Flyways To De-Congest Data Center Networks

Paper # 112

A study of application demands from a production datacenter of 1500 servers shows that except for a few outliers, application demands can be generally met by a network that is slightly oversubscribed. Eliminating oversubscription is hence a needless overkill. In a significant departure from recent proposals that do so, we advocate a hybrid architecture. The *base* network is provisioned for the average case, is oversubscribed, and can be built with any of the existing network designs. To tackle the hotspots that remain, we add extra links on an on-demand basis. These links called *flyways* provide additional capacity where and when needed. Our results show that even a few additional flyways substantially improve performance (by over 50%), as long as they are added at the right place in the network. We consider two design alternatives for adding flyways at negligible additional cost: one that uses wireless links (60ghz or 802.11n) and another that uses commodity switches to add capacity in a randomized manner.

1. INTRODUCTION

As cloud-based services gain popularity, many businesses continue to invest in large data centers. Large datacenters provide economies of scale, large resource pools, simplified IT management and the ability to run large data mining jobs (e.g., indexing the web) [2]. One of the key challenges in building large data centers is that the cost of providing the same communication bandwidth between an arbitrary pair of servers grows in proportion to the size of the cluster [1, 6].

Production networks use a tree like topology (see Fig. 1a) with 20-40 servers per rack, increasingly powerful links and switches as one goes up the tree, and over-subscription factors of 1:2 (or more) at higher levels in the tree [1]. High oversubscription ratios put a premium on communication with non-local servers (i.e., those outside the rack). Application developers are forced to be cognizant of this limitation [3].

In contrast, recent research proposals [1, 6, 7] combine many more links and switches with variants of multipath routing such that the 'core' of the network is not oversubscribed. At any point in the network, sufficient bandwidth is always available to forward all incoming traffic. In such a network any server in the cluster can talk to any other server at full NIC bandwidth, regardless of the location of the servers in the cluster, or any other ongoing traffic. Needless to say, this benefit comes with large material cost (see Table 1) and implementation complexity (see Fig. 1b, c). Some [1] require so many wires that laying out cables becomes challenging while others [6, 7] require updates to server and switch software and firmware in order to achieve multipath routing.

		Tree	FatTree	VL2
Oversubscription Ratio		1:2	1:1	1:1
#Links	**10G**	160	0	640
	1G	3200	10112	3200
#Switches	**Agg**	1	0	5
	Commodity	0	360	0
	Top-of-rack	160	0	160
Network Cost (approx.)		x	2-3x	4-5x

Table 1: Comparison of three data center networking architectures. 3200 Servers, 160x10G agg switches, 1G Server NIC, 1G,10G links, 48port *commodity* switches for FatTree. Notice the number of links required for FatTree topology.

Eliminating oversubscription is a noble goal. For some workloads, such as the so-called "all-pairs-shuffle" [2], it is even necessary. Yet, as the cost and complexity of non-oversubscribed networks is quite high, it is important to ask: how much bandwidth do typical applications really demand? The answer to this question may point towards an intermediate alternative that bridges the gap between today's production network, and the ideal, non-oversubscribed proposals.

To answer the question, we gathered application demands by measuring all network events in a 1500 server production data center that supports map-reduce style data mining jobs [3].

Figure 2 shows a sample matrix of demands between every pair of the top-of-rack switches. A few trends are readily apparent. First, at any time only a few top-of-rack switches are hot, i.e., send or receive a large volume of traffic (dark horizontal rows and vertical columns). Second, the matrix is quite sparse, i.e., even the hot ToRs end up exchanging much of their data with only a few other ToRs. The implications are interesting. Figure 3 shows the completion time of a typical demand matrix in a conventional tree topology that has 1:2 over-subscription at the top-of-racks. The sparse nature of the demand matrix translates into skewed bottlenecks, just a few of the ToRs lag behind the rest and hold back the entire network from completion. Providing extra capacity to just these few ToRs can significantly improve overall performance.

Demand matrices exhibit these patterns because of the characteristics of underlying applications. Specifically, the map-reduce workload that runs in the examined cluster causes, at worst, a few tens of ToRs to be simultaneously bottlenecked. We expect this observation to hold for many data center workloads including those that host web services, except perhaps for rare scientific computing applications.

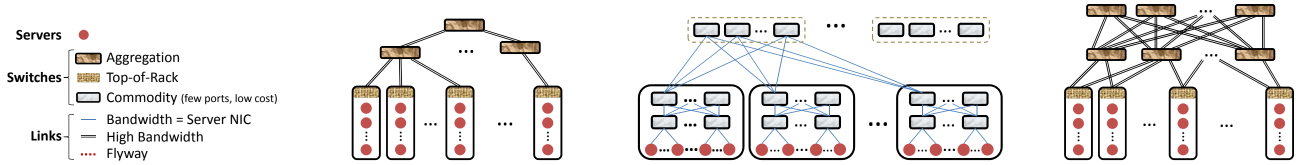
Based on these observations, we advocate a hybrid network. Since the demand matrix is quite sparse, the *base* network need only be provisioned for the average case and can be oversubscribed. Any hotspots that occur can be tackled by adding extra links between pairs of ToRs that can benefit

---

[1] 20 servers with 1Gbps NICs per rack, 24 port Cisco3560s at the top of the rack (ToR) with 10Gbps uplinks and 160port Cisco6509s at the root results in 1:2 oversubscription at the ToR's uplink

[2] Every server sends a large amount of data to every other server.

[3] We note that internal network is rarely the bottleneck for clusters that support external web traffic.



(a) Conventional Tree (b) FatTree [1] (c) VL2 [6]

Figure 1: Tree, VL2 and FatTree topologies

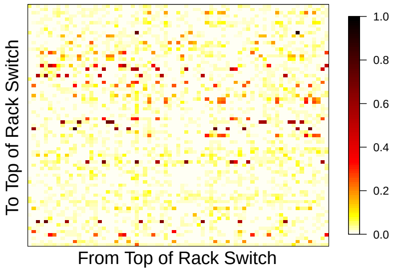


Figure 2: Matrix of Application Demands (normalized) between Top of Rack Switches. Only a few ToRs are hot and most of their traffic goes to a few other ToRs.

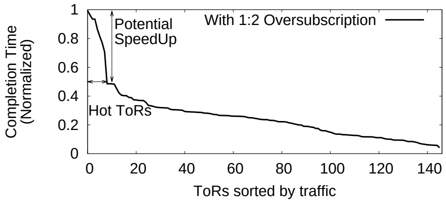


Figure 3: Providing some surplus capacity for just the top few ToRs can significantly speed up completion of demands.

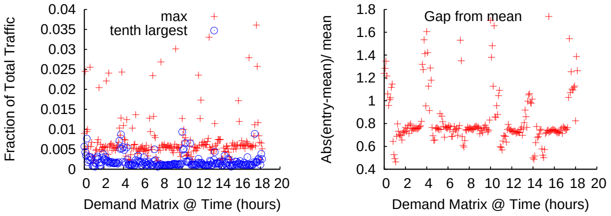


Figure 4: Demand matrices are neither dominated by a few ToR pairs nor uniformly spread out. None of the ToR pairs contributes more than 4% of the total (left) and the typical ToR pair is off the mean by 80%. (right).

from it. We call these links *flyways*. Flyways can be realized in a variety of ways, including wireless links that are set up on demand and commodity switches that interconnect random subsets of the ToR switches. We primarily investigate 60ghz wireless technology for creating flyways. This technology can support short range (1-10 meters), high-bandwidth (1Gbps) wireless links. We also consider 802.11n which can support link bandwidths of about 100-600Mbps.

We now make several observations about flyways, which we will justify in the rest of the paper. First, only a few flyways, with relatively low bandwidth, can significantly improve performance of an oversubscribed data center network. Often, the performance of a flyway-enhanced oversubscribed network is equal to that of a non-oversubscribed network. Second, the key to achieving the most benefit is to place flyways at appropriate locations. Hence, wireless technology, which can be used to form links on an on-demand basis, is more suitable for building flyways. Third, the traffic demands remain predictable at short time scales allowing flyways to keep up with changing demand. Fourth, 60GHz wireless technology appears an apt choice for creating flyways. Finally, we will describe a preliminary design for a central controller that gathers demands, adapts flyways in a dynamic manner, and uses MPLS label switched paths to route traffic.

We stress that this flyway architecture is not a replacement for architectures such as VL2 and FatTree that eliminate oversubscription. Rather, our thesis is that for practical traffic patterns one can get equivalent performance from a slightly oversubscribed network (of any design) that is augmented with flyways. Further, flyways can be deployed today on top of the existing tree-like topologies of production data centers and in many cases, flyways are also likely to be cost-effective.

2. THE CASE FOR FLYWAYS

We examine the traffic demands from a production cluster by instrumenting 1500 servers. Together, these servers comprise a *complete* data mining cluster that supports replicated distributed storage (e.g., GFS [5]) as well as parallel execution of data mining jobs (e.g., MapReduce [3]).

We collected all socket level events at each of the instrumented servers using the Event Tracing for Windows [4] framework. Over a few month period our instrumentation collected several petabytes of data. The topology of the cluster is identical to the conventional tree topology (see Fig. 1a). To compute how much traffic the applications have to exchange (i.e., the demands) independent of the topology that the traffic is currently being carried on, we accumulate traffic at the time scale of the applications (e.g., the duration of a job). For the map-reduce application in our data center, we accumulate over a 5 minute period since most maps and reduces finish within that time [3].

This traffic falls in two categories, the traffic between servers in the same rack, and the traffic between servers that are in different racks. As the backplane of the ToR switch has ample capacity to handle the intra-rack traffic, we focus only on the inter-rack traffic which is subject to oversubscription and experiences congestion higher up the tree.

What do the demand matrices look like? If the matrices are uniform, i.e., every ToR pair needs to exchange the same amount of traffic, then the solution is to provide uniformly high bandwidth between every pair of ToRs. On the other hand, if only a few ToR pairs consistently contribute most of the total traffic, then the network can be engineered to provide large bandwidth only between these few pairs. We find that neither extreme happens often. Fig. 4 (left) plots the maxi-

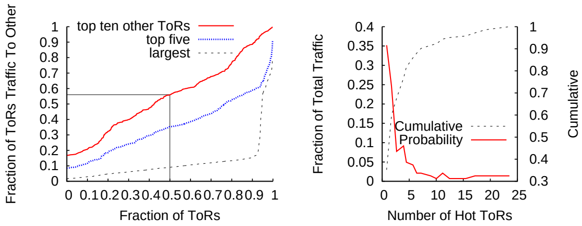


Figure 5: The hot ToRs, i.e., those that either send or receive a lot of traffic, exchange most of it with just a few other ToRs (left) and there aren't too many hot ToRs (right)

mum entry in demand matrices of an entire day. The largest entry contributes 0.5% of the total demand on average and never more than 4%. Fig. 4 (right) plots the average gap between a demand entry and the *mean demand*, which is typically 80% of the mean.

Let us now consider the ToR switches that either send or receive large amounts of traffic and examine the fraction of each ToR's traffic that is exchanged with its top few correspondents (other ToRs). Figure 5 shows that among ToR switches that contribute more than 3% of total traffic, the median ToR exchanges more than 55% of its traffic with just 10 other ToRs. This result has several implications. Providing additional capacity between the hot ToR and the other ToR that it exchanges a lot of data with would improve the completion time for that pair. By removing the traffic of this pair from competing with the other traffic at the hot ToR, completion times for the other correspondents improves as well. Even better, since we picked a hot ToR to begin with, speeding up completion of this ToR's demands (i.e., local improvements) will lower the completion time of the entire demand matrix (global impact). It turns out that the number of hot ToRs that would need the surplus capacity is small–between 5 to 10 switches exchange much more demand than the others (see Fig. 5 right).

Suppose we do want to add flyways to provide extra capacity between hot ToRs and some other ToRs that they exchanging traffic with. We need to answer two questions. First, which pairs should one select to get the most speedup? And second, how much capacity does each flyway need to have?

Placing the first flyway between a ToR that is the most congested and another ToR that it exchanges the most data with is clearly the right choice. But subsequent choices are less clear, for example should one place the next flyway at the same ToR or elsewhere? Fig. 6 examines different ways of placing the same number of flyways. Neither spreading flyways too thinly nor concentrating them at the top few ToRs works well. For example, placing one flyway each between the top 50 ToRs and their largest correspondent does not reduce the completion time of the hot ToR enough. Conversely, placing flyways between the top five ToRs and each of their ten largest correspondents does eliminate congestion at the top five only for the sixth ToR to end up as the bottleneck. Achieving a proper balance between helping *more* ToRs and reducing *enough* congestion at every one of the hot ToRs obtains the most speedup. (See §3.4 for our algorithm).

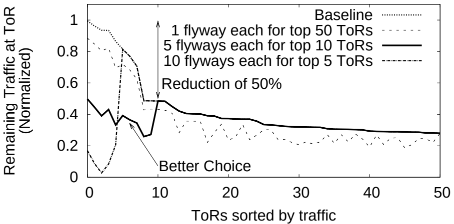


Figure 6: Where to place flyways for the most speedup?

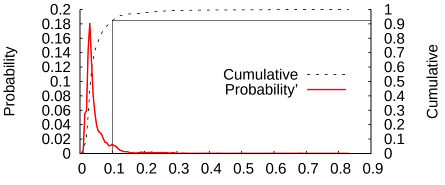


Figure 7: How much capacity should each flyway have?

How much capacity does each flyway need to have? Suppose we add flyways between the top ten ToRs and each of the five other ToRs that they exchange the most data with (i.e., a total of 50 flyways), Fig. 7 plots how much traffic each flyway needs to support. Most flyways need less than 10% of the ToR's uplink bandwidth to be useful. The reason is that while the ToR's uplink carries traffic to all of the other ToRs, a flyway has to only carry traffic to one other ToR.

The usefulness of flyways stems directly from application characteristics that cause sparse demand matrices. In data centers that support web services, the request traffic is load balanced across servers, each of which in turn assembles a response page by perhaps asking a few other servers to generate parts of the response (e.g., advertisements). The reduce part of map-reduce in data mining jobs perhaps comes closest to being the worst case, with each reducer pulling data from all the mappers. The job is bottlenecked until all the reducers complete. Even then, it is rare to have so many mappers and reduces that all ToRs are congested simultaneously. Though flyways will provide little benefit for demands like the all-pairs shuffler, we believe that a large set of practical applications stand to gain from flyways.

3. REALIZING FLYWAYS

In this section, we discuss how a practical network with flyways can be built. We consider both wireless and wired technologies, including 802.11n, which can provide upto 600Mbps bandwidth (although most links are not likely to exceed 100Mbps), the 60Ghz band, which can provide over 1Gbps and using commodity switches. We will argue that the 60ghz technology is an apt fit but since it is relatively new, we provide some background before fleshing out further details.

3.1 60Ghz Background

We briefly overview 60GHz technology and explain how this helps construct flyways in the data center environment. We refer the reader to [12] for further details.

The 60GHz band is a 7Ghz wide band of spectrum (57-64GHz) that was set aside as unlicensed by the FCC in 2001. In contrast to the 80MHz wide ISM band at 2.4GHz which supports the IEEE 802.11b/g/n networks, this band of frequency is 88x wide. The higher band width facilitates higher capacity links. For example, a simple encoding that achieves 1 bit/Hz makes possible links with a nominal bandwidth of 7Gbps. The 802.11b/g/n links use far more complex encodings that achieve 15 or more bits per Hz (e.g., 600Mbps over a 40MHz channel). Most regulators allow 10 to 100 watts of effective radiated power for transmissions in this band and per Shannon's law higher transmission power facilitates higher capacity links. Since this band includes the absorption frequency of the oxygen atom, the signal strength falls off rapidly with distance (1-10 meters). However, in the constrained environs of a datacenter, this short range is helpful; it allows for significant spatial reuse while being long enough to span tens of racks. The wavelength of 60GHz radiation is short (5 mm) which facilitates compact antennas. From the Frii's law, the effective area of an antenna decreases as frequency squared. Thus, a one-square inch antenna can provide a gain of 25dBi at 60GHz [11]. One can purchase 60GHz devices in bulk for about 20$ each. Taken together, these characteristics allow placing one or more 60GHz devices atop each of the racks in a datacenter to provide surplus link capacity, spatial reuse and viable range.

Numerous startups (SiBeam [11], Sayana [10]) have demonstrated prototype 60GHz devices that sustain data rates of 1-15Gbps over a distance of 4 to 10 meters with a power draw between 200mw to 10 watts. Fig. 8 shows a prototype SiBeam device. The most-envisioned usage scenario for 60Ghz networks, so far, has been to replace connecting wires between home entertainment devices and a few industry standards (WiGig [13], Wireless HD [14]) support this usage.

These existing devices are usable in datacenters today. Given standard equipment racks that are 24 inches wide, this range is suitable for communicating across several racks. See Figure 9. The small power draw (<10W) is readily available at each rack and the form factor of the devices (2-3 cubic inches) allows easy mounting on top of the racks. Some devices include electronically steerable phased-array antennas that form beams of about 60 degrees and can be steered with millisecond latency. Further customization of MAC and PHY layers for data center environment (e.g. more sophisticated encodings that provide more bits/Hz, higher power etc.) would result in greater cumulative capacity.

Needless to say, some challenges remain. First, due to the absorption characteristics and also because 60GHz waves are weakly diffracted [12], non-line of sight communication remains difficult to achieve. This is less of a problem in a data center environment where antennas can be mounted atop the racks and out of the way of human operators. Second, the technology to build power amplifiers at these high frequencies is still in flux. Until recently, amplifiers could only be built with Gallium-Arsenide substrates (instead of silicon) causing 60GHz radio front ends to be more expensive [12]. Recent advances in CMOS technologies have allowed companies like SiBeam and Sayana to develop 60Ghz devices using silicon which lowered prices and reduced power draw.

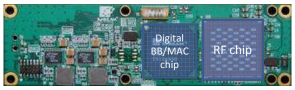


Figure 8: **60GHz wireless NIC. Courtesy SiBeam.**

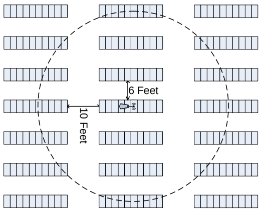


Figure 9: **View from top of a (partial) data center. Each box represents a 24x48 inch rack, which are arranged in rows of ten. The circle represents the 10m range of a 60Ghz device mounted atop a rack in the center, which contains about 70 other racks.**

3.2 Flyway links

Wireless: Using a choice of 802.11n devices or 60GHz devices, one can construct wireless flyways by placing one or a few devices atop each rack in the datacenter. To form a flyway between a pair of ToRs, the devices atop the corresponding racks create a wireless link. The choice of technology affects the available bandwidth, the number of channels available for spatial re-use, interference patterns and the range of the flyway. The antenna technology dictates the time needed to setup and tear down a flyway. We evaluate a few of these constraints in §4 and defer others to future work.

Wired: We suggest that wired flyways be constructed by using additional switches of the same make as today's ToR switches that inter-connect random subsets of the ToR switches. For e.g., one could use Cisco 3560 switches to inter-connect 20 ToRs with 1Gbps links each. To keep links short, we have the flyway switches preferentially connect racks that are close by in the datacenter (see Fig. 9).

Regardless of which of the above technologies one uses for flyways, the additional cost due to flyways is a small fraction of today's network cost. From Table 1, we note that adding a few tens of flyway switches, a few hundreds of 1G links or a few wireless devices per ToR increases cost marginally.

However, the two classes of flyways are qualitatively different. When deploying wired flyways, one does not have to worry about spectrum allocation or interference. At the same time, their random construction constrains wired flyways; ToR pairs that exchange a lot of traffic and can benefit from surplus capacity might end up without a wired flyway.

We do note however that either method of flyway construction is strictly better than dividing the same amount of bandwidth uniformly across all pairs of ToRs. Rather than spread bandwidth uniformly and have much of it wasted, as would happen when the demand matrix is sparse, flyways provide a way to use the spare bandwidth to target the parts of the demand matrix that can benefit the most from surplus capacity.

3.3 A Network with Flyways

Our initial design uses a central controller, which gathers estimates of demands between the pairs of ToRs, perhaps from lightweight instrumentation at the end servers themselves or by polling SNMP counters at switches. Using these estimates, the controller periodically [4] runs the placement algorithm (see §3.4) to usefully place the available flyways.

The topology of a flyway-based network is dynamic, and requires multipath routing. Towards this end, we leverage ideas from prior work that tackles similar problems [1, 6, 8]. The controller determines how much of the traffic between a pair of ToRs should go along the base network or take a flyway from the sending ToR to the receiving ToR, if one exists. The ToR switch splits traffic as per this ratio by assigning different flows onto different MPLS label switched paths. We note however that only a few flyways, if any, are available at each ToR. Hence, the number of LSPs required at each switch is small and the problem of splitting traffic across the base and flyways that are one hop long is significantly simpler than standard multipath routing.

3.4 Placing Flyways Appropriately

The problem of creating optimal flyways can be cast as an optimization problem. Given D_{ij} demand between ToRs i, j and C_l the capacity of link l, the optimal routing is the one that minimizes the maximum completion time:

$$\min \max \frac{D_{ij}}{r_{ij}} \tag{1}$$

such that

$$\sum_{l \in incoming} r^l_{ij} - \sum_{l \in outgoing} r^l_{ij} = \begin{cases} D_{ij} & \text{at ToR } j \\ -D_{ij} & \text{at ToR } i \\ 0 & \text{at all other ToRs} \end{cases}$$

$$\sum_{ij} r^l_{ij} \leq C^l \quad \forall \text{ links } l$$

where r_{ij} is the rate achieved for ToR pair i, j and r^l_{ij} is the portion of that pair's traffic on link l.

Computing the optimal flyway placement involves suitably changing the topology and re-solving the above optimization problem. For example, we could add all possible flyways and the constraint that no more than a certain number can be simultaneously active or that none of the flyways can have a capacity larger than a certain amount. Not all the variants of the above optimization problem that we have to solve for placing flyways are tractable. Instead, our results are based on a procedure that adds one flyway at a time, solves the simpler optimization problem shown above and then greedily adds the flyway that reduces completion times the most. This procedure is not optimal and improving it is future work.

4. PRELIMINARY RESULTS

In this section, we present simulation results that demonstrate the value of flyways under different settings. The simulations are driven from the demand matrices obtained from a production datacenter as described in §2. The 1500 servers in the production network have 1Gbps interfaces and are divided among 75 racks with 20 servers per rack. Hence, in the simulations here, we evaluate different ways of interconnecting the 75 ToR switches and routing the observed demands with the constraint that traffic in or out of a ToR cannot exceed 20Gbps. As metric, we use the completion time of the demands (CTD) defined as the maximum completion time of all the flows in that demand matrix. For ease of comparison, we report the normalized completion times where $NormalizedCTD = \frac{CTD}{CTD_{ideal}}$ and CTD_{ideal} is the completion time with the ideal, non-oversubscribed network. As we present results from different ways of adding flyways to a 1:2 oversubscribed tree network, note that the baseline has $CTD = 2$ and obtaining a $CTD = 1$ implies that with flyways, the network has routed demands as well as the ideal, non-oversubscribed network.

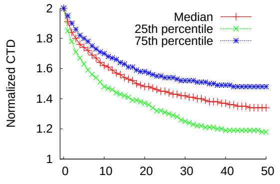


Figure 10: Impact of adding Flyways

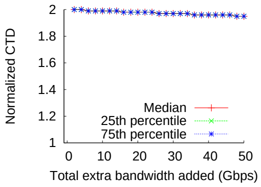


Figure 11: Distributing surplus capacity among all oversubscribed links

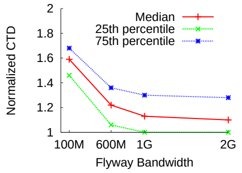


Figure 12: Impact of Flyway bandwidth (50 flyways)

For simulations in this section, we will assume that wireless links are narrow beam, half-duplex and point-to-point. We will ignore antenna steering overhead. We will also assume that given the narrow beamwidth, the limited range and the wide spectrum band available at 60 GHz, the impact of interference is negligible.

4.1 Benefit of using flyways

Fig. 10 shows the median normalized CTD (error bars are 25'th and 75'th percentiles) from adding different numbers of flyways, each of which has capacity 1Gbps, to a 1:2 oversubscribed tree topology over a day's worth of demand matrices.

Without any flyways, the median completion time of the tree topology is twice that of the ideal topology. As more flyways are added the difference between the two topologies narrows. The take-away from this figure is that with just 50 flyways, the median CTD with flyways is within 13% of that from an ideal topology. Observe that the potential cost for establishing these flyways is negligible compared to that of the ideal topologies. For many of the demand matrices just 30 flyways bring CTD on par with that of the ideal topology. Further, Figure 11 shows that distributing equivalent additional capacity uniformly among all the oversubscribed links, achieves little speed up. This simulation validates the key thesis behind flyways: adding low-bandwidth links between ToRs that are congested improves the performance of oversubscribed network topologies.

4.2 How much bandwidth?

How much bandwidth do we really need for each flyway?

---

[4] Or, on-demand, if the traffic matrix changes significantly.

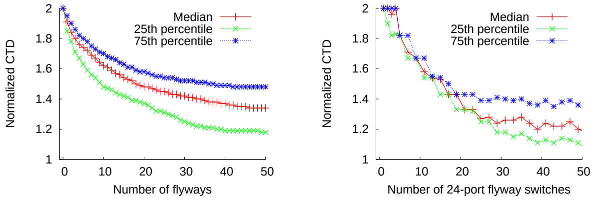


Figure 13: With Technology Constraints: (left) wireless flyways that are no longer than 10m (right) wired flyways that can only provide capacity among the randomly chosen subset

To answer this question, we repeat the above simulations with flyway capacities set to 100Mbps (802.11g, with channel bonding), 600Mbps (the best nominal bandwidth offered by 802.11n) and 2Gbps. As before, Figure 12 shows the median, 25'th and 75'th percentiles of completion times from adding 50 flyways. The graph indicates that while it may be possible to use 600Mbps links to create flyways, performance of 100Mbps flyways would be quite poor. Further, 2Gbps flyways provide little marginal benefit over 1Gbps flyways.

4.3 Constraints due to Technology

So far, we have ignored constraints due to the technology. Wireless flyways are constrained by range and wired flyways which are constructed by inter-connecting random subsets of the ToR switches can only provide surplus capacity between these random subsets. Fig. 13 repeats the above simulations with 1Gbps flyways and also these practical constraints. We assume that 60GHz flyways can span a distance of 10 meters, use the to-scale datacenter layout (see Fig. 9) from a production data-center. For wired flyways, we use 24 port, 1Gbps switches. We see that both constraints lower the benefit of flyways but the gains are still significant.

Note that many more wired flyways need to be added to obtain the same benefit accrued from wireless flyways. For example, when fifty 24-port switches are added, we are adding $50 * 24 = 1200\ duplex$ links to the network. Wireless flyways provide equivalent performance with just 50 half-duplex links. This is because while wireless flyways are added in a targeted manner: they help speed up exactly those pairs of ToRs needs additional capacity. Wired flyways are added at random, and will benefit these pairs, only if they happen to be among the ones chosen.

4.4 Discussion

These simulation results are meant primarily to demonstrate the viability of the flyway concept. While we considered a few practical limits on building flyways, many other issues need to be considered. We list a few here. First, the number of flyways that each ToR can participate in is limited by the number of wireless NIC available at the ToR. We have simulated this scenario, and we find that we need anywhere between 5 and 20 wireless links to and/or from the busiest ToRs. Second, we assumed that capacity of each flyway is constant, and all flyways had the same capacity. In practice, capacity of each flyway is determined by a number of factors, including: interference from other flyways, the amount of spectrum dedicated to the flyway, the antenna gain, and the distance between the two wireless NICs. The flyway construction algorithm will have to take link quality into account. Third, we assumed that there is no interference between flyways. While we believe that this is a reasonable assumption for 60 GHz links, we are working to relax it. To do so, we will need to generate a conflict graph [9] to encode interference information. Given that the data center environment is not particularly dynamic, such a graph will have to be updated only infrequently.

5. CONCLUSION

Prior research has addressed how to scale data center networks, but to the best of our knowledge none has studied application demands. Our data shows that a map-reduce style data mining workload results in sparse demand matrices. At any time, only a few ToR switches are bottlenecked and these ToRs exchange most of their data with only a few other ToRs. This leads us to the concept of flyways. By providing additional capacity when and where congestion happens, flyways improve performance at negligible additional cost. We show that wireless links, especially those in the 60GHz band, are an apt choice for implementing flyways. We expect that pending a revolution in the types of applications that run within datacenters, the sparse nature of inter-rack demand matrices will persist. Hence, the flyways concept should remain useful. We have listed many practical and theoretical problems that need to be solved to make flyway based networks a reality. We are currently working on solving these problems.

6. REFERENCES

[1] M. Al-Fares, A. Loukissas, and A. Vahdat. A scalable, commodity data center network architecture. In *SIGCOMM*, 2008.
[2] L. A. Barroso and U. Hlzle. *The Datacenter as a Computer - an introduction to the design of warehouse-scale machines*. Morgan & Claypool, 2009.
[3] J. Dean and S. Ghemawat. Mapreduce: Simplified data processing on large clusters. In *OSDI*, 2004.
[4] Event Tracing For Windows. http://msdn.microsoft.com/en-us/library/ms751538.aspx.
[5] S. Ghemawat, H. Gobioff, and S.-T. Leung. The Google File System. In *SOSP*, 2003.
[6] A. Greenberg, J. Hamilton, N. Jain, S. Kandula, C. Kim, P. lahiri, D. Maltz, P. Patel, and S. Sengupta. Vl2: A scalable and flexible data center network. In *SIGCOMM*, 2009.
[7] C. Guo et al. Bcube: High performance, server-centric network architecture for data centers. In *SIGCOMM*, 2009.
[8] S. Kandula, D. katabi, B. Davie, and A. Charny. Walking the tightrope: responsive yet stable traffic engineering. In *SIGCOMM*, 2005.
[9] J. Padhye, S. Agarwal, V. Padmanabhan, L. Qiu, A. Rao, and B. Zill. Estimation of link interference in static multi-hop wireless networks. In *IMC*, 2005.
[10] Sayana Networks.
[11] SiBeam. http://sibeam.com/whitepapers/.
[12] P. Smulders. Exploiting the 60GHz Band for Local Wireless Multimedia Access: Prospects and Future Directions. *IEEE Communications Magazine*, January 2002.
[13] Wireless Gigabit Alliance. http://wirelessgigabitalliance.org/.
[14] WirelessHD. http://wirelesshd.org/.