(Draft) IEEE 802 Nendica Report: Intelligent Lossless Data Center Networks

Editor

Name	Affiliation
Guo, Liang	CIACT/ODCC
Congdon, Paul	Huawei

Nendica Chair

Name	Affiliation
Marks, Roger	Huawei

Contributors/Supporters

Name	Affiliation
Li, Jie	CIACT/ODCC
Gao, Feng	Baidu
Gu, Rong	China Mobile
Zhao, Jizhuang	China Telecom
Chen, Chuansheng	Tencent
Yin, Yue	Huawei
Song, Qingchun	Nvidia
Liu, Jun	Cisco
He, Zongying	Broadcom
Sun, Liyang	Huawei
Tang, Guangming	Meituan
Quan, Hao	Meituan
Tao, Chunlei	JD
Wang, Shaopeng	CIACT/ODCC

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.

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-dcn>

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

PDF: ISBN xxx-x-xxxx-xxxx-x XXXXXXXXXX

NOTICE AND DISCLAIMER OF LIABILITY CONCERNING THE USE OF IEEE SA INDUSTRY CONNECTIONS DOCUMENTS

TABLE OF CONTENTS

1 Introduction

This paper is the result of a work item [1] within the IEEE 802 “Network Enhancements for the Next Decade” Industry Connections Activity known as Nendica. The paper is an update to a previous report, “IEEE 802 Nendica Report: The Lossless Network for Data Centers” published on August 17, 2018 [2]. This update provides additional background on evolving use cases in modern data centers and proposes solutions to additional problems identified by this paper.

Scope

The scope of this report is the exploration of networking technologies to support the requirements of modern Data Center Networks that include support for High Performance Computing and Artificial Intelligence applications. Solutions to address challenges created by evolving requirements and new age technologies are proposed. Standardization considerations are identified.

Purpose

The purpose of this report is to frame high-level solutions to issues and challenges with modern Data Center Networks. The report includes background and technical analyses of current Data Center environments as they are applied to the evolving needs of target applications. The report highlights new technologies that are changing the dynamics and operation of the Data Center Network. The results of the analysis lead to identification and recommendation of future standardization activities.

2 Bringing the data center to life

A new world with data everywhere

Digital transformation is driving change in both our personal and professional lives. Workflows and personal interactions are turning to digital processes and automated tools that are enabled by the Cloud, Mobility, and the Internet of Things. The Intelligence behind the digital transformation is Artificial Intelligence (AI). Data centers running AI applications with massive amounts of data are recasting that data into pertinent timely information, automated human interactions, and refined decision making. The need to interact with the data center in real-time is more important than ever in today’s world where augmented reality, voice recognition, and contextual searching demand immediate results. Data center networks must deliver unprecedented levels of performance, scale, and reliability to meet these real-time demands.

Data centers in the cloud era focused on application transformation and the rapid deployment of services. In the AI era, data centers are the source of information and algorithms for the real-time

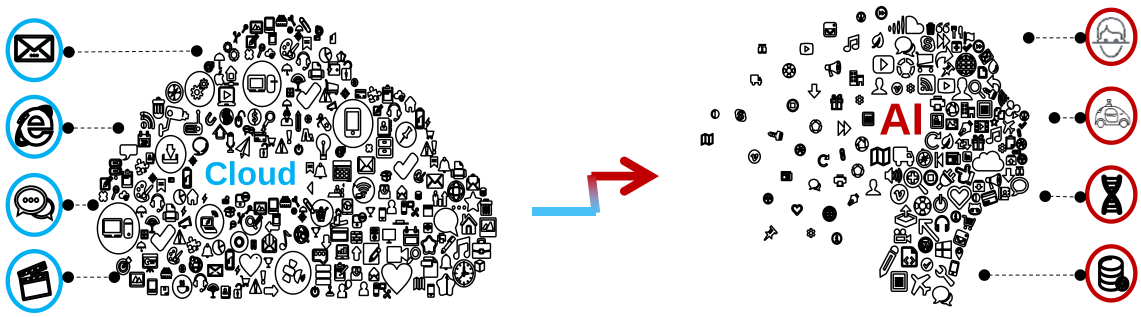


Figure 1 – Digital Transformation in the Era of AI

digital transformation of our digital lives. The combination of high-speed storage and AI distributed computing render big data into fast data, access by humans, machines, and things. A high-performance, large scale data center network without packet loss is critical to the smooth operation of the digital transformation.

For high-performance applications, such as AI, key measures for network performance include throughput, latency, and congestion. Throughput is dependent on the total capacity of the network for quickly transmitting a large amount of data. Latency refers to the total delay for a transaction across the data center network. When the traffic load exceeds the network capacity, congestion occurs. Packet loss is a factor that seriously affects both throughput and latency. Data loss in a network may cause a series events that deteriorate performance. For example, an upper-layer application may need to retransmit lost data in order to continue. Retransmissions can increase load on the network, causing further packet loss. In some applications, delayed results are not useful, and the ultimate results can be discarded, thus wasting resources. In other cases, the delayed result is just a small piece of the puzzle being assembled by the upper-layer application that has now been slowed down to the speed of the slowest worker. More seriously, when an application program does not support packet loss and cannot be restored to continue, a complete failure or damage can be occur.

Today's data center enables the digital real-time world

Currently, digital transformation of various industries is accelerating. It is estimated that 64% of enterprises have become the explorers and practitioners of digital transformation [3]. Among 2000 multinational companies, 67% of CEOs have made digitalization the core of their corporate strategies [4]. The drive towards digital transformation in the real-time world is leading the Data Center Network to support a 'Data-Centric' model of computing.

A large amount of data will be generated during the digitalization process, becoming a core asset, and enabling the emergence of Artificial Intelligence applications. Huawei GIV predicts that the data volume will reach 180 ZB in 2025 [5]. However, data is not the "end-in-itself". Knowledge and wisdom extracted from data are eternal values. However, the proportion of unstructured data (such as raw voice, video, and image data) increases continuously, and will account for 95% of all data in the future. Performance innovations are needed to extract the value from the raw data. At this scale, the current big data analytic methods are helpless. If manual processing is used, the data volume will be far greater than the processing capability of all human beings. The AI approach based on machine computing for deep learning can filter out massive amounts of invalid data and

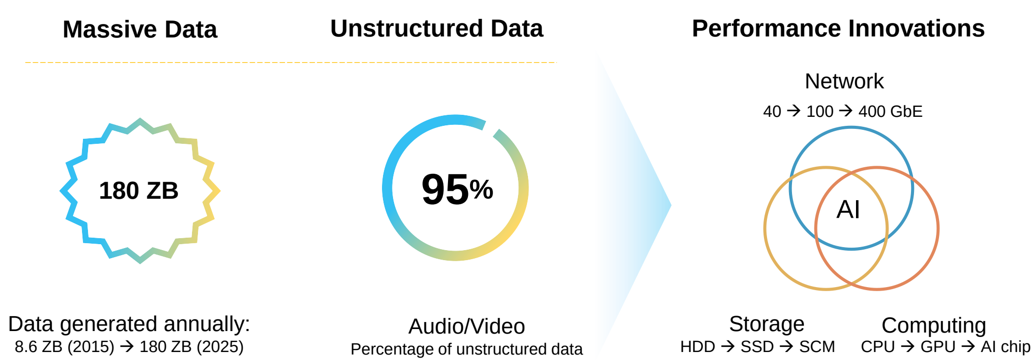


Figure 2 – Emerging Artificial Intelligence Applications

automatically reorganize useful information, providing more efficient decision-making suggestions and smarter behavior guidance.

The cloud data center architecture improved the performance and scale of applications in general. The cloud platform allows rapid distribution of IT resources to create an application-centric service model. In the AI era, the applications are consuming unprecedented amounts of data and the cloud data center architecture is augmented with necessary performance innovations to handle the load. Seamlessly introducing these innovations along with new AI applications can be tricky in an existing cloud data center. Understanding how to efficiently process data based on the needs of AI applications is a key focus area. Orchestrating the flow of data between the storage and computing resources of the applications is a critical success factor.

3 Evolving data center requirements and technology

Previous Data Center Bridging Standards

During the early days of 10 Gbps Ethernet, the IEEE 802.1 Working Group developed a focus on Data Center Bridging (DCB). The DCB task group defined a set of enhancements to the Ethernet, Bridges, and associated protocols for use in data center environments. The use-case and focus were with clustering and storage area networks, where traditionally dedicated technologies such as Infiniband and Fiber Channel were used. Important objectives for Ethernet were to eliminate loss due to congestion and to allocate bandwidth on links for selected traffic. The key contributions at the time included the following:

- **Priority-based Flow Control (PFC)**: A link level flow control mechanism that eliminates packet loss and can be applied independently for each traffic class.
- **Enhanced Transmission Selection (ETS)**: A queue scheduling algorithm that allows for bandwidth assignments to traffic classes.

- **Congestion Notification**: A layer-2 end to end congestion management protocol that detects congestion, signals across the layer-2 network to limit the transmission rate of senders to avoid packet loss.
- **Data Center Bridging Capabilities Exchange Protocol (DCBX)**: a discovery and capability exchange protocol, working in conjunction with the Link Layer Discovery Protocol (LLDP), to convey capabilities and configuration of the above features.

These contributions were important to the expansion of Ethernet into the specialized markets of cluster computing and storage area networks. However, continued evolution is needed as the environments and technologies have changed. Data centers are deployed on massive scale, using Layer-3 protocols and highly orchestrated management systems. Ethernet links have advanced from 10 Gbps to 400 Gbps, with active plans to increase speeds into the Tbps range. New applications, such as Artificial Intelligence (AI) are placing new demands on the infrastructure and driving architectural changes. Continued innovation is needed to further expand the use of Ethernet in modern data center environments.

Requirements evolution

AI applications put pressure on the data center network. Consider AI training for self-driving cars as an example, the deep learning algorithm relies heavily on massive data and high-performance computing capabilities. The training data collected is approaching the P level (1PB = 1024 TB) per day. If traditional hard disk storage and common CPUs were used to process the data, it could take at least one year to complete the training, which is clearly impractical. To improve AI data processing efficiency, revolutionary changes are needed in the storage and computing fields. For example, storage performance needs to improve by an order of magnitude to achieve more than 1 million input/output operations per second (IOPS) [6].

To meet real-time data access requirements, storage media has evolved from hard disk drives (HDDs) to solid-state drives (SSDs) to storage-class memory (SCMs). This has reduced storage medium latency by more than 1000 times. Without similar improvements in network latency, these storage improvements cannot be realized and simply move the bottleneck from the media to communication latency. With networked SSD drives, the communication latency accounts for more than 60% of the total storage end-to-end latency. With the move to SCM drives, this percentage could increase to 85% unless improvements in network performance are achieved. This creates a scenario where the precious storage media is idle more than half of the time. When you consider recent improvements in both storage media and AI computing processors together, the communication latency accounts for more than 50% of the total latency, further hindering improvements and wasting resources [7].

The improvements in storage and computing performance support the AI computing model, which is growing in scale and complexity with the advent of AI cloud-based services. For example, there were 7 ExaFLOPS and 60 million parameters in Microsoft's Resnet of 2015. Baidu used 20 ExaFLOPS and 300 million parameters when training their deep speech system in 2016. In 2017, the Google NMT used 105 ExaFLOPS and 8.7 billion parameters [8]. New characteristics of AI computing are requiring an evolution of data center network.

Traditional protocols are no longer able to satisfy the requirements of new applications that serve our daily lives. In a simple example, the online food take-out industry at Meitan has increased nearly 500% in the last four years. The number of transactions has increased from 2.149 billion to 12.36

billion where those transactions all occur within a few hours at peak mealtimes. The Meituan Intelligent Scheduling System is responsible for orchestrating a complex multi-person, multi-point real-time decision-making process for end-users, businesses and over 600,000 delivery drivers. The drivers report positioning data 5 billion times a day that are used to calculate optional paths for the drivers and deliver optimal solutions within 0.55 milliseconds. When the back-end servers use TCP/IP protocols, the amount of data copied between kernel buffers, application buffers and NIC buffers stresses the CPU and memory bus resources causing increased delay and an inability to meet the application requirements. The newer Remote Direct Memory Access (RDMA) protocol eliminates data copies and frees CPU resources to perform necessary driver path and take-out order calculations at scale. The improved efficiency of RDMA puts more pressure on the network, moving the bottleneck to the data center network infrastructure where low-latency and lossless behavior become the new critical requirements.

Characteristics of AI computing

Traditional data center services (web, database, and file storage) are transaction-based and the calculated results are often deterministic. For such tasks, there is little correlation or dependency between a single transaction and the associated network communication. The occurrence and duration of the traditional transactions are random. AI computing, however, is different. It is an optimization problem with iterative convergence required in the computing process. This causes high spatial correlation within the data sets and computing algorithms, and temporally creates similar correlations with communication flows.

AI computing works on big data and demands fast data. To achieve this it must operate in parallel to “divide-and-conquer” the problem. The computing model and input data sets are large (e.g in a 100 MB node, the AI model with 10K rules requires more than 4 TB memory). A single server cannot provide enough storage capacity and processing resources to handle the problem sequentially. Concurrent AI computing and storage nodes are required to shorten the processing time. The distributed AI computing and storage requirement highlights the need for a fast, efficient, and lossless data center network that has the flexibility to support two distinct parallel modes of operation: model parallel computing and data parallel computing.

Model Parallel Computing

In model parallel computing, each node computes one part of the overall algorithm. Each node processes the same set of data, but with a different portion of the algorithm, resulting in an estimate for a differing set of parameters. The nodes exchange their estimates to converge upon the best estimate for all the data parameters. With model parallel computing, there is an initial distribution of the common data set to a distributed number of nodes, followed by a collection of individual parameters from each of the participating nodes. Figure 3 shows how parameters of the overall model may be distributed across computing nodes in a model parallel mode of operation.

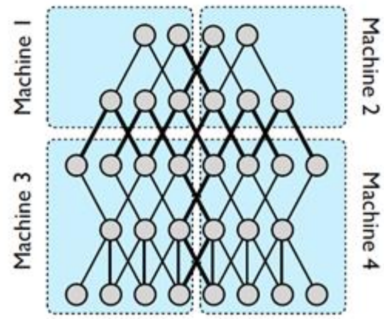


Figure 3 - Model parallel training

Data Parallel Computing

In data parallel computing, each node loads the entire AI algorithm model, but only processes part of the input data. Each node is trying to estimate the same set of parameters using a different view of the data. When a node completes a round of calculations, the parameters are weighted and aggregated by a common parameter server as seen in Figure 4. The weighted parameter update requires that all nodes upload and obtain the information synchronously.

Regardless of the parallel computing approach, data center networks feel the pressure of demanding communication. When the network becomes the bottleneck, the waiting time for computing resources exceeds 50% of the job completion time [9].

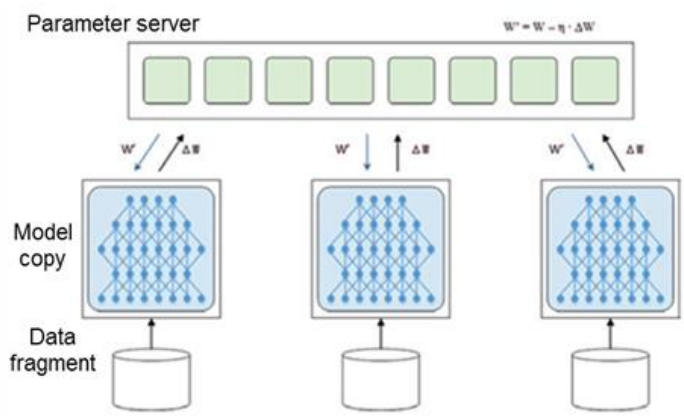


Figure 4 - Data parallel training

Evolving technologies

Progress can be seen when evolving requirements and evolving technologies harmonize. New requirements often drive the development of new technologies and new technologies often enable new use cases that lead to, yet again, a new set of requirements. Breakthroughs in networked storage, distributed computing, system architecture and network protocols are enabling the advancement of the next generation data center.

SSDs and NVMeoF: High throughput, low-latency network

In networked storage, a file is distributed to multiple storage servers for IO acceleration and redundancy. When a data center application reads a file, it accesses different parts of data from different servers concurrently. The data is aggregated through a data center switch at nearly the same time. When a data center application writes a file, the writing of data can trigger a series of storage transactions between distributed and redundant storage nodes. Figure 5 shows an example of data center communication triggered by the networked storage service model.

When an application (i.e. Client in Figure 5) requests to write a file, it will concurrently send data to the object storage device (OSD) servers. There are two types of OSD servers, one type is the primary, and the other type is the replica. When the primary servers receive data that need to be saved, it will transmit the data to the replica servers twice as backup (the orange arrowhead in Figure 5). After receiving the data, the primary OSD server will send an ACK to client while the replica servers will send ACK to the primary server (pink dash line in Figure 5). Each OSD server will then begin to commit the data to the storage medium. It takes a short period time to commit and store data. When the replica servers finish saving data, they will send commit notification to primary server to notify that the writing task is complete. Once the primary server has received all the commit information from all replica servers, the primary server will send a commit message to client. The storage write process is not complete until the primary server has sent the final commit message to the client.

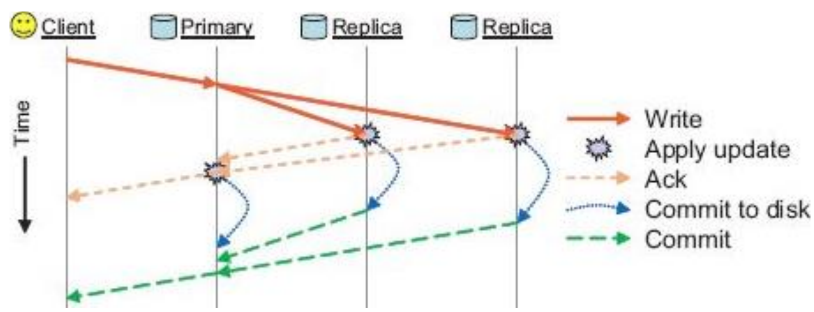


Figure 5 - Networked storage service model

The example highlights the importance of the network enabling both high throughput and low latency simultaneously. The bulk data being written to the primary storage server is transmitted multiple times to the replicas. The small sized acknowledgments and commit messages must be sequenced and ultimately delivered to the originating client before the transaction can complete, emphasizing the need for ultra-low latency.

Massive improvements in storage performance have been achieved as the technology has evolved from HDD to SSD to SCM using the NVMe (Non-Volatile Memory Express) interface specification. Accessing storage media via NVMe has decreased access time by a factor of 1000 over previous HDD technology. Sample seek times between the various technologies include; HDD = 2-5 ms, SATA SSD = 0.2 ms, and NVMe SSD = 0.02 ms. SCM is generally three to five times faster than NVMe flash SSDs. While shorter overall average seek times are better, the performance of drives in each category can still vary [10].

NVMe-over-fabrics (NVMeoF) involves deploying NVMe for networked storage. The much faster access speed of the medium result in greater network bottlenecks and the impact of network

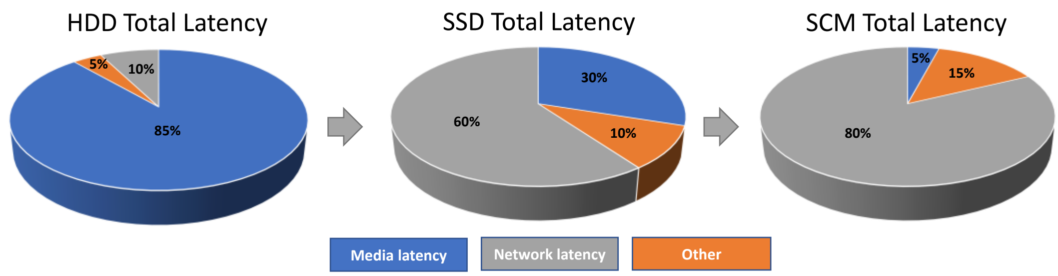


Figure 6 – End-to-end latency breakdown for HDD and SSD

latency becomes more significant. Figure 6 shows how network latency has become the primary bottleneck with faster NVMe based storage. Once upon a time network latency was a negligible part of end-to-end networked HDD storage latency. To maximize the IOPS performance of the new medium, the network latency problem must be resolved first.

An analysis network latency show that it is a combination of two distinct types of latency: static latency and dynamic latency. Static latency includes serial data latency, device forwarding latency, and optical/electrical transmission latency. This type of latency is determined by the capability of the switching hardware and the transmission distance of the data. It usually is fixed and very predictable. Figure 7 shows the current industry measurements for static latency are generally at nanosecond (10^{-9} second) or sub-microsecond (10^{-6}) level, and account for less than 1% of the total end-to-end network delay.

Dynamic latency plays a much greater role in total end-to-end network delay and is greatly affected by the conditions within the communication environment. Dynamic latency is created from delays introduced by internal queuing and packet retransmission, which are caused by network congestion and packet loss. In the AI era, congestion from the unique traffic patterns of high-speed storage and specialized AI computing nodes becomes more and more severe on the network. Packet queuing and packet loss can occur frequently, causing the end-to-end network latency to skyrocket to the level of sub-seconds. The key to low end-to-end network latency is to address dynamic latency.

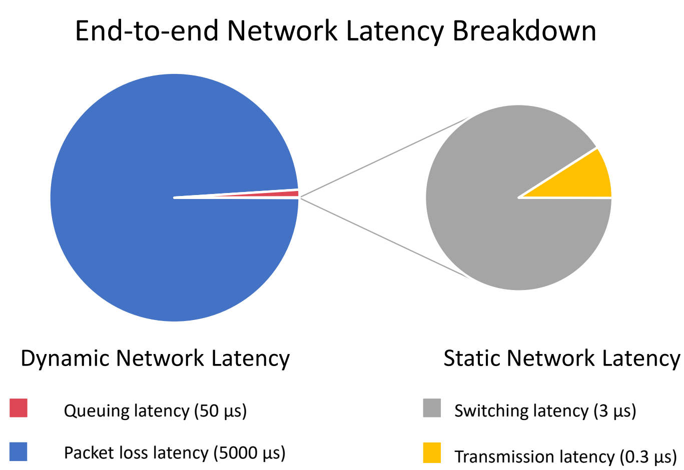


Figure 7 – Network Latency Breakdown

The major component of dynamic latency is the delay from packet retransmission when packets are dropped within the network. Packet loss latency is an order magnitude greater than queuing delay and has proven to have a severe impact on applications. Packet loss occurs when switch buffers are overrun because of congestion (NOTE: we ignore packet loss due low-probability bit errors during transmission). There are two key types of congestion that lead to packet loss: in-network congestion and incast congestion. In-network congestion occurs on switch-to-switch links within the network fabric when the links become overloaded, perhaps due to ineffective load balancing. Incast congestion occurs at the edge of the network when many sources are sending to a common destination at the same time. AI computing models inherently have a phase when data is aggregated after a processing iteration from which incast congestion (many-to-one) easily occurs.

GPUs: Ultra-low latency network for parallel computing

Today's AI computing architecture includes a hybrid mix of Central Processing Units (CPUs) and Graphics Processing Units (GPUs). GPUs, originally invented to help render video games at exceptional speeds, have found a new home in the data center. The GPU is a processor with thousands of cores capable of performing millions of mathematical operations in parallel. All AI learning algorithms perform complex statistical computations and deal with a huge number of matrix multiplication operations per second – perfectly suited for a GPU. However, to scale the AI computing architecture to meet the needs of today's AI applications in a data center, the GPUs must be distributed and networked. This places stringent requirements on communication volume and performance.

Facebook recently tested the distributed machine learning platform Caffe2, in which the latest multi-GPU servers are used for parallel acceleration. In the test, computing tasks on eight servers resulted in underutilized resources on the 100 Gbit/s InfiniBand network. The presence of the network and network contention reduced the performance of the solution to less than linear scale [12]. Consequently, network performance greatly restricts horizontal extension of the AI system.

GPUs provide much higher memory bandwidth than today's CPU architectures. Nodes with multiple GPUs are now commonly used in high-performance computing because of their power efficiency and hardware parallelism. Figure 8 illustrates the architecture of typical multi-GPU nodes, each of which consists of a host (CPUs) and several GPU devices connected by a PCI-e switch or NVLink. Each GPU is able to directly access its local relatively large device memory, much smaller and faster shared memory, and a small, pinned area of the host node's DRAM, called zero-copy memory [13].

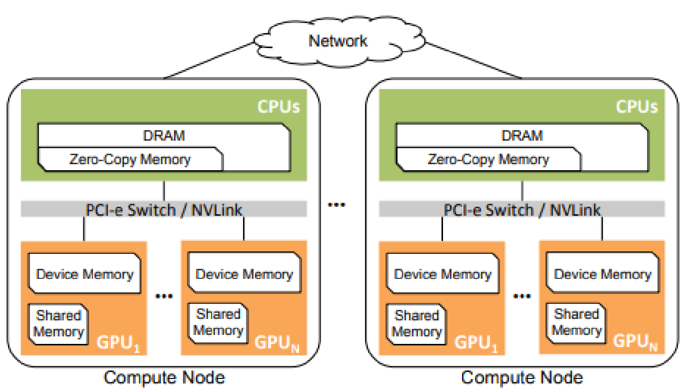


Figure 8 – Distributed AI Computing Architecture

GPUs are inherently designed to work on parallel problems. With AI applications, these problems are iterative and require a synchronization step that creates network incast congestion. Figure 9 shows how incast congestion occurs with AI training. The training process is iterative and there are many parameters synchronized on each iteration. The workers download the model and upload newly calculated results (ΔM) to a parameter server during a synchronization step. The uploading to the parameter server creates incast. When the computing time is improved by deploying faster GPUs, the pressure on the network and resulting incast increases.

The communication between the worker nodes and the parameter server constitutes a collection of interdependent network flows. In the iteration process of distributed AI computing, many burst traffic flows are generated to distributed data to workers within milliseconds, followed by an incast event of smaller sized flows directed at the parameter server when the intermediate parameters are delivered and updated. During the exchange of these flows packet loss, congestion, and load imbalance can occur on the network. As a result, the Flow Completion Time (FCT) of some of the flows is prolonged. If a few flows are delayed, storage and computing resource can be underutilized. Consequently, the completion time of the entire application is delayed.

Distributed AI computing is synchronous, and it is desirable for the jobs to have a predictable completion time. When there is no congestion, dynamic latency across the network is small allowing the average FCT to be predictable and therefor the performance of the entire application is predictable. When congestion causes dynamic latency to increase to the point of causing packet loss, FCT can be very unpredictable. Flows that complete in a time that is much greater than the average completion contributes to what is known as tail latency. Tail latency is the small percentage of response times from a system, out of all of responses to the input/output (I/O) requests it serves, that take the longest in comparison to the bulk of its response times. Reducing tail latency as much as possible is extremely critical to the success of parallel algorithms and the whole distributed computing system. To maximize the use of GPUs in the data center, tail latency should be addressed.

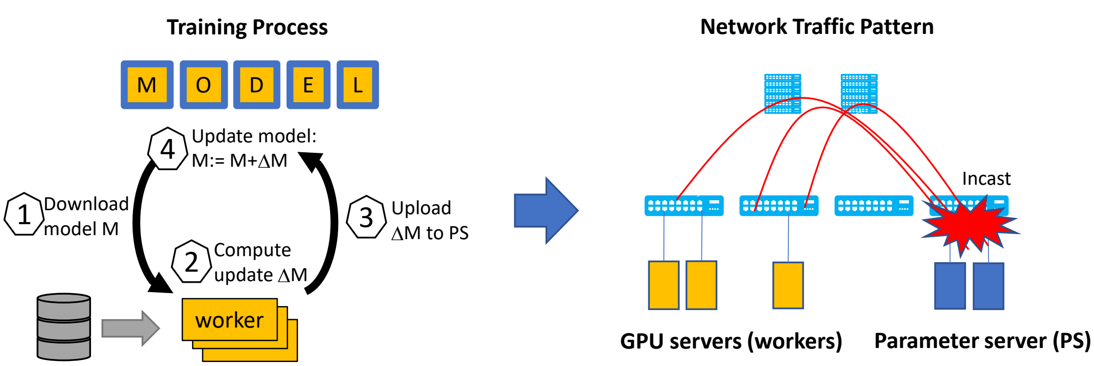


Figure 9 - Periodic incast congestion during training

SmartNICs

Over the years there have been periods of time when performance improvements in CPU speeds and Ethernet links have eclipsed one another. Figure 10 shows the historical performance gains

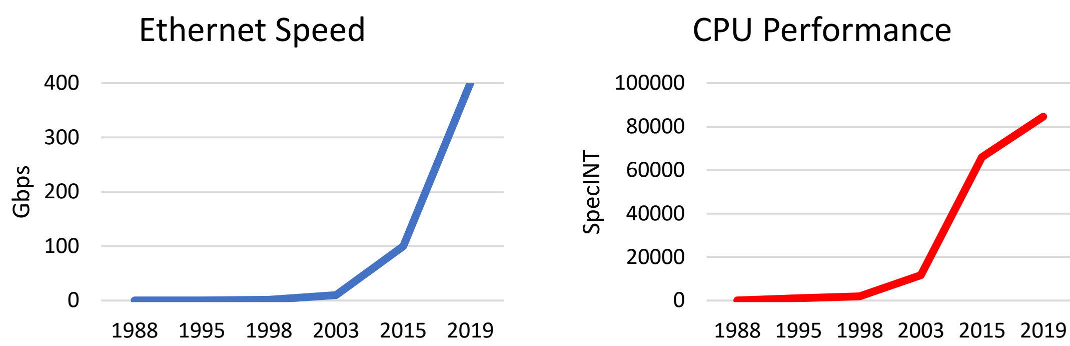


Figure 10 – Historical Performance Comparison

with Ethernet link speeds [14] and benchmark improvements for CPU performance [15]. During some historical periods, the processing capability of a traditional CPU was more than enough to handle the load of an Ethernet link and the cost savings of a simplified network interface card (NIC) along with the flexibility of handling the entire networking stack in software was a clear benefit. During other periods, the jump in link speed from the next iteration of IEEE 802.3 standards was too much for the processor to handle and a more expensive and complex SmartNIC with specialized hardware offloads became necessary to utilize the Ethernet link. As time goes on and the SmartNIC offloads mature, some of them become standard and included in the base features of what is now considered a common NIC. This phenomenon was seen with the advent of the TCP Offload Engine (TOE) which supported TCP checksum offloading, large segment sending and receive side scaling.

In today's world, there are signs of Moore's law fading while Ethernet link speeds continue to soar. The latest iteration of IEEE 802.3 standards is achieving 400 Gbps. Couple this divergence with the added complexity of software-defined networking, virtualization, storage, message passing and security protocols in the modern data center, and there is a strong argument that the SmartNIC architecture is here to stay. So, what exactly is a data center SmartNIC today?

Figure 11 shows a data center server architecture including a SmartNIC. The SmartNIC includes all the typical NIC functions, but also includes key offloads to help accelerate applications running on the server CPU and GPU. The SmartNIC does not replace the CPU or the GPU but rather complements them with networking offloads. Some of the key offloads include virtual machine

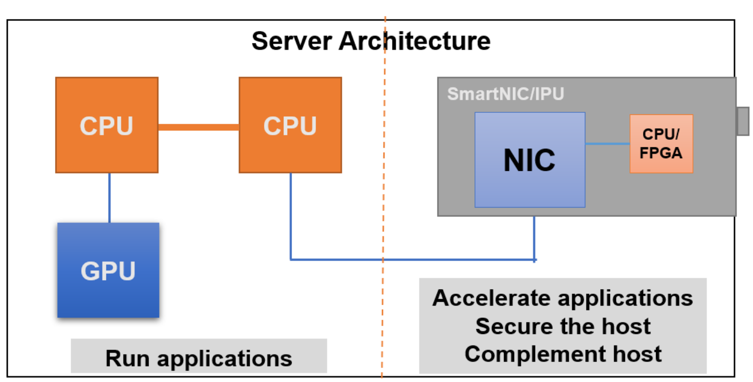


Figure 11 – Server Architecture with SmartNIC

interface support, flexible match-action processing of packets, overlay tunnel termination and origination, encryption, traffic metering, shaping and per-flow statistics. Additionally, SmartNICs often include entire protocol offloads and direct data placement to support RDMA and NVMe-oF storage interfaces.

One new critical component of today's SmartNIC is programmability. A criticism of SmartNICs in the past was their inability to keep pace with the rapidly changing networking environment. The early cloud data center environments favored using the CPU for most networking functions because the required feature set for the NIC was evolving faster than the development cycle of the hardware. Today's SmartNICs however have an open and flexible programming environment. They are essentially a computer in front of the computer with an open source development environment based on Linux and other software-defined networking tools such as Open vSwitch [16]. It is essential that SmartNICs integrate seamlessly into the open source ecosystem to enable rapid feature development and leverage.

SmartNICs in the data center increase the overall utilization and load on the network. They can exacerbate the effects of congestion by fully and rapidly saturating a network link. At the same time, they can respond quickly to congestion signals from the network to alleviate intermittent impact and avoid packet loss. The programmability of the SmartNIC allows it to adapt to new protocols that can coordinate with the network to avoid conditions such as incast.

RDMA

RDMA (Remote Direct Memory Access) is a new technology designed to solve the high latency problem of server-side data processing in network applications. With RDMA data transfers directly from one computer's memory to another without the intervention of either's operating system. This allows for high bandwidth, low latency network communication and is particularly suitable for use in massively parallel computer environments. RDMA allows the transfer of data directly into the storage space of another computer, reducing or eliminating the need for multiple copies of the data during transmission. This frees up memory bandwidth and CPU cycles to greatly improve system performance. Figure 12 shows the principles of the RDMA protocol.

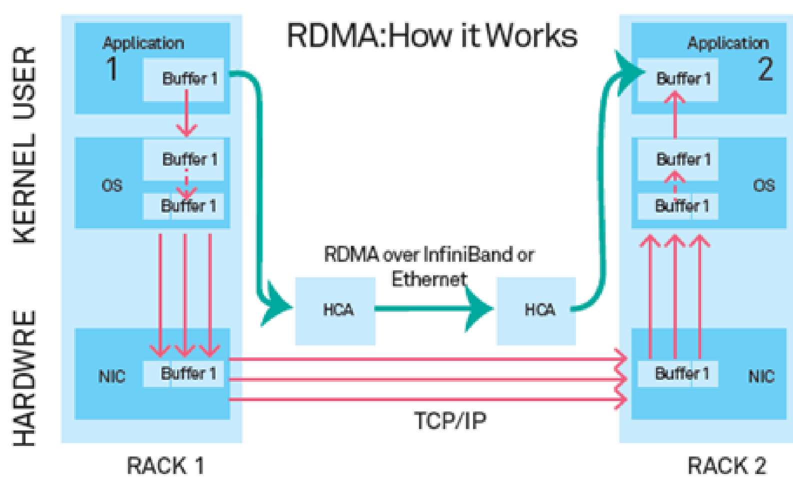


Figure 12 - Working principle of RDMA

There are three different transports for the RDMA protocol: Infiniband, iWarp and RoCEv1/RoCEv2.

Infiniband

In 2000, the InfiniBand Trade Association (IBTA) released the initial support for RDMA, Infiniband, which is a network technology customized for RDMA through a specific hardware design to ensure the reliability of data transmission. InfiniBand allows RDMA to directly read and write the memory of remote nodes. Infiniband is a unique network solution requiring specific Infiniband switches and Infiniband interface cards.

iWarp

An RDMA protocol that runs over TCP, allowing it to traverse the Internet and wide area, has been defined by the IETF and is known as iWarp. In addition to the wide area, iWarp also allows RDMA to run over a standard Ethernet network and within a data center. While iWarp can be implemented in software, to obtain the desired performance of RDMA special iWarp enabled NIC card are used.

RoCE (RDMA over Converged Ethernet)

In April 2010, the IBTA released the RoCEv1 specification, which augments the Infiniband Architecture Specification with the capability of supporting InfiniBand over Ethernet (IBoE). The RoCEv1 standard specifies an Infiniband network layer directly on top of the Ethernet link layer. Consequently, the RoCEv1 specification does not support IP routing. Since Infiniband relies on a lossless physical transport, the RoCEv1 specification depends on a lossless Ethernet environment.

RoCEv2

Modern data centers tend to use Layer-3 technologies to support large scale and greater traffic control. The RoCEv1 specification required an end-to-end layer-2 Ethernet transport and did not operate effectively in a layer-3 network. In 2014, the IBTA published RoCEv2, which extended

Technology	Data Rates (Gbit/s)	Latency	Key Technology	Advantage	Disadvantage
TCP/IP over Ethernet	10, 25, 40, 50, 56, 100, or 200	500-1000 ns	TCP/IP Socket programming interface	Wide application scope, low price, and good compatibility	Low network usage, poor average performance, and unstable link transmission rate
Infiniband	40, 56, 100, or 200	300-500 ns	InfiniBand network protocol and architecture Verbs programming interface	Good performance	Large-scale networks not supported, and specific NICs and switches required
RoCE/RoCEv2	40, 56, 100, or 200	300-500 ns	InfiniBand network layer or transport layer and Ethernet link layer Verbs programming interface	Compatibility with traditional Ethernet technologies, cost-effectiveness, and good performance	Specific NICs required Still have many challenges to
Omni-Path	100	100 ns	OPA network architecture Verbs programming interface	Good performance	Single manufacturer and specific NICs and switches required

Table 1 – Comparison of RDMA Network Technologies

RoCEv1 by replacing the Infiniband Global Routing Header (GRH) with an IP and UDP header. Now that RoCE is routable it is easily integrated into the preferred data center environment. However, to obtain the desired RDMA performance, the RoCE protocol is offloaded to special network interface cards. These network cards implement the entire RoCEv2 protocol, including the UDP stack, congestion control and any retransmission mechanisms. While UDP is lighter weight than TCP, the additional support required to make RoCEv2 reliable adds complication to the network card implementation. RoCEv2 still depends upon the Infiniband Transport Protocol, which was designed to operate in a lossless Infiniband environment, so RoCEv2 still benefits from a lossless Ethernet environment.

Figure 13 shows the most common RDMA protocol stacks and their associated standards bodies. Table 1 compares the details of different implementations. RDMA is more and more widely used to support high-speed storage, AI and Machine Learning applications in large scale cloud data centers. There are real world examples of tens of thousands of servers running RDMA in production. Applications have reported impressive performance improvements by adopting RDMA [17]. For instance, distributed machine learning training has been accelerated by 100+ times compared with the TCP/IP version, and the I/O speed of SSD-based cloud storage has been boosted by about 50 times compared to the TCP/IP version. These improvements majorly stem from the hardware offloading characteristic of RDMA.

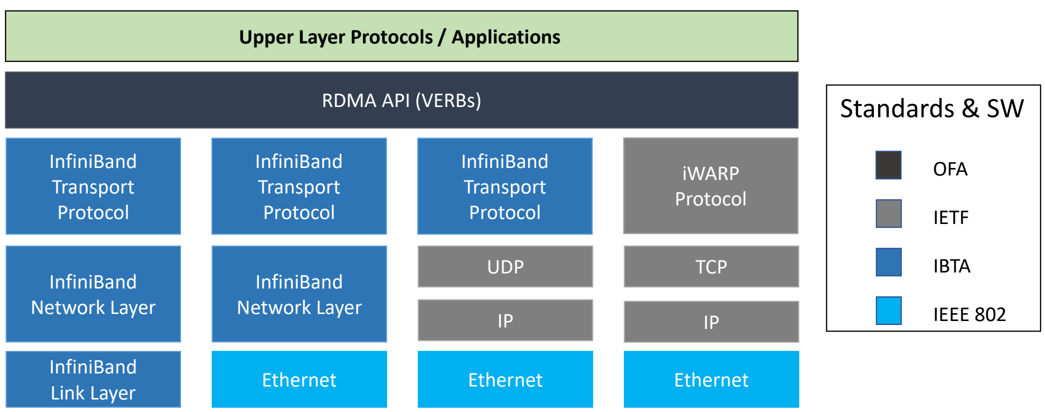


Figure 13 – RDMA protocol stacks and standards

GPU DirectRDMA

Combining two good ideas can often create a breakthrough idea. GPU DirectRDMA comprises the PeerDirect technology of PCIe and the RDMA technology of the network to deliver data directly to the GPU. This technology includes support for any PCIe peer which can provide access to its memory, such as NVIDIA GPU, XEON PHI, AMD GPU, FPGA, and so on.

GPU communications uses “pinned” buffers for data movement. A SmartNIC may also use “pinned” memory to communicate with a remote “pinned” memory across the network. These two types of “pinned” memory are sections in the host memory that are dedicated for the GPU, and separately for the SmartNIC.

Before GPU DirectRDMA, when one GPU transferred data to another GPU in a remote server, the source GPU needed to copy the data from GPU memory to CPU memory which was pinned by the

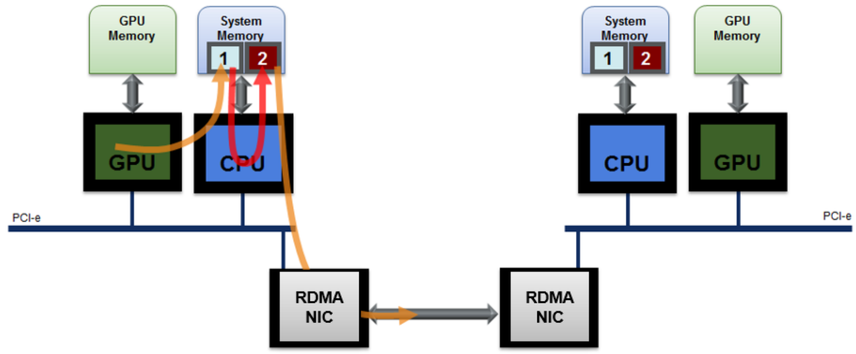


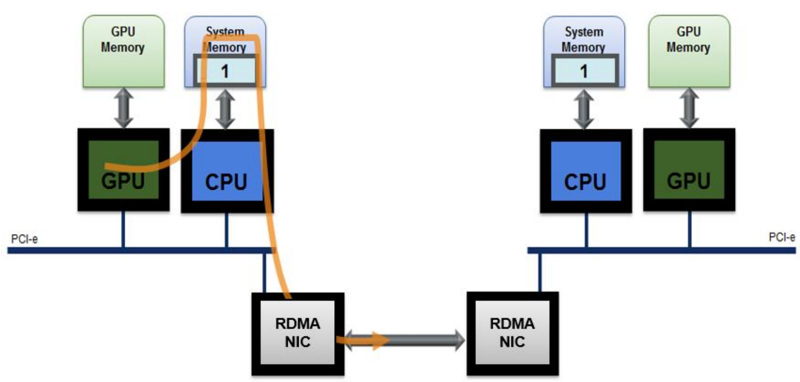
Figure 14: The Data Transfer Before GPU DirectRDMA

GPU. Then the host CPU copied the data from the GPU pinned memory to memory pinned by the SmartNIC. Next, the SmartNIC transmitted the data from the local server to the remote server across the network. On the remote server side, the reverse process took place. The data arrived at the memory pinned by the SmartNIC, then the CPU copied the data to the memory pinned by the GPU, and eventually the data arrived at the remote GPU memory from the host memory. Figure 14 shows the GPU-to-GPU data copy process before the existence of GPU DirectRDMA.

While the cost of copying data between the GPU and CPU is much lower than the cost of using TCP to pass the data between GPUs, it still suffers from a several issues:

1. Consumption of GPU resources. The CPU may become the bottleneck during the data copy.
2. Increased latency and lower bandwidth between the GPU and the remote GPU.
3. Host memory consumption. Consumption of host memory impacts application performance and increases system TCO.

Optimizations such as write-combining and overlapping GPU computation with data transfer allow the network and the GPU to share “pinned” (page-locked) buffers. This eliminates the need to make a redundant copy of the data in host memory and allows the data to be directly transferred via RDMA. On the receiver side the data is directly written to the GPU pinned host buffer after arriving



Picture 15: The Data Transfer Using GPU Direct

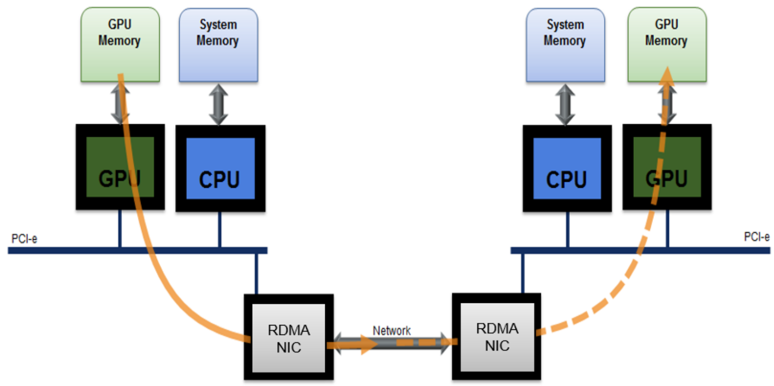


Figure 16: The Data Transfer Using GPU DirectRDMA

via RDMA. This technique eliminates buffer copies between the CPU and the GPU and is known as GPU Direct technology.

A further optimization is to create an RDMA channel between the local GPU memory and the remote GPU memory to eliminate CPU bandwidth and latency bottlenecks. This results in significantly improved communication efficiency between GPUs in remote nodes. For this optimization to work, the CPU prepares and queues communication tasks for the GPU and uses the GPU to trigger the communication on the SmartNIC. The SmartNIC directly accesses GPU memory to send and receive or to read and write data. This technique is known as GPU DirectRDMA technology.

Figure 17 shows how GPU DirectRDMA technology improves GPU communication performance by a factor of 10 over the traditional approach. These improvements have made GPU DirectRDMA technology a mandatory component of HPC and AI applications, improving both performance and scalability. All standard Message Passing Interface (MPIs) and the NVIDIA Collective Communications Library (NCCL) include native RDMA support.

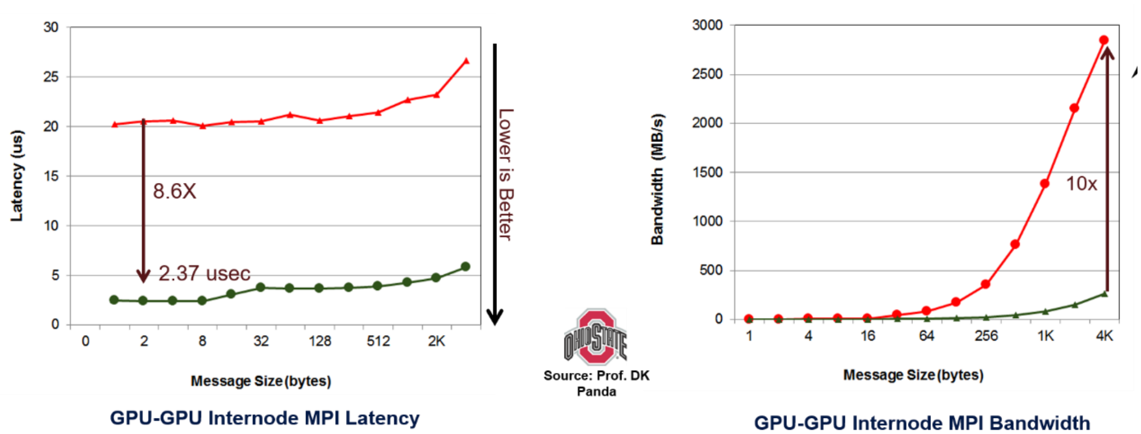


Figure 17: GPU DirectRDMA Performance (From OSU)

4 Challenges with today's data center network

High throughput and low latency tradeoff

Simultaneously achieving both low latency and high throughput in a large-scale data center is difficult. To achieve low latency, it is necessary to allow flows to begin transferring at line rate while at the same time maintaining near empty switch queues. Aggressively starting flows at line rate allows them to consume all available network bandwidth instantly and can lead to extreme congestion at convergence points. Deep switch buffers absorb temporary congestion to avoid packet loss but delay the delivery of latency sensitive packets. While deep switch buffers provide more resources for balancing the tradeoff between low latency and high throughput it is increasingly difficult to build switches with deep buffers. Switch capacity continues to increase with link speeds and higher port density, but the buffer size of commodity switching chips cannot keep pace. Figure 18 shows hardware trends for top-of-the-line data center switches chips manufactured by Broadcom [18].

Using a low ECN marking threshold can help slow aggressive flows and keep switch queue levels empty, but this reduces throughput. High throughput flows benefit from larger switch queues and higher ECN marking thresholds to prevent overreacting to temporary congestion and slowing down unnecessarily.

Experimentation shows the tradeoff between high throughput and low latency exists after varying algorithms, parameters, traffic patterns and link loads [17]. Figure 19 from [17] shows how flow completion times (FCT) are delayed beyond their theoretical minimum FCT when using different

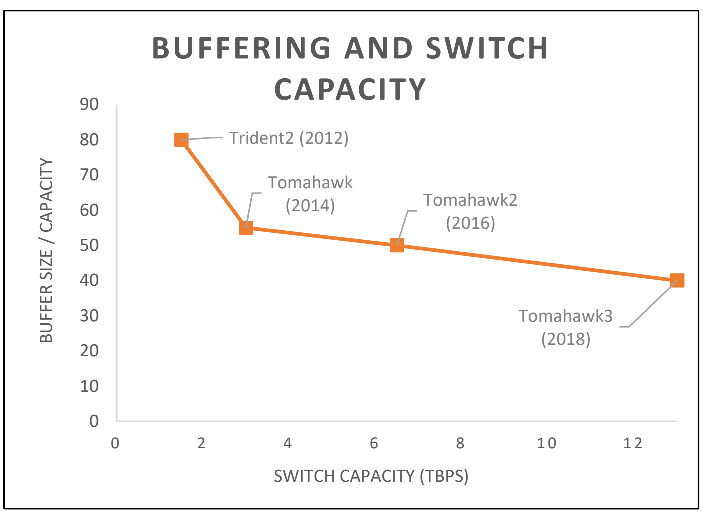


Figure 18 – Switch Chip Buffer Trends

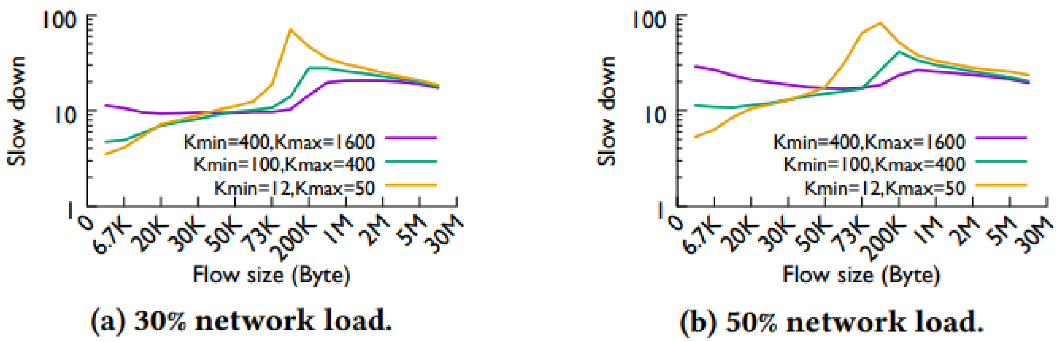


Figure 19 – FCT slowdown distribution with different ECN thresholds, using WebSearch

ECN marking thresholds (Kmin, Kmax) during a controlled experiment using a public RDMA WebSearch traffic workload as the input. Lower values for Kmin and Kmax will cause ECN markings to occur more quickly and force a flow to slow down more aggressively. As seen in the figure, when using low ECN thresholds, small flows which are latency-sensitive have lower FCT slowdown, while big flows which are typically bandwidth-hungry suffer from larger FCT slowdown. The trend is more obvious when the network load is higher (Figure 19-b when the average link load is 50%).

Deadlock free lossless network

RDMA advantages over TCP include low latency, high throughput, and low CPU usage. However, unlike TCP, RDMA needs a lossless network; i.e. there should be no packet loss due to buffer overflow at the switches [19]. The RoCE protocol runs on top of UDP with a go-back N retransmission strategy that severely impacts performance when retransmission is invoked. As such, RoCE requires Priority-based Flow Control (IEEE Std 802.1Q-2018, Clause 36 [20]) to ensure that no packet loss occurs in the data center network. Figure 20 from [21] shows how the RoCE service throughput decreases rapidly with increases in packet loss rate. Loosing as little as one in one thousand packets decreases RoCE service performance by roughly 30%.

Priority-based Flow Control (PFC) prevents packet loss due to buffer overflow by pausing the upstream sending device when the receiving device input buffer occupancy exceeds a specified

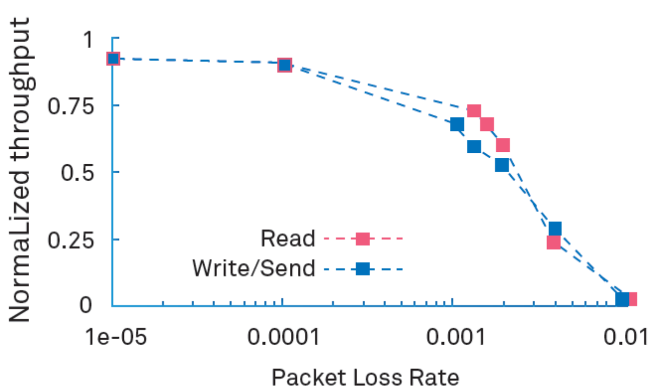


Figure 20 – Impact of packet loss on RDMA throughput

threshold. While this provides the necessary lossless environment for RoCE, there are problems with the large-scale use of PFC. One such problem is the possibility of a PFC deadlock.

Deadlocks in lossless networks using PFC style backpressure have been studied for many years [22, 23, 24]. A PFC deadlock occurs when there is a cyclic buffer dependency (CBD) among switches in the data center network. The CBD is created when a dependent switch in a sequence of switches is waiting for the availability of buffers in other switches in the sequence before transmitting a packet. If the switches involved in the CBD are using PFC and are physically connected in a loop, a PFC deadlock can occur. RDMA flows in a Clos data center network are distributed across multiple equal cost paths to achieve the highest possible throughput and lowest latency. While there are no loops in the logical topology, these paths naturally contain loops in the physical topology. A PFC deadlock in the network can completely halt network traffic.

Consider the example in Figure 21. The figure shows four phases of PFC deadlock creation. In phase 1, four flows are equally load balanced across the Clos fabric and the network is running smoothly. In phase 2, the red cross indicates a transient or permanent fault in the topology, such as link failure, port failure, or route failure. Due to the failure, in the example, traffic between H1 and H7 (green and yellow lines) is re-routed. The re-routing pushes more traffic through leaves 2 and 3 causing a potential overflow in spine 1 and spine 2 as shown in phase 2. In the example we assume the pressure on spine 1 occurs first. To avoid loss, the spine 1 switch issues PFC towards leaf 3, shown in phase 3. Traffic in leaf 3 now backs up, causing further backups around the topology and a cascade of PFC messages along the loop backward towards the original point of congestion. Phase 4 shows the resulting PFC deadlock.

When the network size is small, the probability of PFC deadlock is low. However, at larger scale and with the high-performance requirements of the RoCE protocol, the probability of PFC deadlock increases significantly. Achieving larger scale and optimal performance is a key objective of the intelligent lossless data center network of the future. Section 5 discusses a possible new technology for PFC deadlock prevention.

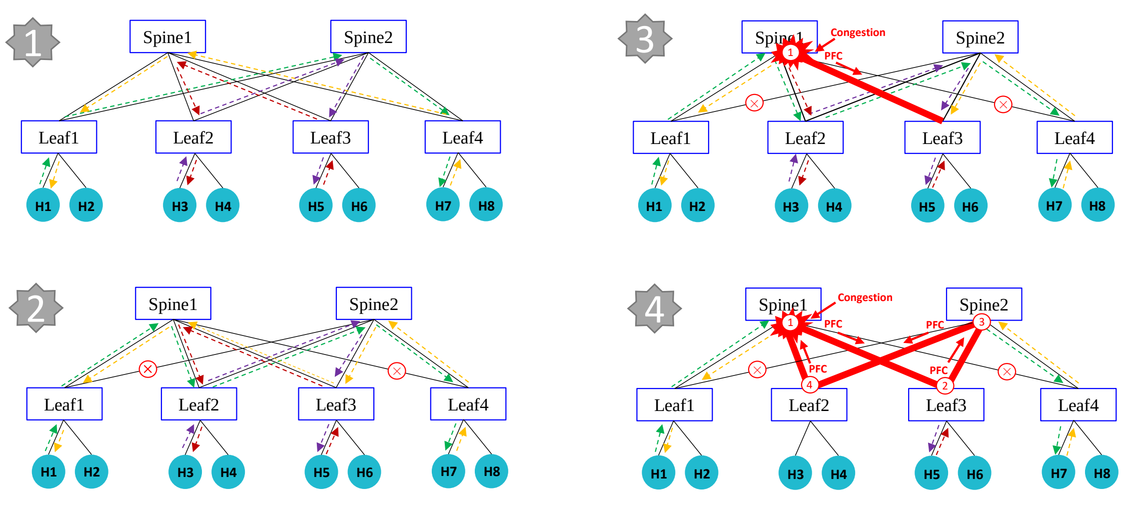


Figure 21 – Example PFC Deadlock

Congestion control issues in large-scale data center networks

RDMA technology was initially used by customers in constrained, conservative, small scale environments such as high-performance cluster computing or targeted storage networks. Tuning the resources required for the dedicated environment was manageable by the network operator, at least to some degree. However, the performance advantages of RDMA have proven useful in many application environments and there is a strong desire to use RDMA in a large-scale. Figure 22 shows an example of a large-scale RoCE network. In the example, the entire data center network is based on Ethernet. The computing cluster and storage cluster use the RDMA protocol while the X86 server cluster uses traditional TCP/IP.

In the large-scale data center network scenario TCP and RoCE traffic can traverse common parts of the network for several different reasons. Traditional web-based applications using high-speed storage backends mix end-user TCP requests with RDMA storage requests to read and write data. The management and software-defined control plane of RDMA devices is typically based on TCP while using RoCE for data communications. AI/ML applications use RoCE to interconnect GPUs and CPUs, but still may be using TCP-based storage solutions. This leads to multiple combinations of TCP and RoCE between computing-and-computing, storage-and-storage, and computing-and-storage systems.

In theory, separating TCP and RoCE traffic within the network should be easy. IEEE Std 802.1Q defines 8 classes of service that can map to 8 queues with differing queue scheduling algorithms. Different switch queues can be used to isolate the different traffic types. While the queues and the buffer management are implemented in hardware on the switch chip, there is a performance and cost tradeoff problem with the memory. Allocating sufficient dedicated memory to each queue on each port to absorb microbursts of traffic without incurring packet loss can be too expensive and technically challenging as the number of ports per switch chip goes up. To address this tradeoff, switch chip vendors implement a smart buffering mechanism that allows for a hybrid of fixed and shared buffers.

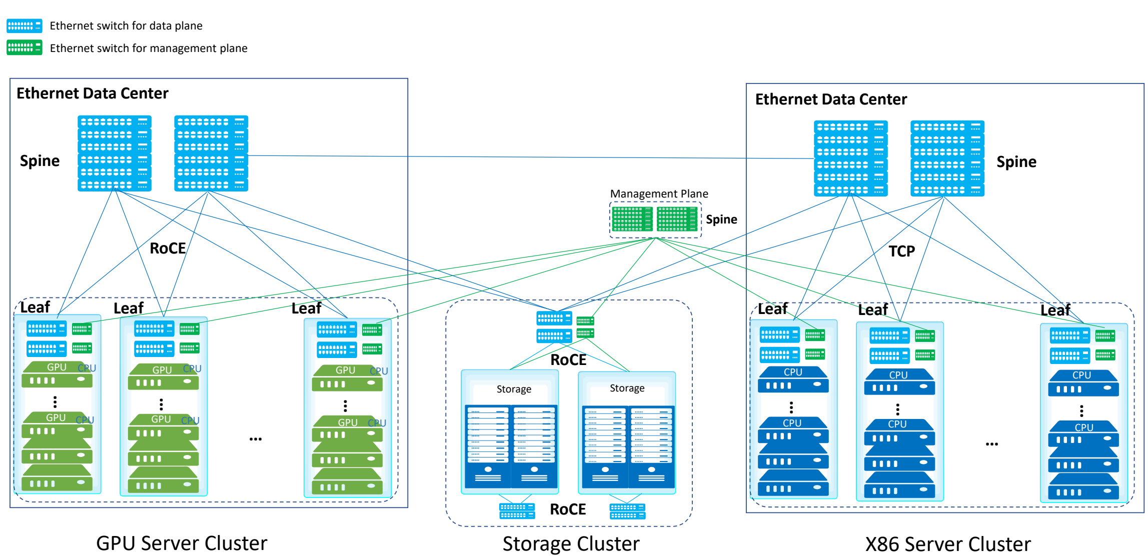


Figure 22 – RoCE application in large-scale data center networks

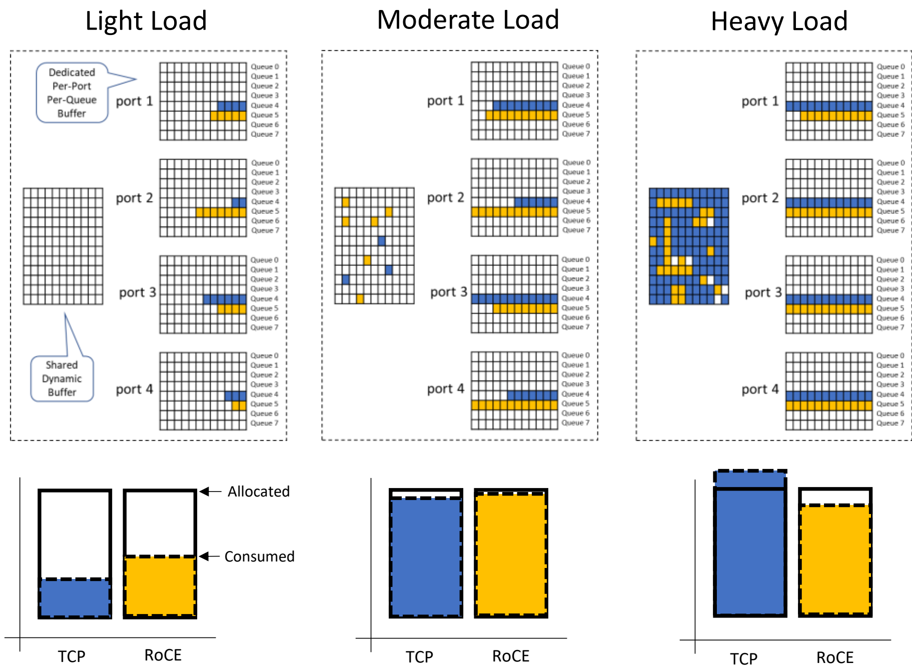


Figure 23 – TCP and RoCE coexistence with smart buffering.

A core idea of smart buffering is the creation of a dynamic shared buffer. The goal is to optimize buffer utilization and burst absorption by reducing the amount of statically dedicated buffers while providing a dynamic and self-tuning shared pool across all ports to handle temporary bursts [25].

An example smart buffer architecture, as shown in Figure 23. Each port has some dedicated buffers for each of its queues and a dynamic pool of centralized surplus buffers. The approach considers that congestion in a typical data center environment is localized to a subset of egress ports at any given point in time and rarely occurs on all ports simultaneously. This assumption allows the centralized on-chip buffer to "right-size" for overall cost and power consumption while still providing resources for congested ports exactly when needed using self-tuning thresholds.

Contrasted with static per-port buffer allocation schemes found in other switch architectures, the smart buffer approach significantly improves buffer utilization and enables better performance for data center applications. However, the shared dynamic pool has consequences on traffic class isolation in congested situations. TCP and RoCE flows may impact one another when they traverse common links, even if they are using separate traffic classes on those links. TCP and RoCE use different congestion control mechanisms, different re-transmission strategies and different traffic class configuration (lossless verse lossy). The algorithms and configurations can lead to the unfair sharing of the common resource. Figure 23 shows the problem when the switch is under heavy load. Network operators allocate the network bandwidth to different traffic classes based on the service requirements of the network. But over time and during periods of congestion the bandwidth allocations cannot be met. The different congestion control methods create different traffic behavior that impacts the smart buffering mechanism's ability to fairly allocated the dynamic shared buffer pool. In this case, TCP preempts RoCE bandwidth, even when it is allocated to separate traffic classes. The RoCE flow completion delay has been seen to increase by 100 times. ODCC conducted several tests to verify the problem of traffic coexistence. The results from testing are available at [26].

Configuration complexity of congestion control algorithms

Historically, HPC data center networks were small in scale and optimized through manual configuration. However, a goal of the Intelligent Lossless Data Center Network is to enable HPC and AI data centers to grow to cloud scale and be provisioned through automation. Manual configuration and hand tuning parameters are not possible at cloud scale, but the proper operation of the HPC data center requires network wide consistent configuration of several attributes. Some of the key attributes include:

- Consistent mapping of network priorities to switch traffic classes (i.e. switch queues).
- Consistent assignment of application traffic to network priorities.
- Consistent enablement of PFC on lossless traffic classes.
- Bandwidth allocations for traffic classes using Enhanced Transmission Scheduling (ETS).
- Buffer threshold settings for PFC and ensuring there is enough headroom to avoid loss.
- Buffer threshold settings for ECN marking.

The IEEE 802.1 Working Group defined the Data Center Bridging eXchange protocol (DCBX) to automate the discovery, configuration, and misconfiguration detection of many of the data center network configuration attributes. DCBX leverages the Link Layer Discovery Protocol (LLDP) to exchange a subset of configuration attributes with a network peer, and if the peer is 'willing' to accept recommended settings, the two peers can create a consistent configuration. This consistent configuration can propagate across the entire data center network if all devices are running DCBX. The protocol, however, does not exchange all key attributes for a data center network. In particular, it does not enable the automatic setting of buffer thresholds, which can be quite complex to determine and critical to the proper operation of the network.

The PFC buffer threshold determines when PAUSE frames are sent as seen in Figure 24. If the receiver's buffer filles past the XOFF threshold, the receiver sends a PAUSE frame. When the buffer drains and empties below the XON threshold, the receiver may send an UN-PAUSE frame canceling the previous pause or it may simply timeout. The XOFF threshold must be set in such a way to allow in-flight frames to be received. The buffer memory available beyond the XOFF threshold is often

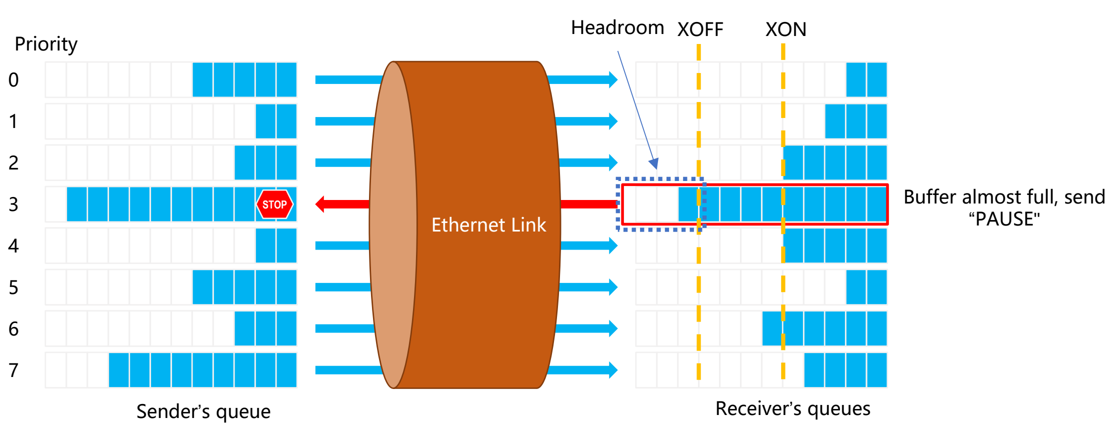


Figure 24 – Priority-based Flow Control (PFC)

called headroom and must be available to ensure lossless operation. Finding the best XON/XOFF thresholds can be tricky. Overestimating the threshold is not practical because it wastes precious switch memory and reduces the number of lossless traffic classes that can be supported. Underestimating the threshold leads to packet loss and poor performance for protocols such as RoCE. Finding the optimal setting is difficult because it requires the complex calculation of many obscure parameters [27]. Some of these obscure parameters include:

- Maximum frame size on the network
- Speed of the link
- The length of the cable
- Internal switch and transceiver latency
- Response time of sender
- Internal memory cell size of the receiver's buffer architecture

Clearly these parameters are not something a network operator can easily obtain. Many are internal to the switch implementation and will differ from vendor to vendor. In addition, the propagation delay, which includes the product of the link speed and cable length, can vary on every port of the network. With thousands of ports to configure a network operator will benefit from an automated solution to configuring PFC headroom.

The threshold for marking Explicit Congestion Notification (ECN) bits in congested packets is another important configuration setting for the smooth operation of the network. As mentioned above in the tradeoff between high throughput and low latency, setting the ECN threshold low helps achieve low latency, but at the cost of high throughput for larger flows. Figure 18 shows that setting a high ECN threshold has better performance for throughput-sensitive large traffic but slows down flow completion time for latency-sensitive smaller flows. As workloads change within the data center network an ideal solution is to dynamically adjust the ECN threshold to balance the tradeoff between high throughput and low latency.

The congestion control algorithms enabled by ECN involve collaboration between network adapters and network switches. The ECN thresholds in switches and rate reduction and response parameters on NICs and protocol stacks on end stations need to be coordinated as the workload changes. This coordination can result in an untenable set of configuration parameters that need to be updated frequently. Many network operators only use a recommended static configuration based on the experience of engineers over time. However, the static configuration does not adapt to real-time changes in network traffic that are driven by measurable fluctuations in an application's I/O and communication profile. Different static settings can result in different service performance for the same application and using the same settings for different applications can result in sub-optimal performance for the aggregate of applications on the data center network. Measuring the characteristics of network traffic for the set of the application I/O and communication profiles can lead to a predictive algorithm that dynamically adjusts the ECN threshold in switches and rate reduction and response parameters at the end-station to optimize the balance between high throughput and low latency.

5 New technologies to address new data center problems

Hybrid transports for low latency and high throughput

Traditional data center transport protocols, such as DCTCP [28] and RoCEv2 with DCQCN [21] are sender driven. They attempt to measure and match the instantaneous bandwidth available along the path by pushing data into the channel and awaiting feedback or measurements from the receiver. They continue to push more and more data into the channel until congestion is experienced, at which point they reduce their sending rate to avoid packet loss. There can be many methods of determining when congestion is experienced and how to adjust the sending rate, but the basic premise of sender driven transports is the same – continue to adjust the sending rate up or down based upon an estimation of the available channel bandwidth. This is a very well-known and mature approach to transport congestion control that has been shown to be successful in highly diverse networks such as the Internet. Accurately estimating of the available bandwidth depends, not only, on detecting congestion, but on creating it. Congestion signal delays and untimely adjustments to the sending rate can cause fluctuations to queue depths, leading to variance in throughput and latency. Large buffers in routers and switches can absorb these fluctuations to avoid packet loss.

A receiver driven transport, such as ExpressPass [29], can be used to avoid fluctuations in queue depths and minimize buffering along the path from sender to receiver. With receiver driven transports, the sender's transmissions are paced by the receiver's schedule. A request-grant or credit-based protocol is used to pace the sender and avoid congestion while fully utilizing network bandwidth. The approach is especially good at handling incast congestion where the receiver is overrun by multiple simultaneous senders. The challenge with receiver driven transports is that the receiver must now estimate the available bandwidth along the path. Similar techniques for congestion detection can be used and the receiver driven approach as the advantage of receiving

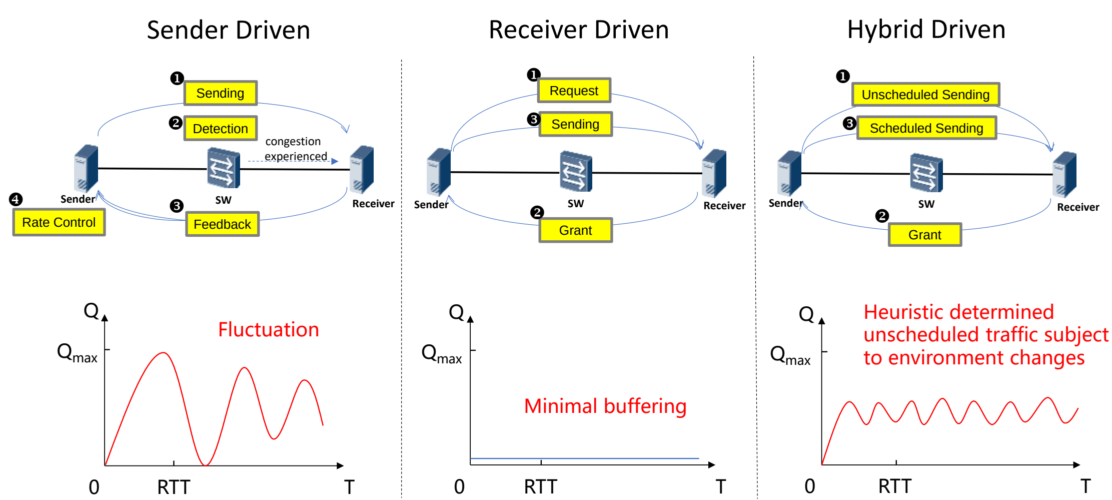


Figure 25 – Transport styles with conceptual network buffering implications

those congestion signals first. Perhaps a more significant challenge with receiver driven transports is the inherent delay built into the initial buffer request by the sender. The initial request-grant exchange penalizes small flows which, in most cases, are latency sensitive and constitute the majority of flows in the data center network.

A hybrid driven transport, such as NDP [30] or Homa [31], attempts to borrow the best qualities from sender driven and receiver driven transports to reduce latency and increase throughput by avoiding congestion. A hybrid approach allows the sender to transmit a certain amount of unscheduled traffic into the network without waiting for a buffer grant by the receiver, but then it must transition to a scheduled receiver driven approach after the unscheduled traffic is sent. The unscheduled traffic has no additional latency penalties and benefits small flows but can create minor fluctuations in buffer occupancy which can lead to moderate packet loss. Since the amount of unscheduled traffic is small, the overall buffer occupancy remains low which leads to more bounded latency and low packet loss. Adjusting the amount of unscheduled traffic based on heuristics helps tune the network for high throughput and low latency while maintaining low buffer utilization. Figure 25 shows the high-level approach to each of the different transport types and a conceptual graph of buffer utilization over time.

PFC deadlock prevention using topology recognition

Traffic on a well-balanced Clos networks is loop free and typically flows from uplink to downlink on ingress and downlink to uplink on egress. However, rerouting occurs when transient link faults are detected, and traffic may be generated from uplink to uplink as shown in Figure 21. According to [24], the probability of rerouted traffic is approximately 10^{-5}. While 10^{-5} is not a high probability, given the large traffic volume and the large scale of data center networks the chance of a deadlock occurring is possible and even the slightest probability of a deadlock can have dramatic consequences. PFC deadlocks are real! The larger the scale, the higher the probability of PFC deadlock, and the lower the service availability from this critical resource.

ODCC proposes a mechanism to prevent the PFC deadlock problem by discovering and avoiding CBD loops. The core idea of the deadlock-free algorithm is to break the circular dependency by identifying traffic flows that create it. The first step in achieving this is to discover the topology and understand the port orientation of every switch port in the network. An innovative distributed topology and role auto-discovery protocol is used to identify network locations and roles of across the data center network.

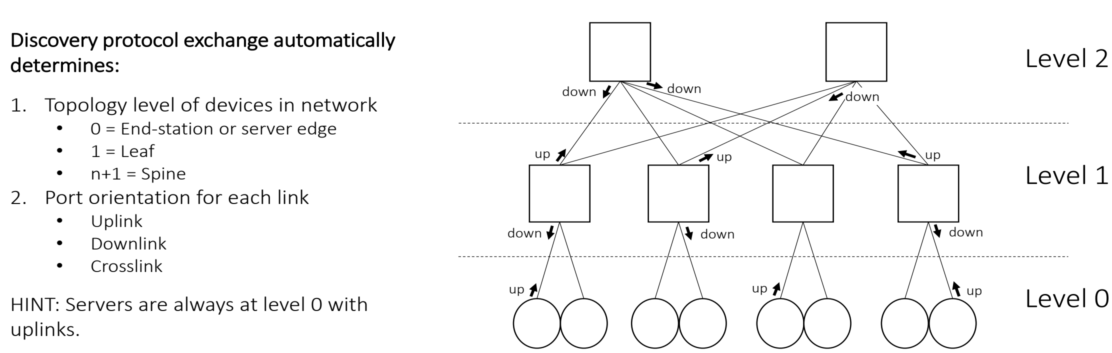


Figure 26 – Topology and Role Discovery

The topology and role discovery protocol automatically determines a device's level within the topology and the orientation of each of the device's ports. The level within the topology is defined as the number of hops from the edge of the network. For example, a server or storage endpoint is at level 0 and the top-of-rack switch connected to that server or storage endpoint is at level 1. The port orientation of a port can be either an uplink, downlink or a crosslink. An uplink orientation, for example, is determined for a port of a device that is connected to another device at a higher level.

The protocol starts out by recognizing known conditions. Servers and storage endpoints are always at level 0 and their port orientation is always an uplink. Switches are initialized without any knowledge of their level or port orientation, but as the information is propagated by a discovery protocol, the algorithm converges upon an accurate view. Figure 26 shows the resulting topology and role discovery in a simple Clos network.

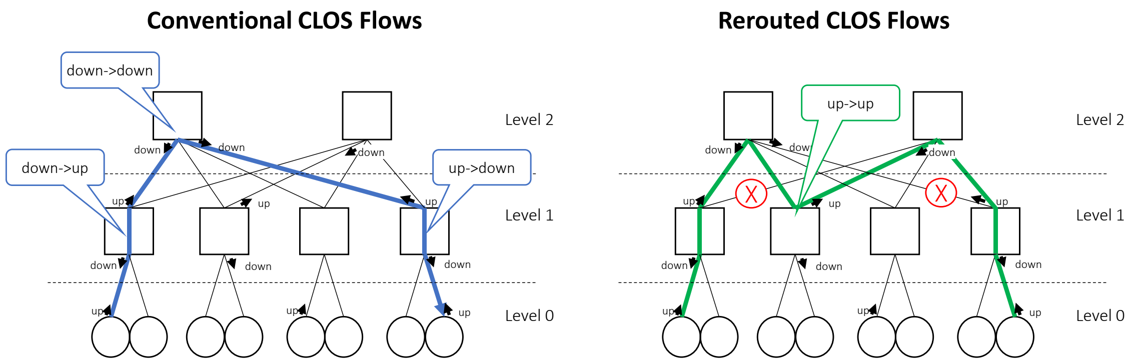


Figure 27 – Identifying CBD points in rerouted flows.

Once the protocol has recognized the topology and port roles, the deadlock free mechanism can identify potential CBD points in the network and then adapt the forwarding plane to break the buffer dependencies. Figure 27 shows how potential CBD points in the topology can be recognized. In a properly operating Clos network, there is no CBD and flows will typically traverse a switch ingress and egress port pair that has three of four possible port orientation combinations. The flow may pass from a port oriented as a downlink to a port oriented as an uplink. In the spine of the network, the flow may pass from a port oriented as a downlink to another port oriented as a downlink. Finally, as the flow reaches its destination, the flow may pass from a port oriented as an uplink to a port oriented as a downlink. A CBD may exist in the case where a flow has been rerouted and now passes from a port oriented as an uplink to another port oriented as an uplink.

After recognizing the CBD point, the forwarding plane is responsible for breaking the CBD. The CBD exists because a set of flows are using the same traffic class and are traversing a series of switches that now form a loop due to the flow rerouting. The buffer dependency is the shared buffer memory of the common traffic class (i.e. switch queue). To break the CBD, packets of the rerouted flow need to be forwarded to a separate queue. These packets can be identified because they are flowing from a port oriented as an uplink to another port oriented as an uplink. Figure 28 illustrates the process of queue remapping within the switch. In the example, the remapping of the green flow to an isolated queue will lead the elimination of PFC deadlock. The different flows can safely pass-through different queues at the point of a potential CBD.

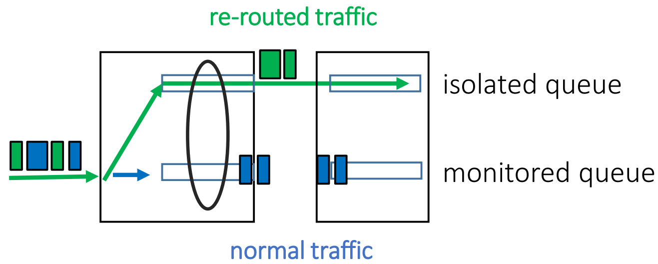


Figure 28 – Queue switch according to CBD reroute flow recognition.

ODCC, in participation with many network vendors, conducted tests to verify the deadlock free algorithm. The results indicate the effectiveness of the approach [26].

Improving Congestion Notification

A state-of-the-art congestion control mechanism for the RoCEv2 protocols in today's data centers is Data Center Quantized Congestion Notification (DCQCN) [21]. DCQCN combines the use of ECN and PFC to enable a large-scale lossless data center network. Figure 29 shows the three key components of DCQCN; a reaction point (RP), a congestion point (CP) and a notification point (NP).

Reaction Point (RP)

The RP is responsible for regulating the injection rate of packets into the network. It is typically implemented on the sending NIC and responds to Congestion Notification Packets (CNP) sent by the NP when congestion is detected within the network. When a CNP is received, the RP will decrease the current rate of injection. If the RP does not receive a CNP within a specified period, it will increase the transmit rate using a quantized algorithm specified by DCQCN.

Congestion Point (CP)

A CP is included in the switches along the path between the transmitter and the receiver. The CP is responsible for marking packets with ECN when congestion is detected at an egress queue. Congestion is determined by looking at the egress queue length and evaluating it against configurable thresholds (K_{min} and K_{max}). When the queue length is less than K_{min}, traffic is not marked. When the queue length is greater than K_{max}, all packets passing through the queue are marked. When the queue length is between K_{min} and K_{max}, the marking probability increases according to the extent of the queue length, as specified by DCQCN.

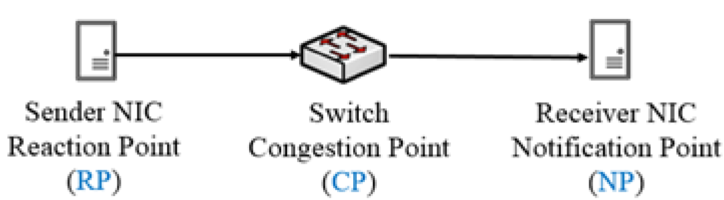


Figure 29 – Three parts of RoCE congestion control using DCQCN

Notification Point (NP)

The NP is responsible for informing the RP that congestion has been experienced by packets of a flow while traversing the network. When a data packet with an ECN flag arrives at a receiver, the NP sends a CNP packet back to the RP at the transmitter if one has not already been sent in the past N microseconds. It is possible to set N to 0 such that the NP will send a CNP for each packet with an ECN flag set.

As data center networks scale to larger sizes and support an increased number of simultaneous flows, the average bandwidth allocated to each flow can become small. Flows experiencing congestion in this environment may have their packets delayed, causing the arrival of ECN markings at the NP to also be delayed. If the rate of arrival of ECN marked packets is greater than the interval the RP uses to increase the rate of injection a problem may occur. The problem is that the RP will begin increasing the rate of injection when it should decrease the rate since the flow is congested and the missing CNP messages have simply been delayed. In this case, the end-to-end congestion control loop is not functioning correctly.

For example, if the link speed of the switch is 25 Gbps and the number of RoCE flows is 300, the average rate of each RoCE flow is 80 Mbps. In this case, a 4 KB message is generated every 400 µs. If the RP waits less than 400 µs to receive a CNP before increasing the rate of transmission a congestion control loop failure will occur. The default time an RP will wait for a CNP before increasing transmission rate is often 300 µs in commercial NICs. This implies that network operators need to tune individual timer settings to support large scale deployments.

The impact of end-to-end congestion control loop failure in a lossless network is further congestion. This congestion causes an increase in the number of PFC packets generated and an increase in the amount of time links are paused to avoid packet loss. These PFC packets further delay the propagation of ECN marked packets and only make the problem worse. The combination of PFC and ECN becomes ineffective.

One possible solution to this problem is for the network to intelligently supplement the CNP packets sent by the NP. The intelligence involves considering the congestion level at the egress port, the interval of the received ECN marked packets, and the interval of the DCQCN rate increase by the RP. After receiving an ECN marked packet, the CP keeps track of the frequency of received ECN marked packets as well as the sequence number. When the CP egress queue is congested and the received flow has been experiencing congestion further upstream, the CP may proactively supplement the CNP depending upon the rate of received ECN marked packets and the interval of the DCQCN rate increase at the RP. The CP is aware that ECN marked packets are delay and that subsequent CNP packets from the NP will be further delayed, so the supplemental CNP messages will prevent the end-to-end congestion control loop failure. The supplemental CNP operation is performed only when the CP egress queue is severely congested, thus latency and throughput are not affected when the DCQCN is operating in a normal non-congested state. The solution is shown in Figure 30.

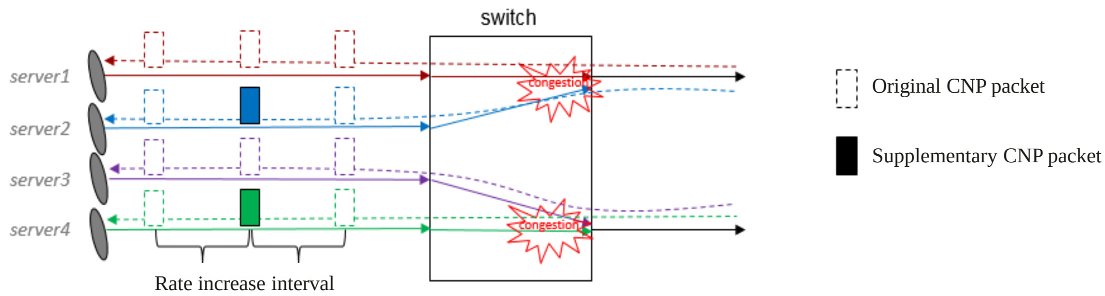


Figure 30 – Intelligent Supplemental CNP

The ODCC tested the enhanced congestion control mechanism and the effect is beneficial [32]. According to the test result, the bandwidth QoS performance is improved by more than 30% (TCP:RoCE = 9:1 scenario).

Addressing configuration complexity of congestion control algorithms

With thousands of switches and tens of thousands of ports to configure, network operators need automated solutions to properly configure the parameters responsible for managing congestion control in the data center network. The Data Center Exchange Protocol (DCBX) defined by IEEE 802.1 made great strides in simplifying some of the configuration and error detection, however, more is needed. Automated solutions for setting and adapting switch buffer thresholds are needed.

Buffer optimization to reduce the complexity of PFC headroom configuration

The key to successful PFC XOFF threshold setting is assuring there is enough headroom to absorb the in-flight data once the PAUSE frame has been issued. There is a natural delay between the time when the PAUSE frame can be sent and the sender actually stops transmitting data. The headroom must receive data during this delay but the calculation for memory needed can be quite complex. Annex N of IEEE Std 802.1Q-2018 [20] provides the technical details of this calculation. Many of the components of the delay are internal to the switch implementations and remain relatively static. The interface delay and higher-layer delay do not vary for a particular configuration and implementation. These static components of delay can be communicated between peers on the network, but currently there is no standard protocol that allows this. The propagation delay for the medium is dependent upon the transmission speed and the length of the cable. To accurately obtain this component of delay a measurement is required.

The Time Sensitive Networking (TSN) Task Group of the IEEE 802.1 Working Group has defined IEEE Std 802.1AS-2020 Timing and Synchronization for Time-Sensitive Application [33]. A small subset of this specification, along with optional time-stamping support in IEEE Std 802.3 can be used to measure cable delay between two peers on a point-to-point link. IEEE Std 802.1AS, however, targets time-sensitive applications in constrained environments such as audio/visual, industrial, and automotive networks. Its primary focus is to enable a Precision Time Protocol (PTP) used to synchronize clocks throughout the computer network. While a fine-grained synchronized clock could be valuable in a data center, the burden for supporting the complete set of IEEE Std 802.1AS functions in data center switching silicon could be onerous. The delay measurement facilities of IEEE Std 802.1AS, on the other hand, are useful in the data center to assist in the auto-configuration of

PFC thresholds. Having the ability to discover and communicate this capability between peers, along with other DCBX attributes, would be necessary for full automation of the configuration settings.

Intelligent ECN threshold optimization

The ECN threshold determines how aggressively a switch will indicate that packets are experiencing congestion and subsequently how frequently the sending station may need to adjust transmission rate. The optimal threshold setting depends on the current state of the network and the types of communication flows that are competing for common resources. As previously discussed, a low threshold setting can benefit latency-sensitive smaller flows and a high threshold setting can have better performance for throughput-sensitive larger traffic flows. The mix of these flows and their communication patterns is constantly changing but have been shown to be predictable using machine learning techniques that model application traffic behavior [34] [35] [36]. A machine learning model that predicts data center network traffic patterns could be used to dynamically adjust ECN thresholds to optimize the trade-off between low-latency and high-throughput. The unfair sharing of the dynamic pool of memory in the smart buffering scheme can also be address by dynamically adjusting the ECN threshold differently for TCP and RoCE traffic.

To train a model of network traffic patterns in the data center an AI/ML system needs an abundance of real-time data from the network. The data acquisition system needs to capture the temporal relationships between network devices across the data center and at large scale. Traditional network monitoring systems based on SNMP and/or NetConf use polling to "pull" data from the devices. This approach has scaling issues, increases network traffic and is more difficult to correlate the collected data. What is needed is a telemetry stream of essential parameters directly from the network devices. Telemetry is a network monitoring technology developed to collect performance data quickly from physical or virtual devices. Telemetry differs from traditional network monitoring technologies as it enables network devices to "push" high-precision performance data to a data repository in real time and at high speeds. This improves the utilization of device and network resources during data collection.

With Telemetry technology, an AI/ML system can build a model that monitors the congestion status of all queues on the entire network. The stream of parameters can be used to train and retrain the network model, allowing inference engines on the network devices to predict changes in the data center environment and self-adjust their ECN threshold. Inputs to the model can extend well beyond the existing counters obtained by traditional network monitoring systems. Essential input parameters might include:

- A snapshot of the incast ratio (N:1) at an egress port
- The mix of mice and elephants flows at an ingress port
- The rate change in switch buffer occupancy

Other more traditional network metrics might include:

- Port-level information
    - Sent and received bytes
    - Sent and received packets
    - Discarded packets in the transmit and receive directions
    - Received unicast packets, multicast packets, and broadcast packets

    - Sent unicast packets, multicast packets, and broadcast packets
    - Sent and received error packets
    - Ingress port bandwidth usage and egress port bandwidth usage
    - ECN packets
- Queue-level information
    - Egress queue buffer utilization
    - Headroom buffer utilization
    - Received PFC frames
    - Sent PFC frames

Another type of telemetry, known as in-band telemetry, provides real-time information about an individual packet's experience as it traverses the network. The information is collected and embedded in the packets by the data plane without involving the control plane. The amount of information collected is more limited than a traditional telemetry stream because it must be included within the data packet. However, the information within the packet is directly related to the network state that the packet observed during its existence within the network. Each hop along the path can be instructed to insert local data representing the switch hop's state. Essential information might contain:

- Ingress and egress port numbers
- Local timestamps at ingress and egress
- Egress link utilization
- Egress queue buffer utilization

An AI model that takes real-time telemetry input from the local device can predict the adjustments needed to the ECN threshold for the desired balance between low-latency and high-throughput. The in-band telemetry signals can be examined with an objective of rapidly communicating appropriate congestion signals to the sending sources to avoid packet loss and long tail latency with flow completion times.

6 Standardization Considerations

Two important standards development organizations for the future technologies discussed above 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 widely used and relevant standards to this report are for Ethernet, Bridging, Virtual Bridged LANs and Time Sensitive Networking. The IEEE 802.1 Working Group provides the focus for Bridging, Virtual Bridged LANs and Time Sensitive Networking.

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 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). A parallel organization to the IETF is the Internet Research Task Force (IRTF) which focuses on longer term research issues related to the Internet. The IRTF is comprised of several focused and long-term Research Groups, of which the most relevant for this report are the Internet Congestion Control Research Group (iccrg) and the Computing in the Network Research Group (coinrg).

The IEEE 802 and IETF/IRTF 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 place [37]. 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.

Transport protocols is typically the domain of the IETF, however providing signals from the network could be provided by specifications from IEEE 802.1. A new hybrid transport that optimized the tradeoff between low latency and high throughput could likely be investigated in the IRTF's Internet Congestion Control Research Group (iccrg). A proposed standard from this research would most likley be developed by the IETF's Transport Area (tsv). The key to success for the hybrid transport is knowning how to best estimate the amount of unscheduled traffic to allow and how to rate control the senders of an incast scenario. Congestion signals and resource status along the communication path could be provided by the network switches themselves. In-band telemetry or enhanced ECN signaling by the network switches could provide the needed information and represents an opportunity for specificaiton by the IEEE 802.1 Working Group.

PFC deadlock prevention requires an awareness of the network topology and an ability to break a CBD caused by re-routed flows. P802.1Qcz Congestion Isolation has specified a mechanism using LLDP to automatically recognize the level of a switch within the topology as well as the orientation of each port (e.g. uplink, downlink, crosslink). A missing specification is how to recognize flows that are at risk of creating a CBD and how mitigate the CBD. The mechanism specified by P802.1Qcz used to adjust the priority of a congesting flow could be used to adjust the priority of a flow at risk of creating a CBD. Further specificion of how to use these mechanisms for PFC deadlock prevention could be done by the IEEE 802.1 WG.

The network supplemented CNPs discussed above augments the DCQCN protocol which currently has no formal specification. Since DCQCN works in conjunction with RoCEv2, the Infiniband Trade Association (IBTA) would be the natural standards organization to complete these enhancements. The general idea of network supplemented CNPs could also be applied to a new IETF hybrid transport protocol and most likely would be investigated by ICCRG and TSVWG. A third alternative is to consider updating the mechanism defined by IEEE Std 802.1Q-2018 in Clause 30 through Clause 32 – Congestion Notification. To make Congestion Notification relevant to today's modern data centers the Congestion Notification Messages (CNM) would need to be Layer-3 and routable.

Automatically setting the PFC XON/XOFF thresholds requires an accurate measurement of the delays between two ends of a link in the data center. An adaptive PFC headroom algorithm could be defined by the IEEE 802.1 Working Group using or augmenting the facilities already defined by IEEE 802.3 for timestamping and IEEE 802.1 TSN for path delay measurements. A solution targetted for the needs of a data center is needed to reduce the overhead of configuring lossless modes of operation and eliminating the chance for configuration errors. A mechanism for communicating this capability between peers and an update to the current specification of how to manually calculate headroom are execellent candidates for an amendment to IEEE Std 802.1Q.

Adjusting the ECN threshold automatically is dependent on recognizing and prediciting the current congestive state of the data center network. A rapid response to changing congestion status is needed and traditional network management approaches can not react quickly enough. Network devices that are armed with an AI model to assist in this prediction rely on the model being well trained from an accurate set of real-time data. Network telemetry can provide a new view for the state of the network, whether that telemetry data is in-band or streamed from the network devices themselves. Standards for telemetry at Layer-3 and above have historically been specified by the IETF. Currently the IP Performance Measurement (ippm) group within the IETF TSV area is defining In-situ Operations, Administration, and Maintenance (IOAM) [38] that provides in-band telemetry at Layer-3. There are also other related and competing specifications for Layer-3 in-band telemetry [39] [40]. In some environments a Layer-2 solution working in conjunction with Layer-3 may be more appropriate and require standard specifications to support interoperatbility. This in-band telemetry could be defined by the IEEE 802.1 Working Group. The Operations and Management Area Working Group (opsawg) in the IETF Operations and Management area (ops) is working on a framework for Network Telemetry [41]. This framework is looking at various techniques for remote data collection, correlation, and consumption. For the framework to be successful it is necessary to specify the information that can be extracted from the network. Supporting specifications by Layer-2 devices will be needed from the IEEE 802 and specifications for Layer-3 and above devices will be needed by the IETF.

7 Conclusion

Data center networks must continue to scale and innovate with new technologies to keep pace with the evolving needs of high-speed computing and storage used for AI and Machine Learning applications. This paper expanded upon the previous report [2] with the exploration of technical challenges and potential new solutions for today’s cloud scale high performance computing data centers. We discussed new hybrid transport protocols that better balance the needs of both high throughput and low latency communications for AI and Machine Learning. We described a solution for PFC deadlock prevention using a topology recognition extension to the existing and widely deployed Link Layer Discovery Protocol (LLDP). We explored ways to reduce the feedback cycle for congestion notification messages by allows the switches to supplement congestion signaling. We also described approaches to reduce the complexity of switch buffer threshold configuration using automated protocols and artificial intelligence models developed from advance telemetry systems. Together these innovations, with the commitment to openness and standardization, can advance the use of Ethernet as the premier network fabric for modern cloud scale high performance data centers.

8 Citations

[1] IEEE, "Nendica Work Item: Data Center Networks," [Online]. Available: https://1.ieee802.org/nendica-DCN/. [Accessed 14 05 2020].

[2] IEEE, "IEEE 802 Nendica Report: The Lossless Network for Data Centers," 17 8 2018. [Online]. Available: https://xploreqa.ieee.org/servlet/opac?punumber=8462817. [Accessed 13 05 2020].

[3] Orange, "Finding the competitive edge with digital transformation," 03 June 2015. [Online]. Available: https://www.orange-business.com/en/magazine/finding-the-competitive-edge-with-digital-transformation. [Accessed 1 09 2020].

[4] J. Wiles, "Mobilize Every Function in the Organization for Digitalization," Gartner, 03 December 2018. [Online]. Available: https://www.gartner.com/smarterwithgartner/mobilize-every-function-in-the-organization-for-digitalization/. [Accessed 10 June 2020].

[5] Huawei, "Huawei Predicts 10 Megatrends for 2025," Huawei, 08 August 2019. [Online]. Available: https://www.huawei.com/en/press-events/news/2019/8/huawei-predicts-10-megatrends-2025. [Accessed 10 June 2020].

[6] J. Handy and T. Coughlin, "Survey: Users Share Their Storage," 12 2014. [Online]. Available: https://www.snia.org/sites/default/files/SNIA%20IOPS%20Survey%20White%20Paper.pdf . [Accessed 14 05 2020].

[7] Huawei, "AI, This Is the Intelligent and Lossless Data Center Network You Want!," 13 March 2019. [Online]. Available: https://www.cio.com/article/3347337/ai-this-is-the-intelligent-and-lossless-data-center-network-you-want.html. [Accessed 14 05 2020].

[8] E. K. Karuppiah, "Real World Problem Simplification Using Deep Learning / AI," 2 November 2017. [Online]. Available: https://www.fujitsu.com/sg/Images/8.3.2%20FAC2017Track3_EttikanKaruppiah_RealWorldProblemSimplificationUsingDeepLearningAI%20.pdf. [Accessed 14 05 2020].

[9] O. Cardona, "Towards Hyperscale High Performance Computing with RDMA," 12 June 2019. [Online]. Available:

https://pc.nanog.org/static/published/meetings/NANOG76/1999/20190612_Cardona_Towards_Hyperscale_High_v1.pdf. [Accessed 14 05 2020].

[10] J. L. Jacobi, "NVMe SSDs: Everything you need to know about this insanely fast storage," 10 March 2019. [Online]. Available: https://www.pcworld.com/article/2899351/everything-you-need-to-know-about-nvme.html. [Accessed 14 05 2020].

[11] M. Alipio, N. M. Tiglao, F. Bokhari and S. Khalid, "TCP incast solutions in data center networks: A classification and survey," *Journal of Network and Computer Applications,* vol. 146, p. 102421, 2019.

[12] T. P. Morgan, "Machine Learning Gets An Infiniband Boost With Caffe2," 19 April 2017. [Online]. Available: https://www.nextplatform.com/2017/04/19/machine-learning-gets-infiniband-boost-caffe2/. [Accessed 14 05 2020].

[13] Z. Jai, Y. Kwon, G. Shipman, P. McCormick, M. Erez and A. Aiken, "A distributed multi-GPU system for fast graph processing," in *VLDB Endownment*, 2017.

[14] Wikipedia, "IEEE 802.3," 5 June 2020. [Online]. Available: https://en.wikipedia.org/wiki/IEEE_802.3. [Accessed 22 July 2020].

[15] K. Rupp, "42 Years of Microprocessor Trend Data," February 2018. [Online]. Available: https://www.karlrupp.net/2018/02/42-years-of-microprocessor-trend-data/. [Accessed 22 July 2020].

[16] The Linux Foundation, "Open vSwitch," 2016. [Online]. Available: https://www.openvswitch.org/. [Accessed 23 July 2020].

[17] Y. Li, R. Miao, H. H. Liu, Y. Zhuang, F. Feng, L. Tang, Z. Cao, M. Zhang, F. Kelly, M. Alizadeh and M. Yu, "HPCC: high precision congestion control," in *Proceedings of the ACM Special Interest Group on Data Communication (SIGCOMM '19)*, New York, NY, USA, 2019.

[18] P. Goyal, P. Shah, N. Sharma, M. Alizadeh and T. Anderson, "Backpressure Flow Control," in *Proceedings of the 2019 Workshop on Buffer Sizing (BS '19)*, New York, NY, USA, 2019.

[19] C. Guo, H. Wu, Z. Deng, G. Soni, J. Ye, J. Padhye and M. Lipshteyn, "RDMA over Commodity Ethernet at Scale," in *In Proceedings of the 2016 ACM SIGCOMM Conference (SIGCOMM '16)*, 2016.

[20] IEEE, IEEE Std 802.1Q-2018, IEEE Standard for Local and Metropolitan Area Networks — Bridges and Bridged Networks, IEEE Computer Society, 2018.

[21] Y. Zhu, H. Eran, D. Firestone, C. L. M. Guo, Y. Liron, J. Padhye, S. Raindel, M. H. Yahia and M. Zhang, "Congestion Control for Large-Scale RDMA Deployments," in *Proceedings of the 2015*

ACM Conference on Special Interest Group on Data Communication (SIGCOMM '15), London, United Kingdom, 2015.

[22] M. Karok, J. Golestani and D. Lee, "Prevention of deadlocks and livelocks in lossless backpressured packet networks," *IEEE/ACM Transactions on Networking,* vol. 11, no. 6, p. 11, 2003.

[23] S. Hu, Y. Zhu, P. Cheng, C. Guo, K. Yan, J. Padhye and K. Chen, "Deadlocks in datacenter networks: Why do they form, and how to avoid them," in *Proceedings of the 15th ACM Workshop on Hot Topics in Networks*, 2016.

[24] S. Hu, Y. Zhu, P. Cheng, C. Guo, K. Tan, J. Padhye and K. Chen, "Tagger: Practical PFC Deadlock Prevention in Data Center Networks," in *In Proceedings of the 13th International Conference on emerging Networking EXperiments and Technologies (CoNEXT ’17)*, 2017.

[25] S. Das and R. Sankar, "Broadcom Smart-Buffer Technology in Data Center Switches for Cost-Effective Performance Scaling of Cloud Applications," April 2012. [Online]. Available: https://docs.broadcom.com/docs-and-downloads/collateral/etp/SBT-ETP100.pdf. [Accessed 24 June 2020].

[26] ODCC, "ODCC lossless network test report (final draft)," 02 September 2020. [Online]. Available: http://www.odcc.org.cn/auth/v-1300974311558307841.html. [Accessed 03 09 2020].

[27] Cisco Systems, Inc, "Priority Flow Control: Build Reliable Layer 2 Infrastructure," 2009. [Online]. Available: https://www.cisco.com/c/en/us/products/collateral/switches/nexus-7000-series-switches/white_paper_c11-542809.pdf. [Accessed 15 12 2020].

[28] M. Alizadeh, A. Greenberg, D. Maltz, J. Padhye, P. Patel, B. Prabhakar, S. Sengupta and M. Sridharan, "Data center TCP (DCTCP)," in *ACM SIGCOMM 2010 conference (SIGCOMM '10)*, New York, 2010.

[29] I. Cho, K. Jang and D. Han, "Credit-Scheduled Delay-Bounded Congestion Control for Datacenters," in *Proceedings of the Conference of the ACM Special Interest Group on Data Communication (SIGCOMM '17)*, New York, 2017.

[30] 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.

[31] 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].

[32] ODCC, "Lossless Network Test Specifications," 03 September 2019. [Online]. Available: http://www.odcc.org.cn/download/p-1169553273830920194.html. [Accessed 01 09 2020].

[33] IEEE, IEEE Std 802.1AS-2020, IEEE Standard for Local and Metropolitan Area Networks — Timing and Synchronization for Time-Sensitive Applications, IEEE Computer Society, 2020.

[34] L. Nie, D. Jiang, L. Guo, S. Yu and H. Song, "Traffic Matrix Prediction and Estimation Based on Deep Learning for Data Center Networks," in *2016 IEEE Globecom Workshops (GC Wkshps)*, Washington, DC, 2016.

[35] X. Cao, Y. Zhong, Y. Zhou, J. Wang, C. Zhu and W. Zhang, "Interactive Temporal Recurrent Convolution Network for Traffic Prediction in Data Centers," *IEEE Access,* vol. 6, pp. 5276-5289, 2018.

[36] A. Mozo, B. Ordozgoiti and S. Gomez-Canaval, "Forecasting short-term data center network traffic load with convolutional neural networks," *PLoS ONE,* vol. 13(2), no. e0191939. https://doi.org/10.1371/journal.pone.0191939, 2018.

[37] IETF, "IEEE 802 and IETF Coordination Guide," 6 7 2017. [Online]. Available: https://trac.ietf.org/trac/iesg/wiki/IEEE802andIETFCoordinationGuide. [Accessed 1 2 2018].

[38] IETF, "Data Fields for In-situ OAM," 17 12 2020. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-ippm-ioam-data/. [Accessed 06 01 2021].

[39] Alibaba; Arista; Barefoot Networks; Dell; Intel; Marvell; Netronome; VMware, "In-band Network Telemetry (INT) Dataplane Specification," 20 04 2018. [Online]. Available: https://github.com/p4lang/p4-applications/blob/e5d0c4f4c9fe548e83ad91adbd38847c7dce6cfe/docs/INT.pdf. [Accessed 06 01 2021].

[40] J. Kumar, S. Anubolu, J. Lemon, R. Manur, H. Holbrook, A. Ghanwani, D. Cai, H. Ou, Y. Li and X. Wang, "Inband Flow Analyzer," 24 04 2020. [Online]. Available: https://tools.ietf.org/html/draft-kumar-ippm-ifa-02. [Accessed 06 01 2021].

[41] IETF, "Network Telemetry Framework," 17 12 2020. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-opsawg-ntf/. [Accessed 06 01 2021].

[42] Huawei, "Configuration Guide - Low Latency Network," [Online]. Available: https://support.huawei.com/enterprise/en/doc/EDOC1100040243/c28a82e4/buffer-optimization-of-lossless-queues. [Accessed 14 07 2020].

