Data Center Switch Architecture in the Age of Merchant Silicon

Nathan Farrington, Erik Rubow, and Amin Vahdat
Dept. of Computer Science and Engineering
UC San Diego
{farrington,erubow,vahdat}@cs.ucsd.edu

***Abstract*—Today, data center networks that scale to tens of thousands of ports require the use of highly-specialized ASICs, with correspondingly high development costs. Simultaneously, these networks also face significant performance and management limitations. Just as commodity processors and disks now form the basis of computing and storage in the data center, we argue that there is an opportunity to leverage emerging high-speed commodity merchant switch silicon as the basis for a scalable, cost-effective, modular, and more manageable data center network fabric. This paper describes how to save cost and power by repackaging an entire data center network as a distributed multi-stage switch using a fat-tree topology and merchant silicon instead of proprietary ASICs. Compared to a fat tree of discrete packet switches, a 3,456-port 10 Gigabit Ethernet realization of our architecture costs 52% less, consumes 31% less power, occupies 84% less space, and reduces the number of long, cumbersome cables from 6,912 down to 96, relative to existing approaches.**

I. Introduction

With the help of parallel computing frameworks such as MapReduce [1], organizations routinely process petabytes of data on computational clusters containing thousands of nodes. For these massively parallel workloads, the principal bottleneck is often not the performance of individual nodes, but rather the rate at which nodes can exchange data over the network. Many data center network (DCN) applications demonstrate little communication locality, meaning that the communication substrate must support high aggregate bisection bandwidth for worst-case communication patterns. Unfortunately, modern DCN architectures typically do not scale beyond a certain amount of bisection bandwidth [2] and become prohibitively expensive well in advance of reaching their maximum capacity [3], in some cases oversubscribed by a factor of 240 [4].

There is interest in replacing these expensive packet switches with many smaller, commodity switches, organized into a fat-tree topology [3]. But as the number of packet switches grows, so does the cabling complexity and the difficulty of actually constructing the network, especially on a tight schedule and with a minimum amount of human error. Fat trees have been used successfully in telecom networks [5], HPC networks [6], and on chips [7], but not yet in data center Ethernet networks. One reason is the fear of the resulting cabling complexity from trying to interconnect thousands of individual switches and the overhead of managing a large number of individual switch elements.

In addition to the cabling problem, networks of discrete packet switches are unnecessarily wasteful in the data center. The relative close proximity of the compute nodes and the single administrative domain provide opportunities for eliminating redundant components, such as packaging, power conditioning circuitry, and cooling. Multiple CPUs and memory banks could be consolidated to save cost and power. Consolidation could also reduce the cost of inter-switch links, which often use expensive and power hungry optical transceivers.

Our goal is to design a multi-stage switch architecture leveraging merchant silicon to reduce the cost, power consumption, and cabling complexity of DCNs, while also increasing the bisection bandwidth available to parallel applications such as MapReduce. In essence, we repackage a fat tree of discrete packet switches as a single distributed multi-stage switch, while also eliminating redundant components to save cost and power. We describe one realization of our architecture, a 3,456-port 10 Gigabit Ethernet (10GbE) switch with 34.56 Tb/s of bisection bandwidth. Using commodity 24-port 10GbE merchant silicon as the fundamental building block, we further reduce cost, power, and cabling complexity by implementing our own Layer 2 Ethernet extension protocol (EEP) in hardware to aggregate four 10GbE links into a single 40GbE link, and vice versa. The combination of custom packaging and the EEP protocol reduces the number of inter-switch cables from 6,912 to just 96. When 64-port 10GbE switch silicon becomes available (likely late 2009), our architecture should generalize to 65,536 ports of 10GbE.

A. Bisection Bandwidth, Cabling Complexity, and Fat Trees

The primary goal when constructing a DCN for a computational cluster is to provide enough bisection bandwidth so that communication-intensive parallel computations can maintain high-levels of CPU utilization. A bisection of a network is a partition into two equally-sized sets of nodes. The sum of the capacities of links between the two partitions is called the bandwidth of the bisection. The *bisection bandwidth* of a network is the minimum such bandwidth along all possible bisections. Therefore, bisection bandwidth can be thought of as a measure of worst-case network capacity.

We define cabling complexity to be the number of long inter-switch cables required to construct a particular DCN. Short intra-rack cables or cables that cross between adjacent racks are not difficult to install and maintain, but long cables

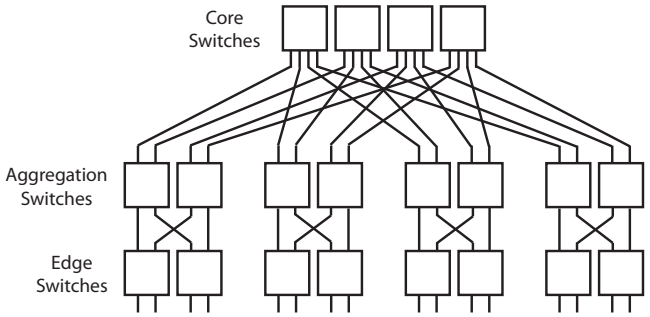

Fig. 1: A 3-tier (5-stage) fat tree using 4-port switching elements. This example is equivalent to a 16-port switch.

require planning and overhead cable trays. They are also more difficult to test and to replace after a break. This in turn increases the cost of the network.

The fat-tree topology is very promising because it provides an enormous amount of bisection bandwidth while using only small, uniform switching elements. Figure 1 shows a small example of the fat tree topology constructed from 4-port switching elements. Bandwidth scaling is achieved by having multiple, redundant paths through the network. However, if we are not careful, these links can translate into a large degree of cabling complexity, making the network impractical to construct and maintain.

II. Technologies

DCNs are composed of racks, switches, cables, and transceivers. This section briefly reviews these four technologies.

A. Racks

Almost all data processing and networking equipment is housed in large metal racks [8]. A typical rack measures 0.6 m wide by 1.0 m deep by 2.0 m high, has an unloaded weight of 170 kg, and can support a maximum load of 900 kg [9]. A 2.0 m high rack is partitioned into 42 vertical rack units (denoted as RU, or simply U). Each RU is 44.45 mm (1.75 in) high. Rack-mountable equipment occupies one or more rack units.

Racks are lined up side-by-side in rows. To assist with cooling, rows face front-to-front and back-to-back to form what are called *cold aisles* and *hot aisles* [10]. A cold aisle is at least 1.22 m (4 ft) wide and allows human access to the front panels of the racks. A hot aisle is at least 0.9 m wide and is heated by the air exhaust from the servers' cooling fans. Most cables are in the hot aisles.

Generally, there are three ways to route cables between racks. If the racks are in the same row, then the simplest way is to run the cables inside the racks. Specialized wire management solutions exist to facilitate this. If there are lots of cables, then it may be best to leave the top one or two RUs empty. If the racks are in different rows, then it is common to run the cables along the ceiling, suspended in overhead cable trays. This is standard practice since it reduces clutter and prevents safety hazards [10]. Some data centers have raised floors with removable tiles. The space underneath the floor can also be used to route cables, including power cables.

B. Switches

The most common Ethernet switch in the data center is the so-called *top-of-rack* (TOR) switch. This is a 1-RU switch that is placed in the top position in a rack and connects to all of the compute nodes in that rack. These switches are becoming commodities due to the recent emergence of merchant silicon. TOR1G in Table I is a 48-port GbE / 4-port 10GbE switch. Prices typically range between $2,500 and $10,000. TOR10G is a 24-port 10GbE switch, with prices between $5,000 and $15,000. Both switches consume approximately 200 W of power.

	TOR1G	TOR10G	EOR
GbE Ports	48	0	0
10GbE Ports	4	24	128
Power (W)	200	200	11,500
Size (RU)	1	1	33

TABLE I: Ethernet switch models. Prices vary.

Also present in data centers are so-called *end-of-row* (EOR) switches. An EOR switch derives its name from the fact that it is placed in a rack, sometimes by itself due to its size, and all nearby switches in other racks connect directly to it. EOR switches are also called *modular* switches, because they accept a variety of modules, called *line cards*, with different Layer 1 interfaces and different combinations of switching fabrics and network processors. EOR in Table I is a 128-port 10GbE switch. Prices range between $500,000 and $1,000,000.

C. Cables and Transceivers

Table II lists properties of Ethernet cables and transceivers.

The term "Ethernet cable" usually refers to unshielded twisted pair (UTP) copper cable. UTP cable is available in different grades, which do not always correspond directly with IEEE Layer 1 standards. For example, Cat-5e allows GbE links of up to 100 m and Cat-6a allows 10GbE links of up to 100 m. Twinax ("InfiniBand") shielded cable allows 10GbE links of up to 15 m, but with different engineering tradeoffs. For example, since twinax is shielded, the manufacturing cost is higher than UTP, but transceivers are less expensive and consume less power.

Optical fiber cables are becoming more common in the data center as 10GbE is introduced. Multimode fiber (MMF) is preferred over single-mode fiber (SMF) because optical transceivers for multimode fiber are significantly less expensive. OM-3 "laser grade" MMF allows 10GbE links longer than 300 m. Distribution fiber cable can pack multiple fiber strands within a single physical cable. Typical densities are multiples of 12. A bidirectional link actually requires two fibers, so a 72-strand fiber cable only has 36 bidirectional communication channels.

A transceiver converts signals between a circuit board and a communications cable. For UTP and twinax copper cable,

	Cat-6	Twinax	MMF120	Units
Cable cost	0.41	11.2	23.3	$/m
Cable weight	42	141	440	g/m
Cable diameter	5.5	7	25	mm
Gigabit Ethernet				
Standard	1000BASE-T	N/A	1000BASE-SX	
Range	100	N/A	550	m
Transceiver cost	10	N/A	36	$
Transceiver power	0.42	N/A	0.5	W
10 Gigabit Ethernet				
Standard	10GBASE-T	10GBASE-CX4	10GBASE-SR	
Range	30	15	300	m
Transceiver cost	100	100	250	$
Transceiver power	6	0.1	1	W
40 Gigabit Ethernet				
Standard	N/A	N/A	40GBASE-SR4	
Range	N/A	N/A	300	m
Transceiver cost	N/A	N/A	600	$
Transceiver power	N/A	N/A	2.4	W

TABLE II: Data center communication cables and transceivers. The Telecommunications Industry Association recommends Cat-6 UTP instead of Cat-5e for new data center installations [10]. MMF120 is 120 strands of MMF in a single cable. Note that the maximum range depends on the signal bandwidth of the cable. Ranges shown here reflect the highest-quality cable at the time of standardization.

transceivers are packaged into a chip and placed directly onto the circuit board. These standards include 1000BASE-T, 10GBASE-CX4, and 10GBASE-T, for GbE over UTP, 10GbE over twinax, and 10GbE over UTP, respectively. UTP uses the popular "RJ-45" connector, whereas twinax uses the Fujitsu "InfiniBand" connector.

There are a number of different Layer 1 protocols for optical fiber, depending on the carrier wavelength, the bit rate, the encoding, multimode vs. single mode, etc. The networking industry has taken the 7-Layer OSI model quite literally by separating the Ethernet MAC functions (Layer 2) from the Ethernet PHY transceiver functions (Layer 1) through the use of pluggable optical transceiver modules. These modules are standardized by various Multi Sourcing Agreements (MSAs) to define standard mechanical and electrical form factors. This allows Ethernet device manufacturers to specialize in Layer 2 and Layer 3 functionality, and allows the end user to choose their desired Layer 1 protocol just by plugging in a module.

The most common optical module standard for GbE is the SFP module [11]. Recently, SFP+ [12] has become the dominant standard for 10GbE. These modules use the LC connector, which couples two fibers together into close proximity. The QSFP transceiver is being developed for 40GbE using an emerging standard to be named 40GBASE-SR4. This module will be slightly larger than SFP and SFP+ modules and will use an MT connector with 8 optical fibers (4 send, 4 receive). MT connectors are available with up to 72 optical fibers.

III. Motivation

Al-Fares et al. [3] proposed the construction of a 3-tier fat tree of 2,880 commodity 48-port GbE TOR switches, providing 27.648 Tb/s of bisection bandwidth. However, this network is almost impossible to construct following their suggested packaging. It would require 1,128 separate cable bundles, each of which must be manually and independently routed and installed. They partitioned the network into 48 separate pods (switches and servers) that communicate through a core switch array of 576 48-port switches. Assuming that a pod has a 3 m diameter and the distance between two pods is 1.5 m, their proposal would require 226,972 m of cable, weighing 9,532 kg.

Rather than dismiss the fat-tree topology outright, we believe that two simple rules will at least allow the construction of small fat trees, thereby gaining the benefit of the fat-tree topology and commodity switches. First, minimize the number of unique cable bundles. The Al-Fares proposal required $k(k-1)/2$ bundles because the top tier of the network was collocated with the k pods. However, by placing the top tier in a central location, only k bundles are required (from the pods to the central location). Second, use optical fiber cable instead of copper cable. Compared to copper, fiber is much thinner and lighter. Cables containing more than 72 fibers are readily available.

Following these rules, we propose an alternative fat-tree network based on the recently commoditized 24-port 10GbE TOR switch. At the time of writing, these switches are available from several manufacturers for between $5K and $15K. Connecting 720 such switches into a 3-tier fat-tree topology will yield a bisection bandwidth of 34.56 Tb/s. Each of the 24 pods will contain 24 switches, co-located in a single rack. The 144 core switches will be distributed over 4 racks. The 3,456 long fiber cables can be combined into just 24 bundles of cable. We estimate the total cost to be about $7.2M, which is $2,000 per port. This network is both less expensive and provides more bisection bandwidth than a traditional DCN constructed from modular packet switches [3].

A. Merchant Silicon and the Commodity Top-of-Rack Switch

The last decade has seen the introduction of merchant silicon: networking ASICs produced for the network equipment mass market. For example, there are currently at least three separate makers of 24-port 10GbE switch ASICs: Broadcom, Fulcrum, and Fujitsu. All three product offerings provide similar degrees of functionality. Table III shows that the three chips also have similar properties.

Maker	Broadcom	Fulcrum	Fujitsu
Model	BCM56820	FM4224	MB86C69RBC
Ports	24	24	26
Cost	NDA	NDA	$410
Power	NDA	20 W	22 W
Latency	< 1 us	300 ns	300 ns
Area	NDA	40 x 40 mm	35 x 35 mm
SRAM	NDA	2 MB	2.9 MB
Process	65 nm	130 nm	90 nm

TABLE III: Comparison of 10GbE switch ASICs. We could not obtain some numbers due to NDA requirements, but we assume they are comparable.

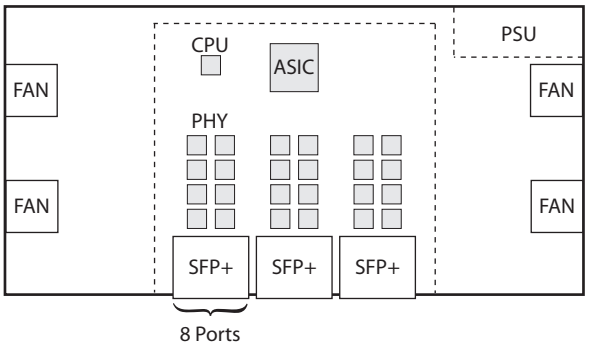

Fig. 2: Layout of a 24-port 10GbE TOR switch. The power supply unit (PSU) and fans are physically separated from the main circuit board for signal integrity. Each SFP+ cage allows for up to 8 SFP+ modules. There is one PHY chip per SFP+ module.

Several companies have used these ASICs to create 24-port TOR switches. Since purchasing merchant silicon is much easier and less expensive than creating a custom switch ASIC, these companies can offer TOR switches at a lower cost to consumers, which was the original motivation for constructing an entire data center network out of TOR switches. Because these manufacturers choose from the same pool of merchant silicon, they differentiate their products primarily through the features in their custom software. But the board and chassis designs are usually very similar.

Figure 2 shows the board layout of a 24-port 10GbE TOR switch; the cost and power consumption of these parts are given in Table VI.

A switch ASIC by itself is not an Ethernet switch; it is only the fast path. To build a functional switch, several other chips are needed, as well as software for handling networking protocols such as Minimum Spanning Tree or OSPF. This software runs on a local CPU. The CPU does not need to be very powerful since the amount of required computation is typically small. Common operating systems include Linux and VxWorks. The CPU also requires supporting chips such as DRAM for system memory and flash for secondary storage.

A switch also contains several PHY chips, which are used to bridge the Layer 1 protocol supported by the ASIC with the Layer 1 protocol exposed by the switch. For example, an ASIC may only support XAUI natively, but by adding a PHY chip, the XAUI protocol can be converted into 10GBASE-T, which allows Cat-6a cables to be connected to the switch.

These three classes of chips: ASICs, CPUs, and PHYs, are the primary building blocks of switches and routers. Sometimes switches will contain other chips such as flash memory for nonvolatile storage or FPGAs for glue logic. Also present are power supply units (PSUs) and fans. Although optical transceiver modules are not technically part of the switch, we include them in our analysis since they contribute greatly to the overall cost and power.

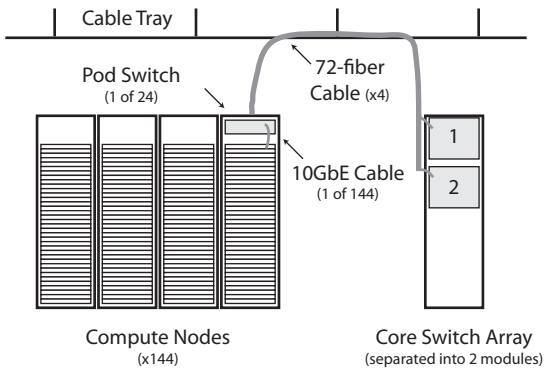

Fig. 3: Example deployment of the 3,456-port switch.

IV. Design of a 3,456-port Switch

This section describes the design of a 3,456-port 10GbE switch. What makes this switch novel is that it is comprised entirely of merchant silicon, connected in a fat-tree topology. In contrast, large Ethernet switches from traditional network equipment manufacturers use multiple proprietary ASICs and a crossbar topology. We introduce one simple and completely optional ASIC called EEP to further reduce the cost, power consumption, and cabling complexity of the overall design. In section V, we show how this switch is strictly better than a fat tree of commodity TOR switches.

A. Overview

Figure 3 shows an overview of the 3,456-port switch. Rather than build one monolithic switch, we separate the design into 24 pod switches and a core switch array. A pod switch occupies 4 rack units of space, whereas each of the two core switch array modules occupies 9 rack units of space. A pod switch is basically the bottom two tiers of the fat tree; the core switch array forms the top tier.

Each pod switch can function as a standalone 144-port 10GbE switch. But when connected to the core switch array, the pod switches act as a single non-interfering [13, p112] switch. The pod switches can also be incrementally deployed as the network is built out. When fully deployed, the switch has 3,456 ports and 34.56 Tb/s of bisection bandwidth. Each pod switch connects to the core switch array with four parallel cables, each cable carrying 72 multimode fibers. These cables can be routed in an overhead cable tray.

The core switch array is not a monolithic switch; it is a collection of 144 individual 24-port switches. We chose to separate the array into two modules to provide fault tolerance and to allow a limited form of incremental deployment. It is also possible to divide the core switch array into 4 modules, but our analysis showed that it would increase the overall cost of the system. We show the core switch array installed into a single rack for illustration purposes, but in actual deployment, the modules would be physically separate to provide fault tolerance of critical network infrastructure, e.g. in the face of localized power failure.

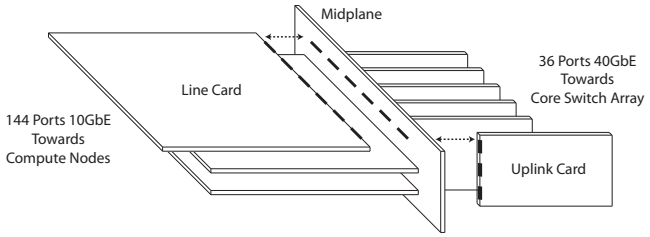

Fig. 4: Pod switch physical organization. The midplane separates and connects the three line cards and the six uplink cards. The horizontal-vertical arrangement allows each line card to connect directly with each uplink card, and vice versa.

B. Pod Switch: Organization

The pod switch is constructed from multiple circuit boards, assembled as shown in Figure 4. At the center of the switch chassis is a single midplane circuit board. The midplane provides three important functions. First, it connects the line cards to the uplink cards through high-density high-speed electrical connectors. Second, it provides power to all of the line cards and daughter cards. Third, it contains a CPU that manages the state of the pod switch. The midplane design is quite simple since communication signals from each line card pass directly to each uplink card, and vice versa. Since the line cards are mounted horizontally and the uplink cards are mounted vertically, we call this a *horizontal-vertical arrangement*.

The electrical connectors are inexpensive and available from several vendors [14]. The male connectors are attached to the midplane and the female connectors are attached to the cards. Each connector has at least 16 pins to support 8 differential channels of 10GbE between one line card and one uplink card. There are 18 connectors per pod switch, plus additional connectors for power distribution.

In addition to these boards, the chassis also contains dual redundant power supply units (PSUs), which have been omitted from Figure 4 for clarity. They are located in the back of the chassis on the left and right sides of the uplink cards.

C. Pod Switch: Line Card

Figure 5 shows the layout of a line card. Each line card essentially replaces four discrete 24-port switches from the edge layer of the network and eliminates redundant components. The four switch ASICs separate the board into two halves. The bottom half of the board contains 48 SFP+ optical transceiver cages and 48 PHY chips, to convert between the SFP+ 10G electrical interface (SFI) and the IEEE XAUI standard for 10GbE. Figure 5 shows 12 solid traces running from the SFP+ cages, to the PHYs, to one of the switch ASICs. The other traces are shown with dotted lines to indicate that they also continue, but have been removed from the diagram for clarity.

The CPU is shown off to the side. It connects directly to the four switch ASICs using the PCI Express bus. This allows the CPU software to configure the switch ASICs, and allows exceptional Ethernet frames to be sent from the switch ASICs over the PCI Express bus to the CPU for further processing.

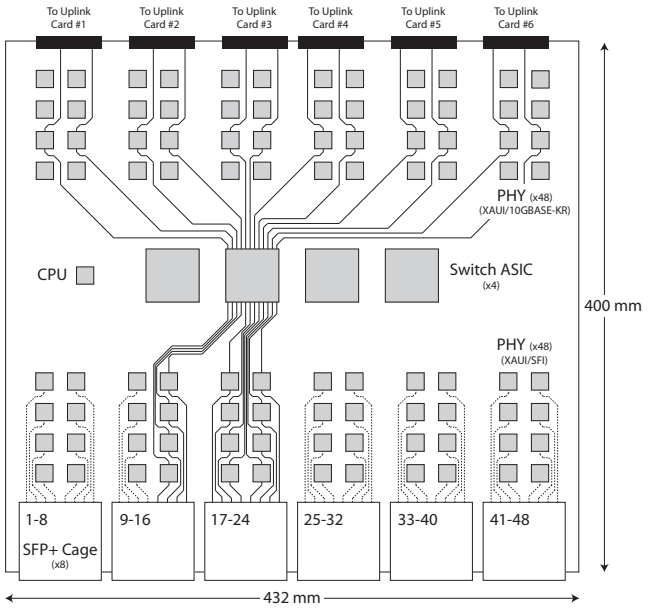

Fig. 5: Pod switch line card. Not all traces are shown. Partial traces are drawn with dotted lines.

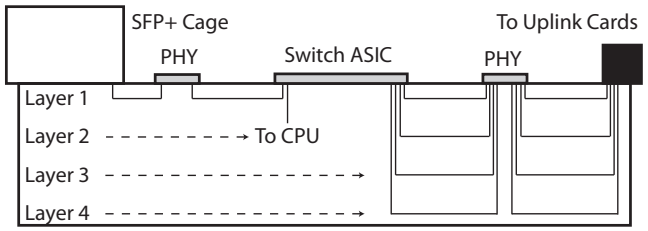

Fig. 6: Layers of the pod switch line card circuit board. This figure is not to scale.

The top half of the board contains an additional 48 PHYs and 6 electrical connectors. These PHYs convert between XAUI and 10GBASE-KR, which is the IEEE standard for 10GbE over backplanes. 10GBASE-KR is more robust than XAUI when passing between different boards. Figure 5 shows an additional 12 traces from the switch ASIC to one row of PHYs on the top half of the board. Two traces from each switch ASIC are routed to each of the 6 uplink cards.

Circuit boards are divided into multiple layers and can be categorized as either signal layers or power/ground layers. Figure 6 shows that the line card requires four separate signal layers to route all copper traces. Congestion occurs in the top half of the board where each switch ASIC must connect to each connector.

D. Pod Switch: Uplink Card

The uplink card performs two functions. First, it acts as a switch fabric for the pod switch, allowing the 12 switch ASICs on the 3 line cards to connect to each other. Second, it forwards traffic to and from the core switch array.

Figure 7 shows the layout of an uplink card. Each uplink card essentially replaces two discrete 24-port switches from

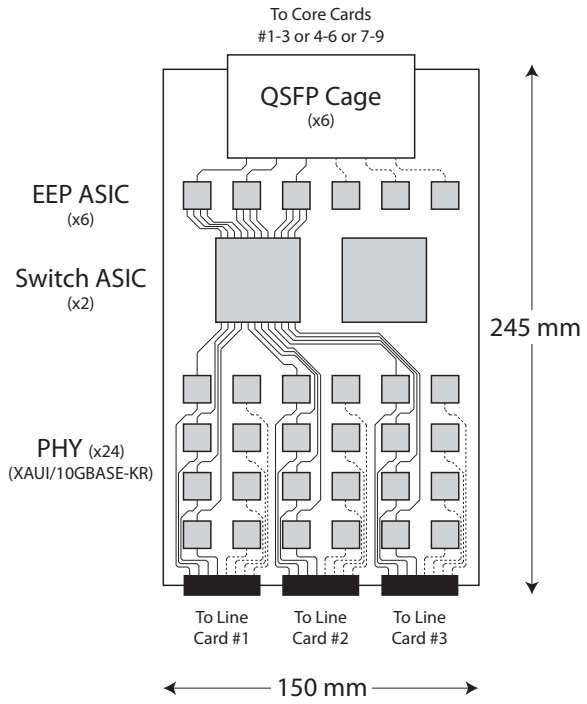

Fig. 7: Pod switch uplink card. Not all traces are shown. Partial traces are drawn with dotted lines.

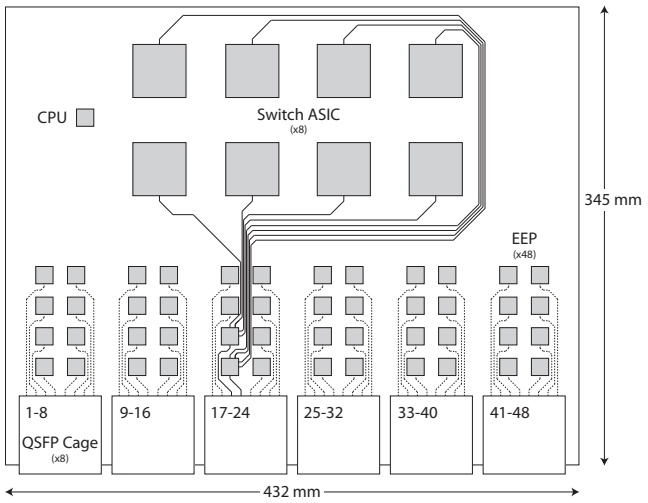

Fig. 8: Core switch array card.

the aggregation layer of the network. Like the line card, the uplink card is also divided into two halves. The bottom half connects to the midplane with electrical connectors. The 8 traces from each line card are routed to 8 PHYs, and then split between the two switch ASICs. In Figure 7, traces connecting to the first switch ASIC are shown as solid lines, whereas traces connecting to the second switch ASIC are partially shown as dotted lines. This half of the board requires two separate signal layers.

The top half of the board contains six EEP ASICs and six QSFP cages. Each EEP ASIC connects to four ports of the switch ASIC using the XAUI protocol and connects to one of the QSFP modules using the QSFP electrical interface. The EEP ASIC is described in more detail in section VI. Essentially, the top half of the board is aggregating the 24 ports of 10GbE into 6 ports running the custom EEP protocol at 40 Gb/s.

Experience from a prototype system indicates that most exceptional packet processing happens either on the line card or in the core switch array, rather than on the uplink card. For this reason, we deliberately omit a dedicated CPU from the uplink card. Instead, all 6 uplink cards are managed by system-level CPU located on the midplane. It is also possible to place a CPU on each uplink card with a modest increase in cost.

E. Core Switch Array

The core switch array contains 18 independent core switch array cards, partitioned into two modules with 9 cards each. The choice of core switch array packaging is largely arbitrary since the 144 separate 24-port switches do not communicate with each other directly. The cards do not connect to a backplane and their co-location is merely a matter of simplifying cable management and packaging. The absence of a backplane greatly reduces the cost of the core switch array and allows our design to scale to even larger fat-tree switches in the future.

Figure 8 shows the layout of a core switch array card. Each card essentially replaces eight discrete 24-port switches from the core layer of the network. Each card contains a CPU, 8 switch ASICs, 48 EEP ASICs, and 48 QSFP modules. Figure 8 shows 8 traces from two EEP ASICs connecting to the 8 different switch ASICs. Although there are a total of 192 connections between the EEP ASICs and the switch ASICs, only 8 signal layers are required, one per switch ASIC. A 9th signal layer is required for the CPU to connect to the switch ASICs. Current circuit board manufacturing techniques make it difficult to design boards with more than 10 signal layers, so we feel that this layout is the most efficient in terms of board area utilization without becoming an engineering challenge. More exotic designs can double the number of signal layers to 20, but the engineering challenge increases correspondingly.

Figure 9 shows the internal topology of the 3,456-port switch. In order to achieve full bisection bandwidth, each pod switch must connect to each of the 144 core switch ASICs. If a mistake is made when constructing the network, a pod switch might end up with two or more parallel links to the same core switch ASIC, which would subtract from the overall bisection bandwidth.

By design, the EEP protocol helps prevent installation mistakes. The four links leaving an ASIC on a pod switch uplink card travel to a single core switch array card, where they connect to four *different* core switch ASICs. The core switch array card is designed such that the top row of 24 QSFP modules connect only to the top row of switch ASICs, and the bottom row of QSFP modules only to the bottom row of switch ASICs. Therefore, as long as each pod switch connects to *exactly* one upper and one lower port on each

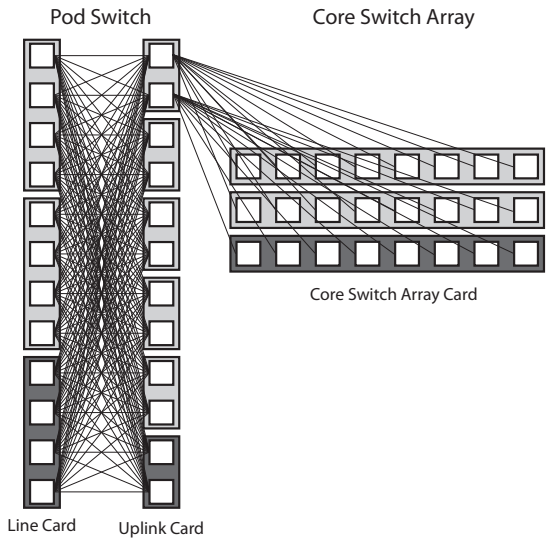

Fig. 9: Topology of the 3,456-port switch. Only one of the 24 pod switches is shown. When fully deployed, each ASIC on each line card connects to each ASIC on each uplink card, and each pod switch connects to each of the 144 core switch ASICs. Each uplink card connects to exactly three core switch array cards, for a total of 18 core switch array cards in a fully deployed system. The fact that this is a small, finite number is useful for scaling to larger fat-tree switches in the future.

core switch array card, the network will achieve full bisection bandwidth.

F. Incremental Deployment

The 3,456-port switch was designed to support incremental deployment. A very small network can start with a single pod switch and no core switch array. Each pod switch line card can be installed as more ports are needed. However, all 6 uplink cards must be installed for non-interfering throughput. Uplink cards can also be installed incrementally to reduce cost or if some degree of oversubscription can be tolerated. Each uplink card provides $1/6$ of the total aggregate pod switch bandwidth. For a single pod switch, the QSFP modules are not needed.

A single pod switch can provide up to 1.44 Tb/s of bisection bandwidth. Adding a second pod switch then requires the installation of the core switch array. However, if fault tolerance is not a concern, then it is possible to install just one of the two core switch array modules. Also, if some degree of oversubscription can be tolerated, then not all nine core switch array cards need be installed. Each core switch array card provides $1/18$ of the total aggregate core switch array bandwidth.

One core switch array module can support up to 12 pod switches with no oversubscription. In this deployment, each pod switch will connect all of its uplink cables to the same core switch array module, rather than splitting them between the two modules. Once the 13th pod switch is installed, both core switch array modules will be required. The four 72-strand fiber cables from each of the pod switches must be distributed evenly among the core switch arrays. This could require disconnecting and reconnecting a small number of cables.

V. Comparison

In this section, we compare the design from section IV with two other possible implementations of a 3,456-port 10GbE fat-tree switch. We call these Network 1, 2, and 3.

Network 1. This network can be constructed immediately without any hardware engineering. It is comprised of 720 discrete 24-port 10G TOR switches (see TOR10G in Table I) connected in the same fat-tree topology described in section IV. All inter-switch links use SFP+ optical transceivers and multimode fiber cable.

Network 2. This network requires board and chassis design. Instead of discrete TOR switches, we use the boards and chassis from section IV. However, Network 2 does not use EEP or 40GbE QSFP modules. Instead, Network 2 simply replaces these components with additional 10GbE PHYs and SFP+ modules.

Network 3. This network is precisely the design described in section IV. It has the highest non-recurring engineering (NRE) costs since it requires board, chassis, and ASIC design. However, we will see that the higher NRE costs translate into lower capital and operational costs.

In order to eliminate non-comparable costs such as software engineering and profit margins, we reduce all three designs to their major hardware components, namely chips and optical transceivers. We exclude certain components such as circuit boards, power supplies, fans, chassis, and cables. We do not expect these components to contribute more than 10% of the total cost or power consumption.

Our rationale for EEP comes from the properties of optical transceivers. From Table IV, it is actually less expensive in terms of cost, power consumption, and physical area to aggregate to higher-rate links, and then to disaggregate back to the original rate.

	SFP	SFP+	QSFP
Rate	1 Gb/s	10 Gb/s	40 Gb/s
Cost/Gb/s	$35	$25	$15
Power/Gb/s	500mW	150mW	60mW

TABLE IV: Comparison of three optical transceiver standards. Higher-rate transceivers are more efficient both in cost and in power consumption. The prices shown here are estimates provided to us by a large OEM manufacturer, and these prices will change over time.

Table V lists part counts for the three networks. Simply by repackaging the components, a significant number of SFP+ modules can be eliminated. Adding EEP and QSFP modules can further reduce part counts.

Table VI lists cost and power consumption estimates for the different parts. Power consumption estimates were obtained from data sheets and product briefs. Cost estimates, which

	Network 1	Network 2	Network 3
ASIC	720	720	720
CPU	720	132	132
PHY	17,280	17,280	10,368
SFP+	17,280	10,368	3,456
EEP	0	0	1,728
QSFP	0	0	1,728

TABLE V: Part counts.

were obtained from distributors, necessarily represent a snapshot in time, so we recommend using current pricing rather than relying on these numbers.

Part	Cost ($)	Power (W)
ASIC	410	22
CPU	130	8
PHY	10	0.8
SFP+	250	1
EEP	10	2
QSFP	600	2.5

TABLE VI: Part costs and power estimates.

Table VII compares the three networks. Although all three networks have equivalent bisection bandwidth, Network 2 is strictly better than Network 1 and Network 3 is strictly better than Network 2. We have not accounted for one-time NRE costs, which will be larger for Network 3 than for Network 2 due to the EEP ASIC. Table VII shows that the largest gains come from repackaging the fat tree. The EEP ASIC also provides a significant improvement.

	Network 1	Network 2	Network 3
Bisection Bandwidth (Tb/s)	34.56	34.56	34.56
Cost ($M)	4.88	3.07	2.33
Power Consumption (kW)	52.7	41.0	36.4
Cabling Complexity	3,456	96	96
Space (Rack Units)	720	192	114

TABLE VII: Comparison of three equivalent fat-tree networks.

Table VII shows that optical transceiver modules account for 21% of the overall power consumption and 81% of the overall cost. Our design would benefit greatly from less expensive optical transceiver technology. Two companies, Infinera [15] and Luxtera [16] are developing less expensive optical transceivers by utilizing recent advances in photonic integrated circuits that allow multiple optical transceivers to be fabricated on a single chip [17]. For example, Luxtera uses their chips to build a 40 Gb/s active cable, which is a fiber cable with transceivers permanently attached to both ends.

VI. EEP: Ethernet Extension Protocol

This section describes the design of the EEP ASIC, an Ethernet traffic groomer. A traffic groomer aggregates frames from multiple low-rate links and tunnels them over a single higher-rate link. There are two existing standards for traffic grooming: SONET/SDH and IEEE 802.1ad. SONET/SDH was designed for transporting both voice and data traffic over long distances of optical fiber while also supporting QoS and providing very low jitter. IEEE 802.1ad [18] (VLAN Tunneling) bridges multiple remote Ethernet networks connected to a single carrier network. We decided that SONET/SDH provided too much functionality for our simple task.

At the same time, IEEE 802.1ad requires an entire Ethernet frame to be stored on chip before forwarding it through a higher-rate port. This is a limitation of the Ethernet protocol, which does not allow frame fragmentation. Store-and-forward can become a significant source of latency, especially when used in a multi-stage switch such as a fat tree.

Thus, we designed a lightweight protocol, called EEP, to eliminate the latency and buffering requirements of IEEE 802.1ad. Although EEP is new, the ideas are not; we draw upon well-known techniques from other synchronous protocols such as ATM. EEP is very simple, requires no configuration, and is completely invisible to the rest of the network. This means that EEP is compatible with current data center bridging efforts such as IEEE 802.1Qau, 802.1Qaz, and 802.1Qbb.

A. EEP Design

EEP provides the abstraction of a set of 16 virtual Ethernet links, multiplexed over a single physical Ethernet link. EEP breaks up an Ethernet frame into a sequence of 64B segments, and encapsulates each segment into an EEP frame. Each EEP frame originating from the same switch port is assigned the same Virtual Link ID, similar to a VLAN tag.

Figure 10 shows the EEP frame format. The very first EEP frame in the sequence (corresponding to the first 64B of an Ethernet frame) has the SOF (Start of Frame) bit set. All other EEP frames in the sequence have the SOF bit cleared. The last EEP frame has the EOF (End of Frame) bit set. The last EEP frame will likely not have a 64B payload. In this case, the LEN (Length) bit is set, which indicates that the second byte in the EEP frame is part of the header rather than the payload. This second header byte records the number of valid bytes in the payload. In the common case, each EEP frame will use one header byte. Only final EEP frames use two header bytes. Three unused bits are reserved for future use.

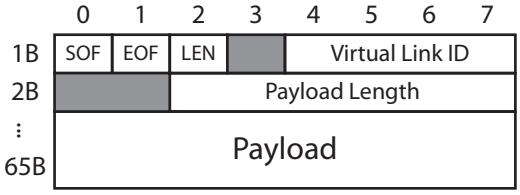

Fig. 10: EEP frame format.

Although IEEE 802.1ad is limited to store-and-forward, EEP can use cut-through switching to begin transmitting an Ethernet frame on an output port after the first EEP frame has been received. However, one must be careful when using cut-through switching. The EEP receiver should never run out of bytes part way through the transmission of an Ethernet frame. This will corrupt the frame. This scenario can be avoided if the EEP transmitter services the input queues in a timely and round-robin fashion. If an EEP frame happens to get dropped

and omitted from a reassembled packet, the frame check sequence (i.e., checksum) will invalidate the larger Ethernet frame, since it is passed on as part of the EEP payload.

One drawback to EEP is that it adds overhead to the link. A standard 1500B Ethernet payload actually requires 1538B for transmission, once the preamble, header, frame check sequence, and inter-frame gap are accounted for. EEP will add an additional 25B of headers to create 24 EEP frames from the original Ethernet frame. Normally, this would reduce the capacity of the link by 1.6%, a small but annoying amount. However, we can overclock the PHYs and the optical transceivers by at least 1.6% to recover the lost capacity. This technique is frequently used in both proprietary and merchant silicon.

B. Implementation

We implemented both IEEE 802.1ad and EEP in hardware using a Xilinx Virtex-5 LX110T FPGA. Due to our limited resources, we simplified the implementation to use four GbE ports and one 10GbE port rather than four 10GbE ports and one 40GbE port. However, our design generalizes to these faster rates.

Our custom logic totaled 3,480 lines of Verilog-2001 and we used Xilinx's IP for the PHY, MAC, and FIFO cores. We verified the designs with a custom test bench totaling 1,200 lines of SystemVerilog-2005 and simulated the designs with ModelSim 6.4. We used Xilinx XST 10.1 for logic synthesis.

Table VIII shows how much of our FPGA was used for both designs. The EEP implementation is slightly smaller because we leverage cut-through switching to reduce the buffer size. Our implementations were actually small enough to fit into Xilinx's second smallest LXT device, indicating that an ASIC conversion would produce an extremely inexpensive and low-power chip.

	IEEE 802.1ad	EEP
Flip Flops	5,427 (7%)	4,246 (6%)
LUTs	6,243 (9%)	5,004 (7%)
BlockRAM/FIFOs	11 (7%)	0 (0%)

TABLE VIII: Virtex-5 device utilization.

It is worth mentioning that some merchant silicon vendors offer custom ASIC design services, where they will integrate customer logic and IP into one of their existing devices. For example, Broadcom offers this service. One could imagine integrating EEP and other PHYs directly onto an existing 24-port 10GbE switch ASIC, to further reduce cost and power. However, such an approach is outside the scope of this paper.

C. Evaluation

Figure 11 shows the round-trip latency of both implementations. Because of store-and-forward, IEEE 802.1ad adds latency proportional to the size of the Ethernet frame. EEP adds a constant amount of latency regardless of frame size. We obtained these measurements using a NetFPGA [19] configured for packet generation and capture. We verified that both implementations work correctly under heavy load without dropping or corrupting frames. We also verified that the IEEE 802.1ad implementation interoperates with other Ethernet devices.

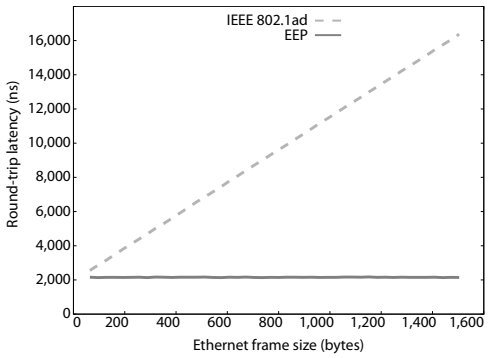

Fig. 11: Latency measurements of the EEP and IEEE 802.1ad implementations.

VII. Related work

Fat-Tree Networks: Fat trees have been used for decades in the HPC field. Perhaps the most famous example is the Thinking Machines CM-5 [6]. The CM-5 used proprietary 8-port switch ASICs with a custom frame format. The bottom two tiers of the fat tree were connected on circuit boards and the upper tiers used copper cables. A fully deployed CM-5 covered an area of 900 m^2.

Al-Fares et al. [3] were the first to suggest constructing a large Layer 3 data center network from a 3-tier fat tree using commodity TOR switches. They overcame the limitations of ECMP multipath routing with both static and dynamic techniques. Their static multipath algorithm relied on a novel use of TCAMs; although it is possible to implement the same search using hashing and commodity SRAM. The two different dynamic routing techniques were both simple and inexpensive to implement, and performed better than static routing.

PortLand [20] and VL2 [4] describe large Layer 2 Ethernet data center networks. The two principal challenges are to limit broadcast traffic and to allow for multipath routing. PortLand is a native Layer 2 network and translates Ethernet MAC addresses inside the network. This feature is supported by all high-performance merchant silicon that we have evaluated. VL2 uses IP-in-IP encapsulation, which adds at least 20B of overhead to each packet and requires end-host operating system modifications.

Fat-Tree Switches: Network equipment vendors have only recently begun building switches from merchant silicon using a multi-stage fat-tree topology internally. The Arista 7148SX [21] is a 48-port 10GbE TOR switch with six Fulcrum switch ASICs connected into a 2-tier fat tree on a single board. The larger Woven EFX-1000 [22] is a 144-port 10GbE modular switch with 18 Fulcrum switch ASICs connected in a 2-tier fat tree. Twelve line cards each contain twelve 10GbE ports and one ASIC. Six additional fabric cards each contain one ASIC. A midplane connects the line cards to the fabric

cards. Both of these designs seek to replace and individual TOR or EOR switch with a fat tree of smaller switch ASICs. Our design seeks to replace an entire data center network.

The Sun Microsystems 3,456-port InfiniBand switch [23] is constructed as a 3-tier fat tree of 720 Mellanox InfiniScale III [24] 24-port switch ASICs. Unlike our design, everything is packaged into a single chassis. Each of the 24 line cards contains 24 switch ASICs and each of the 18 fabric cards contains 8 switch ASICs. A midplane connects the line cards to the fabric cards using 432 connectors with 64 pin pairs each. One advantage of this monolithic design is that it eliminates expensive optical transceivers since all chips are close enough to communicate over copper circuit board traces. However, the high density leads to a cable management problem; proprietary splitter cables are used to aggregate three 10 Gb/s InfiniBand channels over one cable. The monolithic design also makes it difficult to support next-generation higher-density switch ASICs. While omitted for brevity, our pod-based design should generalize to much larger networks, for example 65,536 ports using 64-port 10GbE merchant silicon. In this case, each of the 64 pod switches will support 1,024 ports.

VIII. Conclusion

In this paper, we showed that the construction of fat-tree networks from discrete packet switches is not a scalable solution. We presented one instance of our architecture, a 3,456-port 10GbE switch, built using a fat-tree topology internally and merchant silicon whenever possible. Our design provides 34.56 Tb/s of bisection bandwidth when fully deployed.

We showed how an Ethernet Extension Protocol (EEP) could further reduce the cost, power consumption, and cabling complexity of such a switch by aggregating multiple lower speed links onto a single higher-speed link. We compared EEP to the traditional IEEE 802.1ad standard for Ethernet traffic grooming and showed that EEP is superior in terms of latency and on-chip buffering.

Packaging and cabling is only one aspect of constructing a large fat-tree-based Ethernet switch. There are also other issues which must be addressed such as managing the forwarding tables of individual switching elements, handling ARP and other broadcast traffic in a large Layer 2 domain, efficient multicast, fault tolerance, and routing TCP flows through the switch fabric. We, as well as others, are currently working on these related problems.

IX. Acknowledgments

This work was supported by the National Science Foundation and the Center for Integrated Access Networks through Grant No. 0812072, and by a US Department of Defense SMART scholarship. Brian Kantor provided first-hand knowledge of data center cabling problems and circuit board design techniques. Michael Florea helped guide the implementation of EEP. Glen Gibb and Adam Covington's NetFPGA packet generator was used to measure the performance of EEP.

References

[1] J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," in *USENIX Operating Systems Design and Implementation (OSDI)*, Dec. 2004.

[2] *Cisco Data Center Infrastructure 2.5 Design Guide*, Cisco Systems, Dec. 2007. [Online]. Available: http://www.cisco.com/univercd/cc/td/doc/solution/dcidg21.pdf

[3] M. Al-Fares, A. Loukissas, and A. Vahdat, "A Scalable, Commodity, Data Center Network Architecture," in *ACM SIGCOMM*, Aug. 2008.

[4] 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*, Aug. 2009.

[5] C. Clos, "A Study of Non-Blocking Switch Networks," *Bell System Technical Journal*, vol. 32, no. 5, pp. 406–424, Mar. 1953.

[6] C. E. Leiserson, Z. S. Abuhamdeh, D. C. Douglas, C. R. Feynman, M. N. Ganmukhi, J. V. Hill, D. Hillis, B. C. Kuszmaul, M. A. S. Pierre, D. S. Wells, M. C. Wong, S.-W. Yang, and R. Zak, "The Network Architecture of the Connection Machine CM-5 (extended abstract)," in *ACM Parallel Algorithms and Architectures (SPAA)*, 1992, pp. 272–285.

[7] P. Guerrier and A. Greiner, "A Generic Architecture for On-Chip Packet-Switched Interconnections," in *IEEE Design, Automation and Test in Europe (DATE)*, 2000, pp. 250–256.

[8] *Cabinets, Racks, Panels, and Associated Equipment*, Consumer Electronics Association Std. CEA-310-E, Dec. 2005. [Online]. Available: http://www.ce.org/standards/StandardDetails.aspx?Id=2470&number=CEA-310-E

[9] *Dell PowerEdge Rack Systems*, Dec. 2003. [Online]. Available: http://www.dell.com/downloads/global/products/pedge/en/rack_system.pdf

[10] *Telecommunications Infrastructure Standard for Data Centers*, ANSI Std. TIA-942, Rev. 2005, Apr. 2005. [Online]. Available: http://webstore.ansi.org/RecordDetail.aspx?sku=TIA-942:2005

[11] S. van Doorn, *Specification for SFP (Small Formfactor Pluggable) Transceiver*, SFF Committee Std. INF-8074i, Rev. 1.0, May 2001. [Online]. Available: ftp://ftp.seagate.com/sff/INF-8074.PDF

[12] A. Ghiasi, *Specifications for Enhanced 8.5 and 10 Gigabit Small Form Factor Pluggable Module SFP+*, SFF Committee Std. SFF-8431, Rev. 4.1, Jul. 2009. [Online]. Available: ftp://ftp.seagate.com/sff/SFF-8431.PDF

[13] W. J. Dally and B. Towles, *Principles and Practices of Interconnection Networks*. Morgan Kaufmann, 2004.

[14] The Molex website. [Online]. Available: http://www.molex.com/

[15] L. Geppert, "A Quantum Leap for Photonics," *IEEE Spectr.*, vol. 41, no. 7, pp. 16–17, Jul. 2004.

[16] K. Greene, "Silicon photonics comes to market," *MIT Technology Review*, Aug. 2007. [Online]. Available: http://www.technologyreview.com/Infotech/19261/?a=f

[17] H. Rong, R. Jones, A. Liu, O. Cohen, D. Hak, A. Fang, and M. Paniccia, "A Continuous-Wave Raman Silicon Laser," *Nature*, vol. 433, no. 7027, pp. 725–728, Feb. 2005.

[18] *Standard for Local and Metropolitan Area Networks - Virtual Bridged Local Area Networks, Amendment 4: Provider Bridges*, Amendment to IEEE Std. 802.1Q-2005, IEEE Std. 802.1ad-2005, May 2006.

[19] J. Naous, G. Gibb, S. Bolouki, and N. McKeown, "NetFPGA: Reusable Router Architecture for Experimental Research," in *ACM Workshop on Programmable Routers for Extensible Services of TOmorrow (PRESTO)*, Aug. 2008.

[20] 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*, Aug. 2009.

[21] *Arista 7100S Series Data Center Switches*, Arista Networks. [Online]. Available: http://www.aristanetworks.com/en/7100_datasheet.pdf

[22] *Woven Systems Multi-Chassis EFX Ethernet Fabric Switches*, Woven Systems. [Online]. Available: http://www.wovensystems.com/pdfs/products/Woven_EFX_Series.pdf

[23] *Sun Datacenter Switch 3456*. [Online]. Available: http://www.sun.com/products/networking/datacenter/ds3456/datasheet.pdf

[24] *InfiniScale III*, Mellanox Technologies. [Online]. Available: http://www.mellanox.com/related-docs/prod_silicon/PB_InfiniScale_III.pdf