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	Mellanox
Lui, Jun	Cisco
He, Zongying	Broadcom
Sun, Liyang	Huawei
Tang, Guangming	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

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

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

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

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

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

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

TABLE OF CONTENTS

1 Introduction

<<Editor's notes will be noted inside these marking and removed in future drafts>>

<<short intro and the more detailed background intro is section 2. This will be written near the end>>

This paper is the result of the Data Center Networks 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 includes...

Purpose

The purpose of this report is to ...

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. ~~Work flows~~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 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-

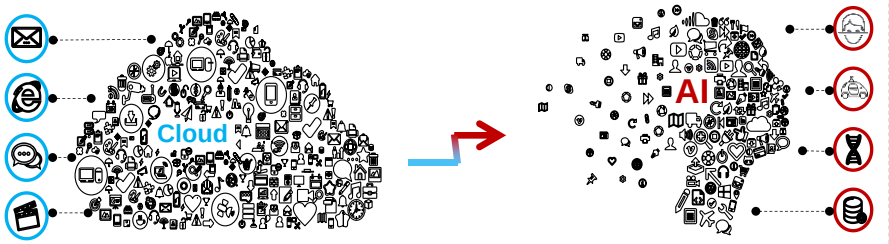


Figure 1 – Digital Transformation in the Era of AI

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 in 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 caused.

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

Currently, digital transformation of various industries is accelerating. According to analysis data, 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].

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

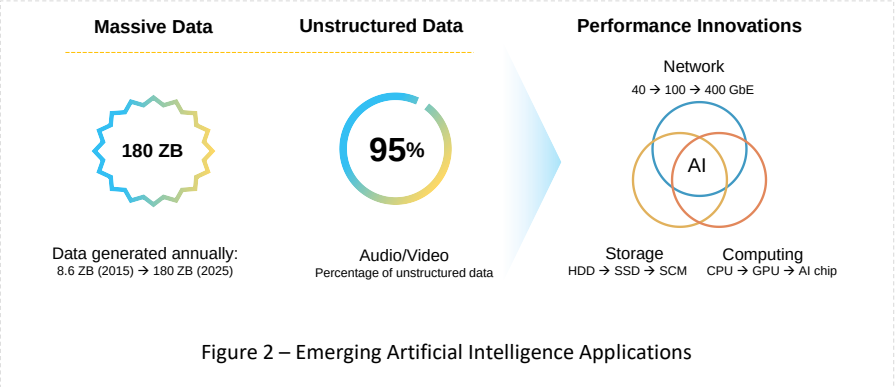


Figure 2 – Emerging Artificial Intelligence Applications

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

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 sample 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].

Storage media has evolved from HDDs to SSDs to meet real-time data access requirements, reducing the medium latency by more than 100 times. Without similar improvements in network latency, these storage improvements are not 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. 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 consequently must "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

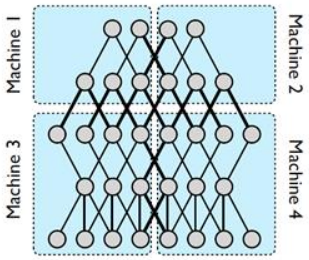


Figure 3 - Model parallel training

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.

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.

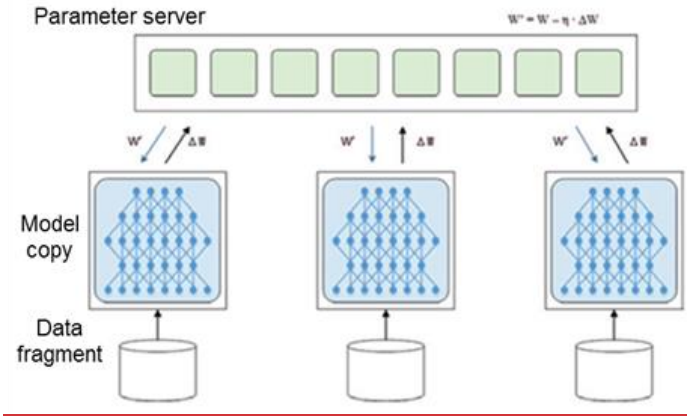


Figure 4 - Data parallel training

No matter the development of distributed storage or distributed AI training, data center network comes to the communication pressure. The waiting time for GPU communication exceeds 50% of the job completion time [9].

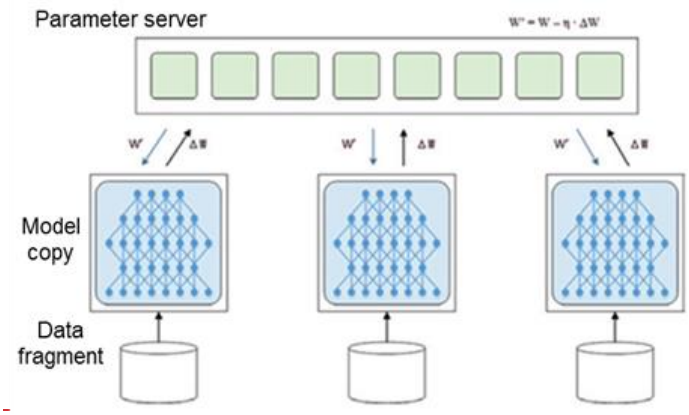


Figure 4 - Data parallel training

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 utility 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 will concurrently accessaccesses different parts of data from different servers, and the concurrently. The data will beis 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.

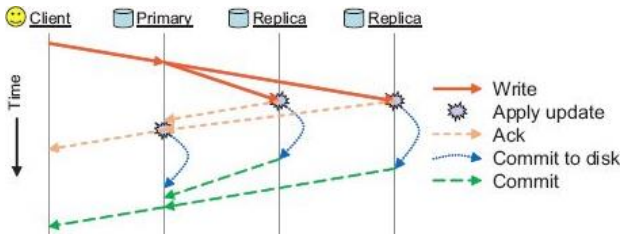


Figure 5 - Networked storage service model

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 SDD to NVMe (Non-Volatile Memory Express). The latest storage media technology, 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. 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 latency becomes more significant. Figure 6 shows how network latency has become the primary bottleneck with networked SSD storage~~, whereas~~. Once upon a time network latency was negligible with networked HDD storage. To maximize the IOPS performance of the new medium, the network latency problem must be resolved first.

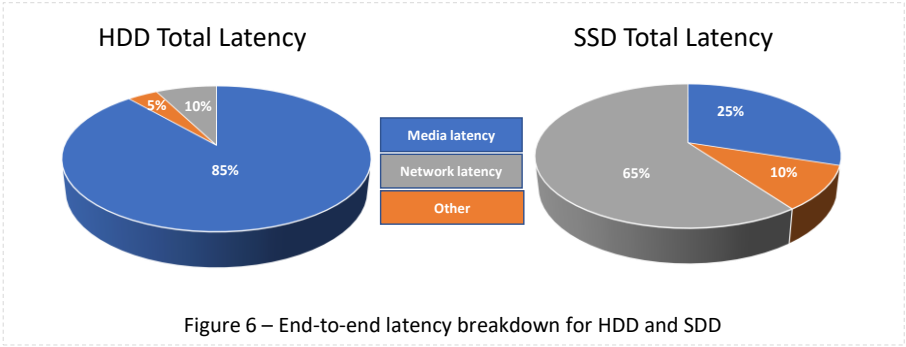


Figure 6 – End-to-end latency breakdown for HDD and SDD

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 improve dynamic latency.

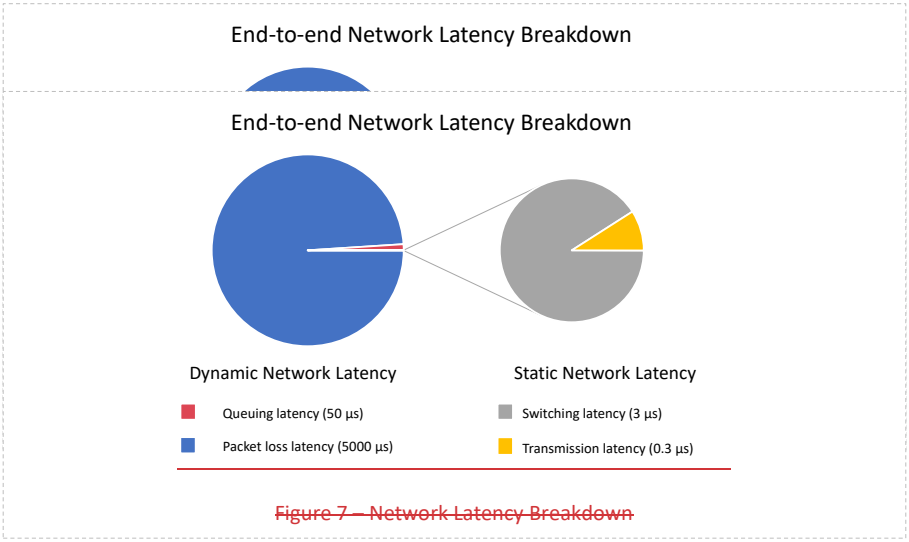


Figure 7 – Network Latency Breakdown

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. Figure 7 shows a typical network latency distribution.

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 within the network: in-network and incast. 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. Incast is a network traffic pathology caused by many-to-one communication patterns that can lead to large packet loss and increased queuing delay. Incast can increase application latency and decrease application throughput to a point well below the characteristics of link bandwidth [11]. The problem especially affects AI training, where distributed processing cannot continue until all parallel threads in a stage complete. Increased application latency degrades the concurrency of the networked storage system which lowers the number of IOPS for the entire solution.

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 algorithms and applications in a data center,

the GPUs must be distributed and networked. This places stringent requirements on communication volume and performance.

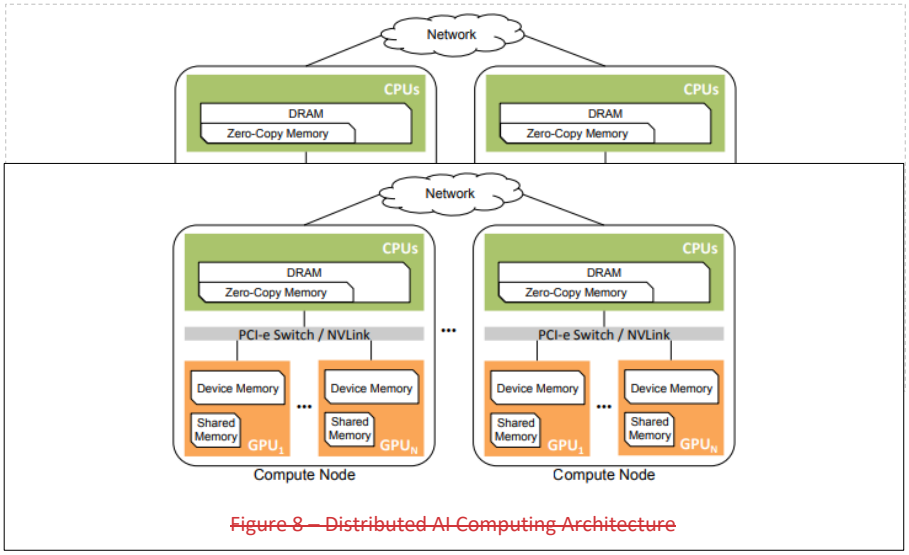


Figure 8 – Distributed AI Computing Architecture

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.

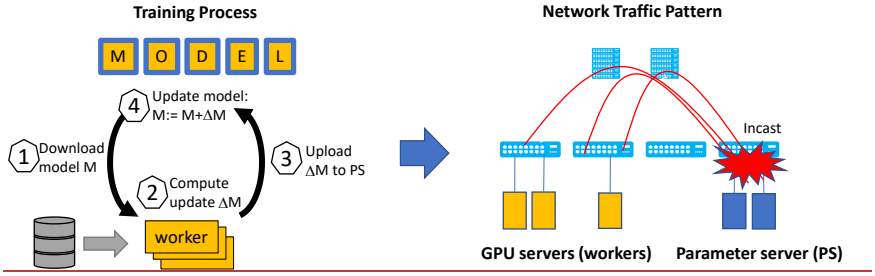


Figure 9 – Periodic incast congestion during training

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].

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.

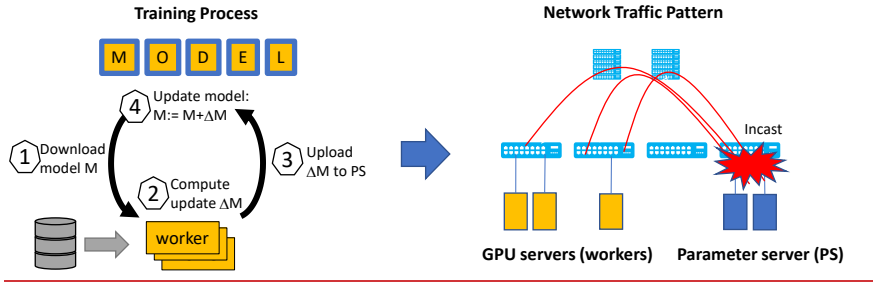


Figure 9 - Periodic incast congestion during training

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.

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 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.

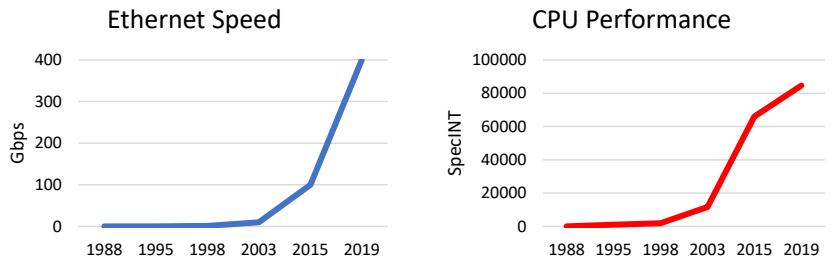


Figure 10 – Historical Performance Comparison

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 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

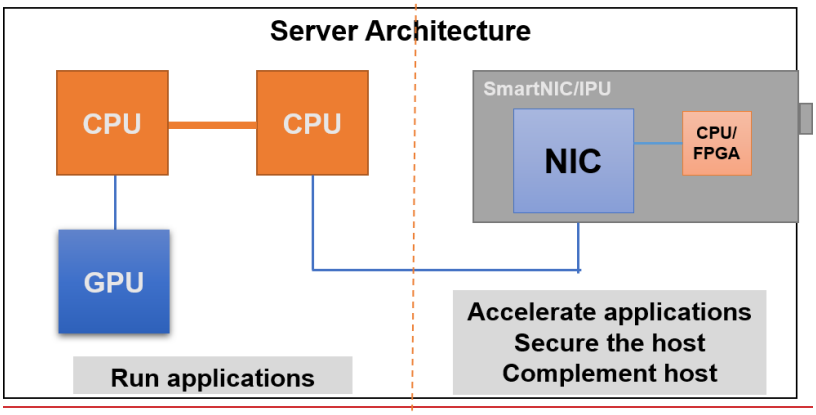


Figure 11 – Server Architecture with SmartNIC

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.

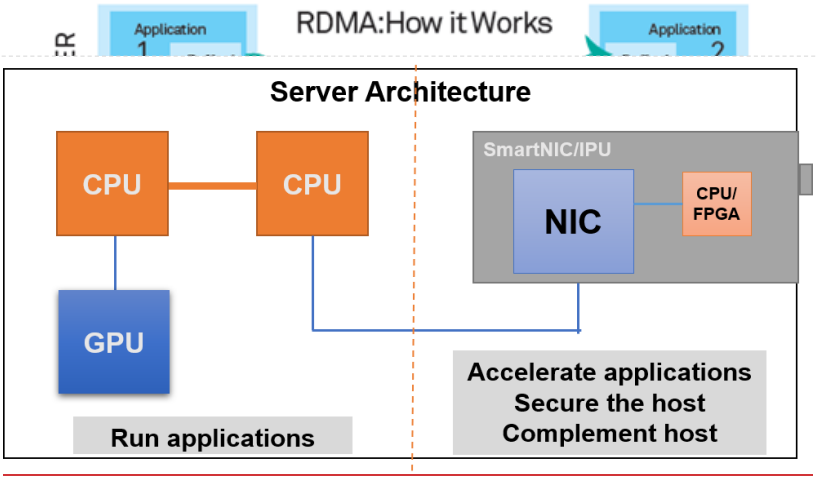


Figure 11 – Server Architecture with SmartNIC

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 problem of server-side data processing latency in network applications, which transfers data directly from one computer's memory to another without the intervention of both operating systems. 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. There are three different transports for the RDMA protocol: Infiniband, iWarp and RoCEv1/RoCEv2.

Infiniband

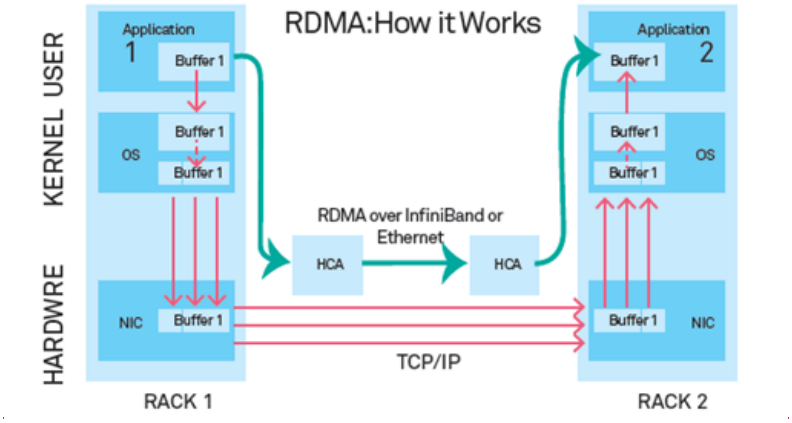


Figure 12 - Working principle of RDMA

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

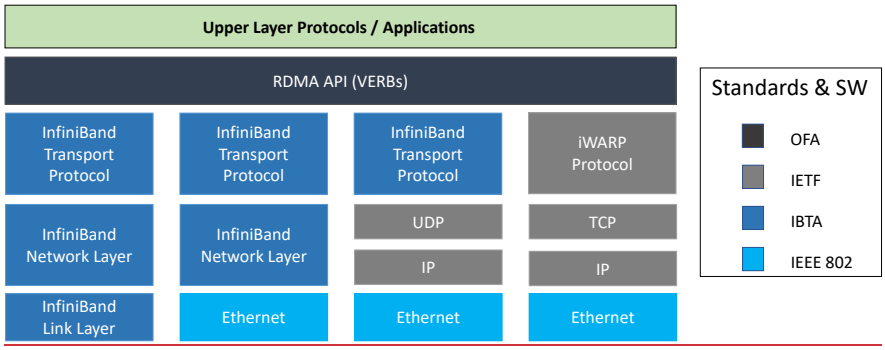


Figure 13 – RDMA protocol stacks and standards

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 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.

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

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 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.

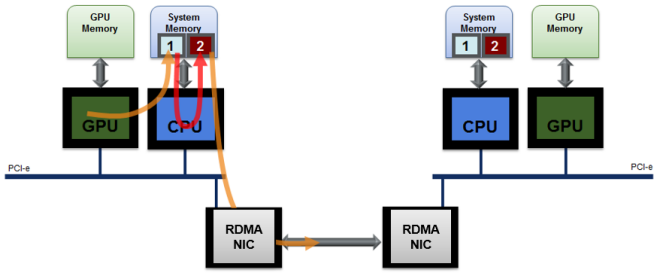
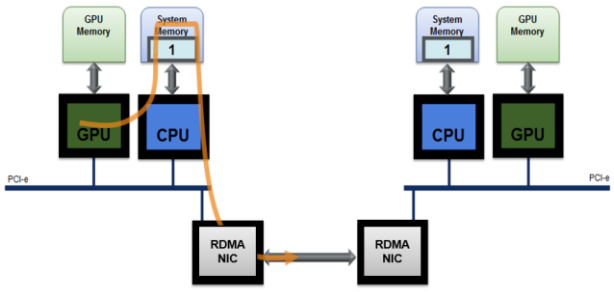


Figure 14: The Data Transfer Before GPU DirectRDMA

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



Picture 15: The Data Transfer Based On GPU Direct

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.

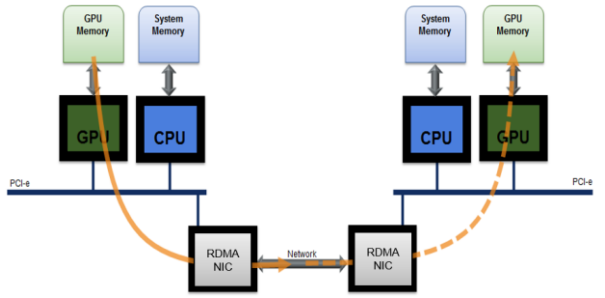


Figure 16: The Data Transfer Based On GPU DirectRDMA

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.

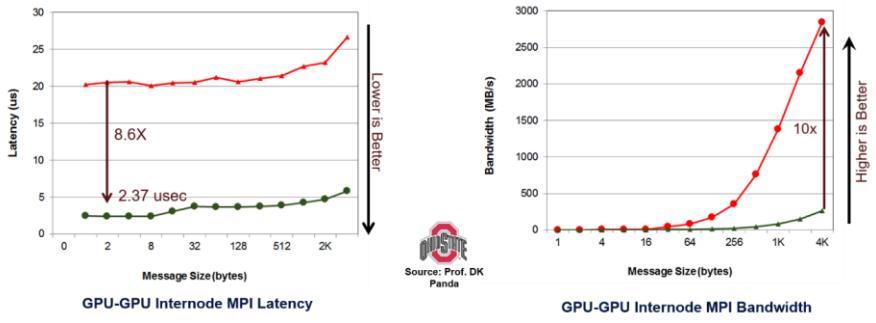


Figure 17: The Performance Of GPU DirectRDMA (From OSU)

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 will allow them to consume all available network bandwidth instantly and can lead to extreme congestion at convergence points in the network. Deep switch buffers absorb temporary congestion to avoid packet loss but delay the delivery of latency sensitive packets. 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 in order to not overreact to temporary congestion and slow down unnecessarily.

Experimentation shows the tradeoff still exists after varying algorithms, parameters, traffic patterns and link loads [17]. Figure ~~14~~18 from [17] shows how flow completion times (FCT) are extended beyond the theoretical minimum FCT when using different ECN marking thresholds (Kmin, Kmax) in switches and using an RDMA WebSearch application as the input traffic load. Lower values for Kmin and Kmax will cause ECN markings to occur more quickly and force a flow to slow down. As seen in the figure, when using low ECN thresholds, small flows which are latency-sensitive have lower slowdown in FCT, 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 14-b when the average link load is 50%).

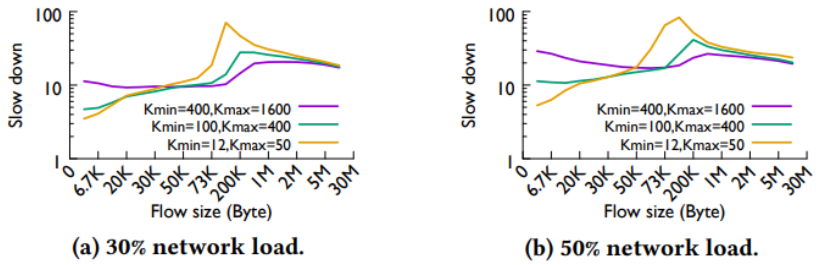


Figure ~~14~~18 – FCT slowdown distribution with different ECN thresholds, using WebSearch

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 [18]. The RoCE protocol runs on top of UDP with a go-back N retransmission strategy that severely impacts performance if invoked. As such, RoCE requires Priority-based Flow Control (IEEE Std 802.1Q-2018, Clause 36 [19]) to ensure that no packet loss occurs in the data

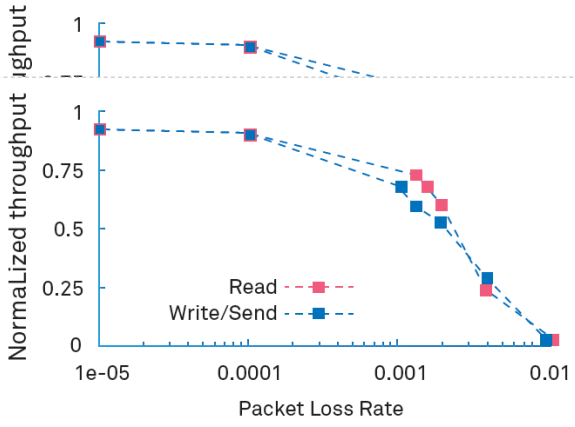


Figure 15 – Impact of packet loss on RDMA throughput

center network. Figure 1519 from [20] shows how RoCE service throughput decreases rapidly with an increase in the 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 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 backpressure flow control such as PFC have been studied for many years [20, 21, 22]. A PFC deadlock occurs when there is a cyclic buffer dependency (CBD) among switches in the data center network. The CBD is created when buffers in a sequence of switches are waiting on buffers in other switches of the sequence to have capacity before a

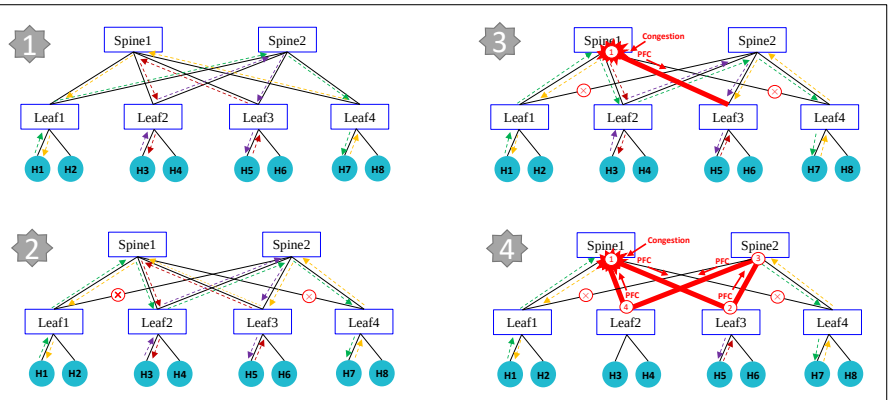


Figure 1620 – Example PFC Deadlock

dependent switch can transmit 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 the 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 1620. 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 line) and between H3 and H5 (purple line) is re-routed. The re-routing pushes more traffic through leaves 2 and 3 causing a potential overflow in spine 1 as shown in phase 3. 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.

Congestion control issues in large-scale data center networks

RDMA technology was initially used by customers in constrained, conservative, small scale environments such as 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 17 shows an example

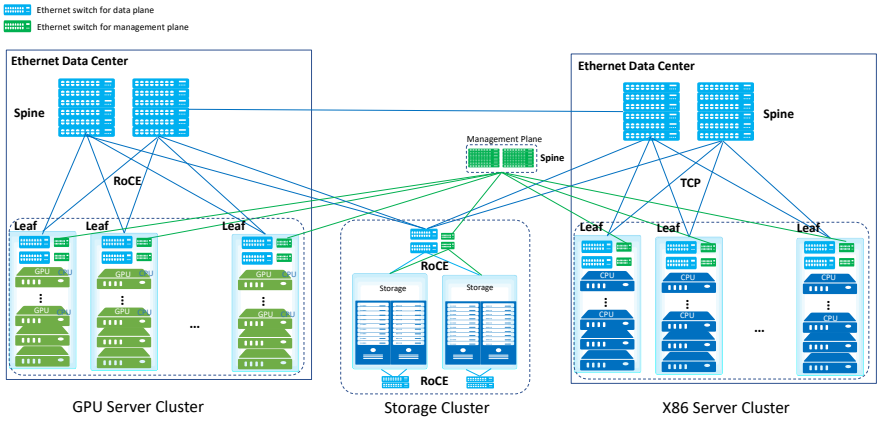


Figure 1721 – RoCE application in large-scale data center networks

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 in several different ways:

Scenario 1: A traditional web-based application with a high-speed storage backend expects an end user to submit a request from the Internet to the web service using TCP. The web service cluster may fetch the shared storage using additional TCP connections. When the storage front-end receives the request, it uses the RoCE protocol to handle the actual reading of the shared data from the medium with the expectation of obtaining extremely high IOPS using RDMA. The shared data will be returned to the end user, again with TCP.

Scenario 2: More highly integrated computing and storage clusters use the RoCE protocol for the bulk of their communication, while the management and any SDN control of the overall infrastructure is based on TCP/IP. All nodes need TCP connections for management and control, so the two types of traffic will traverse common links in the network.

Scenario 3: While the use of RoCE has gradually increased in large scale storage networks, there are still many TCP-based storage solutions used in AI/ML data centers. However, the performance requirements of interconnecting GPUs and CPUs in these data centers demands the use of RoCE. Large-scale ML/AI data center applications lead to multiple combinations of TCP and RoCE between computing and computing, storage and storage, and computing and storage.

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. The queues and the buffer management are implemented in hardware on the switch chip, but there is a performance and cost tradeoff problem. 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.

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 [23].

An example smart buffer architecture, as shown in Figure 18. Each port has some dedicated buffers for each of its queues and a dynamic pool of surplus buffers shown in gray. 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 be right-sized 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

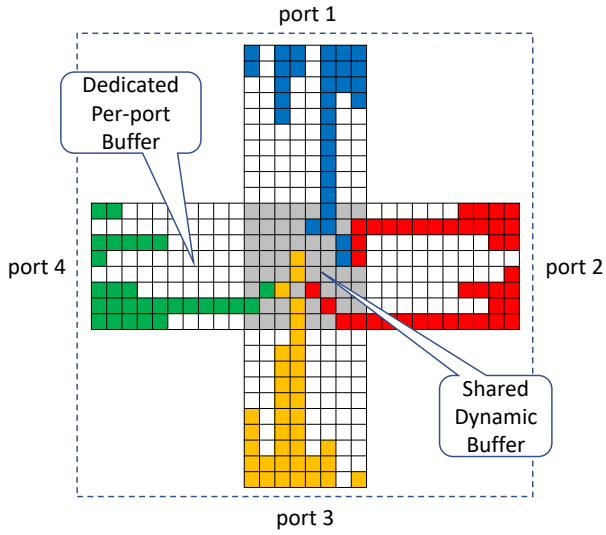


Figure 1822 – Smart-Buffer centralized and dynamic buffer management

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 do not allow a fair share of the common resource. Figure 1923 shows the problem. 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

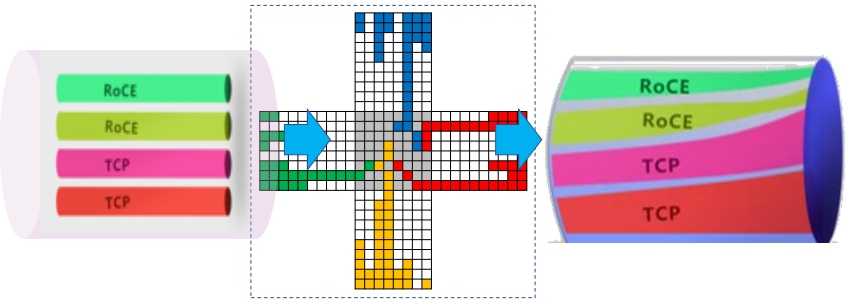


Figure 1923 – TCP and RoCE coexistence with smart buffering

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 [25].

Configuration complexity of congestion control algorithms

- ✓ Tuning RDMA networks is an important factor to achieving high-performance
- ✓ Current method of parameters configuration can be a complex operation. Reference the number of parameters require to tune the configuration
- ✓ Congestion control algorithms usually requires collaboration between the NIC and switch
- ✓ Traditional PFC manual configuration needs complex calculation with lots of parameters
- ✓ Excessive headroom leads to reduce the number of lossless queues while too little headroom leads to packet loss

5 New technologies to address new data center problems

PFC deadlock prevention using topology recognition

Traffic on a well-balanced error free Clos network 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 16. According to [22], 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 CDB 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.

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.

Discovery protocol exchange automatically determines:

1. Topology level of devices in network
   - 0 = End-station or server edge
   - 1 = Leaf
   - n+1 = Spine
2. Port orientation for each link
   - Uplink
   - Downlink
   - Crosslink

HINT: Servers are always at level 0 with uplinks.

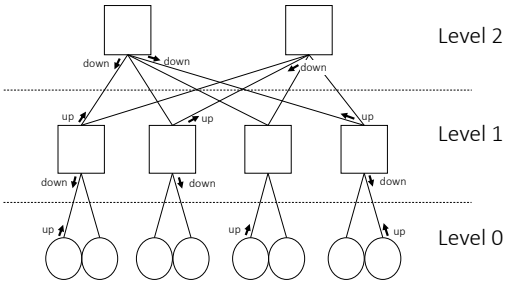


Figure 2024 – Topology and Role Discovery

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 2024 shows the resulting topology and role discovery in a simple Clos network.

Once the protocol has recognized the topology and port roles, the deadlock free mechanism can identify potential CDB points in the network and then adapt the forwarding plane to break the buffer dependencies. Figure 2125 shows how potential CDB points in the topology can be recognized. In a properly operating Clos network, there is no CDB 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 CDB 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 CDB 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

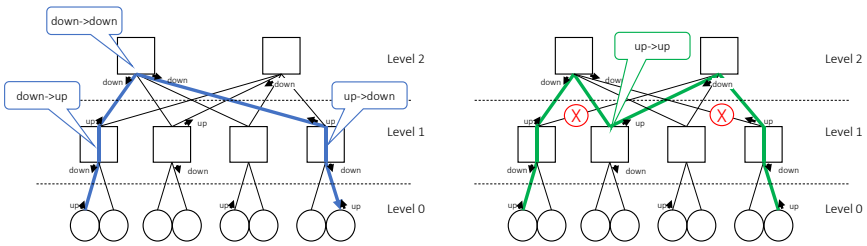


Figure 2125 – Identifying CBD points in rerouted flows

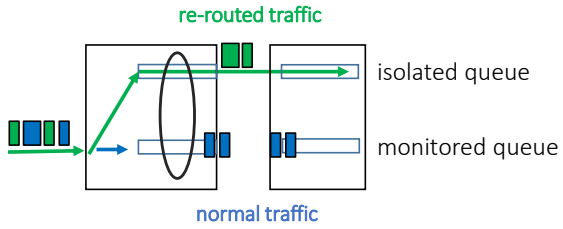


Figure 2226 – Queue switch according to CBD reroute flow recognition

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 2226 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.

ODCC, in participation with many network vendors, conducted tests to verify the deadlock free algorithm. The results indicate the effectiveness of the approach [25].

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) [24]. DCQCN combines the use of ECN and PFC to enable a large-scale lossless data center network. Figure 23 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.

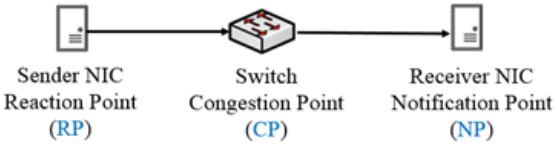


Figure 2327 – Three parts of RoCE congestion control using DCQCN

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.

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 ~~actually~~ 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 ~~24~~28.

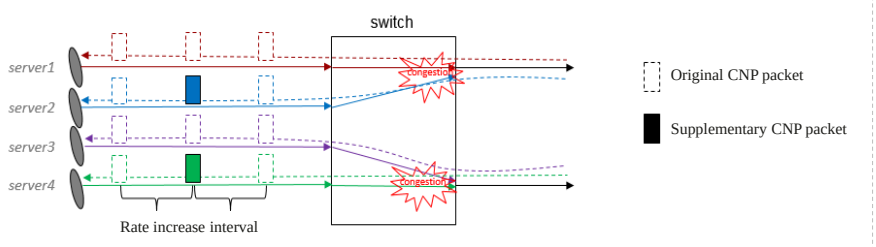


Figure 2428 – Intelligent Supplemental CNP

The ODCC tested the enhanced congestion control mechanism and the effect is beneficial [25]. According to the test result, the bandwidth QoS performance is improved by more than 30% (TCP:RoCE = 9:1 scenario).

Commented [PC1]: This sounds like the test result showed a more fair mix of TCP/RoCE traffic. Was TCP lossy and RoCE lossless?

Configuration complexity of congestion control algorithms

- ✓ Tuning RDMA networks is an important factor to achieving high-performance
- ✓ Current method of parameters configuration can be a complex operation
- ✓ Congestion control algorithms usually requires collaboration between the NIC and switch
- ✓ Traditional PFC manual configuration needs complex calculation with lots of parameters
- ✓ Excessive headroom leads to reduce the number of lossless queues while too little headroom leads to packet loss

Due to the high concurrency feature of distributed application architecture operations, a large number of concurrent data flows exist in data center network, which easily causes network congestion. Network congestion may cause extra delay, resulting in high packet transmission delay, low throughput, and a large amount of resource consumption. How to efficiently control network congestion, obtain higher bandwidth and lower latency, and improve network transmission efficiency is the key to improving data center performance.

As mentioned above, the ECN threshold is set to a low value to achieve low latency. However, a low ECN threshold often leads to low network throughput. Figure X shows the high ECN threshold has better performance for throughput-sensitive large traffic.

In the CC issue section, we also discuss how to minimize latency while maintaining throughput when traffic is mixed. It can be seen that the tune of the RDMA network is a big challenge to achieve the optimal throughput and latency and maximize the performance of the entire network.

The traditional congestion control algorithm commonly used in the industry usually requires network adapter and network collaboration. Each node needs to be configured with dozens of parameters, and the parameter combination of the entire network reaches hundreds of thousands. To simplify the configuration, you can only use the recommended static configuration based on the experience of engineers.

Common static configurations face the following two challenges: Real-time change of network traffic and effects on service performance.

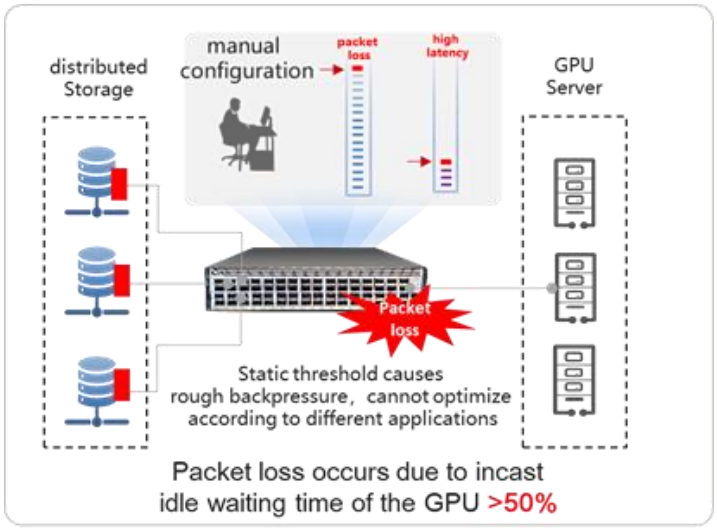
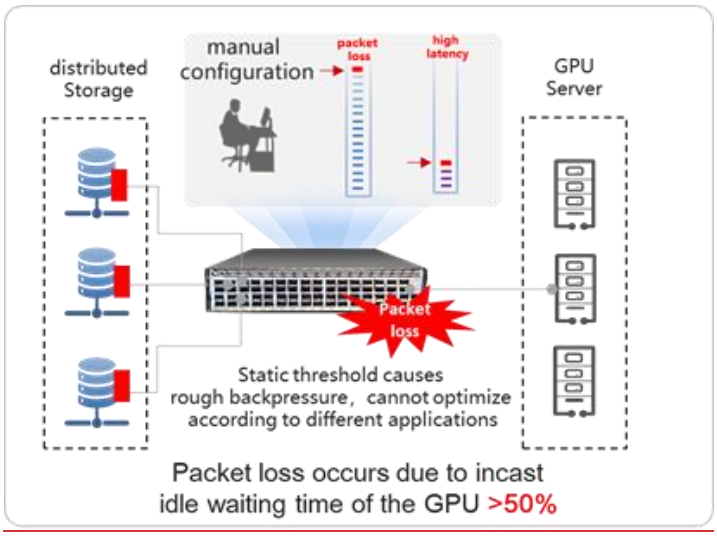


Figure 29 – Packet loss

Real-time change of network traffic

Take distributed block storage services as an example. During the running process, the read/write ratio, I/O block size, and number of concurrent read/write tasks always change, and the network

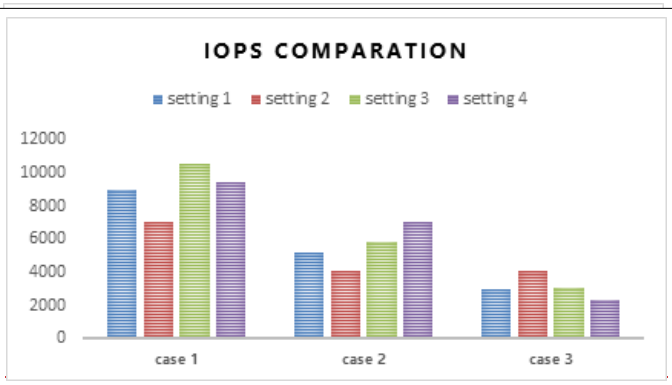


~~Figure Gx – Packet loss~~

traffic mode changes dynamically and continuously. Due to the dynamic traffic changes in the customer's environment, the manually configured static threshold may cause rough back pressure. Static threshold is difficult to adapt to the real-time network traffic changes. As a result, low throughput and high latency may occur, and network performance may deteriorate.

Service performance is affected

The congestion control algorithm parameters configured on the entire network determine the effect of congestion control. The performance difference with different parameters can be as high as 50%. Static experience configuration cannot ensure the optimal performance of most service scenarios in the customer environment. For customers, parameter configuration may lead to a gap of more than 50% in service performance or device investment. Selecting appropriate CC algorithm parameters is significant for improving the service performance of customers. Lab tests show that different congestion control algorithms produce different effects in the same application scenario, shown in Figure X.



~~Figure X – IOPS Comparison~~

Intelligent congestion parameter optimization

Control network congestion to ensure efficient and stable running of DCN services. If incast traffic is sent, traffic bursts occur on the receive end. As a result, a large number of packets are accumulated in the queue, and the number of packets exceeds the capability of the interface on the receive end instantaneously. Consequently packet loss occurs due to network congestion. Based on traditional Ethernet, we use a heuristic algorithm to monitor network traffic bursts and proactively intervene in the network before congestion occurs, ensuring stable and efficient running of DCN services.

The dynamic threshold adapts to network traffic changes, and precise backpressure is used to decrease the rate.

Our algorithm proactively detects network traffic modes and interacts with network environments, greatly improving the adaptability of network congestion algorithms. Uses dynamic threshold, precise backpressure, and proper rate reduction without manual adjustment, reducing O&M costs.

Optimal entire network performance, improving network and application performance.

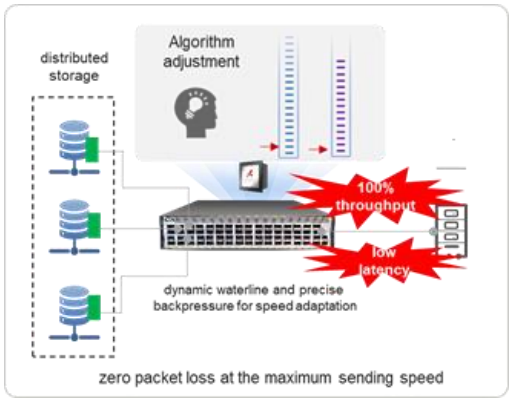


Figure X31 – zero packet loss at max speed

Compared with local optimization policies deployed on CPUs, the algorithm detects global network traffic changes, achieving the highest global network performance and ensuring optimal service performance.

ODCC tests the performance of the intelligent congestion parameter adjustment algorithm. The result shows that the new technical solution improves service throughput and latency simultaneously. For OLTP services, the delay decreases by up to 12%. For video services, the throughput increases by up to 25%.

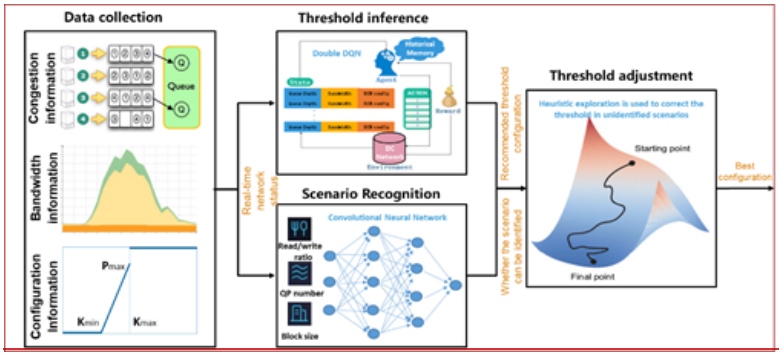


Figure Z – Cool Stuff

Buffer optimization to reduce the complexity of PFC headroom configuration

PFC is a hop-by-hop protocol between two Ethernet nodes. As show in above, the sender's egress port sends data packets to the receiver's ingress port. At the receiving ingress port, packets are buffered in corresponding ingress queues. Once the ingress queue length reaches a certain threshold (XOFF), the switch sends out a PFC pause frame to the corresponding upstream egress

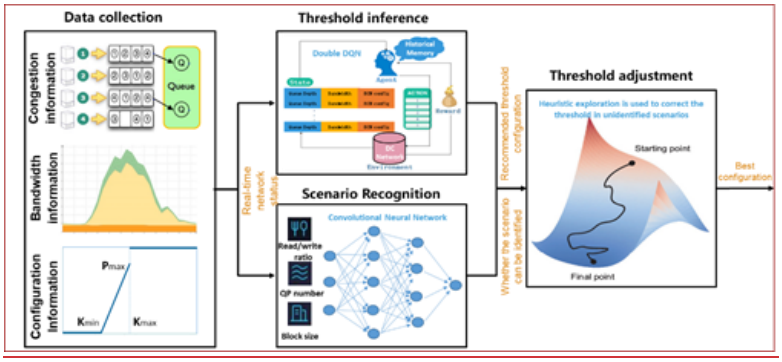


Figure 32 – Cool Stuff

queue. After the egress queue receives the pause frame, it stops sending packets. Once the ingress queue length falls below another threshold (XON), the switch sends a pause with zero duration to resume transmission.

RoCE needs PFC mechanism to achieve lossless Ethernet. Network switch enables PFC to make sure that there's no packet loss in network. Each lossless queue needs to be configured with enough headroom buffer [25].

Originally, configuring the PFC threshold was a very experiential task. The calculation of PFC threshold is complex with lots of parameters (Buffer structure and unit size, switching delay, cable delay and interface delay) (See Clause 36 of [19]). PFC buffer requires both highly usage and implementation dependent.

Pay attention to the fact that different vendors may have different implementations and would imply a different configuration for the headroom. Thus, excessive headroom leads to reduce the number of lossless queues while too little headroom leads to packet loss [18].

To solve this problem of PFC headroom configuration complexity, ODCC provides an adaptive headroom calculation algorithm to simplify the configuration complexity. The core idea of this algorithm is using intelligent and dynamic distance discovery method. A round trip timer determines the latency between two connected switches. According to the MTU size, bandwidth and so on, automatically allocates the desired amount of headroom needed to ensure no frame loss due to congestion. The switch will never allocate more headroom than the maximum needed. Thus, we can reserve enough headroom for more lossless queues. The test result shows that the adaptive headroom algorithm can release more buffer space for more lossless queues and improve the latency by 30% to 50% in long-distance transmission scenarios.

- ✓ Intelligent headroom calculation
- ✓ Intelligent heuristic algorithms for identifying congestion parameters
- ✓ Methods for dynamic optimization based on services
- ✓ Test verification (ODCC lossless DCN test specification and result)
- ✓ Self-adaptive headroom configuration

6 Standardization Considerations

Things for the IEEE 802 and IETF to consider. Possibly others as well – SNIA, IBTA, NVMe, etc..

7 Conclusion

Closing words…

8 Citations

<< format the table later – MS word screws up the format each time your rebuild, so just wait until the end and change the column widths to get it to look correct. >>

[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:

Formatted Table

Inserted Cells

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] 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.

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

[20] 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.

[21] 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.

[22] 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.

[23] 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].

[24] 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.

[25] 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].

