Proteus: A Topology Malleable Data Center Network

Ankit Singla[†][*], Atul Singh[‡], Kishore Ramachandran[‡], Lei Xu[‡], and Yueping Zhang[‡]
[†] University of Illinois Urbana–Champaign, IL, USA
[‡] NEC Labs America, Inc., Princeton, NJ, USA
[†] singla2@illinois.edu, [‡] {atuls, kishore, leixu, yueping}@nec-labs.com

ABSTRACT

Full-bandwidth connectivity between all servers of a data center may be necessary for all-to-all traffic patterns, but such interconnects suffer from high cost, complexity, and energy consumption. Recent work has argued that if all-to-all traffic is uncommon, oversubscribed network architectures that can adapt the topology to meet traffic demands, are sufficient. In line with this work, we propose Proteus[1], an all-optical architecture targeting unprecedented topology-flexibility, lower complexity and higher energy efficiency.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design:**]: Circuit switched networks, network topology

General Terms

Design, Performance

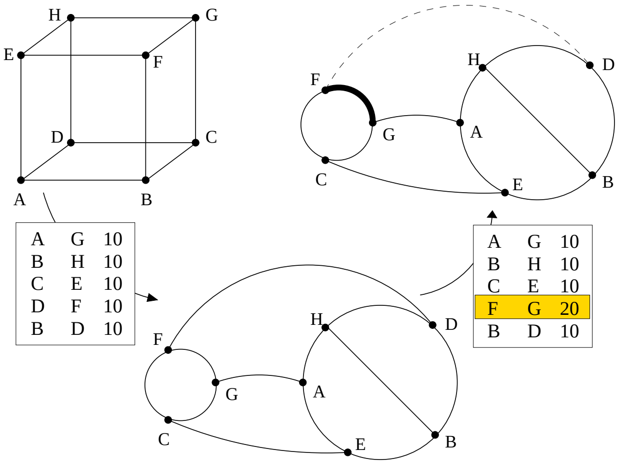
Keywords

Data center networks, optical circuit switching

1. INTRODUCTION

The network interconnect of a data center plays a key role in the performance and scalability of the services it runs. To this end, the quest for a scalable and efficient data center network (DCN) architecture has seen much recent progress [6, 9–12, 16, 19, 22, 23].

The early theme of this research was high bandwidth connectivity between all pairs of servers (or 1:1 over-subscription) [6, 10]. While for certain traffic patterns this is a necessity, recent work has argued that such traffic patterns are, at the least, not ubiquitous [13]. For sparse/skewed traffic patterns, it is desirable to build cheap, energy-efficient interconnects that are *malleable* to traffic – i.e. connect any reasonably-sized subset of the server population with high bandwidth on-demand. Another motivation for malleability stems from new applications and evolving programming models [1], which may lead to unexpected traffic patterns.



A	G	10
B	H	10
C	E	10
D	F	10
B	D	10

A	G	10
B	H	10
C	E	10
F	G	20
B	D	10

Figure 1: Proteus adapts topology and link capacities to the traffic matrix. 'A G 10' denotes demands $A \xrightarrow{10Gbps} G$ and $G \xrightarrow{10Gbps} A$

To illustrate the usefulness of malleability, consider the hypothetical example in Fig. 1. On the left is a hypercube connecting 8 (Top-of-Rack or ToR) switches using 10 Gbps links. The traffic demand is shown in the bottom-left traffic matrix of Fig. 1. For this traffic matrix, with any routing protocol on this hypercube, there will be at least 1 link on which 20 Gbps traffic is desired. One way to tackle this congestion is to reconfigure the switches into a different topology (Fig. 1, center). There are, of course, other topologies that will work too, including differently connected hypercubes. Now, consider that the traffic matrix changes (Fig. 1, bottom-right) with a new (highlighted) entry replacing an old one. If

[*] Work done during an internship at NEC Labs America, Inc.
[1] In Greek mythology, Proteus is a shape-shifting god.

Hotnets '10, October 20–21, 2010, Monterey, CA, USA.

no adjustments are made, at least one link will face congestion. With shortest path routing $F \leftrightarrow G$ will be that link. In this scenario, one solution is to double the capacity of the $F \leftrightarrow G$ link at the expense of reducing capacity at $F \leftrightarrow D$ to 0. (D now has spare capacity of 10 Gbps.) Critically, note that in all three scenarios, the degree and the capacity of nodes (30 Gbps) has remained the same.

To achieve such malleability, we introduce Proteus, a novel DCN architecture which adapts its topology to meet the traffic demands. With a design-time-fixed parameter k, Proteus can assume any k-regular topology and also vary the capacity of each of the k links at each node. To illustrate briefly how many options this gives us, we note that for just 20 nodes, there are over 12 billion (non-isomorphic) connected 4-regular graphs [2]. Not only this, each edge in this 4-regular topology can have variable capacity from a few Gbps to a couple of hundred Gbps (subject to constraints we discuss later).

Proteus achieves malleability by carefully exploiting the *reconfigurability* of optical networking technology – both the ability to change optical circuit configurations (for dynamic topology) as well as optical wavelength provisioning (for dynamic link capacity) at runtime. Proteus completely avoids using electrical equipment other than the ToR switches, enabling high energy-efficiency, easier migration to 40-GigE and beyond, and significantly simplified cabling compared to existing DCN architectures.

In the following, we provide background on optical equipment and describe the design and architecture of Proteus. We then formulate the problem of finding the optimal topology given the traffic matrix, and briefly sketch a heuristic solution. Next, we present feasibility arguments for Proteus and then conclude with future work.

2. OPTICAL TECHNOLOGY

1. Wavelength Division Multiplexing (WDM): Depending on the channel spacing, using WDM, typically 40 or up to 100 channels or wavelengths can be transmitted over a single piece of fiber in the conventional or C-band.

2. Wavelength Selective Switch (WSS): A WSS is typically a $1 \times N$ optical component, consisting of one *common* port and N *wavelength* ports. It partitions the set of wavelengths coming in through the common port among the N wavelength ports. For example, if the common port receives 80 wavelengths then it can route wavelengths 1–20 on port 1, wavelengths 30–40 and 77 on port 2, etc. This mapping is runtime-configurable (in a few ms).

3. Micro-Electro-Mechanical Switch (MEMS): A MEMS achieves reconfigurable one-to-one circuits between its N input and N output ports by mechanically adjusting its micro-mirrors. A few hundred ports are common for commercial products, and >1000 for research prototypes [15]. A MEMS is oblivious to the wavelengths carried across it. Any input port can be connected to any one of the output ports, i.e. the configuration is a bipartite-matching of input and output ports which can be switched within a few ms.

4. Optical Circulators: Circulators enable bidirectional optical transmission over a fiber, allowing more efficient use of the ports of optical switches. An optical circulator is a three-port device: one port is a shared fiber or switching port, and the other two ports serve as send and receive ports.

5. Optical Transceivers: Optical transceivers can be of two types: coarse WDM (CWDM) and dense WDM (DWDM). We use DWDM-based transceivers, which support higher bit-rates and more wavelength channels in a single piece of fiber compared to CWDM.

3. Proteus ARCHITECTURE

In this section, we illustrate how Proteus leverages the above described optical technology in its architecture. Our current design focuses on container-sized data centers. We do not address fault-tolerance issues in this paper.

3.1 Building blocks

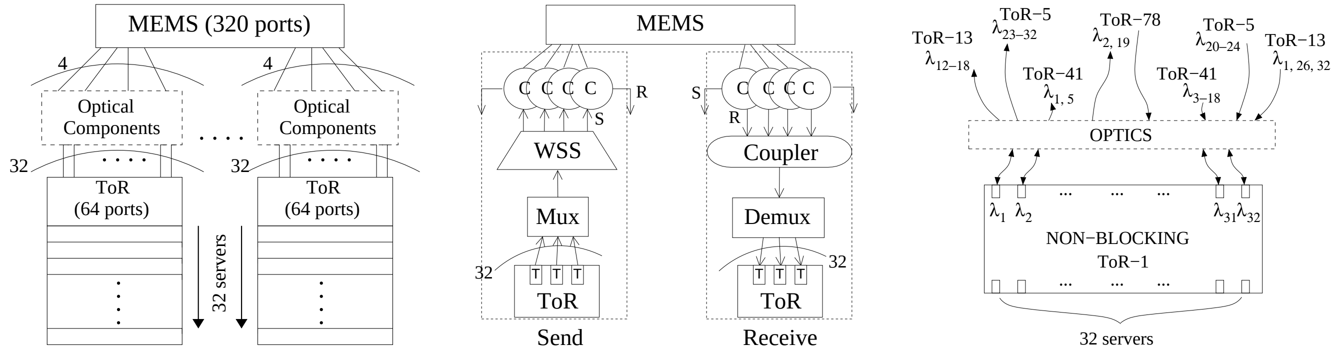
3.1.1 Flexible Topology

Proteus achieves dynamic topology by exploiting reconfigurability of the MEMS. Say we start by connecting each of N ToRs to one port on an N-port MEMS. This implies that every ToR can only communicate with one other ToR at any instant, and the ToR-graph is disconnected. If we connect N/k ToRs to $k > 1$ ports each at the MEMS, each ToR can communicate with k ToRs simultaneously. We note that throughout this paper, the degree of a ToR refers to k, not the port count. The switching configuration of MEMS determines which set of ToRs are connected; thus Proteus must ensure that the entire ToR graph is connected when performing MEMS reconfigurations (see § 4.2).

Given a connected ToR-graph, we use *hop-by-hop* communication to achieve network connectivity. To reach ToRs not directly connected to it through the MEMS, a ToR uses one of its k connections. This first-hop ToR receives the transmission over fiber, converts it to electrical signals, reads the packet header, and retransmits it towards the destination. (Modifications are required in the ToR switch to support this function.) Note that the transit and locally-generated traffic aggregated at a given port can not exceed the port's capacity. Therefore, high-volume connections must use a minimal number of hops. Proteus must manage the topology to adhere to this requirement.

3.1.2 Flexible Bandwidth

Every ToR has degree k. If each edge has fixed bandwidth, then multiple edges would need to be utilized for this ToR to communicate with another ToR at a rate higher than a single edge supports. To overcome this problem, Proteus combines



(a) Overall structure.

(b) Optical components for one ToR.

(c) Wavelength assignment.

Figure 2: (a) shows the overall Proteus architecture. In (b), the send and receive infrastructure are shown for different ToRs for clarity. In (c), ToR-1 has MEMS circuits with 4 other ToRs. Each incoming/outgoing connection to these ToRs has a set of wavelengths associated with it.

the capability of optical fibers to carry multiple wavelengths at the same time (*WDM*) with the dynamic reconfigurability of the *WSS*. Consequently, a ToR is connected to MEMS through a multiplexer and a WSS unit (Fig. 2(b)).

Specifically, suppose ToR A wants to communicate with ToR B using w times the line speed of a single port. The ToR will use w ports, each associated with a (unique) wavelength, to serve this request. WDM enables these w wavelengths, together with the rest from this ToR, to be multiplexed into one optical fiber that feeds the WSS. The WSS splits these w wavelengths to the appropriate MEMS port which has a circuit to ToR B (doing likewise for $k-1$ other sets of wavelengths). Thus, a $w\times$ (*port-line-speed*) capacity circuit is set up from A to B, at runtime. By varying the value of w for every MEMS circuit connection, Proteus achieves dynamic capacity for every edge.

Now, how do we assign wavelengths to ToR ports? We note that a fiber can not carry two channels over the same wavelength in the same direction. Moreover, to enable a ToR pair to communicate using all available wavelengths, we require that each ToR port (facing the optical interconnect) is assigned a wavelength unique across ports at this ToR (illustrated in Fig. 2(c)). The same wavelength is used to receive traffic as well: each port thus sends and receives traffic at one fixed wavelength. The same set of wavelengths is recycled across ToRs. This allows all wavelengths at one ToR to be multiplexed and delivered after demultiplexing to individual ports at the destination ToR. This wavelength-port association is a static, design/build time decision.

3.1.3 Optimization

To make full use of the MEMS ports, we desire that each circuit over the MEMS is bi-directional. For this, we use optical circulators between the ToR and MEMS ports. A circulator connects the send channel of the transceiver from a ToR to the MEMS (after the channel has passed through the WSS). It simultaneously delivers the traffic incoming towards a ToR from the MEMS, to this ToR. Note here that even though the MEMS edges are bidirectional, the capacities of the two directions are independent of each other.

3.2 Putting it all together: `Proteus-2560`

Fig. 2 shows one instantiation, `Proteus-2560`, with a 320-port MEMS and 80 ToRs to support 2560 servers. Each ToR is a commodity electrical switch with 64 10-GigE non-blocking ports [3]. 32 of these ports are connected to servers, while the remaining face the optical interconnect. Each port facing the optical interconnect has a transceiver associated with a fixed and unique wavelength for sending and receiving data. With WDM, this allows data from different ports to be multiplexed into one fiber without wavelength contention. The transceiver uses separate fibers to connect to the send and receive infrastructures.

As shown in the left half of Fig. 2(b), the send fiber from the transceivers from each of the 32 ports at a ToR is connected to an optical multiplexer. The multiplexer feeds a 1×4 WSS. The WSS splits the set of 32 wavelengths it sees into 4 groups, each group being transmitted on its own fiber. These fibers are connected to the MEMS switch through circulators to enable bidirectional traffic through them.

The right half of Fig. 2(b) shows the receive infrastructure. The 4 receive fibers from each of 4 circulators corresponding to a ToR are connected to a power coupler (similar to a multiplexer, but simpler), which combines their wavelengths onto one fiber. This fiber feeds a demultiplexer, which splits each incoming wavelength to its associated port on the ToR.

We point out two key properties of the above interconnect. First, each ToR can communicate simultaneously with *any* 4 other ToRs. This implies that MEMS reconfigurations allow us to construct all possible 4-regular ToR graphs. Second, through WSS configuration, each of these 4 links' capacity can be varied in $\{0, 10, 20, \dots, 320\}$ Gbps.

These configurations are decided by a topology manager (TM). The TM obtains traffic matrix from the ToR switches, calculates appropriate configurations, and pushes them to the MEMS, WSS, and ToRs. This requires direct, out-of-band connections between the TM and these units.

Proteus differs from Helios [9] and c-Through [22] in its degree of flexibility and its architecture. Both these earlier approaches achieve some flexibility in topology through the use of a limited number of single-hop optical links. However, Proteus can choose an arbitrary topology from a large class of graphs (connected k-regular order-N) and also vary the capacity of the edges. We note that the intended scale of Helios is different from that of Proteus and leave scaling to mega-datacenter settings to future work.

4. OPTIMAL TOPOLOGY SELECTION

For optimality, the TM needs to find: a) a MEMS configuration to adjust the topology to localize high traffic volumes, b) a configuration for each WSS to provision the capacity of its outgoing links well, and c) routes between ToR-pairs to achieve high throughput, low latency and avoid congestion. We assume the traffic demand matrix can be estimated in a fashion similar to either c-Through or Helios.

4.1 Mixed Integer Linear Program

Given: A traffic (demand) matrix D between ToRs – D_{ij} is the desired bandwidth from ToR_i to ToR_j.

Variables: We use four classes of variables: $l_{ij} = 1$ if ToR_i is connected to ToR_j through the MEMS and 0 otherwise; $w_{ijk} = 1$ if l_{ij} carries wavelength λ_k in the $i \rightarrow j$ direction and 0 otherwise; a traffic-served matrix S – S_{ij} is the bandwidth provisioned (possibly over multiple paths) from ToR_i to ToR_j; v_{ijk} is the volume of traffic carried by wavelength λ_k along $i \rightarrow j$. Among the latter two sets of variables, S_{ij} have end-to-end meaning, while v_{ijk} have hop-to-hop significance. For all variables, $k \in \{1, 2, \ldots, \lambda_{Total}\}$; $i, j \in \{1, 2, \ldots, NumToRs\}$, $i \neq j$; l_{ij} are the only variables for which $l_{ij} = l_{ji}$ always holds – all other variables are directional.

Objective: A simplistic objective is to maximize the traffic served (constrained by demand, see (6)):

$$Maximize \sum_{i,j} S_{ij}. \qquad (1)$$

Constraints:

A wavelength λ_k can only be used between two ToRs if they are connected through the MEMS:

$$\forall i, j, k : w_{ijk} \leq l_{ij}. \qquad (2)$$

ToR_i can receive/send λ_k from/to at most one ToR (this is illustrated in Fig. 3):

$$\forall i, k : \sum_j w_{jik} \leq 1; \sum_j w_{ijk} \leq 1. \qquad (3)$$

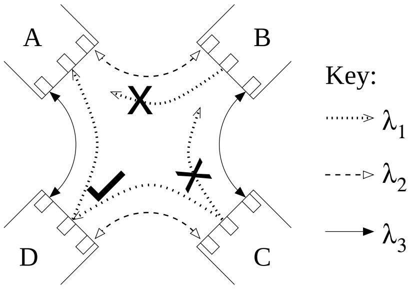


Figure 3: A 4-ToR wavelength contention example: Given the other connections, the X's mark connections that are not permitted by wavelength contention. This necessitates directed edge coloring with wavelengths as colors.

If the number of ports of the WSS units is W, then ToR_i is connected to exactly W other ToRs:

$$\forall i : \sum_j l_{ij} = W. \qquad (4)$$

Hop-by-hop traffic is limited by port capacities (C_{port}), wavelength capacity (C_λ), and provisioning:

$$\forall i, j, k : v_{ijk} \leq min\{C_{port}, C_\lambda \times w_{ijk}\}. \qquad (5)$$

We never provision more traffic than demanded:

$$\forall i, j : S_{ij} \leq D_{ij}. \qquad (6)$$

The outgoing transit traffic (total traffic flowing out, minus total traffic for which ToR_i is the origin) equals incoming transit traffic at ToR_i:

$$\forall i : \sum_{j,k} v_{ijk} - \sum_j S_{ij} = \sum_{j,k} v_{jik} - \sum_j S_{ji}. \qquad (7)$$

The above mixed-integer linear program (MILP) is expected to be NP-Hard. The related problem of constructing an optimal overlay, given latencies between every pair of nodes is known to be NP-hard [8, 21], but we leave the reduction proof for future work. Also, from a practical perspective, it would be necessary to make sure that the graph is connected, even if the current traffic matrix does not require so. This is essential to ensure that new low-latency traffic can be initiated. These observations motivate the need to look at heuristic solutions (§ 4.2).

We note that our problem formulation shares ground with recent work [5] in overlay topology optimization. However, the capabilities, technology constraints, and the problem settings for both systems are different. The overlay problem is constrained by the costs of link-setup, while for us, node-degree and capacity are constraints, together with technology limitations.

4.2 A Greedy Heuristic

In this section, we provide a high-level sketch of one heuristic for finding the optimal topology and omit algorithm details due to space constraints. Our heuristic consists of the following five steps.

Step 1: Assign elephant flows to direct links: We first localize high-volume flows over direct MEMS circuit links. This is accomplished by using a weighted b-matching, where b represents the number of connections that each ToR has to the MEMS (i.e., $b = 4$ in our example scenario). However, this may not provide overall connectivity.

Step 2: Achieve connectivity: Connectivity is simple to achieve – we use the edge-exchange operation (e.g., links $a \rightarrow b$ and $c \rightarrow d$ are replaced by $a \rightarrow c$ and $b \rightarrow d$, also see [20]) on the edges of lowest weight (and which are not cuts themselves) across pairs of components, thus connecting them.

Step 3: Identify paths: Once we have connectivity, the MEMS configuration is known. We proceed to find routes using any of the standard routing schemes – shortest path or preferably, a low congestion routing scheme. Note that some of the routes are single-hop MEMS connection while others are multi-hop MEMS connections.

Step 4: Compute capacity demand: Given these paths and the traffic demand, it is easy to compute the capacity required at each link. To satisfy the capacity demand on each link, multiple wavelengths may be required.

Step 5: Assign λ's: We now desire to provision wavelengths to serve these requirements. We reduce this problem to edge-coloring on a multigraph. Multiple edges correspond to volume of traffic between two nodes; and wavelengths are the colors to be used to color these edges. The need for edge-coloring is illustrated in Fig. 3: for instance, $D \rightarrow A$ and $B \rightarrow A$ can not both use the same wavelength (i.e., color). This constraint stems from the fact that two data-flows encoded over the same wavelength can not share the same optical fiber in the same direction.

Weighted b-matching can be computed for even 1000 nodes in a few hundred milliseconds [18]. Fast edge-coloring heuristics are also known [17]. Recent work [9, 22] demonstrates the feasibility of estimating the traffic matrix and making adjustments in a few hundred milliseconds. Nevertheless, a potentially faster approach we are investigating, is to make incremental changes to the topology. These changes should obey the degree-constant and node-capacity-constant constraints (see [20]). The advantage of such a process is that it is iterative, and each iteration is likely to be inexpensive. It is also likely that a large number of large flows do not change simultaneously, keeping the iterations simple (see § 5.2).

5. PRELIMINARY ANALYSIS

5.1 Benefits

Cost and power consumption: The equipment used by `Proteus-2560` is listed in Table 1. The total cost is approximately USD 6.12 million, with a power consumption of roughly 73 Kilowatts. Note that ToRs and transceivers are responsible for a large portion of the cost and power budget. As a reference-point, a fat-tree connecting roughly the same number of servers costs USD 6.4 million and consumes roughly 160 Kilowatts. This is not an adequate comparison – the above Proteus instance is not fault tolerant. If we simply replicate the optical components (excluding transceivers) for fault-tolerance[2], the power consumption will not increase by more than a Kilowatt, while the total cost will increase to USD 7.6 million. Thus Proteus enables power savings of more than 50% in the network. At 10c/Kilowatt hour [4], this results in savings of USD 0.076 million/year. These savings grow roughly linearly with the number of Proteus containers.

Table 1: Cost (USD) and power consumption (Watt) of `Proteus-2560`

Element	Cost	Power	Qty	Subtotal	
				Cost	Power
ToR	$500\times64^\dagger$	$12.5\times64^\dagger$	80	2.56M	64K
MEMS	$500\times320^\dagger$	$0.24\times320^\dagger$	1	0.64M	0.08K
WSS	$1000\times4^\dagger$	$1\times4^\dagger$	80	0.32M	0.32K
Transceiver	800	3.5	2560	2.05M	8.96K
(DE)MUX	3000	0	160	0.48M	0
Coupler	100	0	80	0.01M	0
Circulator	200	0	320	0.06M	0
Total				6.12M	73.36K

†Cost/power per port × number of ports.

We also note here that the cost of optics is expected to fall with commoditization and production volume. Much of these benefits have already been reaped for electrical technology. There is also scope for packaging multiple components on a chip - the 32 transceivers and the MUX could be packaged into one chip. This will reduce power consumption, cost as well as the number of fibers.

Relatively future-proof: When the servers or ToRs change, the optical interconnect can still remain the same and does not need rewiring – so upgrades are easier. This implies easier migration to 40-GigE and 100-GigE.

Low physical complexity: The number of fibers in our design is very small - 1120 fibers above the MUX layer (compared to a fat-tree's > 5000 cables above the ToR layer). The ToR to MUX/DEMUX connection is very short (and can be packaged with the ToR). This is a significant improvement from the point of view of cost, physical space and associated benefits (cables cover up the cooling).

In addition to the above, Proteus has the potential to simplify virtual machine placement. There is also scope for shutting down parts of the network when the corresponding

[2] We realize that the fault-tolerance of Proteus is still lower than that of a fat-tree. Incorporating fault-tolerance in the DCN architecture is a non-trivial issue and we leave a full treatment for future work.

servers are not being utilized, at the same time reducing the diameter of the rest of the topology.

We also note that Proteus-2560 is just one instance. With a larger number of MEMS and WSS ports, topologies with higher degrees and/or larger numbers of ToRs can be built. It is also possible to make heterogeneous interconnects – a few nodes can have larger degree than the rest.

5.2 Feasibility

Traffic Characteristics: Proteus would work best when high-volume ToR-ToR connections are: a) not numerous and b) stable on the order of seconds. These requirements stem from the low degree of the architecture and the reconfiguration time respectively. It has been noted in [13] over measurements of a 1500-server production datacenter that *"Only a few ToRs are hot and most of their traffic goes to a few other ToRs."* Another study [10], also on a 1500-server production datacenter, shows that more than 90% of bytes flow in "elephant" flows. These observations imply that Proteus' strategy of a small number of high-volume, reconfigurable circuits with hop-by-hop routes over them should work well in practice. Regarding traffic stability, a similarly sized study [7] shows that 60% of ToR-pairs see less than 20% change in traffic demand for between 1.6 to 2.2 seconds on average. It would also be reasonable to believe that higher volume ToR-ToR connections are more persistent.

Electrical-Optical-Electrical Conversion: Proteus utilizes hop-by-hop routing to achieve connectivity for all pairs of ToRs. At each hop, every packet experiences conversion from optics to electronics and then back to optics (O-E-O). The additional latency imposed by this O-E-O conversion, while device/technology dependent and often vendor proprietary information, is small enough to ignore. One measurement [14] of a particular technology pegs this value in the sub-nanosecond region.

6. CONCLUSION AND FUTURE WORK

In this paper, we advocate that a combination of an *all-optical network* and *on-demand, run-time topology reconfigurability* is a more flexible building block for DCN designs than existing solutions. We have presented one design instance of such a DCN but much future work remains to evaluate its performance and practicality. In addition to the multiple container-sized DCN design, our future work includes making Proteus fault-tolerant, designing a routing protocol for hop-by-hop connectivity, and performing extensive experimental evaluation with different traffic patterns and applications using a prototype with real optical devices.

7. REFERENCES

[1] http://googleblog.blogspot.com/2010/06/our-new-search-index-caffeine.html.

[2] http://www.mathe2.uni-bayreuth.de/markus/reggraphs.html.

[3] http://www.broadcom.com/products/features/BCM56840.php.

[4] http://www.eia.doe.gov/electricity/epm/table5_6_b.html.

[5] On graph-based characteristics of optimal overlay topologies. *Computer Networks*, 53(7):913 – 925, 2009.

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

[7] T. Benson, A. Anand, A. Akella, and M. Zhang. The case for fine-grained traffic engineering in data-centers. In *USENIX INM/WREN*, 2010.

[8] Y. Chawathe, S. Mccanne, and E. Brewer. An architecture for internet content distribution as an infrastructure service. Unpublished work, 2000.

[9] N. Farrington, G. Porter, S. Radhakrishnan, H. H. Bazzaz, V. Subramanya, Y. Fainman, G. Papen, and A. Vahdat. Helios: A hybrid electrical/optical switch architecture for modular data centers. In *ACM SIGCOMM*, 2010.

[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 *ACM SIGCOMM*, 2009.

[11] 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. In *ACM SIGCOMM*, 2009.

[12] C. Guo, H. Wu, K. Tan, L. Shi, Y. Zhang, and S. Lu. DCell: A scalable and fault-tolerant network structure for data centers. In *ACM SIGCOMM*, 2008.

[13] S. Kandula, J. Padhye, and P. Bahl. Flyways to de-congest data center networks. In *ACM HotNets*, 2009.

[14] G. Keeler, D. Agarwal, C. Debaes, B. Nelson, N. Helman, H. Thienpont, and D. Miller. Optical pump-probe measurements of the latency of silicon cmos optical interconnects. *IEEE Photonics Technology Letters*, 14(8):1214 – 1216, 2002.

[15] J. Kim, C. Nuzman, B. Kumar, D. Lieuwen, J. Kraus, A. Weiss, C. Lichtenwalner, A. Papazian, R. Frahm, N. Basavanhally, D. Ramsey, V. Aksyuk, F. Pardo, M. Simon, V. Lifton, H. Chan, M. Haueis, A. Gasparyan, H. Shea, S. Arney, C. Bolle, P. Kolodner, R. Ryf, D. Neilson, and J. Gates. 1100×1100 port mems-based optical crossconnect with 4-db maximum loss. *IEEE Photonics Technology Letters*, 15(11):1537 –1539, 2003.

[16] D. Li, C. Guo, H. Wu, K. Tan, Y. Zhang, and S. Lu. FiConn: Using backup port for server interconnection in data centers. In *IEEE INFOCOM*, 2009.

[17] J. Misra and D. Gries. A constructive proof of vizing's theorem. *Inf. Process. Lett.*, 41(3):131–133, 1992.

[18] M. Müller-Hannemann and A. Schwartz. Implementing weighted b-matching algorithms: insights from a computational study. *J. Exp. Algorithmics*, 5:8, 2000.

[19] 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 *ACM SIGCOMM*, 2009.

[20] K. Obraczka and P. Danzig. Finding low-diameter, low edge-cost, networks. Technical report, University of Southern California, 1997.

[21] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker. Topologically-aware overlay construction and server selection. In *IEEE INFOCOM*, 2002.

[22] G. Wang, D. G. Andersen, M. Kaminsky, K. Papagiannaki, T. S. E. Ng, M. Kozuch, and M. Ryan. c-Through: Part-time optics in data centers. In *ACM SIGCOMM*, 2010.

[23] H. Wu, G. Lu, D. Li, C. Guo, and Y. Zhang. MDCube: A high performance network structure for modular data center interconnection. In *ACM CoNext*, 2009.