

# Large Data Center Fabrics Using Intel<sup>®</sup> Ethernet Switch Family

An Efficient Low-cost Solution

**White Paper**

---

*April, 2010*



## Legal

---

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT. Intel products are not intended for use in medical, life saving, life sustaining, critical control or safety systems, or in nuclear facility applications.

Intel may make changes to specifications and product descriptions at any time, without notice.

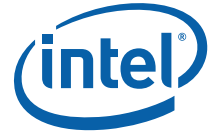
Intel Corporation may have patents or pending patent applications, trademarks, copyrights, or other intellectual property rights that relate to the presented subject matter. The furnishing of documents and other materials and information does not provide any license, express or implied, by estoppel or otherwise, to any such patents, trademarks, copyrights, or other intellectual property rights.

The Controller may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Intel and Intel logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

\*Other names and brands may be claimed as the property of others.

Copyright © 2011. Intel Corporation. All Rights Reserved.



## Table of Contents

---

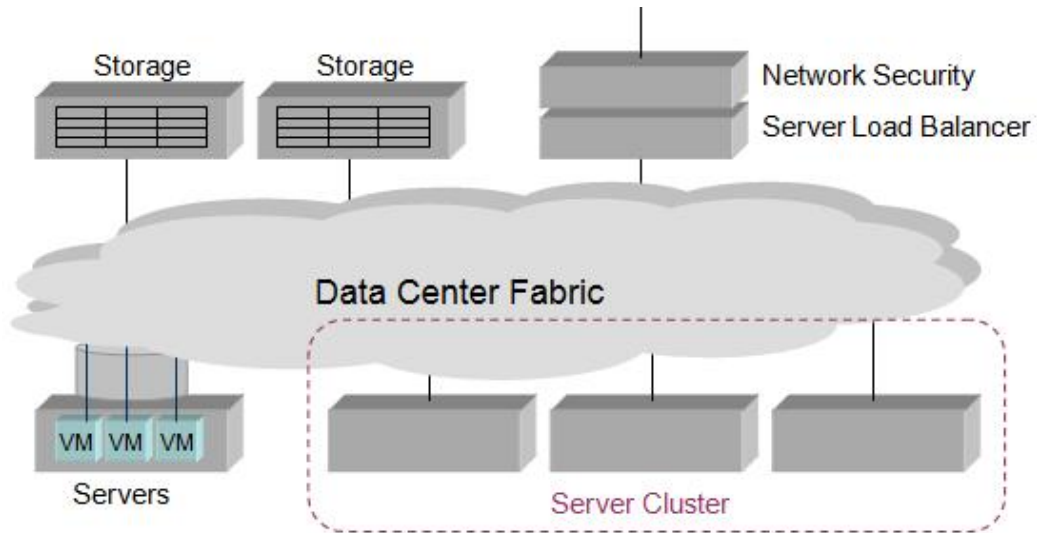
|   |           |
|---|-----------|
| <b>Overview .....</b>   | <b>4</b>  |
| <b>Background .....</b>   | <b>5</b>  |
| <b>Telecom-Style Fabrics in the Data Center .....</b>               | <b>6</b>  |
| <b>Fat Tree Configurations .....</b>                                | <b>7</b>  |
| <b>Ethernet Enterprise Switches in the Data Center .....</b>        | <b>8</b>  |
| <b>Intel® Ethernet Switch Family 10GbE Data Center Fabrics.....</b> | <b>10</b> |
| <b>Bandwidth Efficiency Analysis .....</b>                          | <b>12</b> |
| <b>Comparison with Telecom-Based Fabrics .....</b>                  | <b>13</b> |
| <b>Conclusions .....</b>  | <b>15</b> |

## Overview

The data center needs to evolve to a much more efficient state as it expands to serve the cloud. This is because the large data centers cannot tolerate any wasted cost, power or area as they compete to support cloud-based services. There are multiple industry initiatives under way to support the new efficient data center including:

- Server virtualization to optimize server utilization
- Data Center Bridging (DCB) to support converged data center fabrics
- FCoE to eliminate the need for additional FC fabrics
- TRILL to optimize data center fabric bandwidth utilization
- VEPA to support server virtualization through a single physical link

Another trend is to compartmentalize the data center architecture into atomic units called PODs, which look like shipping containers. Companies like HP and Sun are developing PODs as pre-wired and pre-configured data center building blocks. These are trucked into the data center and ready to run after connection to power, cooling and the network. An example POD block diagram is shown in [Figure 1](#).



**Figure 1.** Data Center POD Block Diagram

For many smaller applications, each sever can be configured as multiple virtual machines (VMs) which are connected to the fabric through a single physical port. In this case, protocols like VEPA will be used to create multiple logical connections to the fabric as shown. In some cases, cloud users will need to run large applications that require multiple servers in a cluster. Here, the fabric must support DCB features along with low latency for high performance.



The servers will also need to access storage through the fabric using protocols such as FCoE, which must support lossless operation and bounded latency using DCB features. The POD must connect to the outside world using multiple high-bandwidth connections. Assuming a homogenous data center, each POD will contain network security. In addition, applications with high user volume may require a server load balancing function.

Today, PODs are being developed that require several hundred fabric connections. Soon, this will scale to over a thousand fabric connections. Data center fabrics must meet these port counts while providing all of the features described above. In addition, they must do this in a very efficient manner, as every dollar, watt and square meter is critical when designing a POD.

## Background

In the late 1990's and early 2000's, proprietary switch fabrics were developed by multiple companies to serve the telecom market with features for lossless operation, guaranteed bandwidth and fine grained traffic management. During this same time, Ethernet fabrics were relegated to the LAN and enterprise, where latency was not important, and QoS meant adding more bandwidth or dropping packets during congestion. In addition, many research institutions developing High Performance Computing (HPC) systems chose InfiniBand (IB), which was the only choice for a low latency fabric interconnect solution.

Time has dramatically changed this landscape. Over the last 3 years, 10Gb Ethernet switches have emerged with congestion management and quality of service features that rival the proprietary telecom fabrics. As evidence of this role reversal, of the 30 or so proprietary telecom fabrics available around the year 2000, only one has survived. Even so, some companies are pushing telecom-style fabrics into the data center, which will be discussed in the next section.

With the emergence of the Intel® Ethernet Switch Family 10GbE switches, IB no longer has a monopoly on low latency fabrics. Many HPC designs are moving to this new cost effective Ethernet solution, pushing IB further into niche applications. Because of this, the two surviving IB switch vendors are even adding Ethernet ports to their multi-chip solution.

The industry needs a cost effective fabric solution for the data center that can scale to POD size requirements. The obvious choice is an Ethernet fabric, with converged features for clustering and storage. This paper will show that adding Ethernet ports to a telecom-style fabric dramatically increases cost size and power compared to an Ethernet switch based solution that has been designed for the data center.

## Telecom-Style Fabrics in the Data Center

A telecom-style switch fabric typically contains a Fabric Interface Chip (FIC) on each line card, which connects to one or several central switch devices. To provide the fine bandwidth granularity required for legacy protocols such as ATM or SONET, the FIC segments incoming packets into fixed size cells for backplane transport, and then reassembles them on egress. Due to the input/output queued nature of this system, Virtual Output Queues (VoQs) must be maintained on ingress to avoid Head of Line (HOL) blocking. The FIC also contains traffic management functions, which hold packets in external memory until they can be segmented and scheduled through the switch.

Today's process technologies allow FIC designs that contain up to 8 10GbE ports on the line side with up to 12 proprietary 10G ports to the backplane. Backplane overspeed is required due to factors such as cell segmentation overhead and fail-over bandwidth margin. The switch can contain up to 64 10G proprietary links to the FICs. Cells can be striped across up to 12 switch chips, providing a maximum of 64 FICs or up to 512 10GbE ports.

Figure 2 shows how this fabric can be used for a top-of-rack switch in the data center. In this case, a mesh fabric cannot be used as it would require at least 24 10G backplane links on each FIC. As can be seen, this is not a cost effective solution for this application as these devices could be replaced with a single 10GbE switch chip.

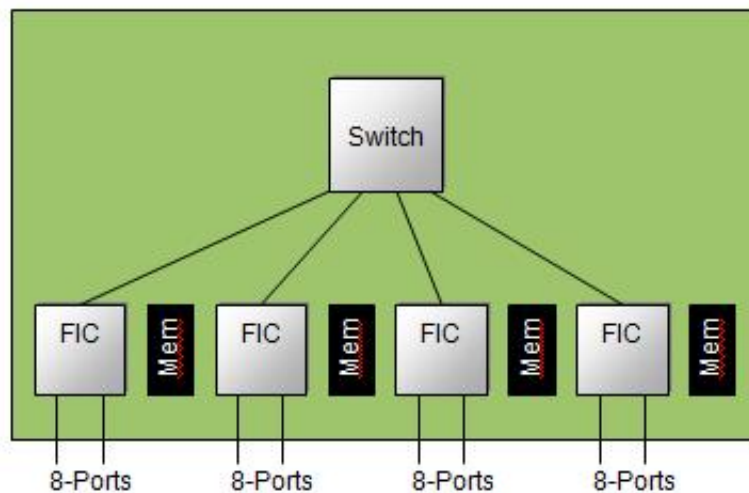
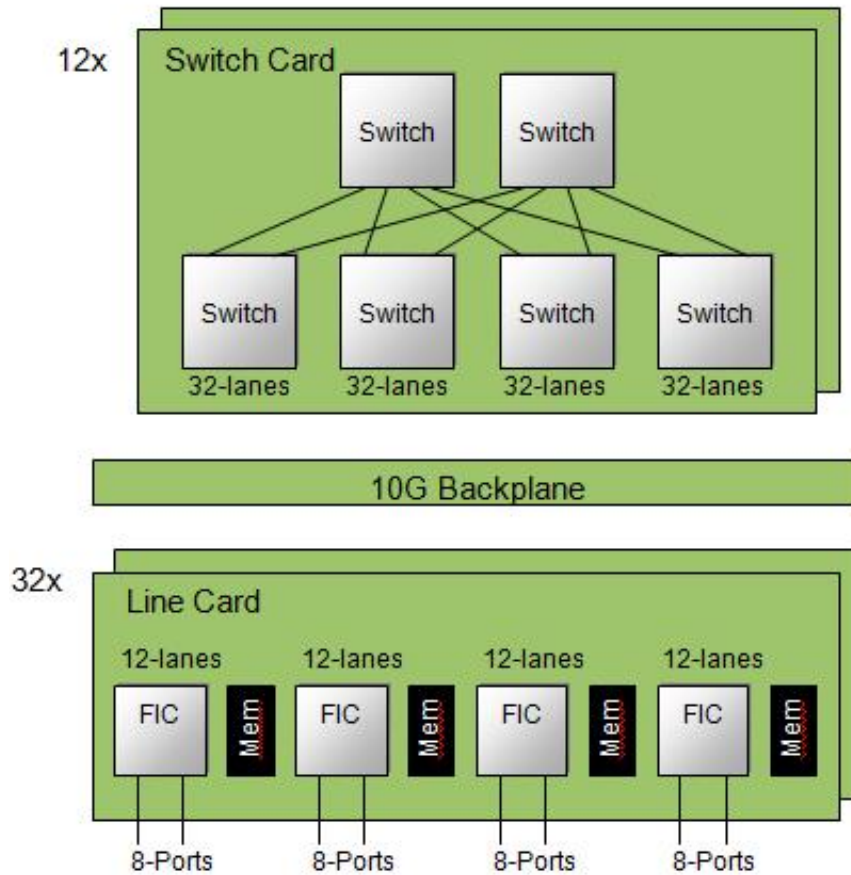
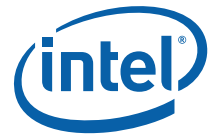


Figure 2. Top-of\_Rack Switch using Telecom-based Fabric

Figure 3 shows how this fabric must be configured to support up to 1024 10GbE ports, which is required for the next generation data center POD. To do this, a small fat tree must be created on each switch card to scale past the 512 port limit described above. This solution has significantly component count and high latency.

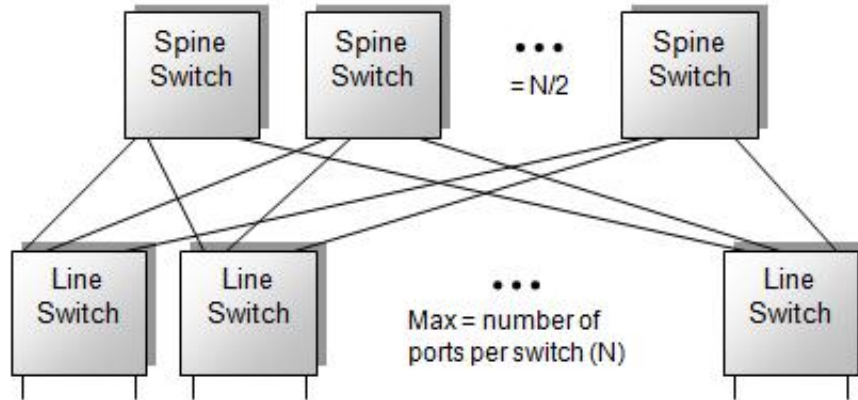


**Figure 3.** 1024-port Data Center POD Switch using Telecom-style Fabric

## Fat Tree Configurations

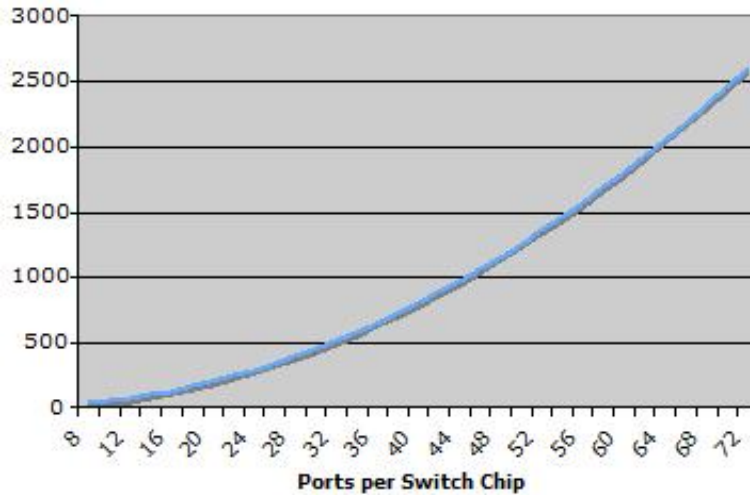
Fat trees can be used to efficiently scale 10GbE data center switch fabrics as shown in Figure 4. Figure 4 shows a 2-tier fat tree that provides bandwidth between stages equal to the total port bandwidth. Although this does not provide backplane overspeed as described in the telecom-style fabric, there are several factors to consider.

- There is no additional cell segmentation overhead like the telecom-style fabric
- Data center traffic is bursty in nature, but the fabric does not need to be fully provisioned like a telecom system. Lower overall average port utilization allows bandwidth margin to absorb bursts.
- Methods such as QCN and adaptive routing can be used to reduce congestion spreading, even under varying traffic loads.
- If QCN or adaptive routing are not used and overspeed is needed for some applications, a small percentage of ports on each line switch can be left unused. This is still a much more cost-effective approach than using a telecom-style fabric as will be shown later in this paper.



**Figure 4.** 2-Tier Fat Tree Structure

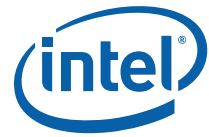
The total ports on a two-tier fat tree fabric scale as  $N^2/2$ , where  $N$  is the number of ports available on each switch chip in the tree. As [Figure 5](#) shows, this allows rapid scaling as the switch chip port count increases. As can be seen, it is very reasonable to achieve 2048 10GbE ports using only a 64-port switch chip. This gives plenty of margin to satisfy the port requirements of the next generation POD.



**Figure 5.** Total Fabric Ports Based on Switch Chip Size

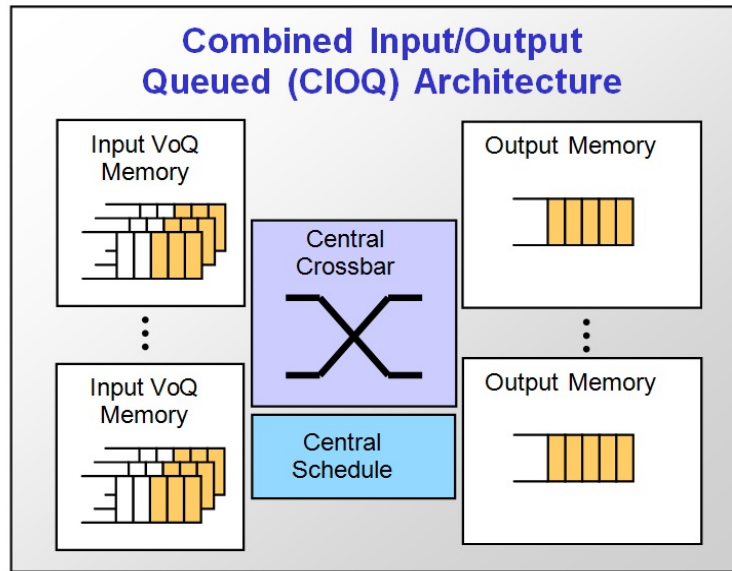
## Ethernet Enterprise Switches in the Data Center

Not just any 10GbE switch chip can function effectively in a 2-tier fat tree, as some are hindered by lack of features to support scalability. In addition, inefficient memory architectures can cause added internal switch congestion and limited multicast performance along with high



latency. Ethernet switches have been traditionally designed for the Enterprise and Telecom access markets where latency is not as important.

The overall port bandwidth of Ethernet enterprise switch chips has increased dramatically over the last few years, surpassing the ability of on-chip memory to support simultaneous access from all ports at full bandwidth. Because of this, these devices employ a Combined Input/Output Queued architecture as shown in Figure 6. This is in effect, a miniature version of the telecom architecture discussed earlier.



**Figure 6.** Traditional Enterprise Switch Chip Architecture

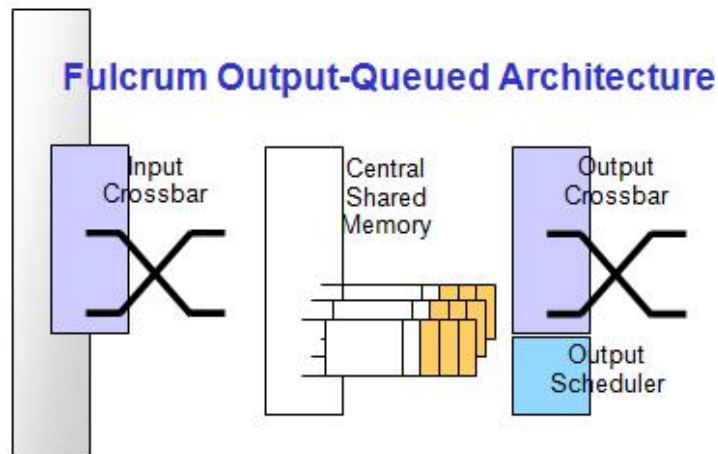
As with the telecom fabric, to avoid Head of Line blocking, VoQs are used at the ingress, adding to overall memory complexity. A sophisticated central scheduler is also required to maintain high fabric utilization. In addition, multicast packets must be stored multiple times in egress memory, adding to the overall on-chip memory requirements.

As can be deduced from Figure 6, in a 2-tier fat tree, the packet must be stored 6 times as it works it's way through the fabric. This not only adds to overall latency, but also adds many more points of congestion in the fabric compared to the Intel® Ethernet Switch Family shared memory switch architecture, which is described in the next section.

## Intel® Ethernet Switch Family 10GbE Data Center Fabrics

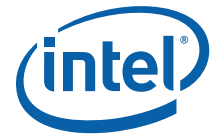
The Intel® Ethernet Switch Family provides a true output queued shared memory architecture. This is enabled by several Intel® patented technologies, which include the Nexus crossbar and the RapidArray memory. By providing full bandwidth access to every output queue from every input port, no blocking occurs within the switch, eliminating the need for complex VoQs and schedulers. The block diagram of the Intel® Ethernet Switch Family shared memory architecture is shown in Figure 7.

With the Intel® Ethernet Switch Family, all packets arriving at any ingress port are immediately queued at full line rate into shared memory. Packets are then scheduled from shared memory to the egress ports. Multicast packets are de-queued multiple times to each egress fan-out port. Each egress port has an independent scheduler design that is much simpler than a central scheduler. In addition, since the packet is queued only once, cut through latencies of a few hundred nanoseconds can be achieved independent of packet size.



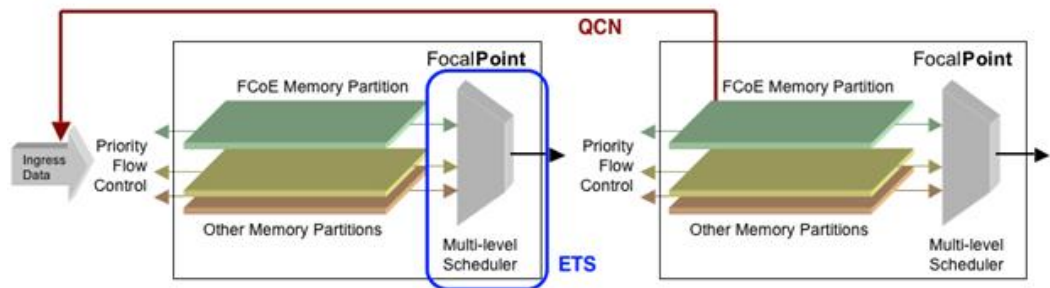
**Figure 7.** Intel® Ethernet Switch Family Output Queued Architecture

The Intel® Ethernet Switch Family has been designed from the ground up to support large fabric topologies including fat trees. Advanced hash functions are employed for load balancing flows across the spine switches in a very uniform manner. Global resource tags (Glorts) are used to identify virtual destination ports using ISL headers. In addition, the software API treats the entire fat tree as a large virtual fabric, easing application software development. All of this is on top of the extremely low cut-through latency of less than 1uS through a 2-tier fat tree.



Data center bridging is being specified by the IEEE to enable converged data center fabrics. The Intel® Ethernet Switch Family supports several DCB features including PFC, ETS, QCN and DCBx. Figure 8 shows how these features are implemented in a multi-stage Intel® Ethernet Switch Family fabric. As ingress data arrives in the Intel® Ethernet Switch Family fabric, an advanced TCAM-based classification engine can inspect the first 128 bytes of the frame and use ACL rules to assign a traffic class to each frame. Based on traffic class, frames can be placed into one of several logical shared memory partitions in the switch. For example, FCoE traffic can be placed into one memory partition, while data traffic can be placed into other memory partitions.

Each memory partition has an independent set of watermarks that can be used to generate link-level flow control back to the ingress link partner using IEEE Priority Flow Control frames. This insures that storage or HPC traffic will not be delayed or dropped if data traffic becomes congested in the switch.



**Figure 8.** Illustration of Intel® Ethernet Switch Family DCB Features

Each switch egress port contains a separate queue per traffic class along with a multilevel scheduler. The scheduler can be programmed to service a given traffic class as strict priority or give it minimum bandwidth guarantees using Deficit Round Robin (DRR). In addition, traffic classes, or groups of traffic classes can be shaped to limit maximum egress bandwidth. Using these mechanisms, storage or HPC traffic can be given a minimum bandwidth guarantee to support IEEE Enhanced Transmission Selection (ETS). By doing this, the maximum latency and latency jitter can be bounded for these flows through the fabric.

In multi-stage fabrics, congestion hot spots can occur which can cause congestion spreading. The Intel® Ethernet Switch Family has been designed to support the IEEE QCN standard. If an egress queue becomes congested (QCN congestion point), QCN frames are generated and sent back to the ingress traffic source (QCN reaction point) through the fabric, which in turn throttles back the offending flows. The traffic source receiving these frames can also use adaptive routing to direct traffic around the hot spots, although this is not defined in the QCN standard.



## Bandwidth Efficiency Analysis

The Intel® Ethernet Switch Family hash functions can be used to evenly distribute flows across multiple spine switches in a multi-stage fabric. The Intel® Ethernet Switch Family devices use a modified Pearson's hash that is highly effective in load distribution while incurring a modest implementation cost. Various parts of the L2/3/4 header can be used as a source for the hashing function. The source is hashed to a 12-bit value (giving 4096 intermediate bins) and the result is distributed among the output links using modulo division. For purposes of load distribution, two hash values can be calculated from the header fields in each frame:

- Layer 3/4 Hash (36 bits)
- Layer 2/3/4 Hash (48 bits)

The keys to these hash functions are constructed in a configurable manner in order to provide the following features:

- Symmetry — Hash value remains the same when source and destination fields are swapped.
- Static field dependence — Support for including a specific set of header fields in the hash function.
- Dynamic field dependence, based on frame type — Certain fields can be omitted or included when a frame is IPv4/IPv6.

The Intel® Ethernet Switch Family hash function was developed using mathematical analysis and was validated in silicon. [Table 1](#) shows the results of this analysis across 4 spine switches for 1000 flows based on the header fields shown. As can be seen, this provides a fairly uniform distribution, which will only improve when using more spine switches in the system.

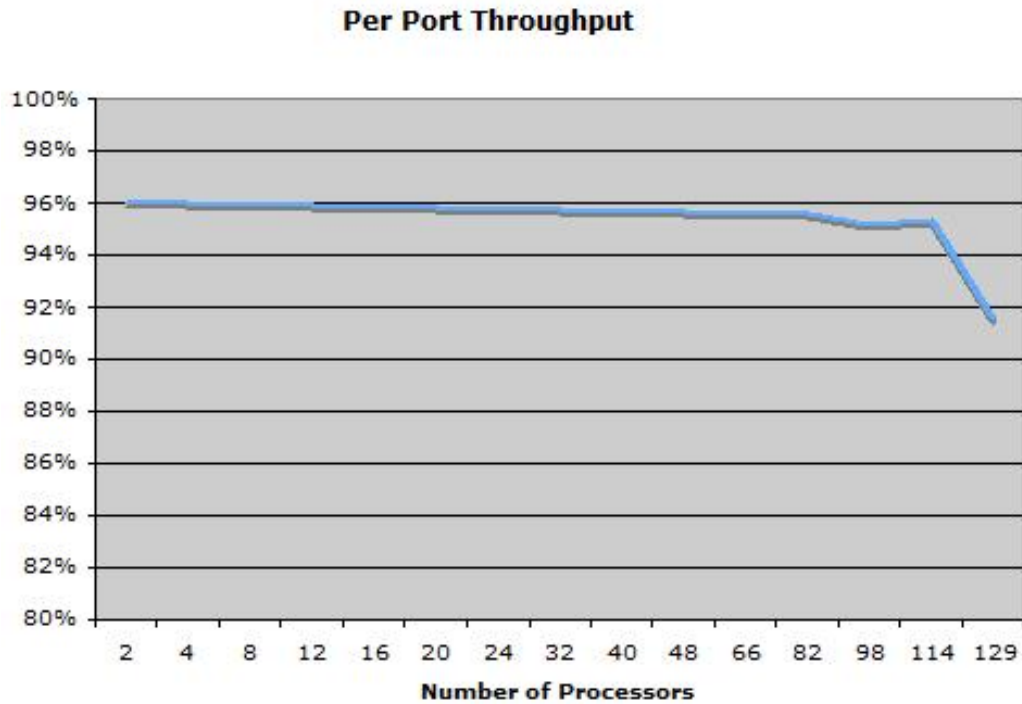
**Table 1. Intel® Ethernet Switch Family Hash Function Mathematical Analysis**

| Condition  | Link 0 | Link 1 | Link 2 | Link 3 |
|--|--------|--------|--------|--------|
| 1000 flows using random SMACs and random DMACs       | 25.1%  | 24.7%  | 23.7%  | 26.5%  |
| 1000 flows using random SIP/SMAC and random DIP/DMAC | 24.1%  | 26.9%  | 23.7%  | 25.3%  |
| 1000 flows using fixed SIP/SMAC and random DIP/DMAC  | 24.0%  | 24.6%  | 27.4%  | 24.0%  |
| 1000 flows using random SIP/SMAC                     | 23.1%  | 23.9%  | 26.0%  | 27.0%  |

Another method that can be used to maximize fat tree bandwidth utilization is adaptive routing. An example of this is the Fortinet two-tier fat tree switch called the FortisSwitch-1000, which uses the proprietary vScale adaptive routing technology. The [Figure 9](#) below



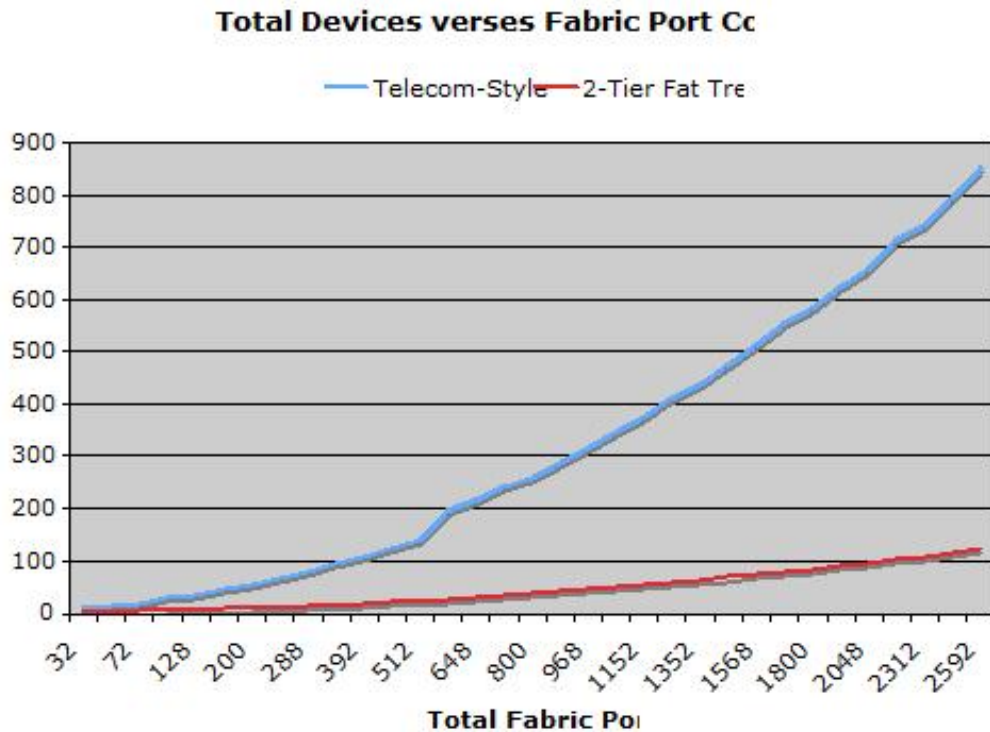
shows the per-port throughput of the switch verses number of ports connected, with up to 129 of the 144 10GbE ports fully utilized.



**Figure 9.** Fat Tree Performance Using Adaptive Routing

## Comparison with Telecom-Based Fabrics

Due to its nature, a telecom-style fabric consumes many more components than a fat tree fabric built with 10GbE switch chips. [Figure 10](#) shows the number of devices required verses 10GbE port count for a telecom-style fabric and a 2-tier fat tree fabric.

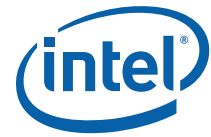


**Figure 10.** Number of Chips Required vs. Fabric Size

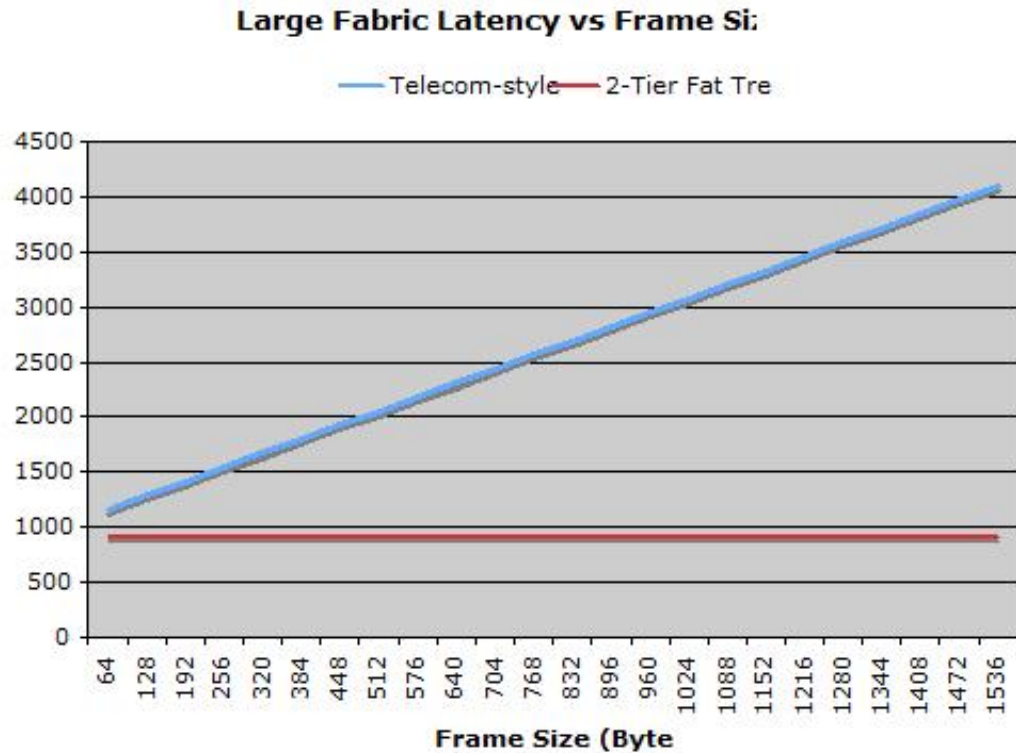
This analysis assumes that each FIC has 8 10GbE ports, 12 10G backplane ports and uses a single external buffer memory chip. It also assumes that the switch chip contains 64 10G backplane ports. For the 2-tier fat tree, it also assumes a 64-port 10GbE switch chip. The step up in the top line is where the telecom style fabric must start using a small fat tree on each switch card in order to scale past 512 ports.

As can be seen in [Figure 10](#), at the fabric sizes required for the next generation data center PODs, the telecom-style fabric needs about 7 times the number of components. Even if the fat-tree design is over-provisioned to account for non-ideal load distribution, it's a much simpler and lower cost solution.

Latency is also a concern for data center applications such as clustering. [Figure 11](#) shows a latency comparison between a telecom-style fabric and a 2-tier fat tree using Intel® Ethernet Switch Family silicon. For the telecom-style fabric, it's assumed that the packet must be queued both on ingress and egress in external memory to accommodate segmentation and re-assembly. It also assumes the backplane switches operate in cut-through mode with latencies of



200nS per stage and there is additional FIC pipeline delay of 200nS. As expected, the latency increases versus packet size for the store-and-forward telecom-style fabric.



**Figure 11.** Fabric Latency vs. Packet Size

## Conclusions

The data center is evolving to more efficiently utilization of system resources. One example of this is in the data center POD, which will soon require a converged DCB fabric supporting over 1000 10GbE ports. Although telecom-style fabrics can scale to high bandwidth, the complexity and component count lead to a very costly solution, which cannot be justified by small improvements in bandwidth efficiencies. Traditional 10GbE fabrics have not been designed to scale in the data center, and like the telecom-style fabrics, have high latency that impacts data center performance. The Intel® Ethernet Switch Family 10GbE switches have been designed for the data center and provide a rich set of features along with low latency and efficient scalability required for large cloud data centers.





**NOTE:**      *This page intentionally left blank.*