AGG_d and an AGG_u or between an AGG_u and a *CORE*. We then use the 144 TMs from the preceding 24 hours, rather than just the TM right before a deactivation action, to plan mitigation actions. We then measure the packet losses during the following 24 hours using the TMs in those hours.

Figure 19 presents the packet losses under each method. As can be seen, NetPilot has the fewest packet losses when using multiple historical TMs to approximate future TMs.

7. RELATED WORK

The design and implementation of NetPilot parallel the ideas from various fields. We group the related work into three categories: failure diagnosis, failure recovery, and what-if analysis.

Failure diagnosis: Automated failure diagnosis is a well-studied topic. Some systems identify failures in IP networks with active probes [6]. Others take measurement data from both end hosts and the network and build probabilistic models to localize the most likely components that are responsible for the observed failure data [15, 19]. Recent work has focused on developing systems that can pinpoint any failure that decreases application performance, whether it be hardware-related or software-related [5, 16]. There is also a large body of work on distributed system diagnosis [3, 7, 23]. NetPilot distinguishes itself by not attempting to find the exact root cause of a failure.

Failure recovery: The idea of automated failure recovery in DCs is not new. Isard [14] proposed an automated server management system called Autopilot based on the concept of recovery oriented computing [9]. When Autopilot detects a server is misbehaving, it takes one of three recovery actions: restart, reimage, or RMA (return merchandise authorization).

R3 [31] is a recovery service that can quickly mitigate link failures by pre-computing forwarding table updates for the loss of each link. Outside the networking domain, Total Recall [18] is a distributed storage system that adapts the amount of redundancy to compensate for host availability changes. Saxons is a peer-to-peer overlay service that can heal itself in the event of a partition [26]. NetPilot has different requirements from the systems above: failures that span multiple devices and constantly changing traffic loads make pre-computation prohibitively expensive. Also, NetPilot must minimize the adverse impact caused by mitigation.

What-if analysis: In order to make an informed decision, it is crucial for NetPilot to reason about the impact of possible mitigation actions – in essence a what-if analysis on the network. Recent work has explored the subject of what-if analysis for content distribution networks, with Tariq *et al.* [28] proposing a statistical approach and Wang *et al.* [30] proposing an empirical approach. Unlike the work above, NetPilot uses a what-if analysis technique that takes advantage of the unique properties of DCNs.

8. CONCLUSION

NetPilot is a system that automatically mitigates DCN failures. It is a departure from the status quo that relies heavily on human intervention. We believe that our work is critical to managing modern DCNs given the ballooning number of devices in these DCNs and the trend towards commodity hardware. NetPilot works by identifying a candidate set of afflicted components that are likely to cause a problem and iteratively taking mitigation actions targeting each one until the problem is alleviated. A key insight that makes this approach viable is the redundancy presented in modern DCN topologies. This redundancy reduces the potential for any single deactivated or rebooted component to disrupt a network. Our experiments show that NetPilot can successfully detect and mitigate several common types of failures both in a testbed and in a real production DCN.

9. ACKNOWLEDGMENTS

The authors would like to thank our shepherd Brad Karp and anonymous reviewers for their help in shaping this paper into its final form. We would also like to thank the infrastructure team at Microsoft Bing whose help was invaluable.

This work was largely done at Microsoft. Part of Xin Wu and Xiaowei Yang's work was supported by NSF awards CNS-0845858 and CNS-1040043.

10. REFERENCES

[1] Cut-Through and Store-and-Forward Ethernet Switching for Low-Latency Environments. `http://www.cisco.com/en/US/prod/collateral/switches/ps9441/ps9670/white_paper_c11-465436.pdf`.

[2] Virtual PortChannels: Building Networks without Spanning Tree Protocol. `http://www.cisco.com/en/US/prod/collateral/switches/ps9441/ps9402/white_paper_c11-516396.html`.

[3] M. K. Aguilera, J. C. Mogul, J. L. Wiener, P. Reynolds, and A. Muthitacharoen. Performance debugging for distributed systems of black boxes. SOSP '03.

[4] M. Al-Fares, A. Loukissas, and A. Vahdat. A scalable, commodity data center network architecture. In *SIGCOMM '08*.

[5] P. Bahl, R. Chandra, A. Greenberg, S. Kandula, D. A. Maltz, and M. Zhang. Towards highly reliable enterprise network services via inference of multi-level dependencies. In *SIGCOMM '07*.

[6] D. Banerjee, V. Madduri, and M. Srivatsa. A Framework for Distributed Monitoring and Root Cause Analysis for Large IP Networks. In *SRDS '09*.

[7] M. Y. Chen, A. Accardi, E. Kiciman, J. Lloyd, D. Patterson, A. Fox, and E. Brewer. Path-based failure and evolution management. In *NSDI '04*.

[8] W. Dally and B. Towles. *Principles and Practices of Interconnection Networks*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2003.

[9] D. P. et. al. Recovery Oriented Computing: Motivation, Definition, Techniques, and Case Studies. Technical report, Berkeley Computer Science, 2002.

[10] 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. In *SIGCOMM '09*.

[11] U. Hoelzle and L. A. Barroso. *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines*. Morgan and Claypool Publishers.

[12] C. Hopps. Analysis of an Equal-Cost Multi-Path Algorithm. RFC 2992, 2000.

[13] IEEE. 802.3ad Link Aggregation Standard.

[14] M. Isard. Autopilot: automatic data center management. *SIGOPS Oper. Syst. Rev.*, 41:60–67, April 2007.

[15] S. Kandula, D. Katabi, and J.-P. Vasseur. Shrink: a tool for failure diagnosis in IP networks. In *MineNet '05*.

[16] S. Kandula, R. Mahajan, P. Verkaik, S. Agarwal, J. Padhye, and P. Bahl. Detailed diagnosis in enterprise networks. In *SIGCOMM '09*.

[17] S. Kandula, S. Sengupta, A. Greenberg, P. Patel, and R. Chaiken. The nature of data center traffic: measurements & analysis. In *IMC '09*.

[18] R. B. Kiran, K. Tati, Y. chung Cheng, S. Savage, and G. M. Voelker. Total Recall: System Support for Automated Availability Management. In *NSDI '04*.

[19] Kompella, R.R and Yates, Jennifer, and Greenberg, Albert and Snoeren, Alex. IP Fault Localization Via Risk Modeling. In *NSDI '05*.

[20] C. Lonvick. The BSD Syslog Protocol. RFC 3164, 2001.

[21] 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. In *SIGCOMM '09*.

[22] R. Presuhn. Management Information Base (MIB) for the Simple Network Management Protocol (SNMP). RFC 3418, 2002.

[23] P. Reynolds, J. L. Wiener, J. C. Mogul, M. K. Aguilera, and A. Vahdat. WAP5: black-box performance debugging for wide-area systems. In *WWW '06*.

[24] S. Nadas, Ericsson. Virtual Router Redundancy Protocol (VRRP) Version 3 for IPv4 and IPv6. *Internet RFC 5798*, 2010.

[25] A. Shaikh, C. Isett, A. Greenberg, M. Roughan, and J. Gottlieb. A case study of OSPF behavior in a large enterprise network. In *IMW '02*.

[26] K. Shen. Structure management for scalable overlay service construction. In *NSDI '04*.

[27] T. Li, B. Cole, P. Morton, D. Li. Cisco Hot Standby Router Protocol (HSRP). RFC 2281, 1998.

[28] M. Tariq, A. Zeitoun, V. Valancius, N. Feamster, and M. Ammar. Answering what-if deployment and configuration questions with wise. In *SIGCOMM '08*.

[29] A. A. Team. Summary of the Amazon EC2 and Amazon RDS Service Disruption in the US East Region. `http://aws.amazon.com/message/65648/`.

[30] Y. Wang, C. Huang, J. Li, and K. Ross. Estimating the performance of hypothetical cloud service deployments: A measurement-based approach. In *INFOCOM '11*.

[31] Y. Wang, H. Wang, A. Mahimkar, R. Alimi, Y. Zhang, L. Qiu, and Y. R. Yang. R3: resilient routing reconfiguration. In *SIGCOMM '10*.