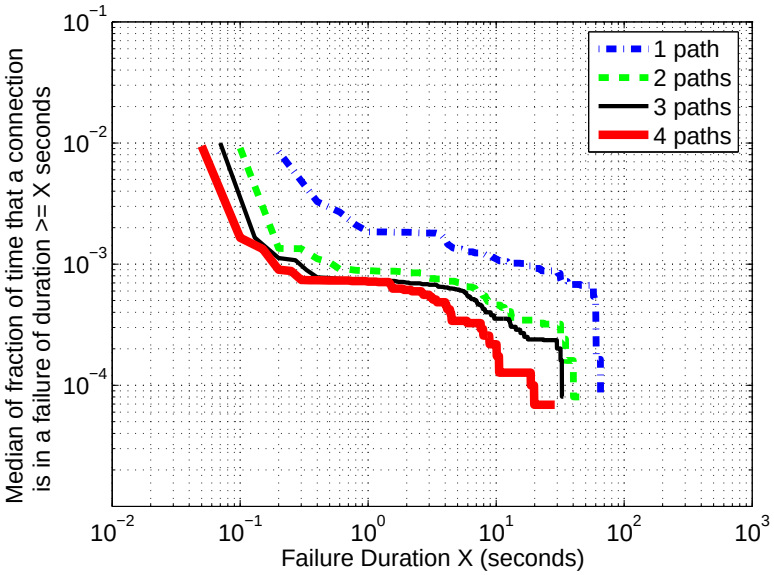
Figure 2 (b) shows the distribution of the median for the second experiment. While the gap between single-path and two-paths is significant, the curves for three-paths and four-paths almost overlap. The gap between two-paths and three-paths implies that there was robustness improvement by raising the degree of multipath from 2 to 3, even though it is not so significant as when raising it from 1 to 2. In comparison with the result of the first experiment, it is inferred that geographically distributing intermediary nodes played a significant role in improving robustness. Nonetheless, the fact that there was improvement simply by selecting intermediary nodes randomly is very encouraging.

These results show that the degree of multipath diversity affects robustness to path failures. A study on how to enlarge the gaps between the curves for different multipath degrees is included in our future work.

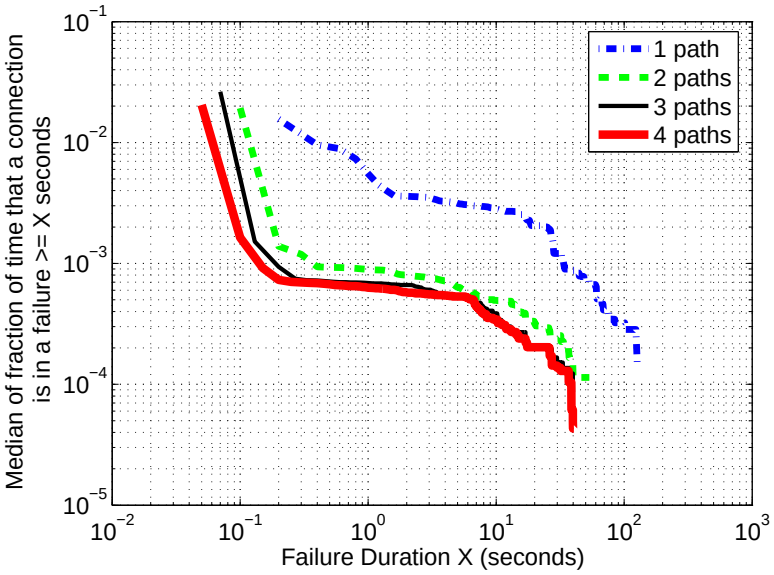
4 Controlling End-to-End Loss Rate

4.1 Exploiting Link Gains to Reduce End-to-End Loss Rate

Depending on the application, besides reducing duration of outages, reducing loss rate may be another main concern. Clearly, if low end-to-end loss rates is the only concern, the network designer would route all packets along the path(s)

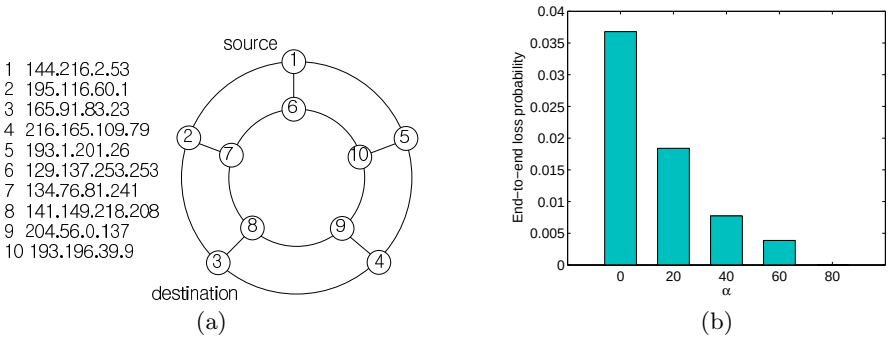


(a) The intermediary nodes are geographically distributed.



(b) The intermediary nodes are selected randomly.

Fig. 2. Distribution of mean outage duration



(a) (b)

Fig. 3. Effect of *link gains* on end-to-end loss rates : (a) Hosts and topology used in the experiments. The links incident to node 8 usually have high loss rates, because node 8 is connected via DSL. (b) Number of lost packets with respect to 5 different α values.

with the lowest loss probability. However, if both robustness to path failures and low loss rates are desired, some trade-off must be achieved.

To this end, we use *link gains* whose goal is to penalize lossy paths and hence robustness is traded for random packet drops. The notion of *link gain* is introduced in Section 3. However, the *link gain* used here is a bit different in the sense that the *link gain* for each link depends on the loss probability of the link and therefore *link gains* are not uniform across all links. Instead, the *link gain* of each link l is set to

$$\frac{1}{(1-p_{loss,\ell})^{\alpha}}$$

where $p_{loss,\ell}$ is the loss probability of link l and α is a positive number given as a network parameter. With this *link gain* definition the flow is amplified by $\frac{1}{(1-p_{loss,\ell})^{\alpha}}$ when it passes through link ℓ. The end-to-end *link gain* along a specific path is $\left(\frac{1}{\prod(1-p_{loss,\ell})}\right)^{\alpha}$. Thus, when $\alpha = 1$, the end-to-end *link gain* is the reciprocal of the probability of successfully delivering the packet. For $\alpha = 0$, the *link gain* for every link is equal to 1 and loss probability has no impact on routing;hence, packets are routed to maximize robustness. On the other extreme, for a very large α value, the *link gain* becomes very large and packets are forwarded along the path(s) with the lowest loss probability.

To show the impact of *link gain* on loss rate, we performed a simple experiment in PlanetLab. We used the topology shown in Figure 3 (a). This topology has 10 nodes and the nodes were connected by a 3-connected graph. Node 8 is connected to the Internet via a DSL line so the loss probabilities on the links to other hosts are typically higher. We selected a source and destination pair so that node 8 is included on one of the paths.

On each host, we performed 5 routing programs, in which five α values were used, namely 0, 20, 40, 60, and 80. At the source node, 5 sender programs send

out packets through distinct routing programs. The results in Figure 3 (b) show that, as expected, the packet loss rate decreases, as α increases.

4.2 Trade-offs between robustness to path failures and loss rate

The downside of this packet distribution scheme, i.e., using *link gains* based on loss probability, is that it may have a negative impact on robustness to path failures, as previously discussed. Favoring forwarding packets through paths with low loss probabilities by setting a large value for α can make the end-hosts experience longer outage duration when the path through which a larger fraction of packets are forwarded fails. The quantitative assessment of how the *link gain* mechanism affects robustness can be easily performed, particularly in the simple case where the *level-of-robustness* of each link (i.e., link capacity in max flow computation) is equally set to c and there are n node-disjoint paths between the source and destination. To do this, we define a *cycle* as the expected time interval between two consecutive packets arriving at the destination through a given path. The cycle of a given path corresponds to the worst-case outage duration that the end-host is supposed to experience when all but one of the paths fail. The cycle is determined by the sending rate, the fraction of packets forwarded over the path, and the loss probability of the path.

Note that for a given α value, the fraction of packets allocated for a path depends on the loss probabilities of other paths as well as the loss probability of the path. Let p_i denote the packet delivery probability for path i, i.e., $p_i = \prod_{\ell:\ell\in\text{path } i}(1-p_{loss,\ell})$, where $p_{loss,\ell}$ is the loss probability for link ℓ. When reaching the destination, the flow along path i will have been amplified by $g_i = \frac{1}{\prod_{\ell:\ell\in\text{path } i}(1-p_{loss,\ell})^\alpha} = \frac{1}{{p_i}^\alpha}$. Since the capacity constraint must be satisfied, the flow allocated over path i at the source node will be $\frac{c}{g_i}$, i.e, $c{p_i}^\alpha$. (Note that setting $\alpha = 0$ is equivalent to not using link gain, i.e., the flow will be equally distributed.) Also since all the capacities are equal, the fraction of the packets forwarded over path i is $f_i = \frac{{p_i}^\alpha}{\sum_{j=1}^{n}{p_j}^\alpha}$.

Given the formula for the fraction of packets over each path, the network designer can find the smallest α to achieve the desired packet delivery ratio (i.e., $1-$loss rate). Figure 4 (a) shows an example where there are two paths between the source and destination. In this example, the loss probability of path 1 is fixed to 0 whereas the loss probability of path 2 varies from 0.05 to 0.2. Figure 4 (b) shows that the way packet delivery ratio varies with respect to α changes quite significantly depending on the loss probability of path 2.

To reckon the change of outage duration by the *link gain*, we consider the expected cycle for a path. Suppose that the sending rate at the source is x (packets/sec) and that $p_1, p_2, \ldots, p_n$ is a sequence of packet delivery probabilities sorted in a decreasing order. To obtain the expected cycle for path i, we count how many times per second we can see a packet given that we observe the path x times for each second. The number of packets being observed is binomially distributed with parameters $(x, f_i p_i)$ because each observation is independent and the probability of a packet being found in each observation is f_i times

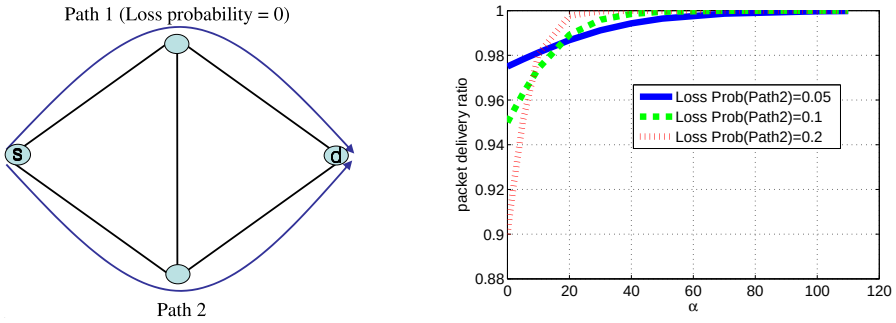


Fig. 4. α versus Packet Delivery Ratio. The *link gain* parameter, α, to achieve a certain packet delivery ratio depends on the loss probabilities of the paths.

p_i. The mean of $B(x, f_i p_i)$ is $x f_i p_i$. That is, the expected number of packets passing path i for each second is $x f_i p_i$. As a consequence the expected interval between consecutive packets passing path i, i.e. the cycle for path i is $\frac{1}{x f_i p_i} = \frac{1}{x} \frac{\sum_{j=1}^{n} {p_j}^{\alpha}}{{p_i}^{\alpha}} \frac{1}{p_i}$. Particularly, the paths with the lowest packet delivery probability (p_n) have the longest cycles, which is $\frac{\sum_{j=1}^{n} {p_j}^{\alpha}}{x {p_n}^{\alpha+1}}$. Note that setting α to 0 gives the longest cycle without using the link gain mechanism, which is $\frac{n}{x p_n}$. Therefore, when using the link gain, the worst-case outage duration is increased by the ratio of $\frac{\sum_{j=1}^{n} {p_j}^{\alpha}}{n {p_n}^{\alpha}}$.

If we want to put a limit on the outage duration when only one path survives, we can find the upper limit on α such that

$$\frac{\sum_{j=1}^{n} {p_j}^{\alpha}}{x {p_n}^{\alpha+1}} \leq t,$$

where t is a threshold for the cycle. The computation can be easily made by Newton's method given that packet delivery probabilities can be obtained through measurement.

5 Conclusions

Robustness is one of the key goals of network protocol design. In this paper the impact of proactive routing based on path diversity on end-to-end robustness is examined. A novel metric to measure robustness at many time-scales was introduced. Through experiments on PlanetLab, we found that proactive multipath routing schemes can significantly improve robustness. However, in our experiments, we observed that little additional robustness is added when more than 3 alternate paths are considered. As future work, we plan to examine why this is the case. Our conjecture is that the main cause is the fact that the underlying network's physical topology may not have enough redundancy. Finally, in order

to balance the sometimes contradictory goals of low loss and high robustness, a technique based on *link gains* was proposed to achieve a specific loss rate goal.

References

1. Savage, S., Anderson, T., Aggarwal, A., Becker, D., Cardwell, N., Collins, A., Hoffman, E., Snell, J., Vahdat, A., Voelker, G., Zahorjan, J.: Detour: Informed internet routing and transport. IEEE Micro **19**(1) (1999) 50–59
2. Andersen, D., Balakrishnan, H., Kaashoek, F., Morris, R.: Resilient overlay networks. SIGOPS Oper. Syst. Rev. **35**(5) (2001) 131–145
3. Gummadi, K.P., Madhyastha, H.V., Gribble, S.D., Levy, H.M., Wetherall, D.: Improving the reliability of Internet paths with one-hop source routing. In: Proc. 6th USENIX OSDI, San Francisco, CA (December 2004)
4. Apostolopoulos, J.G.: Reliable video communication over lossy packet networks using multiple state encoding and path diversity. In: VCIP. (2001)
5. Cha, M., Moon, S., Park, C., Shaikh, A.: Placing relay nodes for intra-domain path diversity. In: Proc. of the IEEE INFOCOM. (2006)
6. Li, Y., Zhang, Y., Qui, L., Lam, S.: Smarttunnel: Achieving reliability in the internet. In: Proc. of the IEEE INFOCOM. (2007)
7. Bohacek, S., Hespanha, J.P., Lee, J., Lim, C., Obraczka, K.: Game theoretic stochastic routing for fault tolerance and security in communication networks. IEEE Trans. on Parallel and Distributed Systems (2007)
8. Markopouloou, A., Iannaccone, G., Bhattacharyya, S., Chuah, C.: Characterization of failures in an ip backbone. In: Proc. of the IEEE INFOCOM. (2004)
9. : Planetlab. http://www.planet-lab.org
10. Blanton, E., Allman, M.: On making TCP more robust to packet reordering. ACM Computer Communications Review **32** (2002)
11. Zhang, M., Karp, B., Floyd, S., Peterson, L.: RR-TCP: A reordering-robust TCP with DSACK. In: Proceedings of IEEE INCP. (2003)
12. Bohacek, S., Hespanha, J.P., Lee, J., Lim, C., Obraczka, K.: A new tcp for persistent packet reordering. IEEE/ACM Trans. on Networking (2006) 369–382
13. Bohacek, S., Hespanha, J., Obraczka, K.: Saddle policies for secure routing in communication networks. In: Proc. of the 41st Conf. On Decision and Contr. (2002) 1416–1421
14. Cormen, T., Leiserson, C., Rivest., R.: Introduction to Algorithms. MIT Press, Boston, MA (2001)