ability are required. To this end, we introduce the Aspen tree — a multi-rooted tree topology with the ability to react to failures locally — and its corresponding failure notification protocol, ANP. Aspen trees provide decreased convergence times to improve a data center's availability, at the expense of scalability (e.g. reduced host count) or financial cost (e.g. increased network size). We provide a taxonomy for discussing the range of Aspen trees available given a set of input constraints and perform a thorough exploration of the tradeoffs between fault tolerance, scalability, and network cost in these trees.

12. ACKNOWLEDGMENTS

We thank the anonymous reviewers and our shepherd, Michael Schapira, for their valuable suggestions and feedback. We also thank John R. Douceur for his feedback and his insight into the complexities of increasing an Aspen tree's depth.

13. REFERENCES

[1] M. Al-Fares, A. Loukissas, and A. Vahdat. A Scalable, Commodity Data Center Network Architecture. ACM SIGCOMM 2008.

[2] A. Banerjea. Fault Recovery for Guaranteed Performance Communications Connections. *ToN*, 7(5):653–668, Oct 1999.

[3] Cisco Systems, Inc. OSPF Design Guide. `https://learningnetwork.cisco.com/docs/DOC-3046`, 2005.

[4] Cisco Systems, Inc. Cisco Data Center Infrastructure 2.5 Design Guide. `www.cisco.com/univercd/cc/td/doc/solution/`, 2008.

[5] C. Clos. A Study of Non-Blocking Switching Networks. *BSTF*, 32(2):406–424, Mar 1953.

[6] W. J. Dally and H. Aoki. Deadlock-Free Adaptive Routing in Multicomputer Networks Using Virtual Channels. *ToPaDS*, 4(4):466–475, Apr 1993.

[7] D. Dao, J. Albrecht, C. Killian, and A. Vahdat. Live Debugging of Distributed Systems. Springer-Verlag CC 2009.

[8] P. Francois. Achieving sub-second IGP convergence in large IP networks. *SIGCOMM CCR*, 35:2005, 2005.

[9] P. Gill, N. Jain, and N. Nagappan. Understanding Network Failures in Data Centers: Measurement, Analysis, and Implications. ACM SIGCOMM 2011.

[10] A. V. Goldberg, B. M. Maggs, and S. A. Plotkin. A Parallel Algorithm for Reconfiguring a Multibutterfly Network with Faulty Switches. *ToC*, 43(3):321–326, Mar 1994.

[11] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta. VL2: A Scalable and Flexible Data Center Network. ACM SIGCOMM 2009.

[12] R. I. Greenberg. *Efficient Interconnection Schemes for VLSI and Parallel Computation*. PhD thesis, MIT, 1989.

[13] R. I. Greenberg and C. E. Leiserson. Randomized Routing on Fat-Trees. In *Advances in Computing Research*, pages 345–374. JAI Press, 1996.

[14] C. Guo, G. Lu, D. Li, H. Wu, X. Zhang, Y. Shi, C. Tian, Y. Zhang, and S. Lu. BCube: A High Performance, Server-Centric Network Architecture for Modular Data Centers. ACM SIGCOMM 2009.

[15] C. Guo, H. Wu, K. Tan, L. Shi, Y. Zhang, and S. Lu. DCell: A Scalable and Fault-Tolerant Network Structure for Data Centers. ACM SIGCOMM 2008.

[16] S. Han and K. G. Shin. Fast Restoration of Real-time Communication Service from Component Failures in Multi-hop Networks. ACM SIGCOMM 1997.

[17] M. Karol, S. J. Golestani, and D. Lee. Prevention of Deadlocks and Livelocks in Lossless Backpressured Packet Networks. *ToN*, 11(6):923–934, Dec 2003.

[18] C. Killian, J. W. Anderson, R. Jhala, and A. Vahdat. Life, Death, and the Critical Transition: Finding Liveness Bugs in Systems Code. USENIX NSDI 2007.

[19] C. Killian, J. W. Anderson, R. B. R. Jhala, and A. Vahdat. Mace: Language Support for Building Distributed Systems. ACM PLDI 2007.

[20] C. Killian, K. Nagarak, S. Pervez, R. Braud, J. W. Anderson, and R. Jhala. Finding Latent Performance Bugs in Systems Implementations. ACM FSE 2010.

[21] A. Kvalbein, A. F. Hansen, T. Cicic, S. Gjessing, and O. Lysne. Fast IP Network Recovery Using Multiple Routing Configurations. IEEE INFOCOM 2006.

[22] K. Lakshminarayanan, M. Caesar, M. Rangan, T. Anderson, S. Shenker, and I. Stoica. Achieving Convergence-Free Routing using Failure-Carrying Packets. ACM SIGCOMM 2007.

[23] F. T. Leighton and B. M. Maggs. Fast Algorithms for Routing Around Faults in Multibutterflies and Randomly-Wired Splitter Networks. *ToC*, 41(5):578–587, May 1992.

[24] C. E. Leiserson. Fat-Trees: Universal Networks for Hardware-Efficient Supercomputing. *ToC*, 34(10):892–901, Oct 1985.

[25] K. Levchenko, G. M. Voelker, R. Paturi, and S. Savage. XL: An Efficient Network Routing Algorithm. ACM SIGCOMM 2008.

[26] J. Liu, A. Panda, A. Singla, B. Godfrey, M. Schapira, and S. Shenker. Ensuring Connectivity via Data Plane Mechanisms. USENIX NSDI 2013.

[27] V. Liu, D. Halperin, A. Krishnamurthy, and T. Anderson. F10: A Fault-Tolerant Engineered Network. USENIX NSDI 2013.

[28] J. C. Mogul, J. Tourrilhes, P. Yalagandula, P. Sharma, A. R. Curtis, and S. Banerjee. DevoFlow: Cost-Effective Flow Management for High Performance Enterprise Networks. ACM HotNets 2010.

[29] J. Moy. OSPF version 2. RFC 2328, IETF, 1998.

[30] R. Niranjan Mysore, A. Pamboris, N. Farrington, N. Huang, P. Miri, S. Radhakrishnan, V. Subramanya, and A. Vahdat. PortLand: A Scalable Fault-Tolerant Layer 2 Data Center Network Fabric. ACM SIGCOMM 2009.

[31] C. Raiciu, S. Barre, C. Pluntke, A. Greenhalgh, D. Wischik, and M. Handley. Improving Datacenter Performance and Robustness with Multipath TCP. ACM SIGCOMM 2011.

[32] A. Singla, C.-Y. Hong, L. Popa, and P. B. Godfrey. Jellyfish: Networking Data Centers Randomly. USENIX NSDI 2012.

[33] L. Song and B. Mukherjee. On the Study of Multiple Backups and Primary-Backup Link Sharing for Dynamic Service Provisioning in Survivable WDM Mesh Networks. *JSAC*, 26(6):84 –91, Aug 2008.

[34] E. Upfal. An O(logN) Deterministic Packet Routing Scheme. ACM STOC 1989.

[35] M. Walraed-Sullivan, R. N. Mysore, M. Tewari, Y. Zhang, K. Marzullo, and A. Vahdat. ALIAS: Scalable, Decentralized Label Assignment for Data Centers. ACM SOCC 2011.