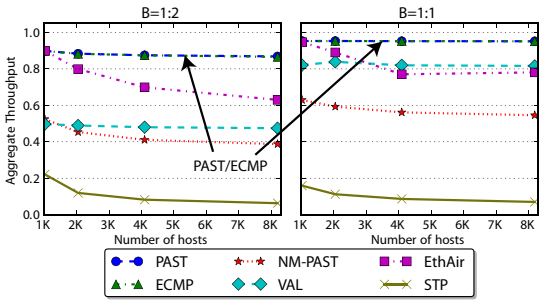

Figure 6: Throughput Comparison of Routing Algorithms Variants for the URand-8 Workload and HyperX Topology

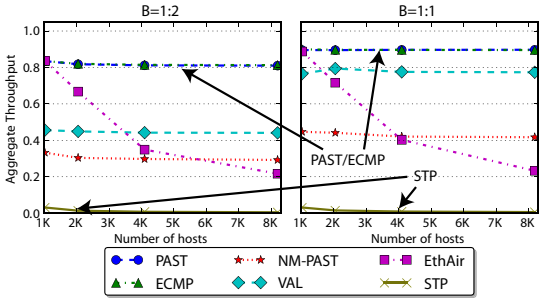

Figure 7: Throughput Comparison of Routing Algorithms Variants for the Shuffle Workload and HyperX topology

sults for the HyperX topology and 1 : 2 and 1 : 1 oversubscription ratio to save space—the other topologies had similar trends.

Our first observation is that the PAST and ECMP routing algorithms perform identically under all workloads. Also, both algorithms perform within 10% of optimal throughput on the 1 : 2 oversubscription ratio topologies for the URand and Shuffle workloads.

NM-PAST uses some minimal paths and does not choose a new random intermediate switch for each flow, so we expect the performance of VAL to be greater than that of NM-PAST. This is the case for the URand and shuffle workloads, but as seen in Figure 5, the NM-PAST and VAL routing algorithms perform similarly on the Stride workload. They are the best performing algorithms, even though the throughput of VAL is far from the optimal throughput, which is equal to the oversubscription ratio [24]. The Stride workload only has a single flow per host, which can cause hash collisions. As the number of flows increase, we anticipate that the throughput of both NM-PAST and VAL will increase.

Overall, the performance of EthAIR is poor. Figure 6 shows that, under the URand workload, the performance of EthAIR is 17%-48% worse than ECMP and PAST. Similarly, under the more demanding shuffle workload shown in Figure 7, EthAIR performs 71%-92% worse than ECMP and PAST at the largest topology size.

The STP algorithm is presented as a strawman to show the baseline performance of traditional Ethernet, even if all broadcasts are disallowed. STP performs significantly worse than all of the other routing algorithms on every topology and bisection bandwidth. This is expected because restricting forwarding to a single tree forces flows to collide.

Table 4 shows the scalability in terms of the number of physical hosts for each of the routing algorithms described earlier, if the routing algorithm were implemented using a network of Trident-based switches. The wildcard ECMP algorithm includes both wildcard ECMP routing on an EGFT topology as well as using EthAir to perform wildcard ECMP on arbitrary topologies. Although both algorithms scale to arbitrary network sizes, the EthAir algorithm restricts the set of usable paths in the network, which reduces performance. Both PAST and STP only require one Ethernet table entry per routable address, so the scalability of both PAST and STP is only limited to the size of the Ethernet table. TRILL ECMP does not scale to networks as large as can be supported by PAST and STP because TRILL ECMP requires ECMP table state, which is exhausted. The scalability of TRILL ECMP is a range because the required number of ECMP table entries varies across topologies and bisection bandwidths.

Wildcard ECMP	PAST	STP	TRILL ECMP
∞	$\sim$100K	$\sim$100K	$\sim$12K-55K

Table 4: Maximum Number of Physical Hosts for Different Eager Routing Algorithms Implemented with Broadcom Trident Chip

6. RELATED WORK

In this section, we discuss the design of PAST in the context of related network architectures. To save space, we omit architectures that have already been discussed, including PortLand, SEATTLE, and Ethernet on AIR. For the sake of discussion, we group related architectures together by the following properties: spanning tree algorithms, link-state routing protocols, and SDN architectures.

MSTP [18], SPAIN [30], and GOE [19] are all architectures that build a spanning tree per VLAN. None of them meet our requirements. MSTP does not achieve high performance because all traffic for a given VLAN is still restricted to a single spanning tree. SPAIN solves this problem by modifying hosts to load balance across VLANs, but SPAIN violates layering and does not scale because each host requires an Ethernet table entry per VLAN. GOE assigns each switch a VLAN and uses MSTP to build a unique spanning tree for each switch. This design limits the total network size to roughly 2K switches and decreases available path diversity and performance compared to PAST. Additionally, all of these architectures limit performance and scalability by requiring broadcast for address learning.

TRILL [20, 36] and Shortest-Path Bridging (SPB) [10] both use IS-IS link-state routing instead of the traditional spanning tree protocol. IS-IS may either use single path or multipath routing. Single path routing limits TRILL to forwarding on switch addresses instead of host addresses, and in SPB it restricts the number of forwarding trees. Using ECMP for multipath routing improves the performance of both architectures, but limits their scalability to the size of the ECMP table. As a result, PAST is more scalable. TRILL and SPB both require specific hardware support that is present in some but not all commodity switch chips, while PAST is designed to use the same hardware as classic Ethernet.

Hedera [4] and DevoFlow [7] are SDN architectures that provide additional functionality compared to PAST. Hedera, DevoFlow, and PAST all eliminate broadcasts and eagerly install routes, but Hedera and DevoFlow both improve performance by explicitly scheduling large flows onto better paths. PAST can complement these traffic engineering mechanisms by efficiently routing flows that are too small to merit explicit scheduling (non-elephant flows). We plan to explore traffic engineering in conjunction with PAST, which will benefit from the fact that PAST does not use the TCAM, so all TCAM entries can be used for traffic engineering.

7. CONCLUSIONS AND FUTURE WORK

Data center network designs are migrating from low-bisection-bandwidth single-rooted trees with hybrid Ethernet/IP forwarding to more sophisticated topologies that provide substantial performance benefits through multipathing. Unfortunately, existing Ethernet switches cannot efficiently route on multipathed networks, so many researchers have proposed using programmable switches (e.g., with OpenFlow) to implement high-performance routing and forwarding. Unfortunately, most OpenFlow firmware implementations and other architectures do not exploit the full capabilities of modern Ethernet switch chips.

In this paper, we presented PAST, a flat layer-2 data center network architecture that supports full host mobility, high end-to-end bandwidth, self-configuration, and tens of thousands of hosts using Ethernet switches built from commodity switch chips. We demonstrate that by designing a network architecture with explicit consideration for switch functionality—in particular the exact-match Ethernet table—it is possible to support heavily multipathed topologies that allow cost and performance tradeoffs. We show that PAST is able to provide near-optimal throughput without using a Fat Tree network. We further show that it is possible to perform efficient multipath routing without using ECMP or similar hashing hardware, which simplifies route computation and installation and could reduce hardware complexity because using ECMP is guaranteed to require more hardware than PAST. Finally, we show that PAST can be easily extended to provide non-shortest-path routing, which benefits adversarial workloads. In the worst case, PAST performs the same as ECMP, while in the best case PAST more than doubles the performance of ECMP.

PAST has implications for the design of future Ethernet switch chips, since our results indicate that layer-2 ECMP is not as useful (or necessary) as previously assumed. We believe PAST will scale well with future networks because the SRAM-based Ethernet table is area-efficient and can easily be increased in size, while it is costly to increase TCAM table size.

Although much has been written about network topologies, we have presented the first three-way comparison between EGFT, HyperX, and Jellyfish. We have also evaluated oversubscribed networks, revealing that in some cases they provide very similar performance to full-bisection-bandwidth topologies but at lower cost. In general, we agree with previous work that Fat Trees are not ideal for any use case. Our work does not provide insight regarding whether HyperX or Jellyfish is the better topology; the outcome is likely to depend on practical considerations, such as ease of cabling, and thus may vary between data centers.

We are excited by the potential of PAST for supporting large enterprise and cloud data centers. We plan to extend it in a number of ways. For example, we are working on an online variant of the per-address spanning tree algorithm that attempts to minimize the amount of new state that needs to be computed and installed when the physical topology or set of addressable hosts changes. We also plan to develop a more detailed cost model for comparing equal-performance HyperX and Jellyfish topologies. Finally, we are exploring ways to integrate traffic engineering, traffic steering, converged storage, high availability, and other advanced networking features into our PAST architecture.

8. ACKNOWLEDGEMENTS

We thank our shepherds, Andrew Moore and Chuanxiong Guo, and the anonymous reviewers for their comments. We also thank Joe Tardo and Rochan Sankar from Broadcom for providing detailed information about the Trident architecture and permission to publish some details here.

References

[1] D. Abts and J. Kim. *High Performance Datacenter Networks: Architectures, Algorithms, and Opportunities*. Morgan and Claypool, 2011.

[2] J. H. Ahn, N. Binkert, A. Davis, M. McLaren, and R. S. Schreiber. Hyperx: topology, routing, and packaging of efficient large-scale networks. *SC Conference*, 2009.

[3] M. Al-Fares, A. Loukissas, and A. Vahdat. A scalable, commodity data center network architecture. In *SIGCOMM*, 2008.

[4] M. Al-fares, S. Radhakrishnan, B. Raghavan, N. Huang, and A. Vahdat. Hedera: Dynamic flow scheduling for data center networks. In *NSDI*, 2010.

[5] M. Alizadeh, A. Greenberg, D. A. Maltz, J. Padhye, P. Patel, B. Prabhakar, S. Sengupta, and M. Sridharan. Data center TCP (DCTCP). In *SIGCOMM*, 2010.

[6] Broadcom BCM56846 StrataXGS 10/40 GbE Switch. http://www.broadcom.com/products/features/BCM56846.php.

[7] A. R. Curtis, J. C. Mogul, J. Tourrilhes, and P. Yalagandula. DevoFlow: Scaling flow management for high-performance networks. In *SIGCOMM*, 2011.

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

[9] N. Farrington, E. Rubow, and A. Vahdat. Data center switch architecture in the age of merchant silicon. In *Hot Interconnects*, 2009.

[10] D. Fedyk, P. Ashwood-Smith, D. Allan, A. Bragg, and P. Unbehagen. IS-IS Extensions Supporting IEEE 802.1aq Shortest Path Bridging. RFC 6329, Apr 2012.

[11] Floodlight openflow controller. http://floodlight.openflowhub.org/.

[12] I. Gashinsky. SDN in warehouse scale datacenter v2.0. In *Open Networking Summit*, 2012.

[13] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel. The cost of a cloud: Research problems in data center networks. In *ACM CCR*, January 2009.

[14] 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*, 2009.

[15] Greg Linden. Make data useful. http://www.scribd.com/doc/4970486/Make-Data-Useful-by-Greg-Linden-Amazoncom, 2006.

[16] C. Guo, G. Lu, H. J. Wang, S. Yang, C. Kong, P. Sun, W. Wu, and Y. Zhang. SecondNet: A data center network virtualization architecture with bandwidth guarantees. In *Co-NEXT*, 2010.

[17] IBM BNT RackSwitch G8264. http://www.redbooks.ibm.com/abstracts/tips0815.html.

[18] IEEE. *Std 802.1s Multiple Spanning Trees*. 2002.

[19] A. Iwata, Y. Hidaka, M. Umayabashi, N. Enomoto, and A. Arutaki. Global Open Ethernet (GOE) system and its performance evaluation. *Selected Areas in Communications, IEEE Journal on*, 2004.

[20] J. Touch and R. Perlman. Transparent Interconnection of Lots of Links (TRILL): Problem and Applicability Statement. RFC 5556, May 2009.

[21] S. Jain, Y. Chen, Z.-L. Zhang, and S. Jain. Viro: A scalable, robust and namespace independent virtual id routing for future networks. In *INFOCOMM*, 2011.

[22] S. Kandula, S. Sengupta, A. Greenberg, and P. Patel. The nature of datacenter traffic: Measurements and analysis. In *IMC*, 2009.

[23] C. Kim, M. Caesar, and J. Rexford. Floodless in SEATTLE: A scalable Ethernet architecture for large enterprises. In *Proceedings of ACM SIGCOMM*, 2008.

[24] J. Kim and W. J. Dally. Flattened butterfly: A cost-efficient topology for high-radix networks. In *ISCA*, 2007.

[25] G. Lu, C. Guo, Y. Li, Z. Zhou, T. Yuan, H. Wu, Y. Xiong, R. Gao, and Y. Zhang. ServerSwitch: A programmable and high performance platform for data center networks. In *NSDI*, 2011.

[26] G.-H. Lu, S. Jain, S. Chen, and Z.-L. Zhang. Virtual id routing: A scalable routing framework with support for mobility and routing efficiency. In *MobiArch*, 2008.

[27] M. Mahalingam, D. Dutt, K. Duda, P. Agarwal, L. Kreeger, T. Sridhar, M. Bursell, and C. Wright. VXLAN: A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks. Internet-Draft draft-mahalingam-dutt-dcops-vxlan-00.txt, IETF Secretariat, Jan. 2012.

[28] A. M. Malcolm Scott and J. Crowcroft. Addressing the scalability of Ethernet with MOOSE. In *DC-CAVES*, 2009.

[29] MC-LAG. http://en.wikipedia.org/wiki/MC_LAG.

[30] J. Mudigonda, P. Yalagandula, M. Al-Fares, and J. C. Mogul. SPAIN: COTS data-center Ethernet for multipathing over arbitrary topologies. In *NSDI*, 2010.

[31] J. Mudigonda, P. Yalagandula, and J. C. Mogul. Taming the flying cable monster: a topology design and optimization framework for data-center networks. In *USENIXATC*, 2011.

[32] J. Mudigonda, P. Yalagandula, J. C. Mogul, B. Stiekes, and Y. Pouffary. NetLord: a scalable multi-tenant network architecture for virtualized datacenters. In *SIGCOMM*, pages 62–73, 2011.

[33] R. N. 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*, 2009.

[34] OFlops. http://www.openflow.org/wk/index.php/Oflops.

[35] S. Ohring, M. Ibel, S. Das, and M. Kumar. On generalized fat trees. *Parallel Processing Symposium, International*, 0:37, 1995.

[36] R. Perlman. Rbridges: Transparent routing. In *INFOCOMM*, 2004.

[37] D. Sampath, S. Agarwal, and J. Gacia-Luna-Aceves. 'ethernet on air' : Scalable routing in very large ethernet-based networks. In *ICDCS*, 2010.

[38] A. Singla, C.-Y. Hong, L. Popa, and P. B. Godfrey. Jellyfish: Networking data centers randomly. In *NSDI*, April 2012.

[39] D. Wischik, C. Raiciu, A. Greenhalgh, and M. Handley. Design, implementation and evaluation of congestion control for multipath TCP. In *NSDI*, 2011.