On the Safety and Security of Path Splicing:

A Case Study for Path Splicing on the GÉANT Network[†]

CHRISTOPHER PAGE
Department of Computer Science
Texas State University-San Marcos
cp1006@txstate.edu

MINA GUIRGUIS
Department of Computer Science
Texas State University-San Marcos
msg@txstate.edu

***Abstract*— Path splicing is a proposed routing architecture for the Internet in which end-hosts could change the path their traffic uses by changing a number of bits in the packet header. Path splicing improves the reliability of the network against link failures since it ensures that physically connected links can be discovered and used. To that end, this paper studies the performance of path splicing in non-adversarial and adversarial environments. In a non-adversarial setting, we investigate the implications behind giving the end-hosts the power to select routes in the absence/presence of errors in the probing mechanisms they are employing to infer the state of the network. In an adversarial setting, we examine the extent to which attackers can exploit path splicing to mount attacks that cause a series of route changes by end-hosts in searching for better paths. Our results are derived from real traffic matrices obtained from the GÉANT network.**

I. INTRODUCTION

Over the past few years, there have been a lot of research efforts dedicated to the design of a clean slate future Internet [1]–[3]. This has prompted a large number of different proposals aiming to "think outside the box" at different levels of the network. To complement these efforts, it is important to investigate the security and safety of those new designs before they are put into action. Security should be given the proper attention as an integral part of the new designs as opposed to an after-thought as with the current Internet. One important aspect in these new designs is Internet routing.

There have been a number of proposals that focus on exploiting multiple paths for Internet routing [4]–[8], among many others. With the potential existence of multiple paths between two points, the network becomes more resilient against link failures, in addition to improving other performance metrics such as reducing congestion and balancing traffic across multiple paths. This paper focuses primarily on path splicing, however, many of the issues addressed here still apply to other multipath routing protocols.

Path splicing [6], [7] is a new proposed architecture for intradomain and interdomain routing. Path splicing allows end-hosts (and intermediate routers) to change the route used by their traffic based on link failures as well as on some perceived performance metrics (e.g., delay and packet loss). Changing the route is achieved by changing the splicing bits in a header that resides between the network and the transport layer.

While path splicing enables fast recovery against link failures and gives the end-host the flexibility to change routes, it opens a back door for exploits to be mounted. In this paper, we examine the safety and security aspects of path splicing in greater details. We seek to tackle the following questions:

1) What is the impact of path splicing on the convergence process of the network?
2) What are the implications behind giving an end-host the power to *change* routes?
3) Can path splicing be exploited by attackers to hinder its convergence and/or introduce oscillations?

To answer the above questions, we use two different models:

Non-Adversarial Model: In this model, we study the convergence properties under normal conditions such as traversing highly utilized links. We investigate the decision making process of end-hosts in changing their routes. For example, an end-host can utilize probing mechanisms to probe the state of the path and based on the results, the end-host may decide to change his/her route. We capture different metrics such as the number of switches performed and how different degrees of probing errors impact the overall routing process.

Adversarial Model: In this model, we study different scenarios, in which attackers can hinder the convergence process of path splicing by introducing oscillations and possibly loops. A group of attackers may collude to impact a small subset of links, causing the end-hosts using them to switch routes. This, in return, may create congestion on other links (causing more end-hosts to switch). Moreover, the attackers can time their attacks in order to exploit the adaptation of the end-hosts.

In order to give an accurate assessment for our study, it is important to work with *real topologies* and with *real traffic matrices*. It is typically hard to obtain both of these metrics for the same network and over the same period of time. For example, the rocketfuel data-sets give us the topology without the traffic matrices [9]. While we can generate synthetic traffic matrices (based on other measured traffic matrices from other networks) on such topologies, the generalization may not always work due to the use of different topologies [10]. Fortunately, Uhlig *et al* [11], have provided the research community with intradomain traffic matrices for the GÉANT network, the European Research and Educational Network.

[†] This work was supported by an REP grant from Texas State University.

We faced two main challenges while working with the GÉANT network. First, the network was highly under-utilized (less than 12% for the busiest hours).[1] This has impacted our results in terms of limiting the impact of path splicing. Despite such under utilization, the results still show relatively significant overhead. With higher utilized networks, the results would even be more significant. The second challenge was the anonymization of the data-sets. Since we needed to compute the weights on the links[2], we had to cross reference the anonymized data set with the GÉANT topology (which contained the capacities of the links) [12]. We then inferred the unanonymized links and we reveal them in the paper for the research community.

Paper Organization: In Section II, we cover background material on path splicing. We also address performance and security issues when end-hosts/attackers are given the power to change/attack routes. Section III represents our experimental evaluation using the GÉANT data sets. We discuss related work in Section IV and conclude the paper in Section V.

II. Security Issues in Path Splicing

In this section we briefly cover some background material on path splicing. Then we study its performance in adversarial and non adversarial environments.

A. Path Splicing

Path splicing is a new intradomain and interdomain routing protocol focused on providing end-hosts with multiple paths to exchange traffic. This would increase the network reliability to approach that of the underlying physical topology. To achieve this, a number of forwarding trees at each node are computed and network traffic is allowed to switch between those trees en route to the destination. The forwarding trees are computed based on perturbed topologies, where the weights on the links are randomized in a controlled manner (to prevent the new paths from being of very high costs). For each perturbed topology, the lowest cost path is computed to obtain a forwarding tree. To switch between slices, a shim header between the transport and network layer is used. The header contains the splicing bits that dictate which forwarding table to be used at each intermediate hop. For an n hop route with k different slices at each hop, the size of the header is $n \times \log k$.

To recover from link failures or to choose a different route, an end-host (or an intermediate router) can select those splicing bits at random. This would cause some routers to choose different forwarding tables along the path, avoiding the original one. Notice that the end-host would not *know* the route used, yet he/she has the power to change it.

Figure 1 shows an illustrative example of two different forwarding trees (computed from node A). The one on the left shows the original weights and the shortest path. The one on the right shows the perturbed topology along with the new shortest path. For additional details on path splicing, we refer the reader to the original papers [6], [7].

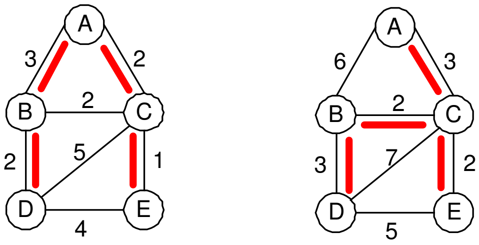

Fig. 1. An illustrative example of path splicing based on node A. Topology with original weights (left) and perturbed weights (right).

B. Non Adversarial Scenarios

In a non-adversarial setting, if an end-host is given the power to choose a different path to reach a particular point, he/she may explore this option for the following reasons: First, the end-host may be experiencing an unacceptable performance on the current path and would like to find a better path to meet some guarantees. Second, the end-host may just be greedy, seeking a better path with additional bandwidth and/or less delay. Third, the end-host may be able to take advantages of multiple paths at the same time.

Splicing Threshold: We model the behavior of an end-host based on perceived performance. This performance can be measured directly or via probing mechanisms. Regardless of the actual method, we define "Splicing Threshold" to be a specific threshold, over which an end-host will try to reroute his/her traffic along an alternate path. Clearly, there could be many instantiations for the Splicing Threshold, however, in this paper, we focus on link utilization as a measure of performance. The motivation is that the higher the link utilization, the more chance that an end-host – using this link – would seek to switch to another link. Notice that even if a link is not fully utilized, due to the statistical multiplexing of packets and the presence of bursty connections [13], [14], the end-host would perceive degraded performance.

The end-host would typically use different probing mechanisms to infer the state of the network. These range from simple packet probes to more heavy weight probes. Regardless of the exact probing mechanism, errors in measurement are typically introduced. We investigate the impact of different degrees of errors on path splicing.

Splicing Overhead: With path splicing, an end-host may use a non-optimal route to carry his/her traffic. In most cases, this would increase the path length which increases the overall load on the network. For example, the path between node A and B, in the perturbed topology in Figure 1, will traverse two segments AC and CB as opposed to the original segment AB. To assess the impact of path splicing on links, we define "Splicing Overhead" for a link to be the percentage of change in utilization:

$$Splicing\ Overhead = \frac{U_a - U_b}{C} \tag{1}$$

[1] Measured as the total traffic divided by the total capacities.

[2] In the GÉANT network, the weights are computed based on the reciprocals of the capacities of the links with minor tweaking.

where U_a is link's utilization after splicing, U_b is the link's utilization before splicing, and C is the link's capacity. Each link in the network has a splicing overhead value, and so does the network as a whole. For some links, the splicing overhead could be negative.

C. Adversarial Scenarios

Path splicing can be exploited in an adversarial setting. The general idea of an attack would be to target (say) a specific link and cause end-hosts using that link to experience degraded performance. This can easily be achieved either by flooding, sending bursty traffic, or mounting a low rate attack [15], [16]. This attack would cause the end-hosts to switch to different paths, causing other paths to be more congested. Notice that those paths are typically longer (as explained above), and thus the impact of the attack ensures a multiplicative factor. Notice also that the newly congested links would likely trigger more splicing for other end-hosts. The question is whether the dynamics induced would subside or would continue. Another question is whether attackers can introduce loops.

Based on the above simple attack, we envision attacks that occur over time causing a series of splicing effects. These attacks resemble the Reduction of Quality (RoQ) attacks on adaptive load balancers [17]. The attack's premise in path splicing is to keep the end-hosts always switching their paths by attacking the correct link at the correct time.

III. Experimental Results

A. The GÉANT Network

For the purposes of our assessment, we required the most realistic representation of a network topology that we could find. We decided to model the network topology used in our simulations after the GÉANT network, the European Research and Educational Network. We had access to information about the GÉANT's topology (although anonymized) and traffic matrices data [11].

Using the anonymized topology data file, we were able to construct an accurate topology modeled after the GÉANT network. We inferred link capacities by cross referencing the anonymized data set with the actual GÉANT network topology [12]. The cross reference was done by hand based on the degrees and connectivity structure of the nodes. Our inferred mappings between node IDs and countries are given in Table I. We have found that Germany is represented by two nodes (10 and 17). Also, node 4 was not present in the GÉANT network topology [12]. This node may have been added later on, or its links may have changed significantly. There were three different link capacities present in our topology: 10 Gbps, 2.5 Gbps, and 155 Mbps. Each link's weight was determined according to its capacity. The weights for the 2.5 Gbps links were three times as much as the 10 Gbps links, and weights for the 155 Mbps links were five times as much as the 10 Gbps links. We made the file that contains the inferred topology, along with weights and capacities, available online [18].

Country	Node ID	Country	Node ID
Switzerland	1	Sweden	12
United Kingdom	2	Netherlands	13
Italy	3	Israel	14
Unidentified	4	Slovakia	15
Slovenia	5	Austria	16
Luxembourg	6	Portugal	18
France	7	Belgium	19
Croatia	8	Czech Republic	20
Hungary	9	Spain	21
Germany	10,17	Poland	22
Greece	11	Ireland	23

TABLE I
The inferred mappings for the GÉANT network.

The authors in [11] were able to collect traffic matrices showing the total amount of traffic between any two nodes in the GÉANT network using the TOTEM toolbox [19]. The traffic matrices are stored in XML data files, each of which contains the traffic matrix for a 15 minute interval of time. These data files range over a period of four months, and allow us to know how much traffic is being transmitted between any two pairs of nodes in the GÉANT network during any 15 minute period.

Using our topology and traffic data, we are able to simulate the traffic on the GÉANT network by loading the topology into our simulation, reading in a file containing the traffic matrix for a 15 minute period, and then creating the traffic data on our topology. We are then able to analyze the state of the network.

B. Path Splicing on the GÉANT Network

We simulated path splicing using the GÉANT topology by first creating the slices. To generate different shortest path trees (slices), we perturb the link weights using random degree-based perturbations as indicated in [7]. So each node's forwarding table contains a number of slices to choose from when routing traffic. For all of our simulations, we set the number of slices to five to allow the reliability of the network to approach the best possible as suggested in [7]. We then generate traffic on the network by loading a network state into our simulation using the traffic matrix for a 15 minute interval. We do this using the original, unperturbed shortest path tree for each node, so no splicing has occurred yet. We now have the network in the state exactly as it is specified in the traffic matrix. We instantiate the "splicing threshold", to be the percentage of bandwidth utilized on a link before some end-hosts using that link will try to reroute their traffic along an alternate path using path splicing. The idea is that links with less available bandwidth will yield less performance for the end-hosts using these links. This creates a situation where it may be an attractive option for these end-hosts to use path splicing to seek better performance along a different path.

The converging process consists of checking the network for links whose utilization crosses the splicing threshold. Once a link whose utilization has crossed the splicing threshold has been identified, one source-destination node pair using this link is selected at random to use path splicing to find an alternate path. To generate the alternate path, we simulate the effect

Splicing Threshold	60	55	50	45	40	35	30	25	20
Maximum Overhead	0.2	0.2	0.2	0.33	0.35	0.41	0.44	0.44	0.48
Link ID	19	19	19	6	19	3	8	19	19

TABLE II

MAXIMUM OVERHEAD ON LINKS DUE TO DIFFERENT SPLICING THRESHOLDS

of the source node randomly setting the splicing bits in the packet headers of the outgoing traffic. This is accomplished by starting at the source node and selecting a random slice to use for finding the next hop to the destination. Each node along the way to the destination behaves in the same manner, selecting a random slice to use for forwarding the traffic until it reaches the destination node. The result is a path from the source node to the destination node which uses multiple slices. We keep the throughput constant for the source-destination node pairs, so that they use the same value on the new path as they were using on the old path, which is the value specified in the traffic matrix.

When we simulated path splicing in our experiments, we noticed that most of the time, the network never really “converges” in the traditional sense. That is, when given a splicing threshold of 60% or less, if at least one link has a utilization which exceeds this threshold, there is a very good chance that splicing will result in at least one oscillating path. This means that due to certain bottleneck links in the network, there will exist at least one source-destination pair for which there is no path between them consisting only of links whose utilizations will not cross the splicing threshold. Because of this effect, the converging process is repeated a limited number of times. For the purposes of our simulations, we have set this to be a maximum of 100. This means that our simulation stops after each link has been checked 100 times to see if its utilization has crossed the splicing threshold.

In our assessment, we tracked the utilization of each link during the converging process along with the number of switches performed by each node. We only present here the maximum overhead on a link since the security of the overall network is typically dictated by its weakest link (the one with the highest utilization) and network administrators typically care more about those links.

C. Non-Adversarial Results

Table II shows the results for a single 15 minute traffic matrix. We ran the simulation with nine different splicing threshold values ranging from 60% to 20% in increments of 5%. For each splicing threshold, the table shows the splicing overhead value for the link experiencing the greatest overhead, and the link’s ID.

Using the same data-set file, Figure 2 shows the usage statistics of all the links on the network during one 15-minute period. For each link in the network we show the initial utilization, which is the utilization as it is specified in the traffic matrix before any splicing occurs. We then show the maximum utilization at any point in time during the converging process, and the utilization after the converging process has ended. Our results show that many of the links experience an increase in utilization as a result of path splicing (splicing overhead). Since we keep the amount of traffic between source-destination pairs constant during the converging process, the reason for any splicing overhead is due to new paths which are longer than the original path (defined as stretch in [7]) , as well as loops in the new paths. Notice that the maximum overhead (the middle bar) indicates *transient overhead* due to splicing that impacts all end-hosts using that link. Overall there was an increase of 11% of the total traffic due to splicing.

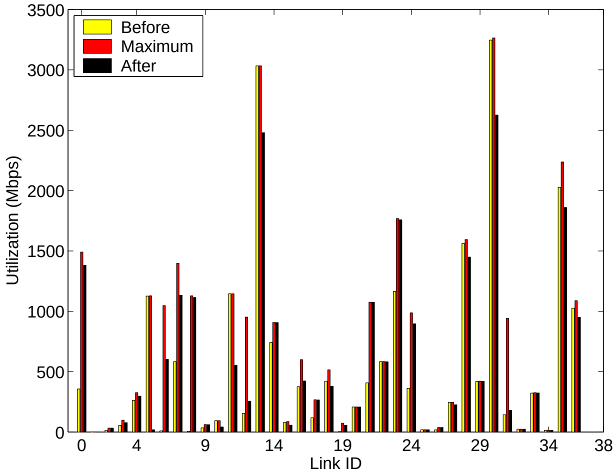

Fig. 2. Utilization of each link, before splicing, during splicing and after splicing. Results are shown for the Splicing Threshold of 30%.

We used the traffic data from ten of the traffic matrices provided in [11], one per day for ten days. More specifically, each of the ten traffic matrices covered a 15 minute period of time, at approximately the same time of day (all within four hours of each other), for one of the ten days. For each traffic matrix we repeated the above experiment over the same range of splicing thresholds. For each simulation we kept track of the link in the network which experienced the greatest splicing overhead. We then averaged the splicing overhead values of these links according to the splicing threshold that was used. Figure 3 (Left) shows both the average and maximum splicing overhead value for each of the splicing thresholds tested over the 10 days. Our results show that even in a network such as the GÉANT network which is highly under-utilized, some links experience splicing overheads ranging from approximately 30% to 50% as the splicing threshold decreases from 60% to 20%. These are significant increases, which we could expect to have a negative impact on a network with a higher overall utilization than that of GÉANT.

We also simulated the effect of probing mechanisms used by the end-hosts, which are prone to a certain amount of error. Instead of using a constant value for the splicing threshold during these simulations, each time we check a link during

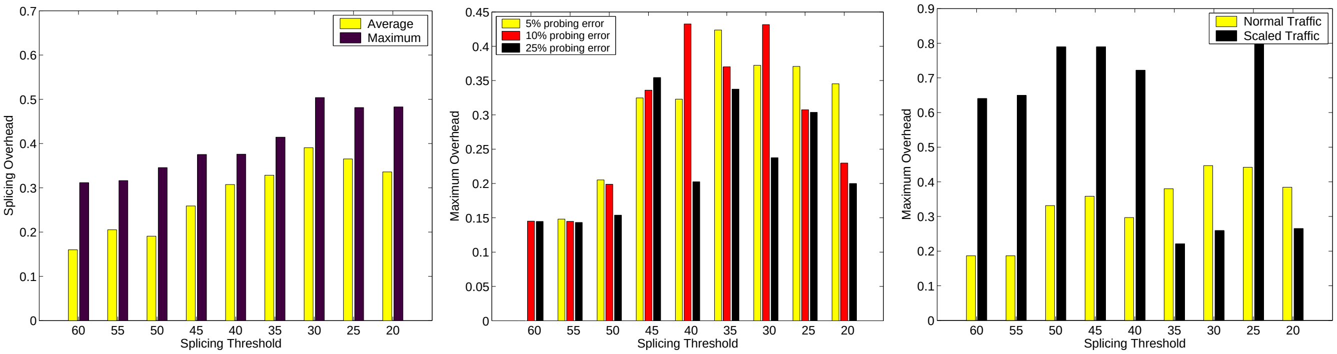

Fig. 3. Left: Maximum and average overhead over a period of 10 days. Center: Effect of probing errors on path splicing. Right: Impact of attacks on link 7 under normal traffic and with scaled traffic with a factor of 2.

the converging process to see if its utilization crosses the splicing threshold, we adjust the splicing threshold by a small amount. Using the same data-set file as shown in Table II, Figure 3 (Center) shows the results when we allow the splicing threshold values to vary by plus or minus 5%, 10%, and 25%. For each splicing threshold value, we kept track of the splicing overhead value of the link in the network experiencing the the maximum overhead. The results indicate that as we allow the perceived splicing threshold to vary to a greater degree, the maximum splicing overhead values in the network *tend to decrease!* This is because the end-hosts are no longer acting in a uniform manner. Each one is deciding based on different measurements which results in a more balanced distribution of traffic.

D. Adversarial Results

We simulated the effect of an attacker flooding a single link on the network with traffic, causing the link to become unusable. In this situation, all of the end-hosts using this link will attempt to reroute their traffic to avoid becoming disconnected. We used the same data-set file as shown for Table II. For each splicing threshold value, we kept track of the link with the maximum splicing overhead.

In Figure 3 (Right), the bars on the left show the maximum splicing overhead values on the network when one link has been attacked. The bars on the right show the same scenario, except that all of the traffic on the network has been scaled up by a factor of two in an effort to show the impact on the network when the utilization is increased. Our results show that the resulting maximum splicing overhead values are slightly greater than those in the non-adversarial model. These values are only slightly greater because much of the traffic utilizing the link which was attacked was actually *unable to be rerouted*. This is because many of the end-hosts were unable to find an alternate path with enough available capacity to carry their traffic. This was the case with other link attacks that we experimented with. In a real scenario, this means that these end-hosts would have to reduce their traffic in order to be accommodated by the network. In order to preserve the integrity of the traffic matrices, we do not modify the traffic occurring between end-hosts to attempt to simulate these decreases. This also explains why some of the maximum splicing overhead values are less for the scaled traffic, since there is less of a chance that there exists an alternate path with enough available capacity to accommodate this traffic. Another reason why some traffic is unable to be routed is that even though end-hosts have multiple shortest path trees to choose from when routing traffic, there are still some cases where the original shortest path may be favored by all of the slices. We explain more on this below.

E. Final Remarks

When generating slices, we have observed that even though there may exist many paths between two nodes physically on a network, a significant number of these paths may not be available for use by path splicing. Since we are only using a limited number of slices, we are limited in the number of possible paths which are available to us. Despite the benefit from path splicing in finding multiple paths between any two nodes, there are still situations where if a link fails, some of the source-destination nodes utilizing that link may become disconnected because an alternate path may not be available. This situation is more of a concern in smaller networks, as there may only be a few possible paths between two nodes, and if any of these are not determined to be attractive paths when generating the slices, they are lost. In larger networks, there are more possible paths between any two points in the network, and this becomes less of an issue. However in either case, we have found that for traffic occurring between nodes that are only 1 to 2 hops away from each other, it is likely that the shortest path will still be favored, even when perturbing link weights and generating multiple slices. This is because we perturb the link weights based on the original weight, making longer paths less attractive. This is in an effort to reduce stretch. Since the shortest path is likely to be favored in these situations no matter which slice is used, if a link which is part of the shortest path fails, then reliability between these source-destination nodes will not improve by using path splicing.

We should note that, in our simulations, throughout the converging process when the utilization of a link is found to be crossing the splicing threshold, node pairs are selected at random *one at a time* to try and reroute their traffic around this link. Only a few end-hosts may need to reroute their traffic in order for the link's utilization to decrease below the splicing threshold. On an actual network using path splicing, end-hosts utilizing a poor performing link may *all* decide to reroute their traffic leaving the link highly under-utilized.

IV. Related Work

The work presented in this paper relates to two main areas of research: multipath routing and security. In this section, we put our work in context within these two areas.

Multipath Routing: Multipath Routing aims to utilize the existence of multiple paths between end-hosts [4], [8], [20]. These paths can either be used simultaneously (to obtain additional bandwidth), or can be used in a fail-over mechanism. The two main proposed approaches for multipath routing is source routing (e.g., [21]) and overlay routing (e.g., [22]). In source routing, the path used is selected by the end-host (or a first-hop edge router). Each packet would contain a list of hops that indicates how routers should route this packet. Clearly, this limits the control over which ISPs route traffic, that is in addition to the overhead in keeping the source updated with recent topologies. In overlay routing, a network overlay is formed and nodes route traffic on the overlay through intermediate nodes, rather than on the physical network. One main drawback of overlay routing is the overhead in bandwidth as data packets traverse longer paths. Path splicing aims to strike a good balance between the above two main approaches. Through the splicing bits, end-hosts can change the route, without the knowledge of the route. Also, those routes are computed based on a perturbed topology, thus limiting the overhead in stretch for the data packets.

Security: The exploits we address in this paper go beyond the traditional Denial of Service (DoS) attacks that aim to flood a link or a service with bogus traffic [23]. In [17], an instantiation of Reduction of Quality (RoQ) attacks is exposed in which attackers can time their malicious traffic to exploit the adaptation employed by load balancing techniques to cause oscillations. Path splicing opens the door for such exploits to be mounted due to the end-host adapting (by changing its route) to conditions on the links he/she is using.

V. Conclusion

This paper examined the proposed concept of path splicing through the GÉANT network. In our study, we have shown that giving the end-host the power to change routes can significantly impact the behavior and state of the network, especially when end-hosts decide to change routes based on probed metrics. In that regard, we have formalized the notion of splicing thresholds to capture the conditions that trigger route changes. We have found that on the GÉANT network, convergence will likely never happen in the traditional sense as selected routes between end-hosts keep oscillating. Moreover, such oscillations can be induced by a clever attacker through exploiting the adaptation at the end-hosts when choosing paths. This research is of importance because it is very possible that some form of path splicing will be a common implementation in future networks, and it is vital that we understand its potential problems and weaknesses along with the security issues involved.

References

[1] GENI, "Global Environment for Network Innovations," http://www.geni.net/.

[2] Stanford University, "Clean Slate Design for the Internet," http://cleanslate.stanford.edu/.

[3] Carnegie Mellon University, "100x100 Clean Slate Project," http://100x100network.org/.

[4] H. Han, S. Shakkottai, CV Hollot, R. Srikant, and D. Towsley, "Multi-path TCP: A Joint Congestion Control and Routing Scheme to Exploit Path Diversity in the Internet," *IEEE/ACM Transactions on Networking (TON)*, vol. 14, no. 6, pp. 1260–1271, 2006.

[5] B. Radunovic, C. Gkantsidis, D. Gunawardena, and P. Key, "Horizon: Balancing TCP over multiple paths in wireless mesh network," in *Proceedings of MobiCom*, San Francisco, CA, September 2008.

[6] M. Motiwala, N. Feamster, and S. Vempala, "Path Splicing: Reliable Connectivity with Rapid Recovery," in *Proceedings of Hotnets*, Atlanta, GA, November 2007.

[7] M. Motiwala and M. Elmore and N. Feamster and S. Vempala, "Path Splicing," in *Proceedings of ACM SIGCOMM*, Seattle, WA, August 2008.

[8] W. Xu and J. Rexford, "MIRO: Multi-path Interdomain ROuting," in *Proceedings of ACM SIGCOMM*, Pisa, Italy, September 2006.

[9] N. Spring, R. Mahajan, and D. Wetherall, "Measuring ISP Topologies with Rocketfuel," in *Proceedings of ACM SIGCOMM*, Pittsburgh, PA, August 2002.

[10] A. Nucci, A. Sridharan and N. Taft, "The Problem of Synthetically Generating IP Traffic Matrices: Initial Recommendations," *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 2, 2005.

[11] S. Uhlig, B. Quoitin, J. Lepropre and S. Balon, "Providing Public Intradomain Traffic Matrices to the Research Community," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 1, 2006.

[12] GEANT, "An Overview Map of teh GEANT Network," http://www.geant.net/upload/pdf/Topology_Oct_2004.pdf.

[13] J. Cao, W. Cleveland, D. Lin, and D. Sun, "The Effect of Statistical Multiplexing on Internet Packet Traffic: Theory and Empirical Study," *Bell Labs Technical Report*, 2001.

[14] H. Jiang, and C. Dovrolis, "Why is the Internet Traffic Bursty in Short Time Scales?," in *Proceedings of ACM SIGMETRICS*, Banff, Alberta, Canada, June 2005.

[15] A. Kuzmanovic and E. Knightly, "Low-Rate TCP-Targeted Denial of Service Attacks (The Shrew vs. the Mice and Elephants)," in *Proceedings ACM SIGCOMM'03*, karlsruhe, Germany, August 2003.

[16] M. Guirguis, A. Bestavros, and I. Matta, "Exploiting the Transients of Adaptation for RoQ Attacks on Internet Resources," in *Proceedings of IEEE ICNP*, Berlin, Germany, Oct 2004.

[17] M. Guirguis, A. Bestavros, I. Matta and Y. Zhang, "Reduction of Quality (RoQ) Attacks on Dynamic Load Balancers: Vulnerability Assessment and Design Tradeoffs," in *Proceedings of IEEE INFOCOM*, Anchorage, Alaska, May 2007.

[18] C. Page and M. Guirguis, "Inferred GEANT Topology with Weights and Capacities," http://www.cs.txstate.edu/ mg65/splicing/.

[19] TOTEM, "A TOolbox for Traffic Engineerng Mehtods," http://totem.run.montefiore.ulg.ac.be/.

[20] A. Akella, J. Pang, A. Shaikh, "A Comparison of Overlay Routing and Multihoming Route Control," in *Proceedings of ACM SIGCOMM*, Portland, OR, August 2004.

[21] H. Kaur, S. Kalyanaraman, A. Weiss, S. Kanwar and A. Gandhi, "BANANAS: An Evolutionary Framework for Explicit and Multipath Routing in the Internet," *ACM SIGCOMM Computer Communication Review*, vol. 33, no. 4, 2003.

[22] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris, "Resilient Overlay Networks," in *Proceedings of the ACM Symposium on Operating Systems Principles*, Banff, Alberta, Canada, October 2001.

[23] CERT Coordination Center, "Denial of Service Attacks," http://www.cert.org/tech_tips/denial_of_service.html.