an always connected control plane that allows it to schedule and route efficiently. On a tangential but still related area, [23] shows that mobility combined with storage increases the capacity of wireless DTNs. *NetStitcher* does the same by using storage to stitch together leftover capacity that would otherwise be wasted due to time misalignments.

Breitgand et al. [15] used store and forward to deliver individual low priority messages of a network monitoring system over a single path. They proposed an online algorithm to deal with unpredictable changes in resource availability. We target multipath bulk transfers in mostly predictable environments, which dictates the reduction of our problem to an efficiently computable maximum flow formulation. We address unpredictability by recomputing the schedule sufficiently often and when component failure is detected.

Point-to-point multipath bulk transfers: Pucha et al.'s dsync [34] is a file transfer system that adapts to network conditions by determining which of its available resources (e.g. network, disk, network peers) is the best to use at any given time and by exploiting file similarity [33]. *NetStitcher* does not rely on network peers that store content of interest in advance, and it has different design goals. Signiant [8] and Riverbed [7] offer point-to-point intra-business content transfer and application acceleration platforms, which utilize WAN optimization technologies. One could view *NetStitcher* as an additional WAN optimization technology. To the best of our knowledge, Signiant and RiverBed do not employ store and forward techniques.

Kokku et al. [27] focus on the TCP implications of multipath background transfers. They do not consider store and forward, but instead aim at making bulk background transfers more TCP-friendly, while maintaining efficiency.

Single-hop and single-path SnF bulk transfers: The closest work to *NetStitcher* is [30, 28], which developed analytical models for transferring bulk data through single-hop and single-path transfers while minimizing 95th-percentile transit costs for Telcos. It quantified the cost for transferring bulk scientific data across time zones and compared it to the use of postal service. That work did not address design and implementation issues, whereas *NetStitcher* is a system deployed in a production setting for SnF bulk transfers. In this work, we extend SnF scheduling beyond the simple case of single-path and single-hop over a single unconstrained storage node presented in [30].

10. CONCLUSIONS AND FUTURE WORK

We have presented the design, implementation, and validation of *NetStitcher*. It is a system for stitching together unutilized bandwidth across different datacenters, and using it to carry inter-datacenter bulk traffic for backup, replication, or data migration applications. *NetStitcher* bypasses the problem of misaligned leftover bandwidth by using scheduled multipath and multi-hop store-and-forward through intermediate storage nodes. As future work, we consider results on bootstrapping deployments and on security. An additional interesting topic is to adapt our system for operations in a P2P setting (higher churn, free-riding, etc).

11. REFERENCES

[1] Amazon Simple Storage Service. aws.amazon.com/s3/.
[2] Datacenter map. www.datacentermap.com/datacenters.html.
[3] Equinix datacenter map. www.equinix.com/data-center-locations/map/.
[4] Facebook Statistics. www.facebook.com/press/info.php?statistics.
[5] James Hamilton's blog: Inter-datacenter replication & geo-redundancy. perspectives.mvdirona.com/2010/05/10/InterDatacenterReplicationGeoRedundancy.aspx.
[6] Personal communication.
[7] Riverbed Products&Services. http://www.riverbed.com/us/products/index.php.
[8] Signiant Products. www.signiant.com/Products-content-distribution-management/.
[9] Telefonica International Wholesale Services Network Map. http://www.e-mergia.com/en/mapaRed.html.
[10] Forrester Research. The Future of Data Center Wide-Area Networking. info.infineta.com/l/5622/2011-01-27/Y26, 2010.
[11] Ovum. CDNs The Carrier Opportunity. reports.innovaro.com/reports/cdns-the-carrier-opportunity, 2010.
[12] B. Ager, F. Schneider, J. Kim, and A. Feldmann. Revisiting Cacheability in Times of User Generated Content. In *IEEE Global Internet'10*.
[13] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris. Resilient Overlay Networks. In *ACM SOSP'01*.
[14] D. Beaver, S. Kumar, H. C. Li, J. Sobel, and P. Vajgel. Finding a Needle in Haystack: Facebook's Photo Storage. In *USENIX OSDI'10*.
[15] D. Breitgand, D. Raz, and Y. Shavitt. The Traveling Miser Problem. In *IEEE/ACM Transactions on Networking'06*.
[16] G. Chen, W. He, J. Liu, S. Nath, L. Rigas, L. Xiao, and F. Zhao. Energy-Aware Server Provisioning and Load Dispatching for Connection-Intensive Internet Services. In *USENIX NSDI'08*.
[17] Y. Chen, S. Jain, V. K. Adhikari, Z.-L. Zhang, and K. Xu. A First Look at Inter-Data Center Traffic Characteristics via Yahoo! Datasets. In *IEEE INFOCOM'11*.
[18] B. Cohen. Incentives Build Robustness in BitTorrent. In *P2P Econ'03*.
[19] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. Introduction to Algorithms. MIT Press'01.
[20] J. Gray. Long Term Storage Trends and You, 2006. http://research.microsoft.com/∼Gray/talks/IO_talk_2006.ppt.
[21] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel. The Cost of a Aloud: Research Problems in Data Center Networks. *ACM SIGCOMM Comput. Commun. Rev.*, 39(1), '09.
[22] 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 *ACM SIGCOMM'09*.
[23] M. Grossglauser and D. Tse. Mobility Increases the Capacity of Ad-hoc Wireless Networks. In *IEEE/ACM Transactions on Networking'01*.
[24] S. Jain, K. Fall, and R. Patra. Routing in a Delay Tolerant Network. In *ACM SIGCOMM'04*.
[25] D. Katabi, I. Bazzi, and X. Yang. A Passive Approach for Detecting Shared Bottlenecks. In *IEEE ICCN'02*.
[26] R. Kohavi, R. M. Henne, and D. Sommerfield. Practical Guide to Controlled Experiments on the Web: Listen to Your Customers not to the HiPPO. In *ACM KDD'07*.
[27] R. Kokku, A. Bohra, S. Ganguly, and A. Venkataramani. A Multipath Background Network Architecture. In *IEEE INFOCOM'07*.
[28] N. Laoutaris and P. Rodriguez. Good Things Come to Those Who (Can) Wait or How to Handle Delay Tolerant Traffic and Make Peace on the Internet. In *ACM HotNets-VII'08*.
[29] N. Laoutaris, M. Sirivianos, X. Yang, and P. Rodriguez. Inter-Datacenter Bulk Transfers with NetSticher. Technical report, 2011. Available at the authors' website.
[30] N. Laoutaris, G. Smaragdakis, P. Rodriguez, and R. Sundaram. Delay Tolerant Bulk Data Transfers on the Internet. In *ACM SIGMETRICS'09*.
[31] T. Leighton, C. Stein, F. Makedon, E. Tardos, S. Plotkin, and S. Tragoudas. Fast Approximation Algorithms for Multicommodity Flow Problems. In *ACM STOC'91*.
[32] S. Mitchell and J. Roy. Pulp Python LP Modeler. code.google.com/p/pulp-or/.
[33] H. Pucha, D. G. Andersen, and M. Kaminsky. Exploiting Similarity for Multi-Source Downloads using File Handprints. In *USENIX NSDI'07*.
[34] H. Pucha, M. Kaminsky, D. Andersen, and M. Kozuch. Adaptive file transfers for diverse environments. In *USENIX ATC'08*.
[35] F. Shahrokhi and D. Matula. The Maximum Concurrent Flow Problem. In *Journal of the ACM'90*.
[36] S. Shalunov and B. Teitelbaum. Qbone Scavenger Service (QBSS) Definition. Internet2 Technical Report'01.
[37] D. Ziegler. Distributed Peta-Scale Data Transfer. www.cs.huji.ac.il/~dhay/IND2011.html.