

The Lossless Network for Data Centers

IEEE-SA Industry Connections



Trademarks and Disclaimers

IEEE believes the information in this publication is accurate as of its publication date; such information is subject to change without notice. IEEE is not responsible for any inadvertent errors.

Copyright © 2018 IEEE. All rights reserved.

IEEE owns the copyright to this Work in all forms of media. Copyright in the content retrieved, displayed or output from this Work is owned by IEEE and is protected by the copyright laws of the United States and by international treaties. IEEE reserves all rights not expressly granted.

IEEE is providing the Work to you at no charge. However, the Work is not to be considered within the “Public Domain,” as IEEE is, and at all times shall remain the sole copyright holder in the Work.

Except as allowed by the copyright laws of the United States of America or applicable international treaties, you may not further copy, prepare, and/or distribute copies of the Work, nor significant portions of the Work, in any form, without prior written permission from IEEE.

Requests for permission to reprint the Work, in whole or in part, or requests for a license to reproduce and/or distribute the Work, in any form, must be submitted via email to stds-ipr@ieee.org, or in writing to:

IEEE-SA Licensing and Contracts
445 Hoes Lane
Piscataway, NJ 08855

Comments on this report are welcomed by Nendica: the IEEE 802 “Network Enhancements for the Next Decade” Industry Connections Activity: <<https://1.ieee802.org/802-nendica>>

Comment submission instructions are available at: <<https://1.ieee802.org/802-nendica/nendica-lldcn>>

Comments will be considered for a future revision of this document and may stimulate the development of a new revision.

*The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA*

*Copyright © 2018 by The Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published Month 2018. Printed in the United States of America.*

IEEE and 802 are registered trademarks in the U. S. Patent & Trademark Office, owned by The Institute of Electrical and Electronics Engineers, Incorporated.

PDF: ISBN 978-0-7381-xxxx-x STDVxxxxx
Print: ISBN 978-0-7381-xxxx-x STDPDVxxxxx

IEEE prohibits discrimination, harassment, and bullying. For more information, visit <http://www.ieee.org/web/aboutus/whatis/policies/p9-26.html>.

No part of this publication may be reproduced in any form, in an electronic retrieval system, or otherwise, without the prior written permission of the publisher.

*To order IEEE Press Publications, call 1-800-678-IEEE.
Find IEEE standards and standards-related product listings at: <http://standards.ieee.org>*

**Notice and Disclaimer of Liability
Concerning the Use of IEEE-SA Industry Connections Documents**

This IEEE Standards Association (“IEEE-SA”) Industry Connections publication (“Work”) is not a consensus standard document. Specifically, this document is NOT AN IEEE STANDARD. Information contained in this Work has been created by, or obtained from, sources believed to be reliable, and reviewed by members of the IEEE-SA Industry Connections activity that produced this Work. IEEE and the IEEE-SA Industry Connections activity members expressly disclaim all warranties (express, implied, and statutory) related to this Work, including, but not limited to, the warranties of: merchantability; fitness for a particular purpose; non-infringement; quality, accuracy, effectiveness, currency, or completeness of the Work or content within the Work. In addition, IEEE and the IEEE-SA Industry Connections activity members disclaim any and all conditions relating to: results; and workmanlike effort. This IEEE-SA Industry Connections document is supplied “AS IS” and “WITH ALL FAULTS.”

Although the IEEE-SA Industry Connections activity members who have created this Work believe that the information and guidance given in this Work serve as an enhancement to users, all persons must rely upon their own skill and judgment when making use of it. IN NO EVENT SHALL IEEE OR IEEE-SA INDUSTRY CONNECTIONS ACTIVITY MEMBERS BE LIABLE FOR ANY ERRORS OR OMISSIONS OR DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO: PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS WORK, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND REGARDLESS OF WHETHER SUCH DAMAGE WAS FORESEEABLE.

Further, information contained in this Work may be protected by intellectual property rights held by third parties or organizations, and the use of this information may require the user to negotiate with any such rights holders in order to legally acquire the rights to do so, and such rights holders may refuse to grant such rights. Attention is also called to the possibility that implementation of any or all of this Work may require use of subject matter covered by patent rights. By publication of this Work, no position is taken by the IEEE with respect to the existence or validity of any patent rights in connection therewith. The IEEE is not responsible for identifying patent rights for which a license may be required, or for conducting inquiries into the legal validity or scope of patents claims. Users are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. No commitment to grant licenses under patent rights on a reasonable or non-discriminatory basis has been sought or received from any rights holder. The policies and procedures under which this document was created can be viewed at <http://standards.ieee.org/about/sasb/iccom/>.

This Work is published with the understanding that IEEE and the IEEE-SA Industry Connections activity members are supplying information through this Work, not attempting to render engineering or other professional services. If such services are required, the assistance of an appropriate professional should be sought. IEEE is not responsible for the statements and opinions advanced in this Work.

CONTENTS

ABSTRACT	1
EDITOR	1
CONTRIBUTORS/SUPPORTERS	1
OUR DIGITAL LIVES ARE DRIVING INNOVATION	2
TRENDS IN THE DATA CENTER	2
OnLine Data Intensive (OLDI) Services	2
Figure 1 – Parallel Application Hierarchy	3
Deep Learning	3
Figure 2 – Deep Learning Training	4
Figure 3 – Parallelism Tradeoff.....	5
NVMe over Fabrics	5
Cloudification of the Central Office	6
Figure 4 – Transition to Cloudified Architecture in the CO	7
Parallelism	7
Figure 5 – The Problem with Network Congestion.....	8
WHY TODAY’S DATA CENTERS AREN’T KEEPING UP	8
Figure 6 – Clos Network	9
Figure 7 – ECMP Load Balancing.....	9
Figure 8 – Current Congestion Management.....	10
ECMP collisions	11
Figure 9 – ECMP Load Balancing Collisions	11
ECN control loop delays	12
PFC head-of-line blocking	12
Figure 10 – PFC Head-of-Line Blocking	12
Lossless configuration complexity	13
Incast congestion	13
TECHNOLOGIES FOR THE FUTURE	14
Virtual Input Queuing	14
Figure 11 – Switch Packet Loss	14
Dynamic Virtual Lanes	15
Figure 12 – Dynamic Virtual Lanes.....	16

Load-Aware Packet Spraying	16
Figure 13 – Load Balancing Design Space	17
Push and Pull Hybrid Scheduling.....	17
Figure 14 – Push and Pull Hybrid Scheduling	18
STANDARDIZATION CONSIDERATIONS	18
CONCLUSIONS.....	20
CITATIONS.....	20

The Lossless Network for Data Centers

Abstract

Modern data centers are tasked with delivering intelligent multi-media responses to real-time human interactions. Massive amounts of data are being churned and sifted by highly parallel applications, such as Online Data Intensive Services (OLDI) and Artificial Intelligence (AI), which historically required specialized High-Performance Computing (HPC) infrastructure. New advancements in high-speed distributed solid-state storage, coupled with remote direct memory access (RDMA) and new networking technologies to better manage congestion, are allowing these parallel environments to run atop more generalized next generation [Cloud-cloud](#) infrastructure. Generalized cloud infrastructure is also being deployed in the telecommunication operator's central office. The key to advancing cloud infrastructure to the next level is the elimination of loss in the network; not just packet loss, but throughput loss and latency loss. There simply should be no loss in the data center network. Congestion is the primary source of loss and in the network, congestion leads to dramatic performance degradation. This paper discusses the need for new technologies to combat loss in the data center network and introduces promising potential solutions.

Editor

Name	Affiliation
Paul Congdon	Tallac Networks

Contributors/Supporters

Name	Affiliation
John Bennett	Microsoft Azure
Jose Duato	Universitat Politecnica de Valencia
Barak Gafni	Mellanox
Feng Gao	Baidu
Liang Guo	CAICT
Russ Housley	Vigil Security, LLC
Jie Li	CAICT
Gu Rong	China Mobile
Richard Scheffenegger	NetApp
Mehmet Toy	Verizon
Sowmini Varadhan	Oracle
Jianglong Wang	China Telecom
Ilan Yerushalmi	Marvell
Yolanda Yu	Huawei

Our Digital Lives are Driving Innovation

For better or worse, our lives ~~are~~have been forever changed by digital technology. Digital technology is increasingly accessed and offered as a service from the cloud—~~and as a result it is becoming more integrated with our natural lives.~~ ~~Our lives and digital technology are coming together as cloud services become more a part of our natural lives.~~

Interacting with cloud services is now done in a human and natural way—through voice commands and visual recognition. ~~Someday, i~~n the not too distant future, as predicted by ~~Futurist-futurist~~ Ray Kurzweil [1], the way we think will be augmented by the cloud. Already ~~today~~, services are personalized to our individual tastes by online data-intensive cloud services. We've come to expect instantaneous access to massive amounts of digital content by our very own voice commands. But how does all this work—~~in the backend~~—~~in the data center~~? How is it that ~~massive-huge amounts-quantities~~ of data can be rendered into useful information within a timeframe that meets real-time human interaction delays?

The requirement to integrate digital technology into our ~~natural-daily~~ lives is ~~driving-inspiring~~ innovation in the data center. This innovation is driving the need for new levels of performance, scale, and reliability from the infrastructure. Enormous quantities of computing cycles are rendering massive amounts of data into real-time information and action. The delivery of information and action from the cloud data center needs to be fast! ~~As—a consequence~~Consequently, the fabric within the data center needs to eliminate loss and deliver low latency and high throughput.

Trends in the Data Center

Application and storage architectures within the data center are continuously evolving to address increasing demands for real-time, interactive digital technology. Currently, four critical data center use cases are stressing ~~today's-the~~ data center network. These include large scale Online Data Intensive (OLDI) services such as automated recommendation systems for online shopping, social media and web search; High performance Deep Learning networks; modern telecommunication central office networks; and ~~high-high~~-speed distributed pools of Non-Volatile Memory Express (NVMe) storage.

Online Data Intensive (OLDI) Services

The fundamental difference between Online Data Intensive services and their offline counterparts (such as MapReduce computations) is that they require immediate answers to requests that are coming in at a high rate. Latency control is a key concern. The end-user experience is highly dependent upon the system responsiveness, and even moderate delays of less than a second can have a measurable impact on individual queries and their associated advertising revenue. A large chunk of unavoidable delay, due to the speed of light, is inherently built into a system that uses the remote cloud as the source of decision and information. This puts even more pressure on the deadlines within the data center itself. To address these latency concerns, OLDI services deploy individual requests across thousands of servers simultaneously.

The responses from these servers are coordinated and aggregated to form the best recommendations or answers. Delays in obtaining these answers are compounded by delayed or ‘straggler’ communication flows between the servers. This creates a long tail latency distribution in the data center for highly parallel applications. To combat tail latency, servers are often arranged in a hierarchy, as shown in Figure 1, with strict deadlines given to each tier to produce an answer. If valuable data arrives late because of latency in the network, the data is simply discarded, and a sub-optimal answer may be returned. Studies have shown that the network becomes a significant component of overall data center latency when congestion occurs in the network [2].

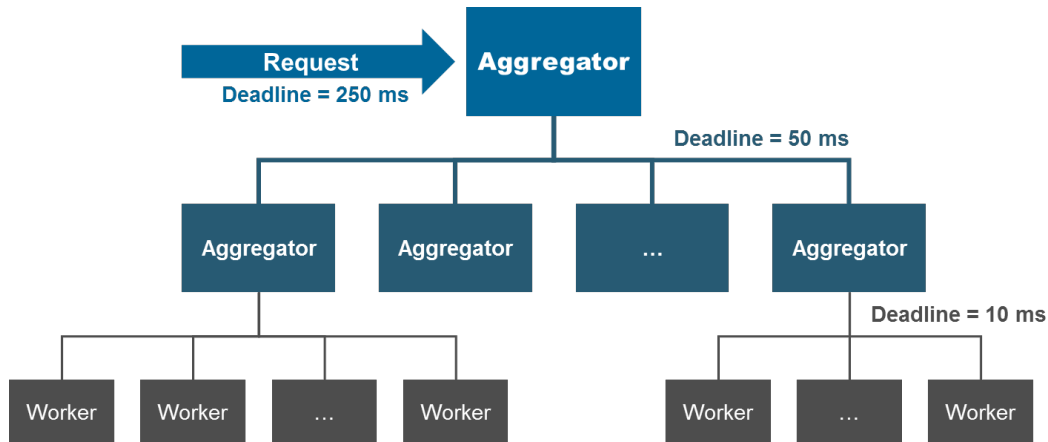


Figure 1 – Parallel Application Hierarchy

The long tail of latency distribution in OLDI data centers can be caused by various factors [3]. One is simply related to the mix of traffic between control messages (mice) and data messages (elephants). While most of the flows in the data center are mice, most of the bytes transferred across the network are due to elephants. [SoTherefore](#), a small number of elephant flows can delay the set-up of control channels established by mice flows. Since OLDI data centers are processing requests over thousands of servers simultaneously, the mix and interplay of mice and elephant flows is highly uncoordinated. An additional complexity is that flows can change behavior over time; what was once an elephant can transform into a mouse after an application has reached steady state. Another cause of latency is due to incast at the lower tiers of the node hierarchy. Leaf worker nodes return their answers to a common parent in the tree at nearly the same time. This can cause buffer over-runs and packet loss within an individual switch. It may invoke congestion management schemes such as flow-control or congestion notification, which have little effect on mice flows and tail latency, as discussed [belownext](#).

Deep Learning

Deep Learning is a branch of Machine Learning that is having tremendous success at allowing computers, applications, and cloud-based services to “see” and “hear”. Everyday human tasks such as speech recognition and image recognition are being mastered by large neural networks, trained with millions and sometime billions of parameters, forming models that can be integrated into an online service. Complex tasks such as social network filtering, fraud detection, and anomaly detection are performed effortlessly once these models are formed. Think of the

deep learning network as equivalent to a brain with its millions of neural interconnections. The larger the deep learning network, built from a larger number of model parameters, the better the network can perform at its job. Current deep learning networks can have billions of parameters and millions of interconnections [4].

Building the neural networks and deep learning models, a process called training, is often accomplished by high-performance computing systems. These systems can include large interconnected pools of virtualized GPUs that are remotely accessed by applications to accelerate computation. Remote GPU virtualization frameworks, such as rCUDA [5], are specified to run in low-latency network-intensive HPC cluster environments. They can significantly reduce execution time with respect to traditional local GPU accelerators. Additionally, training is a highly parallel application that requires low latency and high throughput. Throwing more computing resources at the problem can improve the time it takes to create a model; however, the communication overhead involved in the parallel application can offset the gains of more CPUs or GPUs. As seen in Figure 2, the huge training data sets are partitioned into chunks and distributed across a number of working clusters. Each cluster processes separate chunks of data and returns gradient results to be folded together by a common parameter server or other peers in a coordinated fashion. The process repeats with model parameters being refined, reduced, and redistributed until the model can recognize a known input with an acceptable level of accuracy. Once the models are built, they can be distributed and used as part of a new type of OLDI service that takes complex input such as voice, handwriting, high-resolution images, and video.

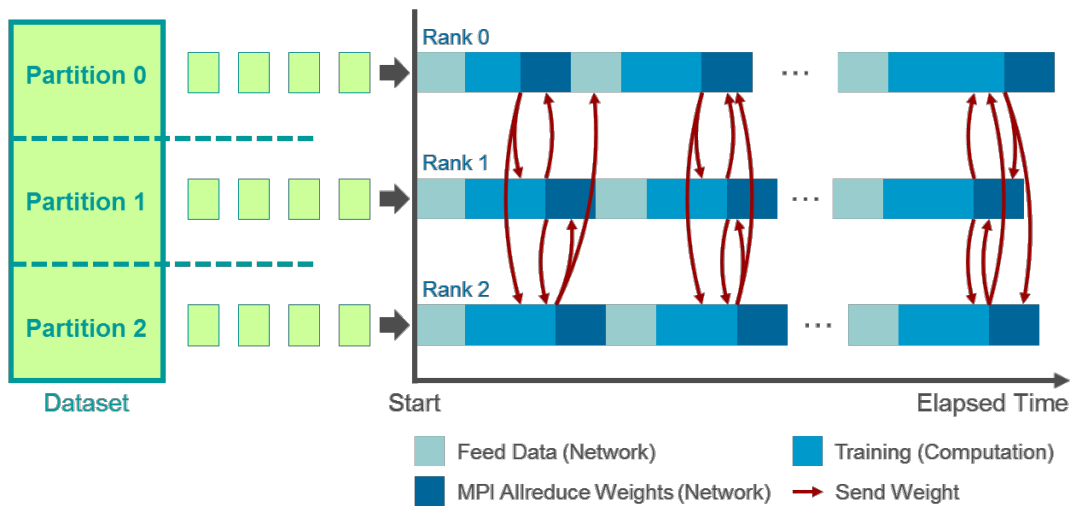


Figure 2 – Deep Learning Training

Deep learning models are constantly being trained and tuned. The challenge with this ongoing process is the high communication cost. Large amounts of data are frequently being shared and computation processes are stalled if synchronization delays occur. The network is often blamed for causing these training delays [6]. When a parameter server is used in the training process an inherent incast problem exists in the network. Clusters of worker nodes return gradient results to the parameter server at nearly the same time. This incast scenario creates congestion at the

switch connecting the parameter server and can result in packet loss and synchronization delays. Further parallelizing the problem only compounds the delay as more communication is required between a larger number of nodes, which multiplies the impact of network congestion. Figure 3 shows that there is an optimal tradeoff between the number of parallel nodes and the time it takes to train a model. Reducing packet loss and improving latency and throughput can allow a larger number of parallel nodes to train the model, thus reducing the overall time.

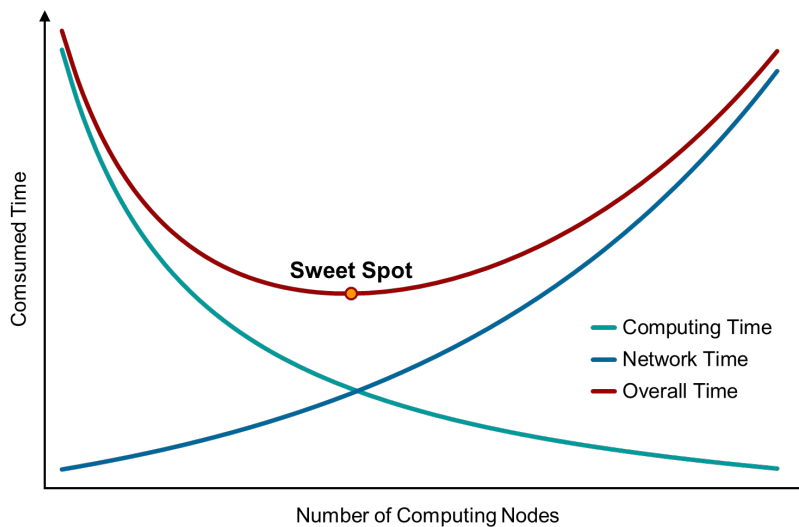


Figure 3 – Parallelism Tradeoff

NVMe over Fabrics

Non-Volatile Memory Express (NVMe) over Fabrics is a storage communications interface and protocol that was designed from conception to capitalize on the low latency and internal parallelism of flash-based storage devices known as solid-state drives (SSDs). NVMe is fast, reliable, and a perfect fit for the highly parallel environments of the future cloud data center. All-Flash-Arrays (AFAs) need NVMe access over the network. They need extremely low latency in order to compete with their on-board counterparts within servers. This latency needs to be on the order of 10 μ s [7] [8]. In the future, NVMe interfaces will only get faster and access latencies will continue to drop.

Cloud data centers are built on converged multi-tenant infrastructure where resources are pooled for lower cost, better manageability, and higher utilization. This means high-speed NVMe storage needs to be accessed on the same infrastructure as virtualized computing and application nodes. However, the latency and reliability requirements of NVMe storage make this access a challenge. To reduce latency, special host adapters utilize remote direct memory access (RDMA) communication semantics. RDMA supports zero-copy networking by allowing the network adapter to transfer data directly to or from remote application memory, bypassing the operating system. This is useful in message passing, and cluster synchronization, as well as storage communication. While extremely fast, bypassing the operating system means the network protocols responsible for reliable transmission and congestion control need to be

implemented in hardware on the adapter. Resources on the adapter can be quite restricted and in order to keep cost and complexity low, some of the support for reliability and congestion control can be passed to the network.

First generation converged infrastructure focused on providing a ~~large-large~~ scale lossless layer 2 fabric to support reliable message passing, Fiber Channel over Ethernet (FCoE), and RDMA over Converged Ethernet (RoCEv1). These layer 2 networks needed to provide a lossless transport because the storage protocols themselves were not tolerant of packet loss and did not provide an adequate congestion control approach. The layer 2 networks implemented priority-based flow control (PFC) and quantized congestion notification (QCN) to support a lossless environment for this first generation of converged infrastructure. These layer 2 congestion management approaches were not explicitly designed to support multi-tenancy. Current cloud data centers are based on multi-tenant layer 3 technology and storage protocols are running over TCP and UDP. The storage protocols over TCP and UDP take advantage of end-to-end congestion control to mitigate congestion, but without the additional support of layer 2 lossless protocols as a last resort, packet loss can still be a problem.

In the converged infrastructure data center, NVMe over Fabrics are specified to run over RoCEv2 (UDP-based) or iWARP (TCP-based). If the network detects congestion, it has the opportunity to mark packets with explicit congestion notification (ECN) indicators. The receiver will signal congestion notification messages back to the sender so that it can reduce the rate of injection in hopes of avoiding packet loss. If the round-trip time for these messages is too long, packet loss may still be unavoidable. Packet loss will require retransmission, which will severely slow down NVMe storage access.

Cloudification of the Central Office

The telecom industry continues to invest in additional infrastructure for the Central Office (CO) to handle the massive growth in mobile and Internet traffic in recent years. The traditional architecture of the CO network involved various dedicated purpose-built devices, optimized for specific functions and burdened with long development lifecycles. While these devices can meet the performance and availability requirements of the telecom industry, they lack the flexibility, openness, and physical characteristics that allow telecommunication companies to scale and adapt quickly to changing requirements. High throughput requirements are being driven by high-definition video, virtual, and augmented reality applications. In addition, low-latency requirements and computational processing needs continue to increase in order to handle complex operations such as interference mitigation of mobile subscribers and security analysis of network traffic. These growth and flexibility requirements are driving the telecommunications companies to consider a new architecture for the CO [9], [10].

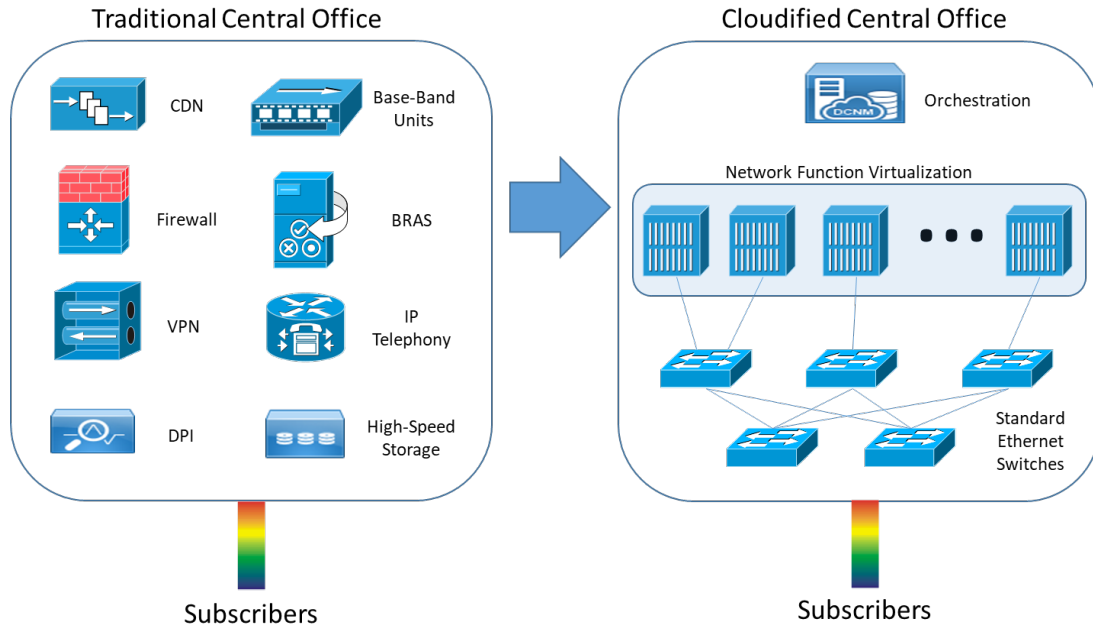


Figure 4 – Transition to Cloudified Architecture in the CO

Figure 4 shows how the CO is transforming from traditional, proprietary, dedicated hardware functions to a “cloudified” cloudified CO that relies upon industry standard servers, virtualization, and industry standard Ethernet switches. Operators are learning from IT vendors to build clusters of virtualized servers, which can be provisioned as needed to address adaptive demand. The virtual servers are running software-based telecommunication functions and must still meet the same low latency, high performance, and high availability requirements of their traditional dedicated hardware brethren. The cloudified central office is challenged to meet these objectives when the network fabric improperly addresses congestion and packet loss.

Parallelism

One attribute that all of the above use cases have in common is parallelism. In order for large-scale cloud services to meet real-time interactive latency requirements, the applications and storage must divide and conquer. There is simply too much data to process, and the true value of data is how quickly it can be rendered into human consumable information and action. As Figure 5 suggests, parallelism in a distributed system depends upon an enormous amount of messaging for synchronization and parameter distribution. Inherent in this messaging are traffic patterns that create congestion due to incast and disorderly flows. Left unattended, congestion leads to overall loss in the network: packet loss, latency loss, and throughput loss. These network issues exacerbate application issues and result in bad user experiences. Successful data centers of the future must minimize these issues.

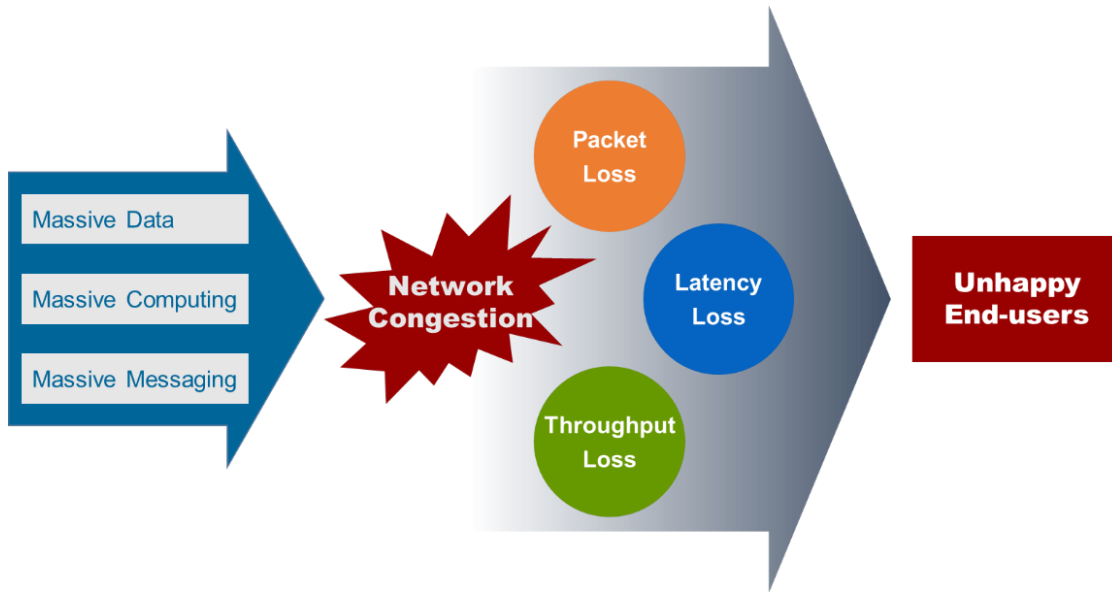


Figure 5 – The Problem with Network Congestion

Why Today's Data Centers Aren't Keeping Up

Whether building a public cloud or a private data center that operates as an internal cloud service for enterprises, a common set of problems need to be addressed. Network designers need to build a highly flexible fabric for rapidly changing environments that carry a diverse set of traffic: application, storage, and control. A common goal is to minimize or eliminate packet loss, provide high throughput, and maintain low latency. These goals are especially important to support the applications of OLDI, Deep Learning, NVMe over Fabrics, and the Cloudified CO.

Today's data centers typically use fat-tree architectures [11] derived from the basic Clos network illustrated in Figure 6. The Clos network achieves non-blocking performance and resiliency through equal cost multi-paths. Layer 3 networking is typically used between the switches because it is scalable, simple, standard, and well understood. In the Clos network, the top of rack (ToR) switches are the leaf switches. They are attached to the core switches, which represent the spine. The leaf switches are not connected to each other and the spine switches only connect to the leaf switches.

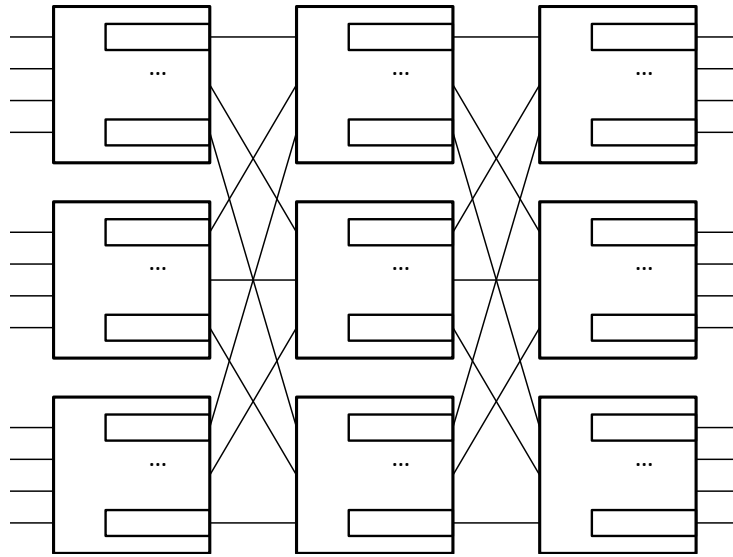


Figure 6 – Clos Network

There are multiple equal cost paths from each ToR switch to any other ToR switch in the network. ~~As a consequence~~Therefore, a ToR switch can spread traffic across the multiple paths in order to balance the load and hopefully avoid congestion. A popular algorithm used for distributing traffic is called Equal Cost Multi-Path (ECMP) routing. As shown in Figure 7, ECMP typically selects a path by hashing the flow identity fields in the routed packet such that all packets from a particular flow traverse the same path.

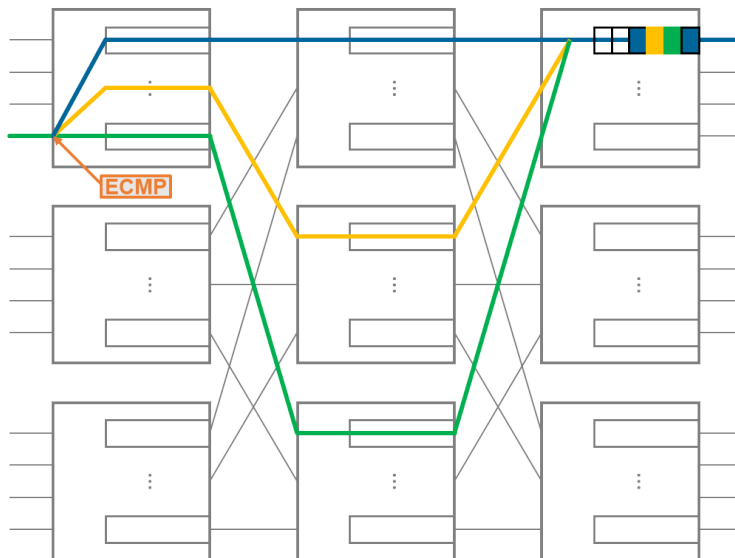


Figure 7 – ECMP Load Balancing

Server-to-server flows in the layer 3 data center are TCP or UDP connections across the fabric. Though the network is statistically non-blocking, congestion can still occur, and packets may be dropped, and the switches may mark IP packets with Explicit Congestion Notification (ECN)

indicators. ECN allows end-to-end notification of congestion before dropping packets. Figure 8 shows how the congestion feedback is returned to the sender via acknowledgement or specific congestion messages so the sender may reduce its rate of traffic injection into the network. The way a sender adjusts its sending rate depends upon the protocols in use. Slight modifications to TCP for data center use are being proposed by the IETF's DCTCP specification [12]. These modifications show that a “one size fits all” approach to TCP parameter tuning is not appropriate for the data center. The traditional TCP transmit window size has grown over the years to adapt to the long latencies and narrow bandwidth of the WAN and Internet. In the data center, bandwidth is high and latency is low, but switch buffers are small and traffic patterns can be unpredictable. These unique data center characteristics are spawning research on additional modifications to TCP as well as new transports specific to the environment [13]. Applications running over UDP are responsible for their own congestion control algorithms and most are using approaches that also recognize ECN indicators. RoCEv2, for example, runs over UDP and adjusts sending rate when it receives explicit Congestion Notification Packet (CNP) from the receiver.

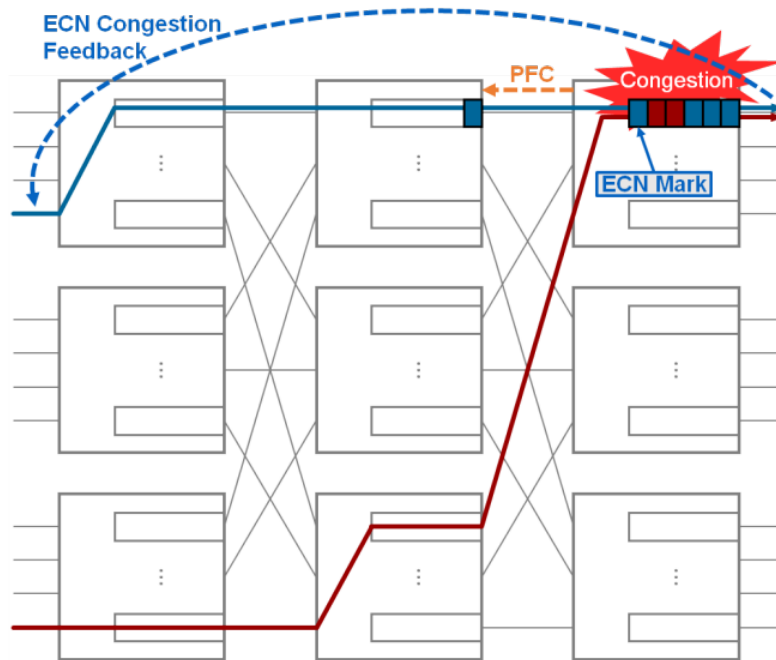


Figure 8 – Current Congestion Management

End-to-end congestion control is effective at getting the sending nodes to reduce their sending rates, but it does not completely eliminate the possibility of packet loss due to congestion. It takes some time for the ECN congestion feedback to make its way back to the source, and for the rate reduction to have an impact. Data that is already in flight can result in buffer overrun in the switches along the path. To avoid packet loss, which can have a dramatic effect on protocols such as FCoE and RoCE, the IEEE 802.1 Working Group developed a standard backpressure message called Priority-based Flow Control (PFC) [14]. A PFC message sent by the downstream switch signals to the immediate upstream switch to pause the sending of packets on a particular priority-/traffic class in order to avoid buffer overrun. To avoid packet loss, the downstream switch needs to [ensure](#) it has enough buffer headroom remaining to absorb the packets

inflight on the link before issuing PFC. While pausing, if the upstream switch buffers fill, it may issue its own PFC message to the next upstream switch, and so on, until eventually the sending node is paused. Typically, these congestion hotspots are temporary, and PFC never has to propagate very far back, but PFC itself is a heavy hammer and has other negative implications—[2](#), which will be discussed later.

The technologies used in today’s state-of-the-art data center are all designed for congestion management. While they have made improvements, they still fall short of providing the lossless data center network required for future use cases. In particular, [2](#) the following issues remain:

ECMP collisions

Selecting a path by hashing the flow identifiers is simple but does not take into consideration whether the selected path itself is congested. It is quite easy for the identifiers of multiple flows to hash to the same selection, resulting in overloaded links, as seen in Figure 9. Additionally, at any point in time, the nature of a flow can be seen to be bi-modal (mice or elephants), with the majority of flows being considered mice, but the majority of bytes transferred being from elephants. ECMP does not consider the nature of a flow when selecting a path. It is unfortunate when ECMP collisions occur on elephant flows because the chance of creating in-network congestion is much greater. Furthermore, ECMP is not effective if the traffic pattern involves incast. The problem of incast congestion, discussed later, occurs when there is a many-to-one traffic pattern creating congestion at the last hop switch to the destination. Load balancing in the core of the network cannot relieve incast congestion. Improvements to ECMP could involve being more congestion, traffic pattern, [2](#) and topology aware when selecting a path and load balancing traffic at a finer granularity.

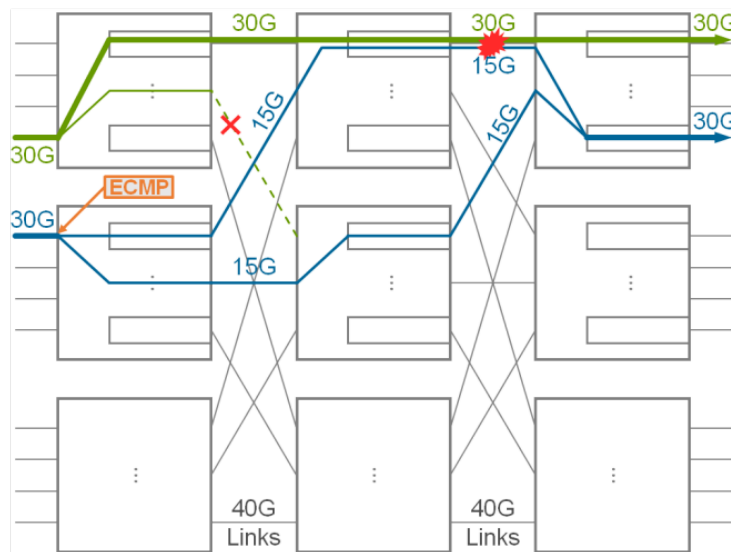


Figure 9 – ECMP Load Balancing Collisions

ECN control loop delays

There is a desire to scale data center networks larger, with more switches and more layers in the Clos network. The goal is to eliminate bottlenecks, simplify workload provisioning, and reduce costs. Large networks have more hops, and ~~as a consequence~~consequently, may lead to longer round-trip-time (RTT) for the ECN control loop. Larger networks can also support more data in-flight, making it difficult to absorb bursts of traffic before ECN congestion control can reduce the sending rate. Adding more switch buffers to absorb bursts is not desirable because it increases cost and increases network queuing delays for lighter flows. End-to-end congestion control is essential to low-loss networks, but additional assistance is needed to ~~assure~~ensure it can be effective and avoid packet loss.

PFC head-of-line blocking

PFC is a technique to avoid packet loss, but it is a heavy hammer and should be used as a last resort. PFC is invoked when switch ingress buffers back-up because of congestion at one of the egress ports. It is common for some of the flows arriving on the ingress port to be destined to other non-congested egress ports within the switch. However, because PFC will stop all traffic in a particular traffic class at the ingress port, the flows destined to other ports will also be blocked. The phenomenon is known as head-of-line blocking, and is illustrated in Figure 10. To avoid head-of-line blocking it is critical to identify the flows that are causing congestion as early as possible and provide congestion mitigation techniques that are specific to the flows. The flows that are causing congestion are most frequently elephant flows.

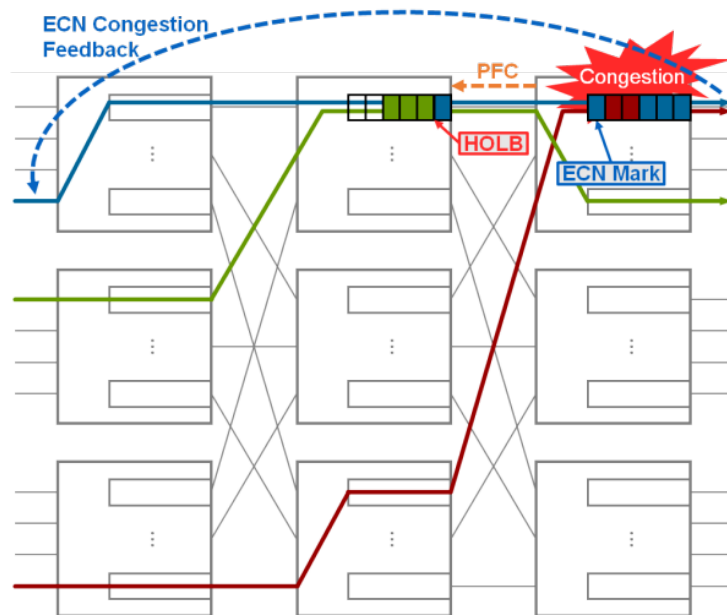


Figure 10 – PFC Head-of-Line Blocking

Head-of-line blocking can cause additional congestion upstream. Since PFC blocks all flows, even those destined to paths that are not currently congested, all flows must queue in the upstream switch. This queuing delay may in-turn create congestion in the next upstream switch. When

input buffers in the upstream switch fill, PFC messages are sent further back into the network, creating more head-of-line blocking and more congestion. This is known as congestion spreading.

Congestion spreading and head-of-line blocking can artificially invoke end-to-end congestion control on flows that are not otherwise a source of congestion. —When an upstream switch traffic class is paused because the downstream switch is experiencing congestion, all flows using the traffic class queue are held. If ECN marking is performed on packets in that queue based solely on the queue occupancy, it is likely that packets of the paused flows will be marked as the queue fills. These markings will eventually cause the sources of those flows to slow down, even if they were not headed into the downstream switch egress queue. In this scenario, isolated PFC head-of-line blocking can reduce the overall fabric-wide performance of the network.

Lossless configuration complexity

Creating a truly lossless network using today's state-of-the-art design with PFC requires precise tuning of configuration parameters. Effectively using PFC requires that differentiated application traffic has been properly allocated to appropriate traffic classes. —Quality of service (QoS) configuration typically occurs at the end-points or top-of-rack switches, and in either case must be consistently configured across the network. Once traffic is flowing on the appropriate traffic class, per-switch buffer tuning is required for PFC to ~~assure~~ ensure no packet drops occur. Reserved buffer “headroom” must be calculated from the product of link speed, link distance, and maximum packet size in order to absorb packets in flight once a per-priority pause frame has been issued by a downstream switch. Additional buffer configuration to manage dynamic shared pools of buffers across different traffic classes may be required on some switch architectures. Consideration of internal switch buffer “cell” size may also be required on many switch architectures [15]. When distance and link speeds are high, multiple small packets may be in flight after a pause frame has been issued and a switch architecture that takes advantage of large internal buffer “cell” sizes may inefficiently utilize the available buffer to absorb individual packets. Tuning QoS and buffer configuration by hand can be quite complex. Some vendors actually recommend a trial-and-error approach to configuration. Future technologies should consider protocols and methods to simplify lossless configuration.

Incast congestion

Incast is a naturally occurring phenomenon in highly parallelized cloud applications and has been shown to be responsible for the majority of packet loss in the data center [11]. Iterative divide and conquer strategies with periodic synchronization require a significant amount of many-to-one communication. Incast congestion occurs at the ToR switch where the node that multiple parties are synchronizing with is connected. Multiple inputs are simultaneously directed to a single output, creating an oversubscription scenario. This type of congestion is an attribute of the application design more than an issue with the network. However, the network can assist by eliminating packet loss both locally within the switch and across the fabric. High-performance multiprocessor systems had addressed the problem of incast congestion using worm-hole switching, synchronization, and packet aggregation along the path [16]. Current data center network equipment simply reacts to incast using a combination of ECN, PFC, and smart buffer management in an attempt to minimize packet loss.

Technologies for the Future

What if there was no loss in the data center network? None whatsoever! No packet loss, no latency loss, and no throughput loss. Networks could be designed to support full utilization without the risk of performance degradation. Parallel applications and datasets would not be delayed or blocked by congestion and could meet the real-time latency requirements needed to meld the intelligence in the cloud with live human interaction. With no information loss or delay, we would be able to create new and unique user experiences from unbounded remote information.

To create such an environment, we must mitigate congestion in the network—~~Not~~ not simply cope with it, like today's networking technologies, but mitigate the effects and create a lossless network. The following new and proposed technologies are aiming to do just that—~~—~~ progressing towards the lossless data center network for the future.

Virtual Input Queuing

The lossless network must begin within the switch itself. There are many different silicon and system architectures available to build a switch, but without coordination between the ingress and egress ports it is difficult to create a lossless environment. Figure 11 shows an example of how incast can create packet loss within a switch if there is no coordination between the ingress and egress ports. PFC is typically implemented on the ingress queues of a switch. When those queues back-up because the egress port is full, they will eventually trigger PFC to the upstream neighbor. However, in the incast scenario without ingress-egress coordination, it is possible that the egress queue will overflow before all the ingress queues have reached their threshold to generate PFC.

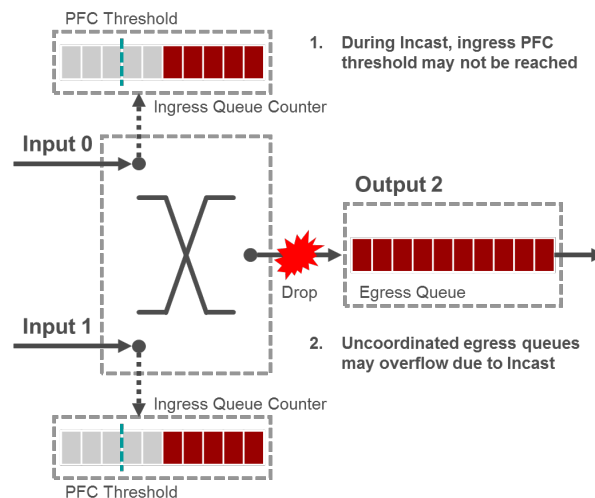


Figure 11 – Switch Packet Loss

Virtual Input Queuing (VIQ) is an example approach to coordinate the resources available on an egress port with the demands of an ingress port to deliver data. With VIQ, the egress port informs the ingress ports of its buffer availability to avoid internal switch transfers that will

result in packet loss. Packets can naturally back-up in the ingress port and PFC can be applied appropriately if needed. VIQ can be modeled as having a dedicated queue at egress for each ingress port and, ~~as a consequence~~consequently, fair share scheduling can be applied to traffic leaving the switch.

VIQ has the advantage of avoiding ~~congestion~~congestion-induced packet loss within the switch itself. In addition, VIQ modeling can allow traffic to exit the switch in a fair and orderly manner to help maintain the foundation of the lossless data center. There are other designs and internal scheduling algorithms, beyond VIQ, that can be implemented to achieve a lossless switching environment [17]. –The key is to support coordination between egress and ingress ports.

Dynamic Virtual Lanes

In today's layer 3 data center networks, traffic can be a mix of various multi-tenant TCP and UDP flows across both the physical underlay and virtual overlay network. Intermittent congestion within the network can be caused by the unfortunate mix of flows across the fabric. A small number of long duration elephant flows can align in such a way to create queuing delays for the larger number of short but critical mice flows. The queuing delays deter the end-to-end congestion control loop and cannot prevent PFC flow control from being invoked. When buffers fill and eventual flow-control kicks in, mice flows can be blocked by the unfortunate burst alignment of elephant flows. If PFC flow control is not being used, packet loss on short mice flows can result in full retransmission timeouts, significantly penalizing the latency of mice flows used for control and synchronization within the parallel application.

Dynamic Virtual Lanes (DVL) is an implementation of Congestion Isolation (CI) that eliminates head-of-line blocking caused by the over-use of PFC while supporting lossless behavior. Similar approaches have been deployed in slightly different context providing a base reference to the approach [18] [19] [20] [21] . DVL identifies the flows that are causing congestion, isolates them to a separate traffic class and potentially signals to the upstream neighbor to do the same. DVL effectively moves the congested flows out of the way, temporarily, while the end-to-end control loop has time to take effect.

Figure 12 shows the operation of DVL. When flows unfortunately collide at the egress port of a switch, congestion is detected, and the offending flows are identified. Subsequent packets from the offending flows are routed through a dedicated congested flow queue (i.e., they are effectively moved out of the way). Once the congested flow queue reaches a threshold, DVL signals to the upstream switch using a Congestion Isolation Packet (CIP) that contains enough information for the upstream switch to identify the same congested flow. The upstream switch also isolates the same flow and begins to monitor the depth of the congested flow queue. The packets in the congested flow queue are drained at a lower priority than other non-congested queues, so when congestion persists, the congested flow queue may fill. A switch implementing DVL may utilize Virtual Input Queuing (VIQ) to coordinate the congested flow queue with the ingress port. When the congested flow queue fills, the ingress port can issue PFC to avoid packet loss. Flow control is only blocking the congested flow queues and other well-behaved mice and elephant flows are free to traverse the fabric via non-congested queues.

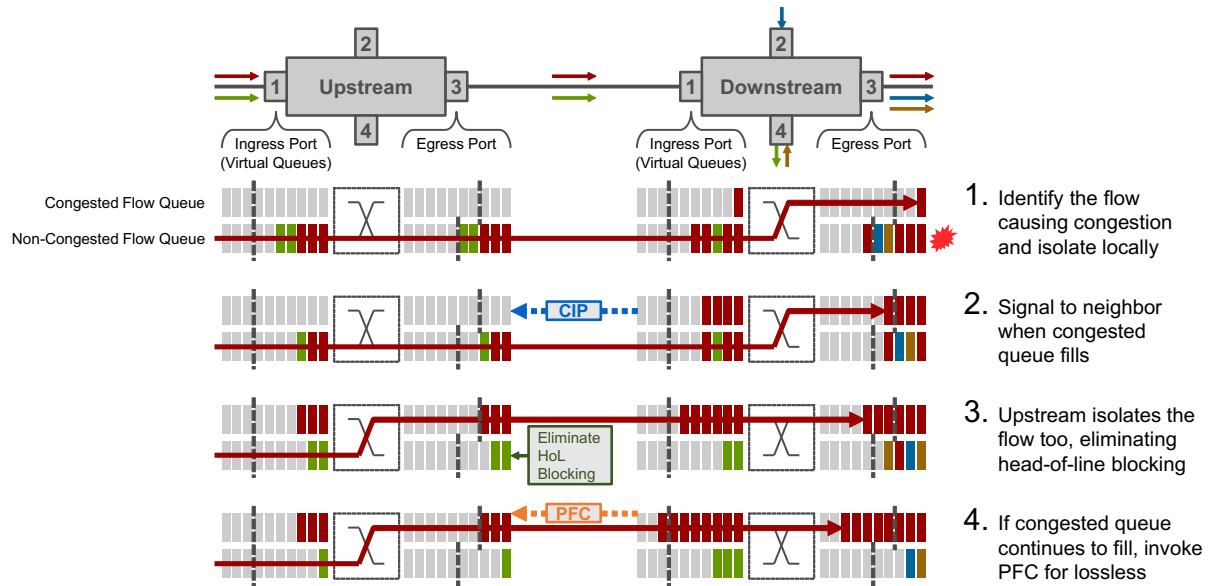


Figure 12 – Dynamic Virtual Lanes

The advantage of DVL is that latency can be reduced for critical control flows and packet loss can be eliminated without head-of-line blocking or congestion spreading. If PFC is needed, it typically is only needed on the congested flow queue. The offending flows will be delayed enough to allow end-to-end congestion control, such as ECN, to take effect. The temporary bursts of the offending flows are absorbed by the coordinated congested flow queues between peers in the fabric. Simulation results have shown that DVL significantly reduces flow completion times by eliminating head-of-line blocking and by dramatically reducing the use of PFC in the network.

Load-Aware Packet Spraying

Load balancing network traffic is a technique to avoid in-network congestion; however, ineffective approaches can actually do the opposite. Figure 13 shows the design space for load-balancing technologies. Centralized approaches have difficulty scaling and meeting real-time latency requirements. A global view of network congestion provides more information than local in-switch decisions and requires an efficient means of obtaining congestion notifications from the network. The granularity of load balancing has trade-offs between the uniformity of the distribution and complexity associated with assuring data is delivered in its original order. From this design space we discuss Load-Aware Packet Spraying (LPS)—a distributed framework with global state and packet level granularity. This congestion aware approach achieves fine grain load balancing without causing packets to be delivered out-of-order. Other selections from the design space have been implemented in proprietary ways, such as a distributed approach using local state and [flowlet-flowlet](#) granularity, but we focus the discussion on LPS to emphasize the challenges of obtaining global state and supporting the re-ordering of per-packet spraying.

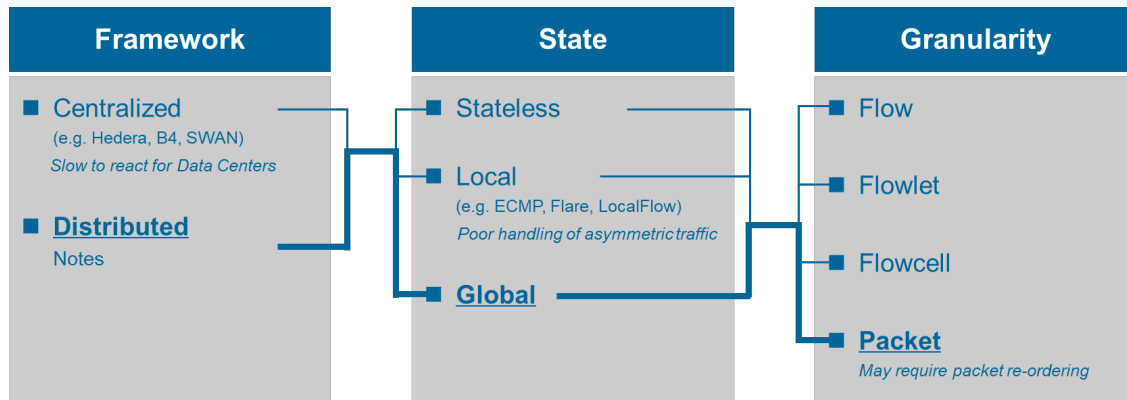


Figure 13 – Load Balancing Design Space

With LPS, packets between a source and destination are sprayed across multiple paths according to the degree of congestion measured on those paths. The spraying and distribution of packets across multiple paths may be initiated, coordinated or performed by transport protocols, virtual switches, NICs, ToR switches, or fabric switches. In layer 3 virtualized environments, flows between a source and destination can be identified by the virtualization encapsulation. LPS includes a sequence number in this encapsulation to allow the destination to reorder packets back into their original sequence. Since a destination may be receiving flows from many sources at the same time, there needs to be an active reordering queue for each source that is spreading traffic across the Clos network. The LPS source maintains an indicator of congestion along the path to other destinations. This indicator can be determined by any number of congestion measurement techniques. The source uses the congestion indicator to determine how to spray packets across the multiple paths—lighter loaded paths will take more packets than congested paths, which may be skipped entirely.

The advantages of LPS over current ECMP load balancing are **threefold**^[CKB1]. LPS avoids elephant flow collisions because it distributes traffic with fine granularity at the packet level. LPS can rapidly adapt to network status changes because it is congestion-aware. LPS is more parallel than ECMP and can reduce flow completion times in lightly loaded networks by distributing a single flow across multiple parallel paths at the same time. Finally, LPS reduces the probability of congestion occurring within the network, thus reducing the frequency of activation of both PFC and ECN-based end-to-end congestion control.

Push and Pull Hybrid Scheduling

While incast congestion is often an artifact of the parallel application design, the application, the transport protocol, and the network can assist in eliminating packet loss at the destination by scheduling traffic delivery when it would otherwise be lost. In the traditional approach, a source sends packets to a destination without knowing the resource availability and processing capacity at the destination. For example, **TCP-TCP**-based applications send windows of data into the network and attempt to assess the bandwidth and resource availability by measuring feedback through acks. This works well when feedback can be delivered because the network is lightly loaded, or congestion is intermittent and moderate. However, once incast congestion appears at the destination, delays increase and buffers overflow, throughput is lost and latency rises. The

traditional feedback ~~which-that~~ causes the TCP application to reduce its sending rate often cannot react quickly enough to handle incast. Requesting (aka pulling) data from the source at a rate that it can be consumed without loss is an alternative. Current research on new transport protocols are exploring such an option [13], [22]. In the pull scenario, the source issues a request to send or simply sends a small amount of unscheduled data, and the destination schedules a grant response when resources are available to receive the entire transfer. The pull approach incurs a request-grant RTT delay, but during incast, the transfers can be scheduled in such a way to avoid queuing delays and packet loss entirely.

The Push and Pull Hybrid (PPH) approach combines both approaches by monitoring the congestion between the source and destination. As seen in Figure 14, if the network load is light, the push approach is used to achieve the lowest possible latency. If the network load is high, the pull approach is used. The source and switches along the path measure congestion to the destination in order to decide which mode to use. Moreover, PPH can be combined with LPS for best results. When there is in-network congestion, LPS is used and the network works in push mode. When incast congestion arises, the network switches to pull mode and LPS is deactivated.

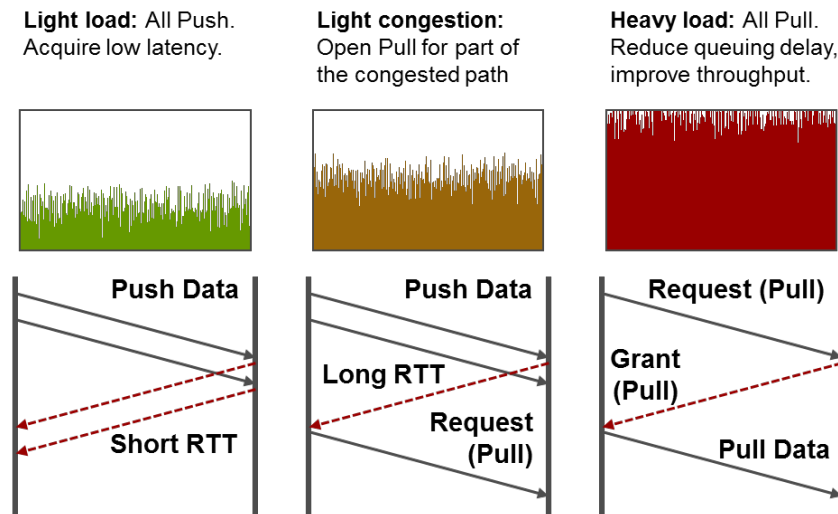


Figure 14 – Push and Pull Hybrid Scheduling

The advantage of PPH is that it can eliminate congestion and packet ~~lost~~^[CKB2] due to incast oversubscription. Current data center networks are unable to avoid packet loss caused by incast congestion without applying a ~~heavy hammer~~^[CKB3] (PFC) that ripples across the network, spreading congestion. With PPH, the traditional push approach is used when possible, but as soon as incast congestion exists, the traffic is scheduled to match the available resources. Thus, similarly to LPS, PPH also reduces the frequency of activation of both PFC and ECN-based end-to-end congestion control.

Standardization Considerations

Two important standards development organizations for the future technologies discussed ~~above previously~~ are the IEEE 802 LAN/MAN Standards Committee and the Internet Engineering Task Force (IETF).

The IEEE 802 LAN/MAN Standards Committee develops and maintains networking standards and recommended practices for local, metropolitan, and other area networks, using an open and accredited process, and advocates them on a global basis. The most relevant and widely used standards are for Ethernet, Bridging, and Virtual Bridged LANs. The IEEE 802.1 Working Group provides the focus for Bridging and Virtual Bridged LANs.

The Internet Engineering Task Force (IETF) is the premier Internet standards body, developing open standards through open processes. The IETF is a large open international community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet. The technical work of the IETF is done in Working-Working Groups, which are organized by topic into several Areas. The most relevant IETF Areas for the future technologies discussed above are likely the Internet Area (int), the Routing Area (rgt), and the Transport Area (tsv).

The IEEE 802 and IETF have a long history of working together on developing inter-related standards and technology. A standing coordination function between the Internet Architecture Board (IAB) of the IETF and the leadership of the IEEE 802 Working Groups is currently in place [23]. Traditionally these two organizations were aligned by layers of the ISO stack, where IEEE 802 focused on layer 2 and IETF on layer 3 and above. The lines have blurred over the years, but the two organizations have continued to work together, sharing information and developing unique and valuable standards.

Virtual Input Queuing is a switch architectural implementation detail. The IEEE 802.1 Working Group would be the most relevant standards organization to address the specification of this technology. However, the IEEE 802.1 Working Group strives to provide implementation flexibility and prefers to specify observable external behavior. When necessary, examples of how packet queuing and scheduling are discussed in Std-IEEE Std 802.1Q, so it could be conceivable that an amendment to this standard could be provided-developed to describe the desired lossless and non-blocking behavior.

Dynamic Virtual Lanes needs to specify how traffic flows causing congestion are identified and how packets of those flows are classified and queued within a switch. Additionally, a protocol to signal congestion to an upstream peer is required. The IEEE 802.1 Working Group would be the most relevant standards organization to address this technology. The Transport area of the IETF would have interest in understanding the interplay of DVL with their end-to-end congestion control protocols such as Explicit Congestion Notification (ECN). DVL marks congestion fields in the layer 3 headers of congested packets in order to cooperate with protocols from the IETF.

Load Aware Packet Spraying requires a combination of layer 2 and layer 3 technologies to function properly in a modern data center. The target data center design involves virtualization overlay networks on top of point-to-point layer 3 connections between switches. Sequence information must be carried end-to-end to support re-ordering and reassembly of flows. This information would likely be included in protocols being standardized by the IETF's Routing Area Network Virtualization Overlays (nvo3) Working Group. Load information used to determine how to spray packets across multiple paths comes from the internal queue status of IEEE 802.1 data center switches. Both the IEEE 802.1 Working Group and the IETF may need to cooperate on different portions of the technology to standardize LPS.

The Push and Pull Hybrid approach is like LPS in that both the IETF and IEEE might be involved to standardize the technology—PPH requires the ability to monitor congestion along the data center network path and switch between push or pull scheduling. The data center switches can assist in providing the status of congestion for scheduling decisions. The end-to-end protocols, however, must involve IETF layer 3 signaling between the top-of-rack switches. PPH will most likely require a deep cooperation between the IEEE and the IETF.

Conclusions

The demands on the data center network will be great. Highly parallelized applications and online services must deliver instantaneous response with very little delay. There is simply no time for loss in the network due to congestion. In this paper we have introduced Load-Aware Packet Spraying, Dynamic Virtual Lanes, Push and Pull Hybrid scheduling, and Virtual Input Queues. Each of these technologies is designed to mitigate congestion in the data center. Load-Aware Packet Spraying provides fine grain load balancing that is congestion aware to avoid the problem of large flow collisions due to simple ECMP load balancing. Dynamic Virtual Lanes reduces the use of PFC in the network and eliminates head-of-line-blocking by moving the flows that are creating congestion to a separate traffic class. Hybrid Push and Pull scheduling eliminates loss due to incast without sacrificing latency in a lightly loaded network. Packets are scheduled for delivery across the fabric with end-to-end congestion awareness. Virtual Input Queues avoid packet loss due to congestion within the switch itself by coordinating ingress and egress queue handling. These new innovations work together to eliminate loss in the cloud data center network.

Citations

- [1] R. Kurzweil, *The Singularity Is Near: When Humans Transcend Biology*, Penguin Publishing Group, 2005.
- [2] R. Kapoor, G. Porter, M. Tewari, G. M. Voelker and A. Vahdat, "Chronos: predictable low latency for data center applications," in *Proceedings of the Third ACM Symposium on Cloud Computing*, San Jose, California, 2012.
- [3] V. Jalaparti, P. Bodik, S. Kandula, I. Menache, M. Rybalkin and C. Yan, "Speeding up distributed request-response workflows," in *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM*, Hong Kong, China, 2013.
- [4] J. Dean, G. S. Corrado, R. Monga, K. Chen, M. Devin, Q. V. Le, M. Z. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang and A. Y. Ng, "Large scale distributed deep networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, Lake Tahoe, Nevada, 2012.
- [5] rCUDA, "rCUDA remote CUDA," [Online]. Available: <http://rcuda.net/>. [Accessed 19 2 2018].
- [6] L. Mai, C. Hong and P. Costa, "Optimizing network performance in distributed machine

learning," in *Proceedings of the 7th USENIX Conference on Hot Topics in Cloud Computing*, Santa Clara, CA, 2015.

- [7] NVM Express, "NVM Express® Moves Into The Future," [Online]. Available: http://www.nvmexpress.org/wp-content/uploads/NVMe_Over_Fabrics.pdf. [Accessed 2 11 2017].
- [8] Cisco; EMC; Intel, "The Performance Impact of NVMe and NVMe over Fabrics," 13 November 2014. [Online]. Available: https://www.snia.org/sites/default/files/NVMe_Webcast_Slides_Final.1.pdf. [Accessed 28 10 2017].
- [9] C.-L. I, J. Huang, R. Duan, C. Cui, X. Jiang and L. Li, "Recent Progress on C-RAN Centralization and Cloudification," *IEEE Access*, vol. 2, pp. 1030-1039, 2014.
- [10] ETSI NFV ISG, "Network Functions Virtualisation (NFV)-Network Operator Perspectives on Industry Progress," 15 10 2013. [Online]. Available: https://portal.etsi.org/NFV/NFV_White_Paper2.pdf. [Accessed 21 2 2018].
- [11] A. Singh, J. Ong, A. Agarwal, G. Anderson, A. Armistead, R. Bannon, S. Boving, G. Desai, B. Felderman, P. Germano, A. Kanagala, J. Provost, J. Simmons, E. Tanda, J. Wanderer, U. Hölzle, S. Stuart and A. Vahdat, "Jupiter Rising: A Decade of Clos Topologies and Centralized Control in Google's Datacenter Network," in *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, London, United Kingdom, 2015.
- [12] IETF, "Data Center TCP (DCTCP): TCP Congestion Control for Data Centers," 17 10 2017. [Online]. Available: <https://datatracker.ietf.org/doc/rfc8257/>. [Accessed 1 11 2017].
- [13] M. Handley, C. Raiciu, A. Agache, A. Voinescu, A. W. A. G. Moore and M. Wojcik, "Re-architecting datacenter networks and stacks for low latency and high performance," in *SIGCOMM '17*, Los Angeles, 2017.
- [14] IEEE Std- 802.1Q™-2014, *Clause 36, Priority-based Flow Control*, IEEE, 2014.
- [15] Cisco Systems, Inc, "Priority Flow Control: Build Reliable Layer 2 Infrastructure," 2009.
- [16] D. K. Panda, "Fast barrier synchronization in wormhole k-ary n-cube networks with multideestination worms," *Future Generation Computer Systems*, vol. 11, no. 6, pp. 585-602, 1995.
- [17] N. McKeown, M. Izzard, A. Mekkittikul, W. Ellersick and M. Horowitz, "The Tiny Tera:1 A Packet Switch Core," *IEEE Micro*, vol. 17, no. 1, pp. 26-33, 1997.
- [18] J. Duato, I. Johnson, J. Flich, F. Naven, P. J. Garcia and T. N. Farinos, "A New Scalable and Cost-Effective Congestion Management Strategy for Lossless Multistage Interconnection Networks," in *HPCA*, 2005.

- [19] P. J. Garcia, F. J. Quiles, J. Flich, J. Duato, I. Johnson and F. Naven, "Efficient, Scalable Congestion Management for Interconnection Networks," *IEEE Micro*, vol. 26, no. 5, pp. 52-56, 2006.
- [20] J. Escudero-Sahuquillo, P. J. García, F. J. Quiles, J. Flich and J. Duato, "An Effective and Feasible Congestion Management Technique for High-Performance MINs with Tag-Based Distributed Routing," *IEEE Transactions on Parallel Distributed Systems*, vol. 24, no. 10, pp. 1918-1929, 2013.
- [21] J. Escudero-Sahuquillo, E. G. Gran, P. J. García, J. Flich, T. Skeie, O. Lysne, F. J. Quiles and J. Duato, "Efficient and Cost-Effective Hybrid Congestion Control for HPC Interconnection Networks," *IEEE Transactions on Parallel Distributed Systems*, vol. 26, no. 1, pp. 107-119, 2015.
- [22] B. Montazeri, Y. Li, M. Alizadeh and J. Ousterhout, "Homa: A Receiver-Driven Low-Latency Transport Protocol Using Network Priorities," 26 03 2018. [Online]. Available: <https://arxiv.org/abs/1803.09615v1>. [Accessed 22 05 2018].
- [23] IETF, "IEEE 802 and IETF Coordination Guide," 6 7 2017. [Online]. Available: <https://trac.ietf.org/trac/iesg/wiki/IEEE802andIETFCoordinationGuide>. [Accessed 1 2 2018].