Multipath at the Transport Layer: An End-to-End Resilience Mechanism

Justin P. Rohrer, Ramya Naidu, and James P.G. Sterbenz
Information and Telecommunication Technology Center
The University of Kansas
Lawrence, KS 66045
Email: {rohrej|ramyanm|jpgs}@ittc.ku.edu

***Abstract*—As society's dependence on network technology increases, the need for resilience and survivability in these services becomes increasingly apparent. Since the user experience is ultimately determined by the dependability of the end-to-end service, we address this issue at the transport layer. In this paper we introduce a resilient multipath selection algorithm, which obtains multiple end-to-end paths in the WAN context through cross-layer interaction with lower layers of the network. This cross-layer interface is provided by a thin internetwork protocol (PoMo) that supports heterogeneity at trust and policy boundaries. The result is a more resilient end-to-end service provided to applications by taking advantage of redundancy in the underlying physical network. We evaluate the efficiency tradeoffs of the multipath approach on both a synthetic topology and a tier 1 ISP's backbone network topology.**

I. Introduction and Motivation

The Internet protocol suite had survivability in the face of failures as a design goal [1]. It has also proven its robustness on a large scale, in large part due to the distributed nature of its operational protocols [2]. In spite of this, it quickly becomes apparent that there is a fragility to the performance of any given network application. Unseen perturbations in the network's operational state result in an end-user experience that is far from optimal. Many applications attempt to disguise these lower-level failures, and some are quite successful, however this is only possible with significant programming overhead on a per-application basis. A fundamentally resilient transport protocol could alleviate the need for this overhead by providing selectable resilience levels in a generic manner. The reason for doing this in the end-to-end (layer 4) context as opposed to a lower layer is that the source or destination nodes are typically first to be aware of a disruption in service, so it makes sense to push control of remediation mechanisms to those hosts.

A. Crosslayering

There are many mechanisms such as diverse end-to-end paths and adaptive erasure coding that may be used to increase the resilience of end-to-end flows. Within the restrictions of the current Internet architecture it is nearly impossible to implement such mechanisms, due to the lack of support for explicit cross-layering. More recently, however, clean-slate approaches to internetworking architectures such as PoMo [3] have provided the necessary support for explicit cross-layer interaction between the transport and lower network layers. This fundamental shift in design philosophy allows us to create a new resilient transport protocol *ResTP*, of which an overview is given in Section III. In this paper we explore the use of the multipath mechanism combined with path diversity to improve the resilience of ResTP over traditional unipath protocols.

In order to evaluate the diversity of potential paths, we present a formal definition of the *diversity* metric as a comparison of two candidate paths. Based on this notion of diversity we then present an algorithm for selecting the most diverse *available* paths to use for a given source-destination pair. We then evaluate the improvement in reliability by comparing it to the conventional unipath approach. We do not evaluate different path discovery mechanisms, but assume the availability of a path database that contains all possible paths.

B. Terminology

Since a number of key terms are used with varying meanings within the networking community, we define them here to avoid confusion:

- Realm: A set of nodes and links that share common mechanisms (addressing, forwarding), trust, and policy.
- Reliability: The ability to perform a required function under stated conditions for a specified period of time. [4]
- Resilience: The ability of the network to provide and maintain an acceptable level of service in the face of various faults and challenges to normal operation. [5] Resilience is a superset of many other metrics.
- Node pair: Any two nodes at the same hierarchical level of a particular network topology; i.e. two core nodes or two subscriber nodes.
- Path: Any complete set of nodes *and* links that form a loop-free connection between a node-pair.
- Path stretch: The ratio of the number of hops on a given path, divided by the number of hops on the shortest path.
- Flow: A data session between a node-pair which may be distributed over one or more paths.
- Application: The higher-level cause that sets the service requirements of a particular flow. This may refer to a traditional software application, or a alternative motivating factor, such as an SLA (service level agreement) in the context of an ISP network.

9781-4244-3941-6/09/$25.00 ©2009 IEEE

C. Design Goals

With Resilience and Survivability as the objective, we have a number a specific goals in mind when selecting end-to-end mechanisms for use in a transport protocol: Once established, a flow should remain stable as long as the underlying physical network is not partitioned. The end systems should have some control over the paths selected. The paths chosen should be the best available given the application's service requirements. Finally, there should not be a negative impact on the network as a whole.

These goals allow us to exploit diversity to the degree that it is present in the underlying network graph. The algorithm used by ResTP to achieve this is formally described in Section III-B.

The remaining sections of the paper are organized as follows. Section II presents background and related research. Section III gives an overview of the features of the ResTP protocol. Section IV explains our simulation methodology and presents our findings, and Section V concludes.

II. Background and Related Work

The current Internet has been dominated by the use of only a few transport protocols. In this section we will take a brief look at their characteristics and shortcomings in terms of designing a resilient transport protocol.

A. Internet Transport Protocols

The most widely used transport protocol in the Internet is the Transmission Control Protocol (TCP) [6], [7], which was designed for terrestrial wired networks. TCP provides a connection-oriented reliable data-transfer service with congestion control, and uses closed-loop feedback control to maintain consistent state at the source and destination. TCP's control loop often proves to be incompatible with multipath routing solutions, especially if they cause sudden changes in RTT, result in asymmetric routes, or deliver packets out of order. The other commonly used Internet transport protocol is the User Datagram Protocol (UDP) [8]. UDP is far simpler than TCP, but does not offer any assurance or notification of correct delivery.

B. Multipath Routing

Most of the existing research related to multiple path discovery has taken place at the routing level. As such it is not concerned with finding complete end-to-end paths; instead alternative routes are discovered that protect a subset of network edges. This research contributes to the establishment of counterpart mechanisms at the transport layer.

Path Splicing [9] is one such approach that uses multiple destination-rooted routing trees to provide multiple alternative paths that may be switched between at any intermediate node. The source node is allowed to select paths at will, however no metrics about the alternative paths are transferred to the source by which an intelligent selection could be made. This is to avoid the situation in which all sources choose the same path and congest it while leaving alternative links underutilized. The benefit to giving this control to the source is that when packet losses are detected it can randomly choose a different set of path indices much faster than routing can reconverge. There is, however, no assurance that the new path chosen will map to a different set of *physical* links. A similar approach is Routing Deflections in which the source node is given some control without detailed information [10]. Both of these approaches are an enhancement to a purely routing-level mechanisms for pre-computing back-up entries in local forwarding tables in case of a link failure [2]. Without involving the source or destination nodes it generally takes more time for nodes at the location of the failure to detect it and initiate routing reconvergence.

C. Link-Layer Protection Mechanisms

Another large body of research addresses the issue of survivability by protecting the network the face of single and in some cases multiple random link failures. These generally function by reserving capacity on a connected backbone of links such that traffic can be re-routed around failed links using the reserved capacity. Some of these, such as the p-cycle approach [11], and the shared backup path protection approach [12] are able to protect all the network links while requiring only a relatively small percentage of the overall network capacity to be held in reserve. Such mechanisms lend themselves to implementation within a given ISP's network, because an intimate knowledge of the topology and utilization (peak and average) of each link in the network is required. For this reason they are not suitable from a subscriber's perspective, because they do not have the information or control over such details of the core networks across which their data is flowing. To make matters worse, even a proactive customer who is multi-homed to two independent providers has no assurance that those providers do not form a shared risk link group (SLRG) as was the case when multiple ISP's experienced failures during the Baltimore Tunnel Fire [13], [14]. These factor emphasize the need for end-to-end path diversity to be established, supported by the cross-layer transfer of relevant information.

D. Postmodern Internetwork Architecture

As mentioned previously, this work assumes the presence of the Postmodern Internetwork Architecture (PoMo) [3], which provides cross-layer information from lower layers with which to make intelligent path selection decisions. PoMo is a greenfield architecture for the future Internet which seeks to separate policy implementation from packet forwarding mechanisms and to support heterogeneous internetworking, by explicitly providing a realm interconnection layer that provides translation services at mechanism, trust, and policy boundaries.

1) PoMo Architecture: PoMo seeks to promote heterogeneity of mechanism through the use of knobs and dials. Dials expose characteristics of the underlying network to higher layers, and knobs allow higher layers to influence the behavior of the lower layers. The current Internet consists of a homogeneous network and transport layer environment operating over what is assumed to a be a stable and well connected physical

and data-link environment. These assumptions are increasingly false with the increased use of mobile and wireless physical layers. We assert that the cross-layer dissemination of control information is necessary to allow for more informed decisions to be made at the higher layers.

2) Transport Layer and PoMo: PoMo is not an end-to-end protocol nor does it define constraints on the implementation of one. However, several aspects of the architecture, such as the *knobs and dials*, provide the opportunity to design a resilient transport-layer protocol utilizing a much greater level of interaction with the underlying network than is possible in the current Internet. For example, in the PoMo architecture it is possible to query the geographic location of nodes and thus determine whether paths are physically link and node disjoint. This interaction will allow us to overcome the limitations imposed by the lack of cross-layering in the current Internet architecture.

III. End-to-End Mechanisms for Resilience

The ResTP protocol is designed with adaptive resilience mechanisms in order to support various application requirements and network operating conditions. It is essentially a generic version of the domain-specific transport protocol AeroTP [15]. While this paper is not intended to discuss all the end-to-end resilience mechanisms used in ResTP in detail, we give an overview of those features here to provide context for using the multipath algorithm. The first part of this section presents an overview of the main protocol features, and the second part covers the details of the multipath mechanism.

A. Transport Service Types and Reliability Modes

The ResTP header (Figure 1) is designed to support cross-layer parameters and shows the fields used to indicate the service parameters required to the lower network layers.

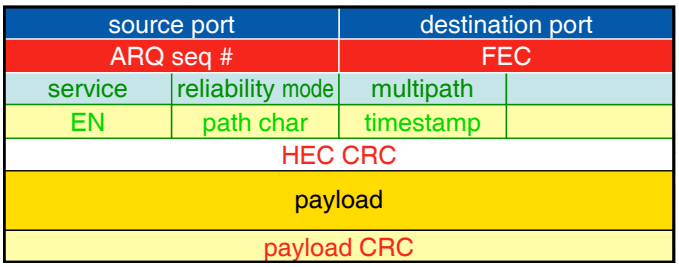

Fig. 1. ResTP header showing knob and dial fields

1) Service Types: In the past the transport layer has had little instrumentation from the network and lower layers about path conditions. In this approach we apply the principle of *translucency* [5] by making key pieces of information upwardly visible from the network to allow the transport layer to make intelligent decisions about the E2E data transfer. In doing this we have several service types in mind, with the selection indicated by the active application:

- ***Delay-bounded*** data is that for which the utility curve decreases over a relatively short period of time. An example of this is VoIP, in which the data is no longer useful after more than a few hundred milliseconds. This kind of traffic requires a low-latency path with high reliability since retransmissions are not generally an option, but the data rate is often low enough to allow for some additional overhead in the form of FEC or erasure coding.
- ***Bandwidth-specified*** traffic has a primary requirement in terms of the peak and average data rate for the flow. A large file transfer is an example of this type of service requirement, and the transport protocol will send the data over a path composed of high-capacity uncongested links, aggregating bandwidth from multiple disjoint paths if possible. Due to the high data-rates involved it may be preferable to correct errors via retransmission as opposed to incurring the overhead of FEC.
- ***Best effort*** service is for delay and bandwidth insensitive applications, such as email, in which the data should be delivered before the user gets impatient, but is not as time sensitive as a packetized telephone call. An important consideration for this type of communication is minimal resource usage at the end nodes since a server could be managing tens of thousands of connections at any given time. UDP [8] is essentially designed to provide this kind of *best effort* service, but because it does not use cross-layer information it cannot provide the application layer with any details about the service being provided, nor can it make intelligent decisions on how to deal with lost or delayed packets [16].

2) Reliability Modes: Based on the application requirements, there may be a number of data classes being transferred over the network. For this reason we define multiple *reliability modes* that are mapped from different service types and for the generic counterpart of the AeroTP reliability modes [15]. The first two modes are connection-oriented, and the last two are connectionless:

- ***Reliable*** mode uses end-to-end acknowledgements from the destination to the source as the only way to *guarantee* delivery. This carries the penalty of requiring the end nodes to maintain state regarding each packet in flight over the entire E2E path, which can be substantial in high bandwidth-$\times$-delay product environments.
- ***Near-reliable*** mode is highly reliable, but does not *guarantee* delivery, instead using the custody transfer [17] approach, which splits the ACK loop at intermediate realms at the cost of buffering ResTP segments in each PoMo gateway until acknowledged by the next realm along the path. Since the gateway uses split ARQ and immediately returns TCP ACKs to the source, the assumption is that ResTPs reliable ARQ-based delivery will succeed using SNACKs (selective negative acknowledgements) [18] supplemented by a limited number of (positive) ACKs. This can be more bandwidth-efficient than full source–destination reliability. However, the possibility exists of confirming delivery of data that the gateway cannot actually deliver to its final destination.

- ***Quasi-reliable*** mode uses only open-loop error recovery mechanisms such as FEC and erasure coding across multiple paths if available [19], thus eliminating ACKs and ARQ entirely. In this mode the strength of the coding can be tuned using cross-layer optimizations based on the quality of the channel being traversed, available bandwidth, and the application's sensitivity to data loss. This mode provides an arbitrary level of statistical reliability but without absolute delivery guarantees.
- ***Unreliable*** mode relies exclusively on the FEC of the link layer to preserve data integrity and does not use any error correction mechanism at the transport layer. Cross-layering is used to vary the link FEC strength.

The multipath mechanism may be useful in implementing any of the *reliability modes*, depending on the *service type* selected, and the graph of the underlying topology.

B. Path Selection

In this section we address the selection process for individual end-to-end paths, several of which may be chosen for use by a single connection or flow. We assume that available paths are provided by the PoMo topology server, which maintains a database of the physical network topology. The path to be taken by a given packet is then embedded into the *forwarding directive* field of the PoMo header.

1) Measuring Diversity: With the goal of increasing flow resilience in mind, we want to choose paths which will not experience correlated failures by selecting diverse paths. To this end, we define a *diversity* metric which quantifies the degree to which alternate paths share the same nodes and links. Note that because we are concerned with events and connections on a large geographic scale, a node may be thought of as representing an entire collocated data center, and a link as the physical bundle of fibers buried in a given right-of-way.

Definition 1 (Path): Given a (source, destination) pair, a path P between them is the combination of the vector of links L and the vector of intermediate nodes N traversed by that path, or

$$P = L \cup N \tag{1}$$

and the length of this path, $|P|$ is the combined total number of elements in L and N.

Definition 2 (Diversity): Let the shortest path between a given (source, destination) pair be P_0. Then, for any other path P_k between the same source and destination, we define the diversity function $D(x)$ with respect to P_0 as

$$D(P_k) = 1 - \frac{|P_k \cap P_0|}{|P_0|} \tag{2}$$

which will result in a value of 1 if P_k and P_0 are completely disjoint and a value of 0 if P_k and P_0 are identical.

In [9], Motiwala et. al. claimed that the *novelty* metric, which is measured with respect to *either* nodes *or* links is sufficient, however we assert that this is not the case. In our example topology (Figure 2) we see the shortest path, P_0, along with the alternate paths P_1 and P_2 both of which have a novelty of 1. However, both P_0 and P_2 will be taken out by the failure of node 1. In our approach, $D(P_2) = \frac{2}{3}$, which reflects this vulnerability. P_1 on the other hand has both a novelty of 1 and a diversity of 1, and does not share any common point of failure with P_0. A similar vulnerability may be found when the wavelengths or fibers from multiple nodes are in fact be spliced into a single physical corridor such as was the case in the Baltimore Tunnel Fire [13], [14], and resulting in a single point of failure. These factors motivated our decision to include both nodes and links into the diversity measure.

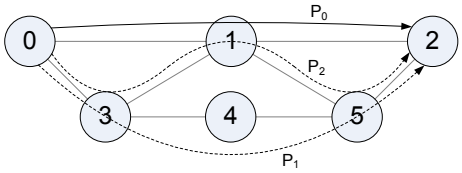

Fig. 2. Shortest path P_0 and alternatives P_1 and P_2

2) Path Selection Algorithm: Given that the number of possible paths existing between a common (source, destination) pair is z:

Step 1: Let A be the set of available paths between a given (source, destination) pair, in decreasing order by diversity value, where $|A| = z$

Step 2: Let n be the number of diverse paths required by the transport layer.

Step 3: Let B be the smallest subset of highly diverse paths, where $|B| = k$ and $k \geq n$.

$$B = \{i \in A : D(P_i) > D(P_j), \forall j \in A\} \tag{3}$$

If $k = n$, B is the set of exactly n diverse paths required by the transport layer and the algorithm is finished, otherwise we continue with steps 4 through 8.

Step 4: Let D_{min} be the minimum diversity amongst all paths in set B.

$$D_{\text{min}} = \min[D(P_i), \forall i \in B] \tag{4}$$

Step 5: Select a set C out of B which contains all the paths with a diversity greater-than D_{min}, where $|C| = m$

$$C = \{i \in B : D(P_i) > D_{\text{min}}\} \tag{5}$$

Step 6: Let D be the remaining paths in B after removing C, where $|D| = k - m$.

$$D = B - C \tag{6}$$

Step 7: Select set E, to be the shortest length paths from D, where $|E| = n - m$

$$E = \{i \in D : |P_i| \leq |P_j|, \forall j \in D\} \tag{7}$$

This step allows us to choose shorter paths when path diversities are equivalent.

Step 8: The final set S of n diverse paths is

$$S = C \cup E \tag{8}$$

This algorithm yields the required number of paths with the constraint that they will include the shortest path and the maximally diverse paths with the least stretch.

IV. Simulations and Results

Using the ns-2 simulator [20] we have implemented the multipath mechanism of ResTP and compared its performance to traditional single-path data transmission.

A. Simulation Setup

Each source sends n redundant (identical) data packets over the n paths it has requested. Because the backup paths are *already in use* there is no loss of data in the event of a failure unless *all* n paths are compromised. In order to limit the computational resources required to run the simulations, the data rates and bandwidths were scaled down from their actual values by two orders of magnitude. Each data point was averaged over 100 runs to eliminate aberrations caused by randomness in the simulation. The plots presented in this paper represent a combined total of 17,600 simulation runs.

B. Fault-Tolerant Topologies

It is intuitively obvious that using a multipath transport protocol will not yield any benefit in terms of resilience unless multiple logical and physical paths are present. For this reason we have confined ourselves to simulating on topologies which are bi-connected or better.

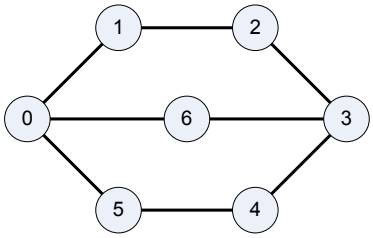

Fig. 3. Synthetic seven-realm interconnection topology

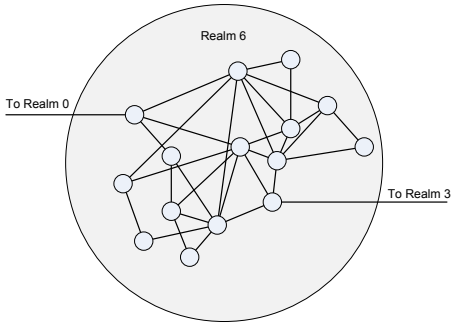

Fig. 4. Internal topology of realm six

1) Synthetic Topology: The first topology we are considering is synthetic and created in order to characterize the behavior of the multipath mechanism on a small scale. It consists of seven distinct realms, interconnected as shown in Figure 3. Each of the links has a bandwidth of 1 Mb/s and a propagation delay of 10 ms. Each of the realms has a randomly generated (using BRITE [21]) well-connected internal network of 15–20 nodes interconnected by 1 Mb/s links. Figure 4 shows the internal network of realm 6 as an example.

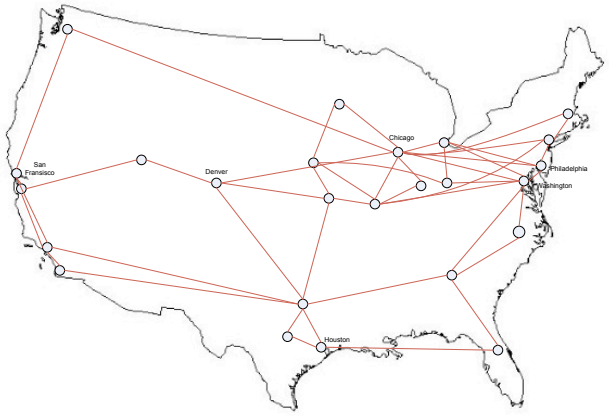

Fig. 5. Simplified AT&T backbone topology

2) ISP Backbone Topology: To further characterize the multipath behavior in a more connected environment we used a map of AT&T's backbone network, Figure 5. This was obtained from the Rocketfuel project [22], which uses a probing technique to determine physical topology. We removed links that were deemed improbable to exist on routes geographically distinct from other existing fiber paths. We also removed any stub-nodes since they could not be part of any end-to-end disjoint paths. While it may not be a perfect representation of the AT&T network, we believe that it is an accurate enough representation for our purposes. The 25 nodes in the topology are interconnected with 1 Mb/s links.

C. Traffic Patterns

On the synthetic topology, two nodes from each realm were randomly selected as traffic sources, each with a randomly selected destination in a different realm (no destination node occupied the same realm as its respective source). In each of the failure scenarios, the application sending data-rate was set to 50 kb/s so that congestion would not be a source of loss. For each of the the load scenarios, the application sending data-rate was varied between 50 and 500 kb/s to observe the effect of multipath on congestion. Each path was selected at the realm level, with traffic traversing the realm via one randomly-selected node and the shortest available path.

For the ISP topology (Figure 5), we are concerned with performance in the presence of a greater number of available diverse paths. For this topology we defined (source, destination) pairs as follows: (Chicago, Houston), (Washington, San Francisco), and (Philadelphia, Denver). These were selected

because each pair has a minimum of 3 link-disjoint paths between them. All the ISP scenarios were run with an application sending data-rate of 50 kb/s. The paths for this scenario specified each transit node.

For each scenario, we compared the performance to a baseline UDP flow routed using Dijkstra's shortest-path algorithm. This is the curve labeled "sp" on the graphs. The other three curves show the performance of ResTP using the path selection algorithm described in Section III-B2 while requesting n diverse paths. It should be noted that the $n = 1$ case does not carry any implication of improved reliability over the sp case, it is only included here for completeness and to show that our algorithm does not cause any degradation in performance. The simulation was run with a 20 second warm-up time after traffic began, at which time the link failure scenario was applied. Trafic was sent for an additional 480 seconds, and all queued traffic was allowed to propagate to the destination before the simulation terminated.

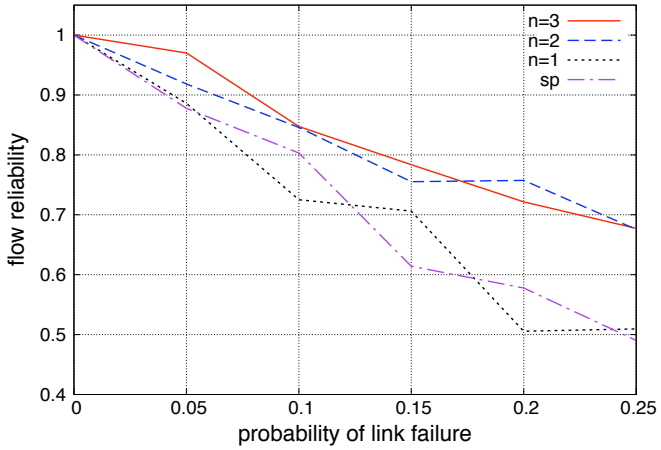

Fig. 6. Comparison of flow reliability for synthetic topology

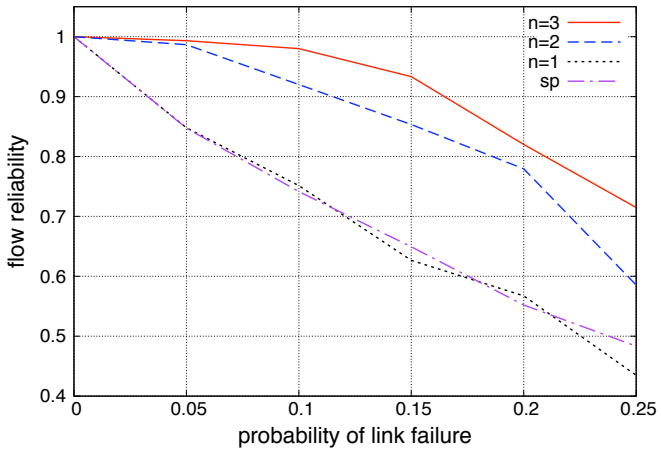

Fig. 7. Comparison of flow reliability for AT&T topology

D. Flow Reliability with Link Failures

The link failure scenario was defined as follows. After the simulation warm-up period, each link failed with a uniform independent probability. This probability was varied between 0 and .25 to evaluate performance under varying severities of failure.

Our primary metric for evaluating the performance of multipath is *flow reliability*. This is shown as the fraction of flows which continue delivering data during a link failure scenario. On the synthetic topology we observe a maximum reliability improvement of 20% over single-path routing, shown in Figure 6. We also note that the performance of multipath with $n = 2$ and $n = 3$ is nearly identical. We attribute this to the low degree of connectivity in the network, which results in fewer high-diversity paths being available to ResTP. On the AT&T topology we observe a maximum improvement of nearly 30% over single-path routing, shown in Figure 7. In this case the performance of multipath with $n = 3$ is consistently 5–10% better than the performance with $n = 2$.

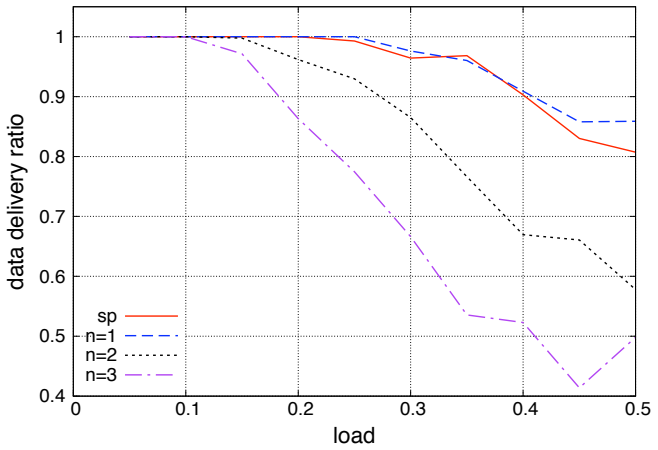

Fig. 8. Reduction in performance due to congestion as traffic load increases

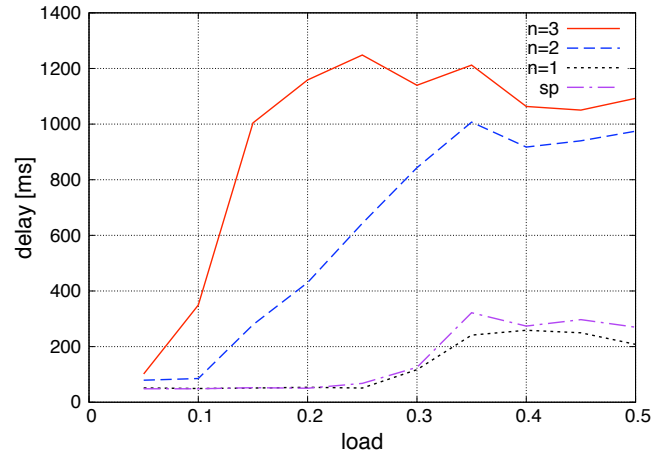

Fig. 9. Increase in delay as traffic load increases

E. Performance with Respect to Load

Clearly there is a cost to increasing reliability with this mechanism, and the tradeoff we are making in increased traffic in the network, and eventually increased congestion. We used the 7-realm topology to evaluate the effects of increased load. Figure 8 shows the decrease in performance due to congestion losses as the load increases in the network. Figure 9 shows the corresponding increase in end-to-end delay caused by queuing in the network as congestion increases. Again these results are topology dependent, in that a better connected graph will have the additional load of multipath spread across a greater number of alternate links.

V. Conclusions and Future Work

This paper introduced the ResTP protocol, along with the *diversity* metric, as well as the design and evaluation of an end-to-end multipath resilience mechanism. This work assumes the presence of the Postmodern Internetwork Architecture to provide cross-layer information with which to make intelligent path selection based on the *diversity* metric. We have shown a 20–30% performance improvement in the presence of link failures when diversity is available in the underlying network graph.

There are a number of aspects of this work which we intend to examine further in future work. Finding all possible paths between a (source, destination) pair is an NP-hard problem and we intend to examine heuristic methods for bounding this process to make it less computationally intensive. There are also many other erasure coding schemes which could be used across multiple paths to improve efficiency, and we intend to examine more of these in the future.

Acknowledgments

The authors would like to thank Abdul Jabbar for his help in formalizing the steps of the path selection algorithm, and all the members of the ResiliNets group for discussions and insights along the way. This work was supported in part by the National Science Foundation FIND (Future Internet Design) Program under Grant No. CNS-0626918.

References

[1] D. D. Clark, "The design philosophy of the darpa internet protocols," in *SIGCOMM '88: Symposium proceedings on Communications architectures and protocols*. New York, NY, USA: ACM, 1988, pp. 106–114.

[2] A. Kvalbein, A. F. Hansen, T. Cicic, S. Gjessing, and O. Lysne, "Multiple routing configurations for fast IP network recovery," *IEEE Transactions on Networking*, 2008.

[3] B. Bhattacharjee, K. Calvert, J. Griffioen, N. Spring, and J. Sterbenz, "Postmodern internetwork architecture," Information and Telecommunication Center, 2335 Irving Hill Road, Lawrence, KS 66045-7612, Technical Report ITTC-FY2006-TR-45030-01, February 2006.

[4] T. W. Group, "Atis telecom glossary 2000," Alliance for Telecommunications Industry Solutions (ATIS), American National Standard for Telecommunications T1.523-2001, February 2001.

[5] J. P. G. Sterbenz and D. Hutchison. (2008, April) Resilinets: Multilevel resilient and survivable networking initiative wiki. http://wiki.ittc.ku.edu/resilinets. [Online]. Available: http://wiki.ittc.ku.edu/resilinets

[6] J. Postel, "Transmission Control Protocol," RFC 793 (Standard), Sep. 1981, updated by RFC 3168. [Online]. Available: http://www.ietf.org/rfc/rfc793.txt

[7] K. Ramakrishnan, S. Floyd, and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP," RFC 3168 (Proposed Standard), Sep. 2001. [Online]. Available: http://www.ietf.org/rfc/rfc3168.txt

[8] J. Postel, "User Datagram Protocol," RFC 768 (Standard), Aug. 1980. [Online]. Available: http://www.ietf.org/rfc/rfc768.txt

[9] M. Motiwala, M. Elmore, N. Feamster, and S. Vempala, "Path splicing," in *SIGCOMM '08: Proceedings of the ACM SIGCOMM 2008 conference on data communication*. New York, NY, USA: ACM, August 17-22 2008, pp. 27–38.

[10] X. Yang and D. Wetherall, "Source selectable path diversity via routing deflections," *SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 4, pp. 159–170, 2006.

[11] W. D. Grover and D. Stamatelakis, "Cycle-oriented distributed preconfiguration: Ring-like speed with mesh-like capacity for self-planning network restoration," in *Proceeding of the IEEE International Conference on Communications (ICC'98)*, vol. 1, June 1998, pp. 537–543.

[12] B. G. Jozsa, D. Orincsay, and A. Kern, "Surviving multiple network failures using shared backup path protection," in *Proceedings of the Eighth IEEE International Symposium on Computers and Communication (ISCC 2003)*, vol. 2, June 2003, pp. 1333–1340.

[13] H. C. Styron, "Csx tunnel fire: Baltimore, md," Federal Emergency Management Administration, Emmitsburg, MD, US Fire Administration Technical Report USFA-TR-140, 2001.

[14] M. R. Carter, M. P. Howard, N. Owens, D. Register, J. Kennedy, K. Pecheux, and A. Newton, "Effects of catastrophic events on transportation system management and operations, howard street tunnel fire, baltimore city, maryland – july 18, 2001," U.S. Department of Transportation, ITS Joint Program Office, Washington DC, Tech. Rep., 2002.

[15] J. P. Rohrer, E. Perrins, and J. P. G. Sterbenz, "End-to-end disruption-tolerant transport protocol issues and design for airborne telemetry networks," in *Proceedings of the International Telemetering Conference*, San Diego, CA, October 27–30 2008.

[16] S. Farrell, V. Cahill, D. Geraghty, I. Humphreys, and P. McDonald, "When TCP breaks: Delay- and disruption- tolerant networking," *IEEE Internet Computing*, vol. 10, no. 4, pp. 72–78, July-Aug. 2006.

[17] K. Scott and S. Burleigh, "Bundle Protocol Specification," RFC 5050 (Experimental), Nov. 2007. [Online]. Available: http://www.ietf.org/rfc/rfc5050.txt

[18] R. C. Durst, G. J. Miller, and E. J. Travis, "TCP extensions for space communications," in *MobiCom '96: Proceedings of the 2nd annual international conference on Mobile computing and networking*. New York, NY, USA: ACM Press, November 1996, pp. 15–26.

[19] A. J. McAuley, "Reliable Broadband Communication Using a Burst Erasure Correcting Code," *SIGCOMM Comput. Commun. Rev.*, vol. 20, no. 4, pp. 297–306, 1990.

[20] (2007, December) The network simulator: ns-2. http://www.isi.edu/nsnam/ns/. [Online]. Available: http://www.isi.edu/nsnam/ns/

[21] (2008, September) Boston university representative internet topology generator (BRITE). [Online]. Available: http://www.cs.bu.edu/BRITE/

[22] (2008, September) Rocketfuel: An ISP topology mapping engine. [Online]. Available: http://www.cs.washington.edu/research/networking/rocketfuel/