Cross-Layer Framework with Geodiverse Routing in Software-Defined Networking

Yufei Cheng*, Md. Moshfequr Rahman*, Siddharth Gangadhar*, Mohammed J.F. Alenazi*§, and James P.G. Sterbenz*†‡
*Information and Telecommunication Technology Center
Department of Electrical Engineering and Computer Science
The University of Kansas, Lawrence, KS, 66045, USA
§Department of Computer Engineering, King Saud University, Riyadh, Saudi Arabia
† School of Computing and Communications (SCC) and InfoLab21
Lancaster LA1 4WA, UK
‡Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong
{yfcheng|moshfequr|siddharth|malenazi|jpgs}@ittc.ku.edu, jpgs@comp.{lancs.ac.uk,polyu.edu.hk}
www.ittc.ku.edu/resilinets

***Abstract*—We propose a cross-layer routing framework in the SDN (software-defined networking) domain to cope with regionally-correlated challenges. By taking advantage of the failure detection model, GeoDivRP calculates multiple geodiverse paths for resilient network communications. Coupled with the optimization model, it realizes the minimized delay-skew product when decoupling traffic onto multiple paths. We evaluate our framework using MPTCP (Multipath TCP) in the face of regionally-correlated failures and it presents better performance compared to the single path routing. We further demonstrate our web framework to automate the OpenFlow experiment by programmatically importing network topologies and execute challenge emulations using the user-provided challenge regions.**

I. Introduction and Motivation

The demands for Internet resilience, survivability, and dependability have been increasing tremendously. Telecommunication networks are widely used for carrying Internet traffic and they rely heavily on the physical infrastructure such as optical fibers, routers, and switches to maintain normal operation; therefore, it is important to evaluate their resilience in the face of various faults and challenges [1]. Survivable optical networks under random edge and non-correlated failures have been a popular research domain [2], [3]. Recently, the research community has become more concerned about the potential damage caused by large-scale challenges and intentional attacks; efficient mechanisms have been proposed to mitigate their impacts [4]–[6]. However, none of the work has focused on a resilient cross-layer network architecture to cope with large-scale challenges.

To deal with the aforementioned challenges, a novel flow-diverse Internet protocol stack has been proposed to provide network protection and resilience by taking advantage of multiple geodiverse paths [7], [8]. It provides network protection mechanisms by preallocating multiple geodiverse paths for each communicating node pair for resilience purposes. GeoDivRP (GeoPath Diverse Routing Protocol) analyzes the network statistics collected from the link layer using the failure detection module, in particular the failed link set along with the links' delay and congestion information. GeoDivRP processes the failed link set and calculates a distance separation criteria d, which is used for the geodiverse path calculation. The link delay information, along with the skew requirement for the path set P_k, are passed to the optimization engine for traffic allocation optimization. The path set P_k, along with the optimized traffic allocation set X_k, are passed up to ResTP [8], our resilient transport protocol. ResTP can establish multiple flows between a pair of communicating hosts using the geodiverse paths provided by GeoDivRP for its data transmission. Applications can also pass down threat models to ResTP and further to GeoDivRP; diverse routes are created based on the given threat model.

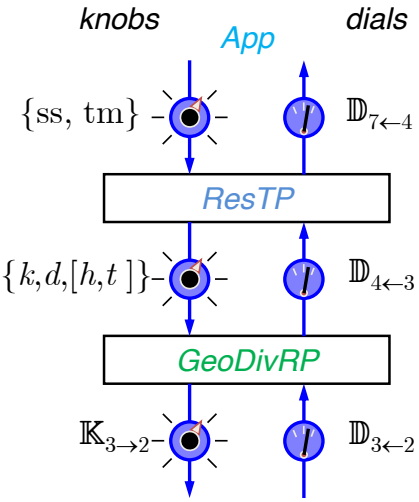


Fig. 1. Block diagram of GeoDivRP and ResTP

Network optimization is a popular research domain, and it performs best with a centralized topology view [9]. Direct control of the underlying network switches by a centralized OpenFlow controller in the SDN (software-defined network-

978-3-901882-77-7

ing) enables a number of mechanisms requiring the global view of the network topology. The data-centric design of SDN is inherent with the optimization model; it enables the controller to make optimized routing decisions using the centralized topology view. We have implemented our delay-skew minimization mechanism [10] along with our GeoDivRP routing algorithm [7] in SDN to realize optimized routing.

The paper is organized as follows. Section II describes the background and related work. Section III presents the novel cross-layer design within the flow-diverse protocol stack. Section IV presents experimental results using our web framework with regional challenges. Section V concludes this paper.

II. Background and related work

A large number of failures in a geographical region can result in catastrophic damage to network communications [6]. When regional challenges occur, a series of nodes and links in the vicinity can be damaged and removed from the network; these are regionally-correlated challenges. Since the challenge effect is frequently long-term [11], a set of backup paths are required for survivable routing. The single-location physical challenge scenario has been analyzed [5], [12], as well as correlated and simultaneous challenges [13]. A random line-cut mechanism has been used to assess the vulnerability to regional-based challenges [4]. Both correlated failures and targeted attacks with simulation results have been presented [14]. Our previous work has studied different mechanisms that identify the vulnerability area and routing algorithms to bypass the impact zone with a threat model [6]. Two heuristics are proposed for solving the d-distance separation paths problem (in which any two nodes on disjoint paths are separated by greater than d distance) and their effectiveness under regional challenges has been demonstrated [15]. A delay-skew minimization mechanism has been proposed for rerouting traffic [16]. However, a unified protocol stack is lacking to systematically remediate regional failure aftereffects. Therefore, it is important to understand the mechanism to statistically direct the rerouted traffic onto multiple d-distance separated paths and to then better cope with network congestion when large-scale challenges occur.

Resilient protocols are important for Future Internet design. Flow-diverse routing mechanisms have been proposed to solve the optical network diversity problem. A SRLG (shared risk link group) is a set of links that share a common physical resource, and it has been proposed to address single or multiple physical failures [17]. Minimum-cost diverse SRLG routing has been proven NP-complete and an ILP (Integer Linear Programming) formulation has been used to solve the routing problem [18]. Path protection has been proposed to provide two SRLG-disjoint paths using graph transformation techniques [19]. Furthermore, INLP (integer nonlinear programming) has been proposed to solve the problem of finding two disjoint paths with minimum-joint path failure probability in the face of probabilistic physical failures [20]. However, most of the work has focused on diverse routing in optical networks calculating only 2-diverse paths.

Flow-diverse routing mechanisms require multipath routing. Multipath routing is advantageous for small networks in the all-commodity traffic scenario [21] and it can be done in multiple layers. ECMP (equal-cost multipath) is achieved in the networking layer as a multipath routing strategy, which uses multiple paths with equal cost for better load-balancing in OSPF (Open Shortest Path First) [22]. Optimization maximizes the flow on each path in an ECMP routing algorithm [23]. Furthermore, multipath can be accomplished at the transport layer as well. MPTCP (Multipath TCP) [24], [25] is proposed as an extension for the current TCP to utilize multiple subflows. It is an ongoing effort of the IETF MPTCP working group [24]. By creating multiple subflows using different network paths and combining the received data, MPTCP can potentially improve throughput and resilience to network failure. Taking advantage of the centralized design of SDN, multipath routing can benefit from its efficient failure detection as well as fast response to topology changes.

SDN is the concept of using programmable components to control network behavior. By dividing the network control and forwarding functions, network services are abstracted from the underlying infrastructure. This enables rapid innovation as new versions of network software can be easily deployed. OpenFlow [26] is the first open standard southbound interface for SDN. It provides an open protocol to program the flow tables in the switches and enables researchers to test new network services along with real-world traffic without significant changes to the infrastructure. It realizes flexible and programmable data transmission through defined actions for each flow entry, while may include forwarding packets to ports, dropping packets. OpenFlow has a tiered architecture in which the southbound interface directly controls the network devices, and the northbound interface presents abstraction to the application for easier development.

III. Framework design

We design our cross-layer framework with the frontend representing topology and a backend OpenFlow module emulating network challenges. We have presented a demo [27] for the real-time operation of our framework at the 23rd GENI Engineering Conference [28]. As shown in Figure 2, the frontend system reads the adjacency matrix from KU-TopView [29], [30] and creates the topology automatically by overlaying it on top of the map with realistic delay and bandwidth configurations. OpenFlow switches are used to represent the network nodes in the physical topologies. The users interact with the system through a drag-and-drop polygon representing the challenge region. The polygon can be modified to any shape or size by users causing links or nodes that fall in the polygon area to fail. The challenge information is then passed to the backend system which runs the OpenFlow experiments. Physical OpenFlow switches are deployed in the KanREN testbed, while Mininet-emulated [31] topologies are used for all the other networks.

GeoDivRP (GeoPath Diverse Routing Protocol) [7] is implemented as an OpenFlow controller, which powers our

framework's backend. GeoDivRP is responsible for calculating d-distance separated paths. The optimization model is responsible for providing optimized traffic distribution for the path set calculated by GeoDivRP. The failure detection module is responsible for identifying failures that occur in the network and notifies GeoDivRP, while GeoDivRP dictates the detection interval.

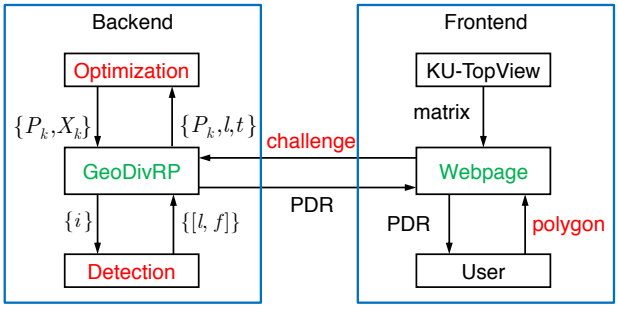


Fig. 2. Web framework for challenge emulation

A. GeoDivRP

GeoDivRP [7] provides multiple geodiverse paths to the higher layer for resilient traffic transmission. In order to decrease the complexity of the geodiverse path calculation, the iWPSP (iterative WayPoint Shortest Path) heuristic [15], [32] is used. As shown in Figure 3, for the case when the number of path for each node pair denoted as $k = 3$, iWPSP first calculates the shortest path p_s connecting source node v_{s} and destination node v_{d}. Assuming the next hop node for v_{s} and v_{d} on the shortest path are v_{s_0} and v_{d_0}, the algorithm selects neighbor nodes v_{s_1} and v_{d_1} that are d-distance separated from v_{s_0} and v_{d_0}, respectively. For simplicity, this work assumes that such nodes exist; otherwise, the nodes with the closest distance compared to the requirement will be chosen, iterating until nodes distance d apart are located. iWPSP selects waypoint nodes m' and m'' in the opposite direction that are distance $d+\delta$ apart from the middle node m in the shortest path, where the segment $m'mm''$ intersects the shortest path. Dijkstra's algorithm [33] is performed for the two branches $v_{\mathrm{s}_1}m'$ and $v_{\mathrm{d}_1}m'$. By connecting the shortest path returned from the two branches, the heuristic obtains the first geodiverse path. The same mechanism repeats for waypoint node m'' for the second geodiverse path. Variable d is a user-chosen parameter based on the threat model, and δ is experimentally chosen for different network topologies to increase the probability of the heuristic to return a d-separated path. The δ parameter is also useful in preventing the links of the two geodiverse paths from interleaving and creating routing loops. By adjusting the value of δ, the heuristic can select a nearby waypoint node if the previous one fails running Dijkstra's algorithm. When the heuristic cannot select paths within the skew bound t, the model increases or decreases δ accordingly. Geodiverse paths are passed to the optimization model along with the latency l and skew t requirement. The path-set P_k along with its flow distribution information X_k are passed to GeoDivRP for optimized traffic transmission.

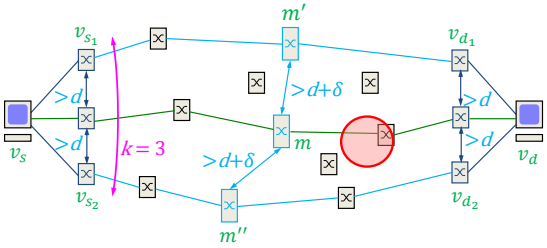


Fig. 3. Iterative waypoint shortest path heuristic

B. Failure detection module

Failure detection module (Figure 2) is an essential component for our customized controller to monitor link failures. The module provides network statistics, such as failed link set and link congestion level to GeoDivRP. GeoDivRP acts on this information and makes routing decisions such as the path to choose to reach the destination. Network statistics are acquired using OpenFlow Discovery Protocol (OFDP) [34]. The network devices advertise their link capacity and the controller constructs a centralized layer-2 network topology. As shown in Figure 2, the detection module collects network statistics and provides the link information l and the failure information f to GeoDivRP. GeoDivRP passes the detection interval i to the detection module.

C. Optimization model

The optimization model is implemented to solve the delay-skew optimization problem [10], [16] using the *IPOPT* solver [35] that comes with the *Pyomo* optimization framework [36], [37]. *IPOPT* (Interior Point OPTimizer) is an open-source solver for the large-scale linear and nonlinear optimization problems. The total variable size for the optimization problem is $D + L$, where D represents the total number of demands and L the number of links. It returns the optimized path set along with the flow distribution (P_k, X_k) to GeoDivRP for optimum network communication. P_k is the k number of d-distance separated geodiverse paths and X_k the optimized flow allocation on these paths. Optimization requires centralized view of the topology and SDN provides natural support with the controller collecting network statistics.

IV. Results and Discussion

Both OpenFlow testbed and Mininet-emulated topologies are used in our experiments. Mininet [31] is a network emulation tool that can create a complex OpenFlow supported topology. The experiments are used to demonstrate the protocol stack's performance in the face of regional challenges. Three geodiverse paths provided by GeoDivRP are used in the example topology for resilient routing.

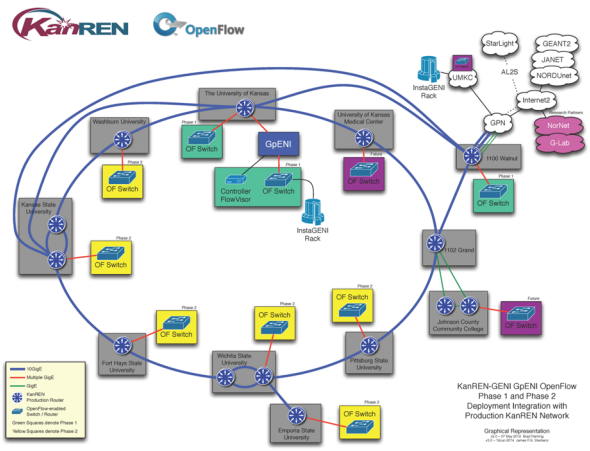


Fig. 4. KanREN OpenFlow switches deployment

A. OpenFlow testbed

In our testbed, we have deployed OpenFlow-enabled switches in KanREN (Kansas Research and Education Network) [www.kanren.net], which is a logical ring throughout the state of Kansas connecting institutions of higher education. Eight Brocade NetIron CES 2024C [38] OpenFlow switches have been deployed at these institutions, as shown in Figure 4. A full-mesh topology is deployed as an OpenFlow overlay and any arbitrary virtual topologies can be initialized through MPLS tunnels. A ring topology is used in our experiment. As shown in Figure 5a, the blue dots represent the Brocade OpenFlow switches and green solid lines the links. The red polygon represents the challenge region tunable by the users. ICMP (Internet Control Message Protocol) messages are used to evaluate the performance, with its PDR (packet delivery ratio) displayed on our website when the experiment is running. Floodlight is an OpenFlow controller based on Java [39], and it works with both physical- and virtual-switches. Our resilience routing framework is implemented based on Floodlight as a customized controller.

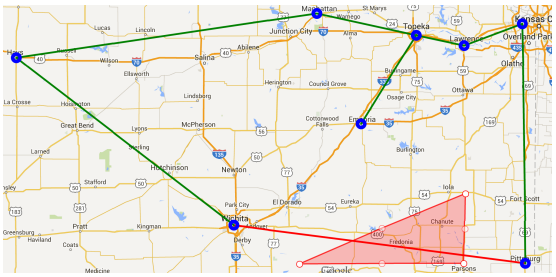
As shown in Figure 5a and 5b, the nodes and links that have been challenged are shown in red. The traffic is sent from Lawrence, KS to Kansas City, KS in our first scenario and Wichita, KS to Pittsburg, KS in the second. When the challenge takes down Wichita–Pittsburg and Lawrence–KC link as part of our challenges for each of our aforementioned scenarios, the traffic reroutes around the failure regions and the average rerouting delay is presented in Figure 5e. The trend for the end-to-end delay is similar when comparing both challenge scenarios. We observe an early high delay for the initial sample which is due to the initial packet trying to find the path to the destination. The next 24 packets have an average delay of 1 ms for Lawrence–KC and 4 ms for Wichita–Pittsburg. The challenge is applied at the 26th packet and is clearly shown by the middle delay spike in both of the challenge scenarios. Rerouting by the controller occurs and packets are routed through an alternate path with higher hops and higher delays than the unchallenged case in both of our scenarios.

B. Mininet experiment

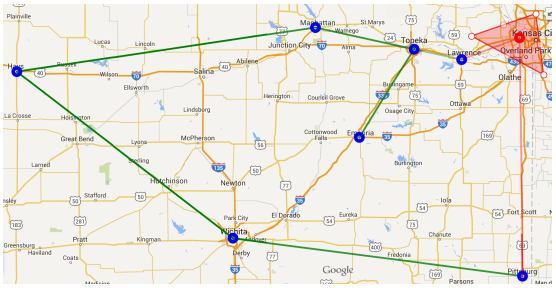
The second experiment begins with reading the adjacency matrix for different physical topologies and creating Mininet experiments programmatically with realistic delay and bandwidth configurations. The bandwidth used for this experiment is a uniform 1 Mb/s across all links and realistic delay parameters are chosen based on the physical distance between the respective hosts. OpenFlow switches are used to represent network nodes in the physical topologies (topology data from KU-TopView). The Sprint physical topology is used in this experiment with nodes shown in blue dots and links in green straight lines in Figure 5c and 5d. The traffic is sent from Seattle, WA to New York City, NY and Los Angeles, CA to Miami, FL for each of our scenarios. When the regional challenge occurs at Chicago and later at Dallas, the traffic is rerouted around the challenge and new path is calculated by the controller. The end-to-end delay for the above experiment is shown in Figure 5f. The result is similar to the KanREN challenge scenario discussed earlier. The initial delay spike is caused by path discovery in both of the cases. The delay for both scenarios is in the range of 50–60 ms for the next 24 packets when the network is unchallenged. The challenge is applied at the 26th packet and the new path discovery causes the delay spike shown in the middle of the graph. The spike is not as pronounced as the previous KanREN scenario as Sprint has a much larger topology with more links and nodes than KanREN has. This means that finding alternate paths is easier in the Sprint network thus reducing the rerouting delay. Once the challenge is applied, due to rerouting delay, the delay for the next set of packets is higher than the unchallenged ones.

We now study the physical switch topologies with MPTCP (Multipath TCP) [24]-enabled routers and a single sender and receiver. All the links' bandwidth are 10 Mb/s. The topology for the experiment is presented in Figure 6 where multiple paths exist between Lawrence and Wichita. A challenge profile in the Midwest is applied over the topology starting from Pittsburg and moving towards Topeka. The initial challenge takes effect at 30 seconds bringing down the Pittsburg switch. The next challenge occurs at Emporia starting at 60 seconds with the Pittsburg switch brought up. For the final challenge, the challenge circle then encompasses Topeka at 90 seconds with the Emporia switch brought up again. Finally, the challenge circle moves away from the topology with all switches up at that time. Since the paths calculated by GeoDivRP are d-distance separated, the challenge cannot affect two paths at the same time. Therefore, when transport-layer erasure coding [40] is applied, our protocol maintains normal communication throughout the regional challenge.

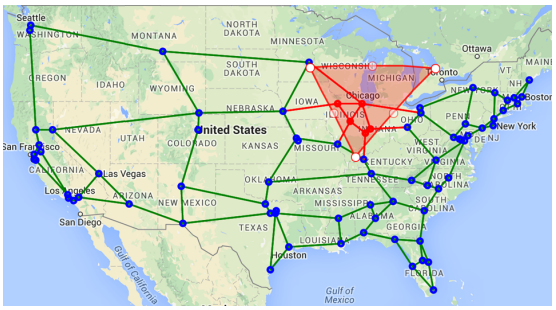
Results from the above challenge profile is shown in Figure 7. The traffic is generated using iPerf, a network framework for evaluating the network's maximum bandwidth. For the first 30 seconds, the throughput for different cities are close to 10 Mb/s, the link capacity. Starting at 30 seconds, the throughput of Pittsburg drops as the challenge is over Pittsburg. At the end of 60 seconds, Emporia drops off the



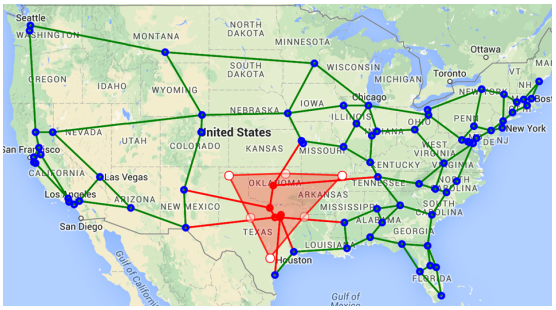
(a) KanREN OpenFlow switches failure scenario one



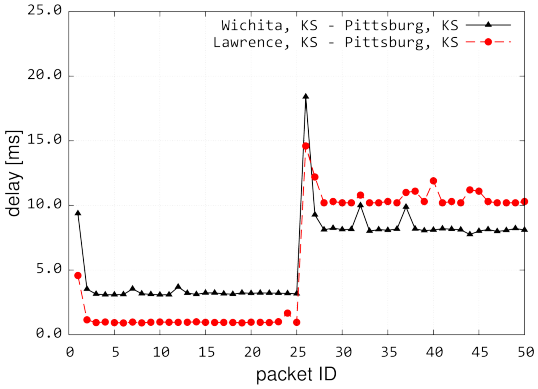
(b) KanREN OpenFlow switches failure scenario two



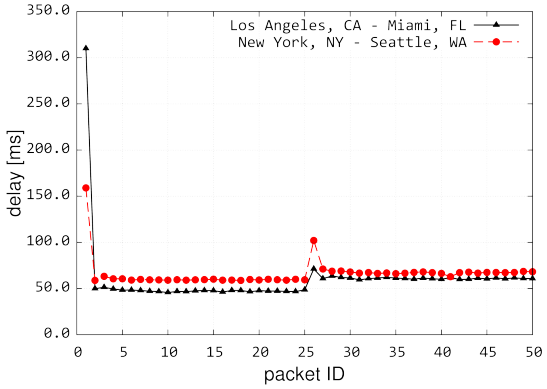
(c) Sprint network failure scenario one



(d) Sprint network failure scenario two



(e) KanREN OpenFlow switches delay



(f) Sprint OpenFlow switches delay

Fig. 5. Network failure scenarios and end-to-end delay result

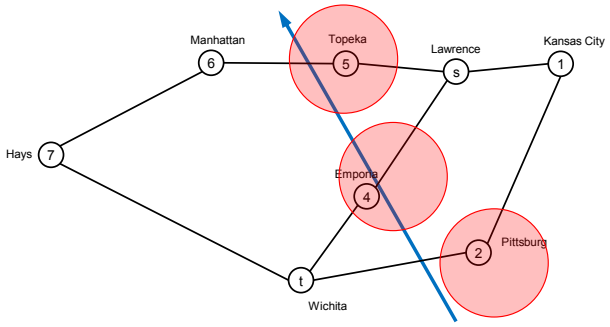


Fig. 6. KanREN OpenFlow network regional challenges

network while Pittsburg is brought up, which explains the rise in throughput for Pittsburg. After another 30 seconds, the challenge moves away from Emporia shown by the rise in throughput at 90 seconds and Topeka is challenged. After 30 seconds, the challenge moves away from Topeka shown by the rise in throughput at 120 seconds.

V. Conclusion

We presented our cross-layer routing framework considering geodiversity. By calculating and selecting single or multiple geographically diverse paths, it meets the requirements from higher network layers and demonstrates efficiency in routing around the challenged areas. Our framework takes advantage of the failure detection capability in OpenFlow and presents great flexibility and efficiency for implementing new routing mechanisms in OpenFlow. We plan to carry out more evaluation experiments comparing our framework to other multipath protection mechanisms, and different measurement metrics are planned to be employed.

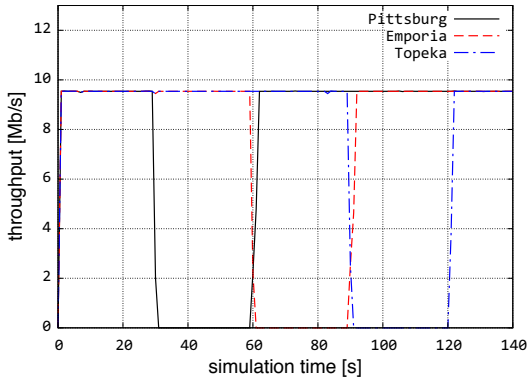


Fig. 7. KanREN challenge throughput result

ACKNOWLEDGMENT

The authors would like to thank the members of the ResiliNets group for discussions to led to this work. This research was supported in part by NSF grant CNS-1128122 and CNS-1219028.

REFERENCES

[1] J. P. G. Sterbenz, D. Hutchison, E. K. Çetinkaya, A. Jabbar, J. P. Rohrer, M. Schöller, and P. Smith, "Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines," *Computer Networks*, vol. 54, no. 8, pp. 1245–1265, 2010.

[2] R. Cohen, K. Erez, D. ben Avraham, and S. Havlin, "Resilience of the Internet to Random Breakdowns," *Phys. Rev. Lett.*, vol. 85, pp. 4626–4628, November 2000. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevLett.85.4626

[3] D. Magoni, "Tearing down the Internet," *IEEE J.Sel. A. Commun.*, vol. 21, no. 6, pp. 949–960, September 2006.

[4] S. Neumayer, G. Zussman, R. Cohen, and E. Modiano, "Assessing the Vulnerability of the Fiber Infrastructure to Disasters," *IEEE/ACM Transactions on Networking*, vol. 19, no. 6, pp. 1610–1623, 2011.

[5] S. Neumayer and E. Modiano, "Network Reliability With Geographically Correlated Failures," in *Proc. of IEEE INFOCOM*, March 2010, pp. 1–9.

[6] Y. Cheng, J. Li, and J. P. G. Sterbenz, "Path Geo-diversification: Design and Analysis," in *5th IEEE/IFIP RNDM*, Almaty, September 2013.

[7] Y. Cheng, M. T. Gardner, J. Li, R. May, D. Medhi, and J. P. Sterbenz, "Analysing GeoPath Diversity and Improving Routing Performance in Optical Networks," *Computer Networks*, vol. 82, pp. 50–67, May 2015.

[8] T. A. N. Nguyen, J. P. Rohrer, and J. P. G. Sterbenz, "ResTP–A Transport Protocol for FI Resilience," in *10th CFI*, June 2015.

[9] D. Medhi and K. Ramasamy, *Network Routing: Algorithms, Protocols, and Architectures*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2007.

[10] Y. Cheng and J. P. Sterbenz, "Geodiverse routing with path jitter requirement under regional challenges," in *6th RNDM*, Nov 2014, pp. 179–186.

[11] M. J. F. Denise M. B. Masi, Eric E. Smith, "Understanding and Mitigating Catastrophic Disruption and Attack," *Sigma Journal*, pp. 16–22, September 2010.

[12] W. Wu, B. Moran, J. Manton, and M. Zukerman, "Topology Design of Undersea Cables Considering Survivability Under Major Disasters," in *International Conference on WAINA*, May 2009, pp. 1154–1159.

[13] P. K. Agarwal, A. Efrat, S. Ganjugunte, D. Hay, S. Sankararaman, and G. Zussman, "Network Vulnerability to Single, Multiple, and Probabilistic Physical Attacks," in *2010 MILCOM*, 2010, pp. 1824–1829.

[14] E. K. Çetinkaya, D. Broyles, A. Dandekar, S. Srinivasan, and J. P. G. Sterbenz, "Modelling Communication Network Challenges for Future Internet Resilience, Survivability, and Disruption Tolerance: A Simulation-Based Approach," *Telecommunication Systems*, vol. 52, no. 2, pp. 751–766, 2013.

[15] Y. Cheng, M. T. Gardner, J. Li, R. May, D. Medhi, and J. P. Sterbenz, "Optimised Heuristics for a Geodiverse Routing Protocol," in *Proceedings of the IEEE 10th DRCN*, Ghent, Belgium, April 2014, pp. 1–9.

[16] Y. Cheng, D. Medhi, and J. P. Sterbenz, "Geodiverse Routing with Path Delay and Skew Requirement under Area-based Challenges," *Networks (Wiley)*, June 2015.

[17] D. Papadimitriou, F. Poppe, J. Jones, S. Venkatachalam, S. Dharanikota, R. Jain, R. Hartani, D. Griffith, and Y. Xue, "Inference of Shared Risk Link Groups," Internet Draft, May 2002. [Online]. Available: https://tools.ietf.org/html/draft-many-inference-srlg-02

[18] J. Q. Hu, "Diverse routing in optical mesh networks," *Communications, IEEE Transactions on*, vol. 51, no. 3, pp. 489–494, March 2003.

[19] P. Datta and A. K. Somani, "Graph transformation approaches for diverse routing in shared risk resource group (srrg) failures," *Computer Networks*, vol. 52, no. 12, pp. 2381–2394, 2008.

[20] H.-W. Lee, E. Modiano, and K. Lee, "Diverse Routing in Networks With Probabilistic Failures," *IEEE/ACM Transactions on Networking*, vol. 18, no. 6, pp. 1895–1907, 2010.

[21] X. Liu, S. Mohanraj, M. Pioro, and D. Medhi, "Multipath Routing from a Traffic Engineering Perspective: How Beneficial Is It?" in *22nd ICNP*, Oct 2014, pp. 143–154.

[22] C. Hopps, "Analysis of an Equal-Cost Multi-Path Algorithm," RFC 2992 (Informational), Internet Engineering Task Force, Nov. 2000. [Online]. Available: http://www.ietf.org/rfc/rfc2992.txt

[23] M. Dzida, M. Zagozdzon, M. Zotkiewicz, and M. Pioro, "Flow Optimization in IP Networks with Fast Proactive Recovery," in *Telecommunications Network Strategy and Planning Symposium, 2008. Networks 2008. The 13th International*, vol. Supplement, Sept 2008, pp. 1–9.

[24] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure, "TCP Extensions for Multipath Operation with Multiple Addresses," RFC 6824 (Experimental), Internet Engineering Task Force, Jan. 2013. [Online]. Available: http://www.ietf.org/rfc/rfc6824.txt

[25] C. Paasch and S. Barré. (2013, January) Multipath TCP in the Linux Kernel. http://www.multipath-tcp.org.

[26] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling Innovation in Campus Networks," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, Mar. 2008. [Online]. Available: http://doi.acm.org/10.1145/1355734.1355746

[27] Y. Cheng, M. J. Alenazi, M. M. Rahman, S. Gangadhar, and J. P. Sterbenz, "GpENI, KanREN, US Ignite Future Internet Testbed & Experiments (demo)," http://www.ittc.ku.edu/resilinets/GEC_DEMO/, June 2015.

[28] (2009, December) GENI: Global environment for network innovations. http://www.geni.net/.

[29] J. P. Rohrer, M. J. F. Alenazi, and J. P. G. Sterbenz. (2011, January) ResiliNets Topology Map Viewer. http://www.ittc.ku.edu/resilinets/maps/v1.

[30] A. Cosner, J. P. Rohrer, M. J. F. Alenazi, and J. P. G. Sterbenz. (2015, January) ResiliNets Topology Map Viewer Version 2. http://www.ittc.ku.edu/resilinets/maps/v2.

[31] (2010, July) An Instant Virtual Network on your Laptop (or other PC). http://www.mininet.org.

[32] M. Gardner, R. May, C. Beard, and D. Medhi, "Using Multi-Topology Routing to Improve Routing during Geographically Correlated Failures," in *10th DRCN*, April 2014, pp. 1–8.

[33] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959, 10.1007/BF01386390. [Online]. Available: http://dx.doi.org/10.1007/BF01386390

[34] OpenFlowDiscoveryProtocol. http://groups.geni.net/geni/wiki/OpenFlowDiscoveryProtocol.

[35] (2006, July) COmputational INfrastructure for Operations Research. http://www.coin-or.org/.

[36] (2009) Pyomo Optimization Framework. http://www.pyomo.org.

[37] W. E. Hart, J.-P. Watson, and D. L. Woodruff, "Pyomo: Modeling and Solving Mathematical Programs in Python," *Mathematical Programming Computation*, vol. 3, no. 3, pp. 219–260, 2011.

[38] (1995) http://www.brocade.com/. [Online]. Available: http://www.brocade.com/

[39] (2015) The Floodlight Controller. http://www.projectfloodlight.org/.

[40] J. W. Byers, M. Luby, M. Mitzenmacher, and A. Rege, "A digital fountain approach to reliable distribution of bulk data," *SIGCOMM Computer Communication Review*, vol. 28, no. 4, pp. 56–67, 1998.