# Loss-Optimized Routing in Overlay Networks

David Andersen, Hari Balakrishnan, Alex C. Snoeren
MIT Laboratory for Computer Science
{dga,hari,snoeren}@nms.lcs.mit.edu
http://nms.lcs.mit.edu/ron/

*Abstract*— **Path diversity exists in many overlay networks. How can we take advantage of this diversity to reduce packet loss? If bandwidth is infinite, the answer is simple—we could send a copy of the packet down every link, and hope it gets through. If we have perfect knowledge of the network state, we could always choose to send our packet down the "best" path. Since neither of these approaches is feasible, we study the trade-offs for more practical variants: Multi-path delivery with redundant encodings, and probe-based reactive overlay routing. Our measurements on a 17-node Internet testbed show that reactive routing can provide 20% reductions in the number of outages and the observed loss rates, and that 2-redundant multi-path routing can eliminates 30% of the outages while reducing standing loss rates by a factor of two.**

## I. Introduction

The routing infrastructure in the Internet does not attempt to provide a loss-free abstraction, and, as a result, derives significant benefits in terms of scalability and statelessness. End-to-end transfers observe packets losses, typically due to network congestion, routing anomalies, and packet corruption. The result is that applications and transport protocols have to cope with these packet losses; usually this involves retransmissions and congestion control, with the net effect that communication latency and throughput suffer, and transport connections are occasionally aborted as they encounter long outages lasting several minutes [1], [2].

Over the past few years, overlay networks layered on top of the Internet routing substrate have emerged as a way to provide new routing services for specific applications. Examples of overlays include Web content distribution networks, overlays for streaming media distribution, application-layer multicast networks, resilient overlay networks (RONs) for alleviating the effects of Internet path outages [2], and peer-to-peer networks (*e.g.*, Chord, CAN, Pastry, Tapestry) that provide a distributed hash table abstraction. In all these networks, nodes cooperatively route packets between each other, usually selecting paths based on domain-specific criteria.

In this paper, we consider an overlay network of $N$ nodes that cooperatively route packets for each other to reduce the effective packet loss rate observed by a transport protocol using the overlay network for data transfer. Reducing the packet loss rate observed by a transport protocol often has significant benefits: it greatly improves throughput (*e.g.*, bulk TCP transfers), reduces communication latency caused by retransmissions (*e.g.*, short Web transfers), and improves the fidelity of received data (*e.g.*, streaming media, Internet conferencing and telephony, real-time data feeds [3]) particularly where retransmissions are expensive or hard to achieve.

The traditional way to reduce the effective loss rate of a packetized data transfer is to use *packet diversity*, either through retransmissions, forward error correction (FEC), or a combination of the two. For example, if the overlay network were to itself use an ARQ protocol like TCP for forwarding packets, then no losses would be observed by the transport protocol using the overlay network. However, this choice would cause end-to-end latencies for some packets to be dramatically larger than the normal RTT, which may not be appropriate for the applications mentioned above. Inspired by these applications, we focus on loss-optimized routing strategies that do not dramatically increase end-to-end round-trip latencies; specifically, we restrict our attention to schemes whose latency overhead is at most the one-way latency.

Overlay networks offer a new degree of freedom—*path diversity*—for loss-optimized data delivery. By taking advantage of the existence of multiple paths, nodes in the overlay network can now select paths that optimize packet loss rates. This strategy can prove beneficial if the underlying Internet paths between different nodes in the overlay don't share common points of outage or congestion. In this paper, we address the question of how best to take advantage of path diversity to reduce overall packet loss rates for unicast traffic, without adversely increasing end-to-end latency.

If bandwidth were plentiful, the solution to this problem is "obvious"—simply transmit each packet along as many different disjoint paths in the overlay as possible (*e.g.*, via all possible one-hop intermediate nodes between a source and destination node). In practice, however, there is a trade-off between how much excess bandwidth is used by any strategy for improving loss-resilience across an overlay network, and how much improved loss-resilience it achieves. This paper studies the trade-off between observed loss rate improvement and the bandwidth overhead, for two radically different strategies for taking advantage of path diversity—*probe-based overlay routing* and *multi-path FEC*.

Probe-based overlay routing is a reactive strategy where the overlay nodes constantly probe the $O(N^2)$ paths between them, and send packets either directly over the Internet, or forward it via a sequence of other nodes in the overlay when the latter path provides better performance. The overhead in this approach is due to the probes, which periodically estimate the quality of several alternate paths. These probes are required to ensure that when a problem occurs with the current path or when a better path presents itself, traffic is rerouted appropriately to reduce the observed loss rate. Inspired by the approach used in RON [2], we focus on a simple (but effective) overlay routing method that uses at most one intermediate node in the overlay network to forward packets.

Multi-path FEC applies a (simple) packet-level FEC code on the data stream by adding redundancy packets to the stream. It then splits the resulting encoded stream across two (or more) paths. The overhead here is due to the redundant packets, but the scheme does not require any probes provided "reasonable" multiple paths are known in advance (or a random overlay path is picked in addition to the direct Internet path). Inspired by recent work on mesh-based routing [3], we pay special attention to a particularly simple instance of the general idea, where the FEC scheme is simple packet replication and all packets are sent along two different paths in the overlay.

Our goal is not to design the most efficient routing protocol, but to compare these two different alternatives based on real-world Internet experiments. To do this, we use seventeen geographically diverse nodes of the RON testbed overlay network spread across three continents. We use a path probing process that allows us to evaluate the performance of these two strategies and compare their loss rate reduction to the direct Internet path connecting every node pair. We notice that losses occur both in isolated fashion (likely due to transient congestion), and long bursts, often several seconds to minutes (likely due to persistent congestion, route failures, and path outages).

We find that periods of high loss rate ($>$10%) are a dominating source of loss in our experiments. Probe-based routing is able to mitigate about 20% of these outages, and packet replication mitigates about 30% of them. Probe-based routing is ineffective at reducing the already-low loss rates observed on most Internet paths, but packet replication can reduce effective loss by more than a factor of two during "normal" conditions.

The remainder of this paper is organized as follows. Section II surveys related work. In Section III we discuss the design of a simple probe-based overlay routing protocol and a replication-based simple multi-path FEC protocol. Section IV presents our experimental results. We conclude in Section V with a summary of our findings.

## II. Background and Related Work

The Internet was designed as a best-effort medium; hence, it is not surprising that Internet paths often exhibit packet loss. While congested routers, packet collisions, and noisy data links combine to cause various levels of packet loss, researchers have observed that severe burst losses or *outages* may be exacerbated by link failures, routing problems, or both. Labovitz *et al.* shows that routers may take tens of minutes to stabilize after a fault, and that packet loss is high during this period [1]. They also note that route availability is not perfect, causing sites to be unreachable some fraction of the time [4]. Paxson notes that packets are often subject to routing loops and other pathologies [5].

### A. Reliable transmission

Several systems use retransmission schemes to recover from packet loss, such as TCP's acknowledgment (ACK) based automatic repeat request (ARQ) scheme. By adaptively adjusting their redundancy rates, ARQ schemes attempt to approach the information rate of the channel but introduce a potentially unbounded delay. End-to-end ARQ schemes introduce at least a RTT delay for lost packets; hop-by-hop ARQ schemes can reduce the delay for certain topologies [6], but require buffering at intermediate nodes. Furthermore, most ARQ schemes are tuned for certain loss characteristics, and function poorly over channels outside of their design space (*e.g.*, TCP is known to break down at loss rates greater than about 30% [7])). Due to ARQ schemes' inability to provide latency guarantees in general, we do not consider them further in this paper.

In contrast, FEC adds redundant information to a stream, allowing the stream to be reconstructed at the receiver even if some of the information is corrupted or missing [8]. They are widely used in wireless systems to protect against bit corruption [9], and more recently in multicast and content distribution systems to protect against packet loss [10], [11]. The latter applications require *packet-level*, rather than bit-level FEC, and these is the approach we consider in this paper.

### *B. Internet Performance Studies*

Unfortunately, sending redundant data along the same Internet path is often not as effective as one might hope due to high packet loss correlation. Bolot examined the behavior of packet trains on a single link between INRIA and the University of Maryland in 1992 [12]. He found that the conditional loss probability of back-to-back packets is high when the packets are closely spaced ($\sim$ 8 ms), but returns to normal when the gap is $\sim$ 500 ms. Similarly, Paxson examined TCP bulk transfers between 35 sites in 1997 [13]. In this work, he found that the conditional loss probability of data packets that were queued together was 10–20 times higher than the base loss rate. We compare our loss probabilities with those of Paxson and Bolot in Section IV.

### *C. Improved Routing*

While ARQ and FEC schemes can improve the effective loss rate of a particular Internet path, there may exist alternative paths that provide lower loss rates. Early ARPANET routing attempted to optimize path selection for congestion [14], but this was removed for scalability and stability reasons. Today, a wide variety of traffic engineering approaches are employed to refine path selection in an attempt to decrease congestion, packet loss, and latency, and increase available bandwidth [15]. Unfortunately these techniques generally operate over long timescales. As a result of current backbone routing's ignorance of short-term network conditions, the route taken by packets is frequently sub-optimal, a fact noted by the Detour study [16].

Recent research in overlay networks has attempted to improve path conditions through indirect routing. The RON project uses active measurements to take advantage of some of these alternate paths [2]. Various Content Delivery Networks (CDNs) use overlay techniques and caching to improve the performance of content delivery for specific applications such as HTTP and streaming video. Overcast and other application level multicast projects attempt to optimize routes for bandwidth or latency [17].

### *D. Multi-path routing*

The success of traffic engineering and overlay routing indicates the presence of redundant routes between many pairs of Internet hosts. A variety of approaches have been developed to leverage the existence of multiple, simultaneous paths through multi-path routing. Dispersity routing [18] and IDA [19] split the transfer of information over multiple network paths to provide enhanced reliability and performance. Simulation results and analytic studies have shown the benefits of this approach [20], [21]. Chen evaluated the use of parallel TCP flows to improve performance, but did not examine failures, or real Internet paths [22]. In addition, researchers have suggested combining redundant coding with dispersity routing to improve the reliability and performance of both parallel downloads [23] and multicast communication [3]. Akamai is reported to use erasure codes to take advantage of multiple paths between sites [24], and the designers of the Opus overlay system have proposed the future use of redundant transmission in an overlay, but, to our knowledge, have not yet evaluated this technique [25].

Unfortunately, most multi-path routing schemes make assumptions about path diversity that may not hold when considering typical Internet paths. IP networks are not constructed with the degree of independence required to achieve the theoretical gains that redundant routing could deliver. Single-homed hosts share the same last-mile link to their provider, creating an obvious shared bottleneck and non-independent failure point. Even multi-homed hosts may have unexpected sources of shared failures. Many providers have some degree of shared physical infrastructure. In 2001, a single train derailment in the Howard Street tunnel in Baltimore, MD, impacted Internet service for at least 4 major US backbone carriers, all of whom used fiber running through the same physical location [26]. Network failures are not only caused by exogenous factors, but may be the result of network traffic or other failures. These cascading logical failures can cause widespread outages that affect multiple paths or providers [27]. Finally, denial of service attacks or other global Internet problems such as worms and viruses can cause correlated, concurrent failures. For instance, the "Code Red" worm, as a side-effect of its scanning, could crash certain Cisco DSL routers and other products, causing correlated failures based solely upon network access technology [28]. We provide an empirical evaluation of the independence of a particular set of Internet paths in Section IV.

## III. Design

In this section, we consider two enhanced mechanisms for packet routing: Probe-based reactive overlay routing, and multi-path redundant routing. These techniques probably would not be used completely alone. For instance, it's necessary to choose which intermediate nodes to use in redundant routing, and the logical way in which to do so is based on network measurement. The difference is one of degree and focus—redundant routing would likely use a very low probing rate, and rely upon duplication to combat losses. We examine the utility of combining redundant routing and active probing in our evaluation.

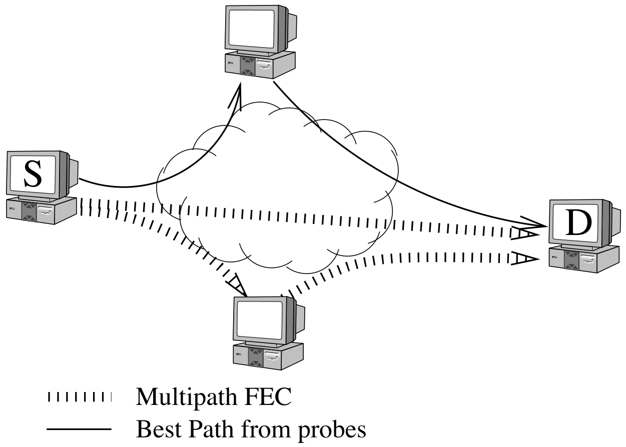

Fig. 1. Multi-Path FEC routing and best path routing. In this diagram, probing has determined that the best path is to travel indirectly via the top node. Multi-Path FEC routing sends a packet down the direct path and via a random alternate hop, in this case, the bottom node.

An application's requirements from the network are similar to Service-Level Agreements (SLAs) provide by ISPs. Applications need a certain average loss rate, average latency, maximum latency, and a maximum amount of outage time. In this work, we consider applications whose maximum latency is less than one round-trip time; in short, applications that cannot afford retransmission.

The underlying network has loss rates, and loss rate variance, but also has different loss mechanics (burst losses vs. random losses, for instance). The latency of the network is affected both by routing and by traffic. Poor routing choices in underlying protocols can result in paths with excessive latency [2], [16]. Congestion on links causes variable queuing delays.

## A. *Probe-based Reactive Overlay Routing*

RON-like systems send frequent probes to determine the availability, latency, and loss rate of the paths connecting the nodes in the overlay. A reactive routing scheme must choose a probing rate $R$, and a network size $N$. A generalized scheme would also need to choose the sets of nodes that probe each other. In this work, we consider only $N^2$ all-pairs probing. Higher probing rates permit quicker reaction to network change, with more overhead. Larger networks have more paths to explore, but create scaling problems. In the system we evaluate, every node probes every other node once every 15 seconds. When a probe is lost, the node sends an additional string of up to 4 probes spaced 1 second apart, to determine if the remote host is down. The paths are selected based upon the average loss rate over the last 100 probes. These are the same parameters used in an earlier evaluation of reactive overlay routing [2].

Reactive routing assumes that *some* path through the overlay can provide good service; FEC and redundant routing attempt to construct a good path out of only bad paths. This creates clear differences between the failure scenarios that these methods can handle. If no individual paths are good, reactive routing does not help. On the other hand, if failures result in only a small subset of functional paths through the network, a probe-based reactive mechanism is better positioned to utilize these paths.

**Benefit:** Reactive routing circumvents path failures in time proportional to its probing rate. For the $N$ possible one-hop paths from a source to the destination, where each has a loss probability $p_i$,

$$p_{reactive} = min_i\,(p_i)$$

Reactive routing is constrained to the latency of the best path, as well. In general, the path with the best loss rate may not have the best latency [2].

**Cost:** The cost of all-paths probing and route dissemination is fixed—each host must send and receive $O(N^2)$ data. The cost is not dependent upon the amount of traffic in the flow; hence, it can be large in comparison to a thin data stream, or negligible when used in conjunction with a high bandwidth stream.

## B. *Redundant Multi-Path Routing*

Redundant multi-path routing sends redundant data down multiple, ideally independent, paths, such that a certain fraction of lost packets can be recovered. The designer must specify a redundancy rate, $R$, and a window size $W$ over which the redundant packets are transmitted. The redundancy rate is the fraction of lost packets that can be tolerated. The window size is the number of packets between which the redundant data is split. Larger values of $W$ permit more flexible choices of $R$, but at the cost of increased recovery latency.

Redundant encoding is generally accomplished through the use of a forward error correction (FEC) technique that adds extra packets to the data stream, rather than increasing the size of individual packets, since increasing the packet size may bump into path MTU limitations. An efficient FEC sends the original packets first, to avoid adding latency in the no-loss case—so called *standard codes*. Reed-Solomon erasure codes are a standard

FEC method that provide a framework in which to pick fractional values of $R$ [11]. As a simpler case, however, for $R = W = 1$, packets can simply be duplicated and sent along multiple paths. This is the process used in mesh routing [3]. We restrict our evaluation to using this simple encoding over two paths, so-called 2-redundant routing, since we believe the number of truly loss-independent paths between two points on the Internet is relatively low [1]. FEC without path diversity can avoid random losses, but cannot tolerate large burst losses or path failures.

**Benefit:** When paths are completely independent, redundant routing can handle the complete failure of up to $(R-1)$ paths per node. When packet losses are independent, redundant routing on $N$ paths whose loss probability is $p_i$ can improve the overall loss rate to the product of their individual loss rates

$$p_{redundant} = \prod_{i=1}^{N} p_i$$

2-redundant routing on random paths achieves, in expectation, the square of the average loss rate:

$$E\left[p_{2-redundant}\right] = \left(E\left[p_i\right]\right)^2$$

When used in conjunction with the direct Internet path, multi-path routing has good expected latency. If the alternate paths have similar latency, then multi-path routing can provide a smaller expected latency [3], while still providing reduced loss.

**Cost:** The cost of simplistic $N$-redundant routing is a factor of $N$. A 2-redundant routing scheme results in a doubling of the amount of traffic sent. The cost does not depend on the size of the network.

### *C. Design Space and Internet Limitations*

There are some situations where redundant routing is not appropriate. Running an unmodified bulk-flow TCP directly over a redundantly-enhanced path would be problematic, because the apparent low loss rate will trick TCP into taking far more than its fair share of the bandwidth. However, running low-rate TCPs (or any application where it's known that the application won't exceed its share of the channel) should be fine.

In general, reactive overlay routing is appropriate for any kind of traffic, though its overhead may be extremely large for low-bandwidth flows. For low-bandwidth flows, redundant approaches can offer good benefits with low overhead. For high-bandwidth flows, FEC approaches result in overhead proportional to the size of the flow, whereas alternate-path routing has constant overhead.

| Probe-based | 2-Redundant Multi-Path |
|---|---|
| $1 + \frac{N^2}{Bandwidth}$ | 2 |

TABLE I

OVERHEAD AS A FUNCTION OF THE FLOW'S BANDWIDTH. PROBE-BASED OVERLAY ROUTING IMPOSES OVERHEAD THAT GROWS RAPIDLY WITH NETWORK SIZE, BUT DOES NOT DEPEND ON FLOW SIZE. 2-REDUNDANT MULTI-PATH ROUTING REQUIRES TWICE THE BANDWIDTH.

Table I lists the overhead for each scheme. The total bandwidth consumed is the product of the overhead and the flow bandwidth. The overhead of each method grows with a different parameter (network size vs. flow size), which suggests the domains in which each would be more appropriate.

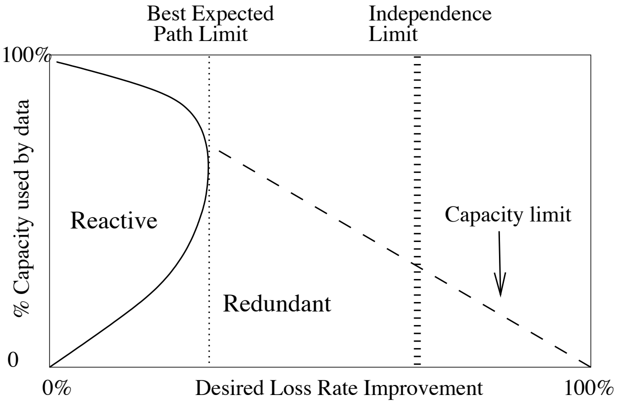

Fig. 2. When to use reactive or redundant routing. Reactive routing asymptotically approaches the performance of the best expected path. Its probes require some network bandwidth. Redundant routing is limited by the capacity of the network relative to the flow bandwidth, and by the degree of independence of its paths. Within these bounds, the flow's bandwidth determines whether reactive or redundant routing provides the required improvement with smaller overhead.

To understand the space in which each method is applicable, we compare two parameters, the desired improvement in loss rate, vs. the amount of the network bandwidth being used by the original data stream. We define "loss rate improvement" as

$$\frac{Loss_{Internet} - Loss_{method}}{Loss_{Internet}}$$

Figure 2 depicts this space graphically, and shows the bounds that limit the performance of each scheme. We consider three major bounds.

[1] For example, if there are three such paths, any redundant encoding must be able to tolerate a loss of at least $\frac{1}{3}$ of the packets in a window, which would require at least 50% overhead anyway.

| Dataset | Samples | Dates |
|---|---|---|
| $RON_{narrow}$ | 4,763,082 | 8 Jul 2002 – 11 Jul 2002 |
| $RON_{wide}$ | 2,875,431 | 3 Jul 2002 – 8 Jul 2002 |

TABLE II
THE TWO DATASETS USED IN OUR EXPERIMENTS. THE NARROW DATASET CONTAINS ONE-WAY SAMPLES FOR THREE ROUTING METHODS. THE WIDE DATASET HAS ROUND-TRIP SAMPLES FOR ELEVEN METHODS.

**Best Expected Path Limit**: Probing can only find the best network path at any given time; its performance is limited to this amount. As the probing frequency increases, the achieved performance asymptotically approaches the performance of the best expected path.

**Capacity Limit**: Both schemes face a capacity limit. If the original data stream is using 100% of the available capacity, neither scheme can make an improvement: Probing cannot send probes, and redundant routing cannot duplicate packets. The bandwidth required by redundant routing is linear with the flow rate. The "constant" bandwidth required by reactive routing decreases slightly with a relaxation in loss rate demands, because when less improvement is required, the probing rate can be reduced.

**Independence Limit**: Finally, redundant routing is limited by the loss and failure independence of the network. The actual values of this limit are unknown. We use an Internet-based evaluation to empirically determine the expected path limit and the independence limit with a fixed probe rate and low flow bandwidth.

## IV. EVALUATION

We evaluate packet duplication and reactive routing on a deployed Internet testbed. Table III lists the 17 hosts used in our experiments. The hosts are concentrated in the US, but span 5 countries on 3 continents. More importantly, the testbed hosts have a variety of access link technologies, from OC3s to cable modems and DSL links. While we don't claim that this testbed is representative of the Internet as a whole, the 272 distinct one-way paths that we can measure between the hosts do provide a diverse playground in which to evaluate various routing mechanisms.

Table IV lists the two datasets we examine. $RON_{narrow}$ measures the three most promising methods with frequent one-way probes, sampling each path (for each method) every 45 seconds on average. $RON_{wide}$ more combinations, at a lower probing frequency, to obtain a broad picture of the landscape.[2]

| Name | Location | Description |
|---|---|---|
| `Aros` | Salt Lake City, UT | ISP |
| `CCI` | Salt Lake City, UT | .com |
| `Cisco-MA` | Waltham, MA | .com |
| `* CMU` | Pittsburgh, PA | .edu |
| `* Cornell` | Ithaca, NY | .edu |
| `Greece` | Athens, Greece | .edu |
| `Korea` | KAIST in Korea | .edu |
| `Lulea` | Lulea, Sweden | .edu |
| `MA-Cable` | Cambridge, MA | AT&T Cable |
| `Mazu` | Boston, MA | .com |
| `* MIT` | Cambridge, MA | .edu |
| `CA-DSL` | Foster City, CA | 1Mbps DSL |
| `NC-Cable` | Durham, NC | RoadRunner Cable |
| `* NYU` | New York, NY | .edu |
| `PDI` | Palo Alto, CA | .com |
| `* Utah` | Salt Lake City, UT | .edu |
| `VU-NL` | Amsterdam, Netherlands | Vrije Univ. |

TABLE III
THE HOSTS BETWEEN WHICH WE MEASURED NETWORK CONNECTIVITY. ASTERISKS INDICATE U.S. UNIVERSITIES ON THE INTERNET2 BACKBONE.

We focus primarily on the $RON_{narrow}$ dataset, as it studies the three most promising methods:

- *Loss*: Probe-based reactive routing that attepts to minimize loss. Requires only probing overhead.
- *Direct Rand*: 2-redundant multi-path routing, without requiring any probing overhead. One copy of each packet is transmitted on the direct Internet path; the second over a random indirect overlay path. From the first packet in this train, we can also infer the behavior of `direct` packets.
- *Lat Loss*: Probe-based 2-redundant multi-path routing. In theory, this combination should be able to achieve the best of both worlds. It sends the first copy of each packet over a path selected to minimize loss, and the second over a path selected to minimize latency. We also use this to infer the `lat` packet.

We present three major findings. First, **outage avoidance is critical.** Most losses happen during longer periods of high loss, which is precisely when a user needs extra reliability. Second, **both probe-based and redundant routing avoid many outages (periods with $>$10% loss)**—about 20% and 30% of them, respectively, depending on the tolerable loss threshold. Finally, **only redundant routing improves low loss rates**. It reduces effective loss by a factor of two, whereas probe-based routing makes almost no difference. In the following sections, we explain how we collected our data, and present a detailed analysis thereof.

[2] Our datasets will be available online when we finalize the paper.

### A. Method

Each node periodically initiates probes to other nodes. A probe consists of a `request` packet from the initiator to the target. In the second batch of experiments, the target sends a `reply` packet, so we can verify the round-trip time. Each probe has a random 64-bit identifier, which the hosts log along with the time at which packets were both sent and received. This allows us to compute the one-way reachability between the hosts. Most, but not all, hosts had GPS-synchronized clocks, but to avoid time-keeping problems, we focus on round-trip times, or one-way latency summaries that average out possible errors. Each probing host periodically pushes its logs to a central monitoring machine, where this data is aggregated. Our post-processing finds all probes received within 1 hour of when they were sent. We disregard probes to hosts during host failures; our numbers only reflect failures that affected the network, while leaving hosts running. It is possible that we slightly under count outages caused by power failures or other events that affect both host and network.

### B. Base Network Statistics

In contrast with earlier studies, the paths we measured were relatively low-loss. Our paths' average loss rates ranged from 0% on many Internet2 or otherwise very fast connections, to about 6% between Korea and a US DSL line. Figure 3 shows the distribution of average loss rates (over several days) on a per-path basis. The overall loss rate we observed on directly-sent single packets was 0.74%, similar to what we observed in earlier experiments on the RON testbed [2], but lower than the loss rates observed 5 years ago by Paxson. Most of the time, the 10-minute average loss rate was close to zero; the `direct` line in Figure 5 shows the distribution of 10-minute loss samples. Over 95% of the samples had a 0% loss rate. The sampling granularity for the CDF is relatively coarse, so it groups low loss-rate conditions into the zero percent bin.

One host, `CA-DSL`, experienced persistent congestion during all of our experiments. This host was responsible for a disproportionate quantity of the losses—36%—that we observed over several days of monitoring. We therefore present outage and loss numbers for both the whole network, and with this host removed.

### C. Avoiding Long Outages

An extended period of high loss is extremely deleterious to application performance. Because different applications may fail at different loss rates, or different lengths of outages, we define $outage(\tau, p) = 1$ if the observed packet loss rate averaged over an interval $\tau$ is larger than $p$ on the path, and 0 otherwise. A small reduction in the overall loss rate could result from finding insignificantly better paths all of the time, or it could result from finding much better paths when it really counts. During our experiments, $\frac{2}{3}$ of the lost packets were lost during periods of 10% or greater loss, which emphasizes the importance of avoiding major network problems.

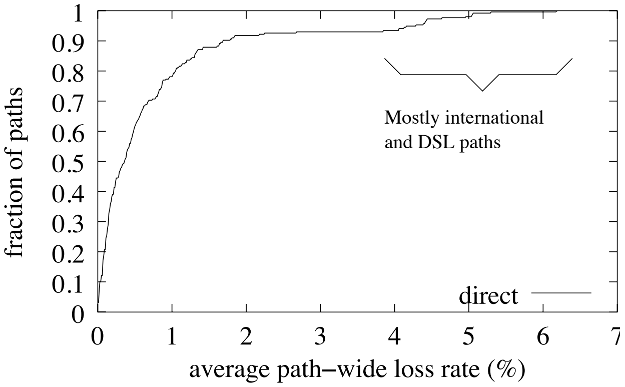

Fig. 3. Cumulative distribution of long-term loss rates, on a per-path basis. 80% of the paths we measured have an average loss rate less than 1%.

| Loss % | direct | loss | direct rand | lat loss |
|---|---|---|---|---|
| $> 0$ | 5907 | 5631 | 2589 | 2503 |
| $> 10$ | 2397 | 2172 | 1268 | 1195 |
| $> 20$ | 900 | 820 | 611 | 602 |
| $> 30$ | 534 | 430 | 370 | 341 |
| $> 40$ | 311 | 252 | 214 | 201 |
| $> 50$ | 171 | 150 | 124 | 123 |
| $> 60$ | 129 | 101 | 96 | 97 |
| $> 70$ | 85 | 69 | 66 | 68 |
| $> 80$ | 43 | 43 | 33 | 38 |
| $> 90$ | 12 | 9 | 7 | 11 |

TABLE IV

For a given loss rate, the number of 10-minute periods in which this rate occurred on one of the $N^2$ paths between the measurement hosts. There were about 119,000 such intervals in the dataset; the no-loss case is omitted for clarity.

To evaluate the routing methods performance with sustained high loss, we first look at $outage(10 \text{ minutes}, p)$ for varying loss rates $p$. On the 272 distinct paths between our hosts, over 3 days, there were 119,100 different 10-minute time-bins. Our major results are shown in Tables IV and V, which list the number of 10-minute periods of high-loss for each path. For example, disregarding the DSL host, Table V tells us that there were 2410

| Loss % | direct | loss | direct rand | lat loss |
|---|---|---|---|---|
| $> 10$ | 1541 | 1283 | 1017 | 948 |
| $> 20$ | 701 | 625 | 508 | 507 |
| $> 30$ | 432 | 345 | 298 | 282 |
| $> 40$ | 241 | 196 | 157 | 152 |
| $> 50$ | 120 | 112 | 77 | 82 |
| $> 60$ | 90 | 71 | 55 | 56 |
| $> 70$ | 49 | 41 | 31 | 36 |
| $> 80$ | 20 | 22 | 13 | 15 |
| $> 90$ | 5 | 3 | 3 | 3 |

TABLE V

The outage statistics without the congested host.

| | All hosts | | Without DSL | |
|---|---|---|---|---|
| **Method** | **Outage** | **Normal** | **Outage** | **Normal** |
| Direct | 7854 | 3279 | 6405 | 1660 |
| Loss | 6914 | 3272 | 4523 | 1600 |
| Direct Rand | 4574 | 1227 | 3618 | 666 |
| Lat Loss | 4448 | 1228 | 3468 | 605 |

TABLE VI

Packets lost during periods of high loss (10% or greater). **Outage** is the number lost during such periods, and **Normal** is the number lost outside of high-loss periods. The first two columns contain the results for all hosts, the latter two columns present the same numbers with the congested DSL host removed.

| Type | 1lp | 2lp | totlp | ulp | clp | Lat |
|---|---|---|---|---|---|---|
| direct* | 0.74 | - | 0.74 | 0.74 | - | 69.54 |
| lat* | 0.75 | - | 0.75 | 0.75 | - | 69.43 |
| loss | 0.67 | - | 0.67 | 0.67 | - | 76.07 |
| direct rand | 0.74 | 1.85 | 0.38 | 1.30 | 51.17 | 68.33 |
| lat loss | 0.75 | 1.53 | 0.37 | 1.14 | 49.82 | 66.73 |

TABLE VII

One-way loss percentages. Items marked with an asterisk were inferred from the first packet of a two-packet train. **1lp** and **2lp** are the probabilities of losing the first and second packets. **totlp** is the probability of losing both. **ulp** is the overall probability of losing either packet. **clp** is the conditional loss probability for the second packet. **Lat** is the average one-way latency.

| Type | 1lp | 2lp | totlp | ulp | clp | Lat |
|---|---|---|---|---|---|---|
| direct* | 0.52 | - | 0.52 | 0.52 | - | 66.51 |
| lat* | 0.52 | - | 0.52 | 0.52 | - | 65.78 |
| loss | 0.44 | - | 0.44 | 0.44 | - | 72.29 |
| direct rand | 0.52 | 1.71 | 0.31 | 1.12 | 58.23 | 65.12 |
| lat loss | 0.52 | 1.38 | 0.29 | 0.95 | 55.53 | 63.24 |

TABLE VIII

One-way loss percentages without the congested DSL host.

“path minutes” (241 intervals of 10 minutes each) where the loss rate between two hosts on the direct Internet was over 40%. Probing reduced this to 1960 path-minutes, and redundant routing reduced it to 770 path-minutes.

We observe that redundant routing is particularly effective at reducing the loss rate when failures are either brief relative to the probe time, or during periods of sustained, low loss rate. For more severe, lasting outages, redundant and reactive routing both offer viable solutions. Table IV-C shows the number of packets lost for each routing method during high-loss periods, still at $\tau = 10$ minutes. This data confirms that confirms that probing can avoid high-loss situations (from the reduction in during-outage losses), but has almost no effect during low-loss situations.

Figure 4 explores the dependence on the length of the outage, for $outage(\tau, \{10\%, 30\%, 50\%, 80\%\})$. For these four distinct loss rates, it shows the number of outages as $\tau$ increases from 10 minutes to 60 minutes. The number of outages decreases as the $\tau$ increases, but the proportions of outages corrected seems to remain the same between the different methods, except at very high loss rates, where all of the alternate-hop methods converge. *Extreme* loss appears to be caused by actual outages that either all schemes can route around, or none can, but lower levels of loss may be caused by both outages that can be routed around, and by congestion.

Tables VII and VIII examine the overall loss and latency results. These tables provide an explanation for the limited improvement during low-loss periods: the overall loss rate, outside of particularly bad periods, was quite low. The 10% overall reduction in loss from reactive routing came primarily from avoiding large outages. On the other hand, redundant routing shines during low, stationary loss periods, and it reduced the overall loss rate by half. Figure 5 shows the distribution of these 10-minute losses; note the large distinction between redundant and non-redundant methods at low loss rates.

### *D. Conditional Loss Probabilities*

In the $RON_{wide}$ dataset, we examined a wider number of probe types, including back-to-back `direct` packets.

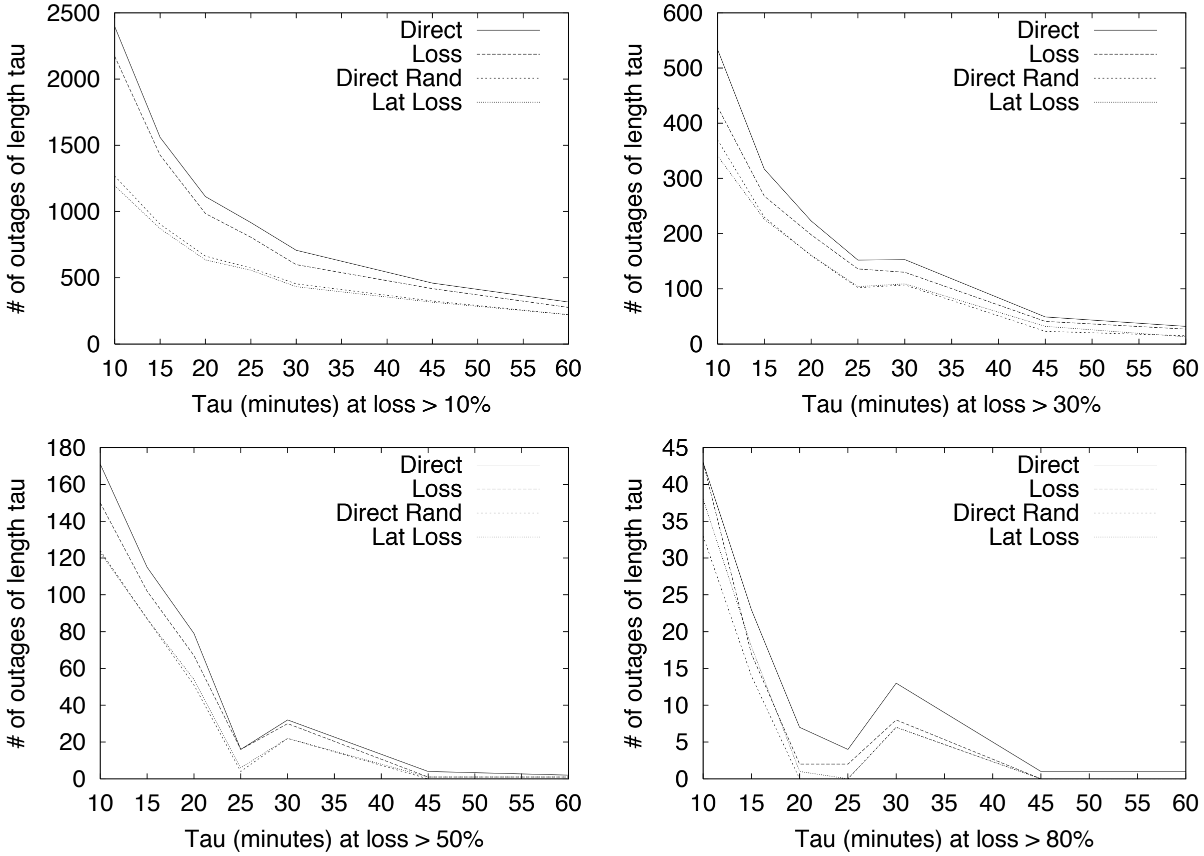

Fig. 4. The number of outages at particular loss rates (10%, 30%, 50%, and 80%) when varying the length of time over which the loss rate must persist. Note that the Y axis is different for each figure, as there are fewer outages of longer duration. This data is for all hosts, including the congested DSL link.

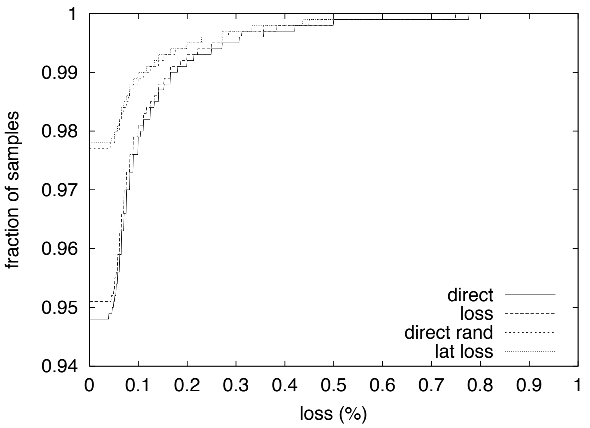

Fig. 5. Cumulative distribution of 10-minute loss rates, per-path.

Bolot [12] examined packets separated by 8ms, and found that their conditional loss probability was 60%. Paxson [13] examined TCP packets that queued together at a router, finding their conditional loss probability to be about 50%. In our experiments, back-to-back packets had a higher conditional loss probability–about 73%, probably because we sent them with *no* intervening delay. The conditional loss probability of a packet sent through an intermediate was only 50%. Taken relative to two direct packets, this indicates an appreciable difference in conditional loss probabilities when traversing an intermediate host. Figure 6 shows the distribution of cumulative loss probabilities across hosts, on the 115 paths on which we observed first-packet losses. With doubly-direct packets, more than half of the hosts had a 100% conditional loss probability. This data suggests that redundant routing on the same path is likely to fall prey to burst losses in a way that multi-path avoids [3].

### *E. Other combinations of methods*

Our broader examination confirmed that the three routing methods upon which we focused—`loss, direct rand,` and `lat loss`—are the most interesting. Some other methods had a few noteworthy features, however. The loss probability for `rand rand` was as low as `direct rand`, though its latency was far worse. The latency of `direct lat` was better than any other method,

[3] These numbers are derived from relatively few losses, so there are likely excessive samples at 100% that should be in the 90s.

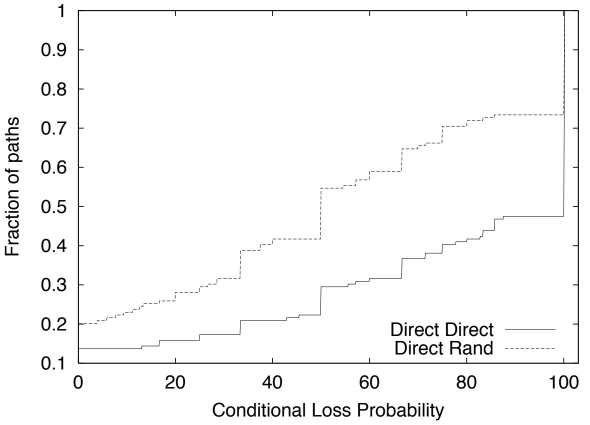

Fig. 6. Cumulative distribution of conditional loss probabilities for the second packet in a back-to-back packet transmission. Two back-to-back direct packets have a higher CLP than two back to back packets where one is sent through a random intermediate. Note that the highest average loss rate for a direct path was 6%; the conditional loss probabilities are much higher.

| Type | 1lp | 2lp | totlp | ulp | clp | RTT |
|---|---|---|---|---|---|---|
| direct | 0.27 | - | 0.27 | 0.27 | - | 133.5 |
| rand | 1.12 | - | 1.12 | 1.12 | - | 283.0 |
| lat | 0.34 | - | 0.34 | 0.34 | - | 137.0 |
| loss | 0.21 | - | 0.21 | 0.21 | - | 151.9 |
| direct direct | 0.29 | 0.49 | 0.21 | 0.39 | 72.7 | 134.3 |
| direct rand | 0.29 | 1.20 | 0.12 | 0.75 | 39.2 | 130.1 |
| direct lat | 0.29 | 0.95 | 0.11 | 0.62 | 39.3 | 123.9 |
| direct loss | 0.27 | 1.06 | 0.11 | 0.66 | 40.0 | 130.5 |
| rand rand | 1.08 | 1.12 | 0.12 | 1.10 | 11.2 | 182.9 |
| rand lat | 1.15 | 0.41 | 0.11 | 0.78 | 9.3 | 131.3 |
| rand loss | 1.11 | 0.28 | 0.11 | 0.69 | 9.9 | 140.4 |
| lat loss | 0.36 | 0.79 | 0.10 | 0.58 | 29.0 | 128.8 |

TABLE IX

ONE-WAY LOSS PERCENTAGAGES FOR THE EXPANDED SET OF ROUTING SCHEMES. THIS TABLE PRESENTS ROUND-TRIP LATENCY NUMBERS, NOT ONE-WAY LATENCY NUMBERS.

by several miliseconds. Table IX shows the results of this more broad examination.

### *F. Effects on Latency*

Figure 7 shows the cumulative distribution of one-way latencies in $RON_{narrow}$. The direct Internet path latency is 66.5ms. Latency optimized overlay routing shaves off about 1ms overall (by reducing the latency of a few poorly performing paths). Loss optimized overlay routing *increases* the latency, because the lower-loss indirect paths may take a longer amount of time. The multipath schemes improve the latency a bit more, with `lat loss` resulting in a 3ms savings relative to the direct path. Loss-based overlay routing actually increases the standard deviation of the latency, but multipath routing *decreases* the variation, from 112ms mean standard deviation to 100ms.

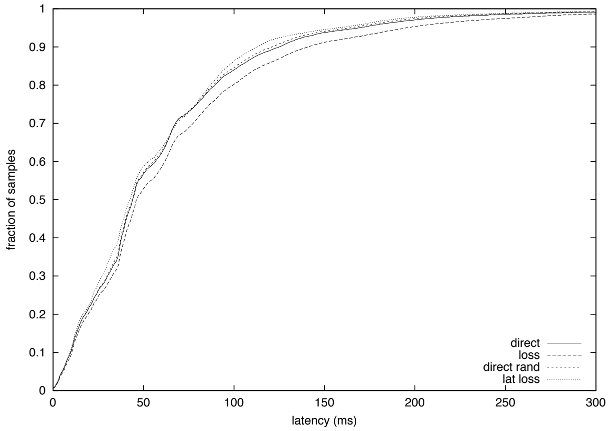

Fig. 7. Cumulative distribution of packet one-way latencies.

## V. CONCLUSION AND FUTURE WORK

Overlay routing is an increasingly popular way to deploy new Internet services. In this paper, we have examined two techniques that reduce end-to-end loss rates by leveraging *path diversity* in the underlying network. Probe-based reactive overlay routing takes advantage of this dispersity by trying to find the best path among its nodes. 2-Redundant multi-path routing sends a duplicate copy of packets along an alternate path, in the hope that the paths will fail independently.

We find that probe-based routing and redundant routing both have niches in which they excel. With its constant degree of overhead, reactive routing can provide cheap benefits for high-throughput flows in small networks, avoiding 20% of the outages that cause the most degraded network performance. Low-bandwidth flows can obtain even greater benefits from using redundant routing. For the cost of doubling their small bandwidth requirements, these flows can reduce their loss rate by a factor of two, and avoid 30% of crippling outages. These methods can act in concert to provide slight additional gains in loss rate and latency.

Our evaluation shows that there is a reasonable—but not total—degree of loss and failure independence in the underlying Internet links. This work is only a first step; a deeper understanding of this independence, on more links and on shorter and longer timescales, would facilitate the design of highly-reliable systems on the Internet. The independence we observed is on *in situ* networks that were not necessarily designed for resilience. We do not yet know what degree of failure independence can be achieved at reasonable cost when actually creating such a network, but it's an interesting question for future study.

## References

[1] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian, "Delayed Internet Routing Convergence," in *Proc. ACM SIGCOMM*, Stockholm, Sweden, September 2000, pp. 175–187.

[2] D. Andersen, H. Balakrishnan, M. Kaashoek, and R. Morris, "Resilient Overlay Networks," in *Proc. 18th ACM SOSP*, Banff, Canada, Oct. 2001, pp. 131–145.

[3] Alex C. Snoeren, Kenneth Conley, and David K. Gifford, "Mesh-based content routing using XML," in *Proc. 18th ACM SOSP*, Banff, Canada, Oct. 2001, pp. 160–173.

[4] C. Labovitz, R. Malan, and F. Jahanian, "Internet Routing Instability," *IEEE/ACM Transactions on Networking*, vol. 6, no. 5, pp. 515–526, 1998.

[5] V. Paxson, "End-to-End Routing Behavior in the Internet," in *Proc. ACM SIGCOMM '96*, Stanford, CA, Aug. 1996, pp. 25–38.

[6] H. Balakrishnan, S. Seshan, and R.H. Katz, "Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks," *ACM Wireless Networks*, vol. 1, no. 4, Dec. 1995.

[7] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP Throughput: A Simple Model and its Empirical Validation," in *Proc. ACM SIGCOMM*, Vancouver, Canada, September 1998, pp. 303–323.

[8] Robert G. Gallager, *Low-Density Parity-Check Codes*, Ph.D. thesis, Massachusetts Institute of Technology, 1963.

[9] A. J. McAuley, "Error Control for Messaging Applications in a Wireless Environment," in *Proc. INFOCOM Conf.*, Apr. 1995.

[10] John W. Byers, Michael Luby, Michael Mitzenmacher, and Asutosh Rege, "A digital fountain approach to reliable distribution of bulk data," in *Proc. ACM SIGCOMM*, Aug. 1998, pp. 56–67.

[11] Luigi Rizzo and Lorenzo Vicisano, "RMDP: An FEC-based reliable multicast protocol for wireless environments," *Mobile Computing and Communications Review*, vol. 2, no. 2, 1998.

[12] J.C Bolot, "End-to-End Packet Delay and Loss Behavior in the Internet," in *Proc. ACM SIGCOMM*, San Francisco, CA, Sept. 1993.

[13] V. Paxson, "End-to-End Internet Packet Dynamics," in *Proc. ACM SIGCOMM*, Cannes, France, Sept. 1997, pp. 139–152.

[14] Atul Khanna and John Zinky, "The Revised ARPANET Routing Metric," in *Proc. ACM SIGCOMM*, Austin, TX, Sept. 1989, pp. 45–56.

[15] Daniel O. Awduche, Angela Chiu, Anwar Elwalid, Indra Widjaja, and XiPeng Xiao, *Overview and Principles of Interent Traffic Engineering*, Internet Engineering Task Force, May 2002, RFC 3272.

[16] Stefan Savage, Andy Collins, Eric Hoffman, John Snell, and Tom Anderson, "The End-to-End Effects of Internet Path Selection," in *Proc. ACM SIGCOMM*, Boston, MA, 1999, pp. 289–299.

[17] John Janotti, David K. Gifford, Kirk L. Johnson, M. Frans Kaashoek, and James W. O'Toole Jr., "Overcast: Reliable multicasting with an overlay network," in *Proc. 4th USENIX OSDI*, San Diego, California, October 2000, pp. 197–212.

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

[19] Michael O. Rabin, "Efficient dispersal of information for security, load balancing and fault tolerance," *J. ACM*, vol. 36, no. 2, pp. 335–348, Apr. 1989.

[20] Anindo Banerjea, "Simulation study of the capacity effects of dispersity routing for fault tolerant realtime channels," in *Proc. ACM SIGCOMM*, Aug. 1996, pp. 194–205.

[21] Azer Bestavros, "An adaptive information dispersal algorithm for time-critical reliable communication," in *Network Management and Control, Volume II*, I. Frish, M. Malek, and S. Panwar, Eds., pp. 423–438. Plenum Publishing Co., New York, New York, 1994.

[22] Johnny Chen, *New Approaches to Routing for Large-Scale Data Networks*, Ph.D. thesis, Rice University, 1999.

[23] John W. Byers, Michael Luby, and Michael Mitzenmacher, "Accessing multiple mirror sites in parallel: Using tornado codes to speed up downloads," in *Proc. IEEE Infocom*, Mar. 1999, pp. 275–283.

[24] Daniel Lewin, "Systems issues in global internet content delivery," 2000, Keynote Address at 4th USENIX OSDI Conference.

[25] Rebecca Braynard, Dejan Kostic, Adolfo Rodriguez, Jeffrey Chase, and Amin Vahdat, "Opus: an overlay peer utility service," in *Proc. 5th International Conference on Open Architectures and Network Programming (OPENARCH)*, June 2002.

[26] Sean Donelan, "Update: CSX train derailment," `http://www.merit.edu/mail.archives/nanog/2001-07/msg00351.html`.

[27] Di-Fa Chang, Ramesh Govindan, and John Heidemann, "An empirical study of router response to large bgp routing table load," Tech. Rep. ISI-TR-2001-552, USC/Information Sciences Institute, December 2001.

[28] "Cisco Security Advisory: Code Red Worm - Customer Impact," `http://www.cisco.com/warp/public/707/cisco-code-red-worm-pub.shtml`, 2001.