

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

Trademarks and Disclaimers

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

Copyright © 2020 IEEE. All rights reserved.

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

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

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

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

IEEE SA Licensing and Contracts
445 Hoes Lane
Piscataway, NJ 08854

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*

*Copyright © 2020 by The Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published April 2020. Printed in the United States of America.*

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

PDF: ISBN xxx-x-xxxx-xxxx-x XXXXXXXXXXX

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

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

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

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

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

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

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

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

TABLE OF CONTENTS

1-20-0030-0405-ICne-pre-draft-dcn-

1. INTRODUCTION.....	2
Scope	2
Purpose.....	2
2. BRINGING THE DATA CENTER TO LIFE	2
A new world with data everywhere	2
Today's data center enables the digital real-time world.....	4
3. EVOLVING DATA CENTER REQUIREMENTS AND TECHNOLOGY	5
Technology evolution.....	5
Network requirements.....	8
4. CHALLENGES WITH TODAY'S DATA CENTER NETWORK.....	17
High bandwidth and low latency tradeoff.....	17
Deadlock free lossless network.....	18
Congestion control issues in large-scale data center networks	21
Configuration complexity of congestion control algorithms.....	25
5. NEW TECHNOLOGIES TO ADDRESS NEW DATA CENTER PROBLEMS	25
Approaches to PFC storm elimination.....	25
Improving Congestion Notification	28
Intelligent congestion parameter optimization	33
6. STANDARDIZATION CONSIDERATIONS	35
7. CONCLUSION	36
8. CITATIONS.....	36

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 new 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

<<

- ✓ Enterprise digital transformation needs more data for using AI
- ✓ Machine translation and search engines need to be able to process huge data simultaneously
- ✓ The era of internet celebrity webcast, all-people online games, data explosion
- ✓ Consumption upgrade in the new era of take-out, online takeout platform schedule and deliver massive orders
- ✓ The XX service of the carrier has higher requirements on data center network
- ✓ Data-based New World Requires Ubiquitous Data Center Technologies.

>>

Digital transformation is driving change in both our personal and professional lives. Work flows 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 and reliability to meet these real-time demands.

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 on the network when performing 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 of 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.

Data centers ultimately deliver the services in this era of digital transformation to our real-time 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 modern data center.

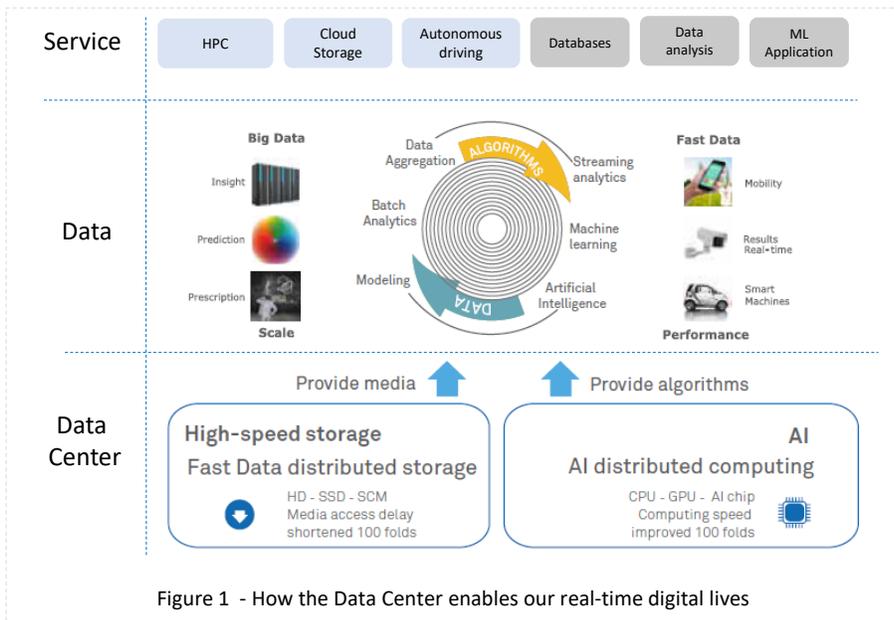


Figure 1 - How the Data Center enables our real-time digital lives

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 <<IDC reference>>. Among 2000 multinational companies, 67% of CEOs have made digitalization the core of their corporate strategies [3].

A large amount of data will be generated during the digitalization process, becoming a core asset, and enabling a new emergence of Artificial Intelligence Applications as seen in Figure W. Huawei GIV predicts that the data volume will reach 180 ZB in 2025 [4]. 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 reach over 95% in the future. The current big data analytics method is helpless. If manual processing is used, the data volume will be far greater than the processing capability of all human beings. The AI algorithm based on machine computing for deep learning can filter out massive invalid data and automatically reorganize useful information, providing more efficient decision-making suggestions and smarter behavior guidance.

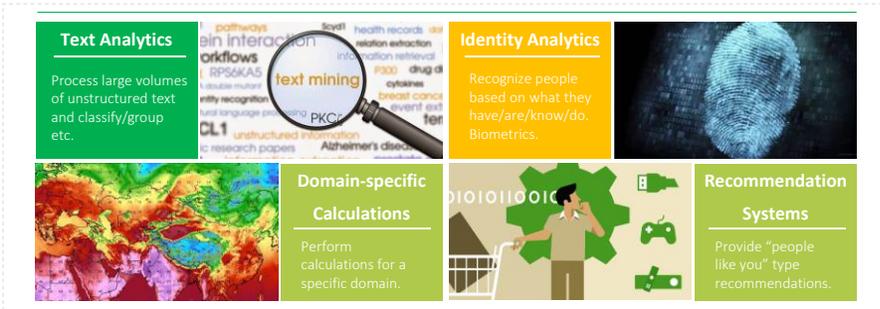


Figure 2 – Emerging Artificial Intelligence Applications

AI applications are emerging everywhere, as shown in Figure 2.

Cloud data centers improve the performance of these applications. Cloud data centers are designed to scale and act more like a service support center. They are application-centric and use the cloud platform to quickly distribute IT resources. While the data centers are application centric, they are founded on big data as shown in Figure 3.

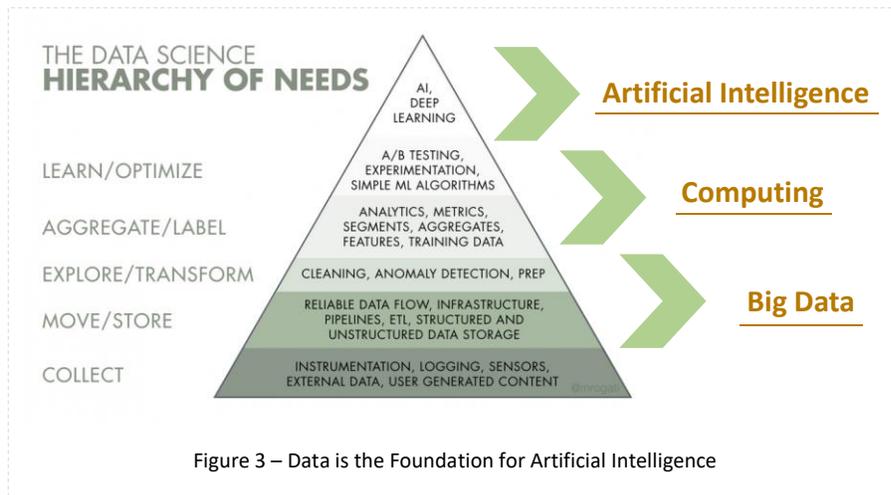


Figure 3 – Data is the Foundation for Artificial Intelligence

So, within data centers, understanding how to efficiently process data based on the needs of different applications is a key focus area. Data centers must know where to reserve storage to efficiently transmit the data to the computing engines of the applications.

3

Evolving data center requirements and technology

Requirements evolution

<< First discuss the new and evolving requirements for data center networks hosting AI applications. These include: 1. huge amounts of data for AI learning. What is the data and why is it so large? Why is AI better with more data? 2. To hold the huge amounts of data, it must be fast and distributed. The latency needs to compete with local storage. 3. Huge amounts of computing cycles needed to work on that data. Describe the AI computing models; data parallelism vs model parallelism and how these differ in network communication requirements. >>

Take AI training of self-driving cars as an example, the deep learning algorithm relies heavily on massive sample data and high-performance computing capabilities. Training data collected is close to the P level (1PB = 1024 TB) per day. If traditional hard disk storage and common CPUs are used

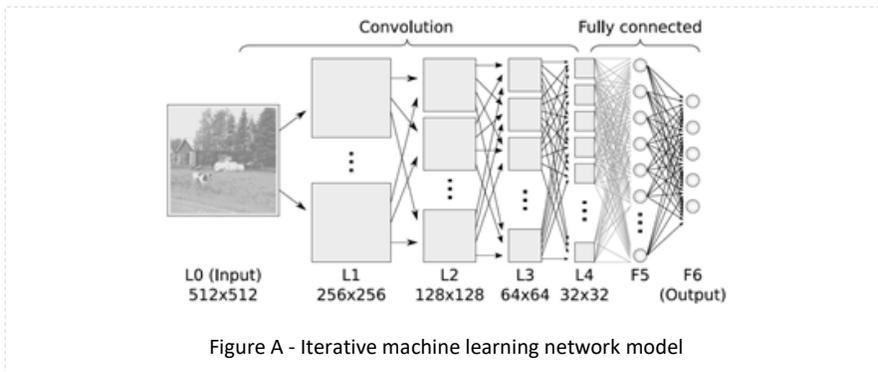
to process the data, it takes at least one year to complete the training, which is almost impossible. To improve AI data processing efficiency, revolutionary changes are occurring in the storage and computing fields. The development of high-speed storage technology will help users to access the content more conveniently. Other data center technologies should be evolved together with distributed storage to ensure customers can obtain high input and output speed. Storage performance needs to improve by an order of magnitude to achieve more than 1 million input/output operations per second (IOPS) [5].

Storage media evolve from HDDs to SSDs to meet real-time data access requirements, reducing the medium latency by more than 100 times. With the significant improvement of storage media and computing capabilities, the current network communication latency becomes the bottleneck of further performance improvement in high-performance data center clusters. The communication latency accounts for more than 60% of the total storage E2E latency, that is, more than half of the time of precious storage media is idle.

In general, with the evolution of storage media and computing processors, the communication duration accounts for more than 50% of the total communication duration, hindering the further improvement of computing and storage efficiency [6].

- ✓ The development of fast storage provides necessary media for big data (distributed storage)
 - Storage performance needs to improve by an order of magnitude to achieve more than 1 million input/output operations per second (IOPS).
 - Communication latency has recently increased from 10% to 60% of storage E2E latency.
- ✓ Computing speed improvement (distributed computing)

AI computing model complexity is exploding



AI training is becoming increasingly complex with the development of services. For example, there are 7 ExaFLOPS and 60 million parameters in the Microsoft Resnet in 2015. The number came to 20 ExaFLOPS and 300 million parameters when Baidu trained their deep speech system in 2016. In 2017, the Google NMT used 105 ExaFLOPS and 8.7 billion parameters [7].

AI inference is the next great challenge so there must be an explosion of network design. The new characteristics of AI algorithm and huge computing workload require evolution of data center network.

Characteristics of AI computing

<< explain that AI computing is iterative, not a single pass, so communication is critical and the application runs for a long time. Describe the different modes for AI computing; data parallel vs model parallel and what that means to the network >>

Traditional data center services (web, video, and file storage) are transaction-based and the calculation results are deterministic. For such tasks, there is no correlation or dependency between single calculation and network communication, and the occurrence time and duration of the entire calculation and communication are random. AI computing is based on target optimization and iterative convergence is required in the computing process, which causes high spatial correlation in the computing process of AI services and temporally similar communication modes.

A typical AI algorithm refers to an optimization process for a target. The computing scale and features mainly involve models, input data, and weight parameters.

To solve the Big Data problem, the computing model and input data need to be large (for a 100 MB node, the AI model for 10K rules requires more than 4 TB memory), for which a single server cannot provide enough storage capacity. In addition, because the computing time needs to be shortened and increasingly concurrent AI computing of multiple nodes is required, DCNs must be used to perform large-scale and concurrent distributed AI computing.

Distributed AI computing has the following two modes: model parallel computing and data parallel computing. For model parallel computing, each node computes one part of the algorithm. After computing is complete, all data fragmented across models needs to be transferred to other nodes, as shown in Figure C.

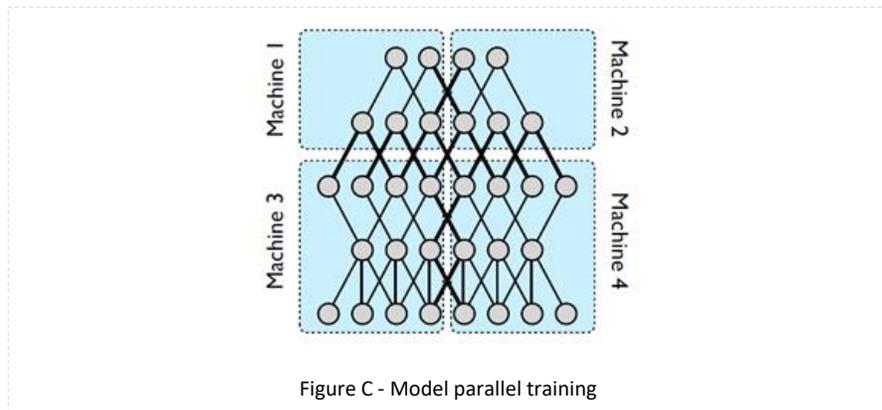


Figure C - Model parallel training

For parallel data computing, each node loads the entire AI algorithm model. Multiple nodes can calculate the same model at the same time, but only part of the input data is input to each node. When a node completes a round of calculation, all relevant nodes need to aggregate updated information about obtained weight parameters, and then obtain the corresponding globally updated data. Each weight parameter update requires that all nodes upload and obtain the information synchronously.

Commented [PC1]: A good summary of the differences is:

'Data parallelism' and 'model parallelism' are different ways of distributing an algorithm. These are often used in the context of machine learning algorithms that use stochastic gradient descent to learn some model parameters, which basically means that:

- The algorithm is trying to estimate some parameters from the given data.
- Parameters are estimated by minimizing the gradient against some loss function.
- Algorithm iterates over data in small batches.

In the data-parallel approach:

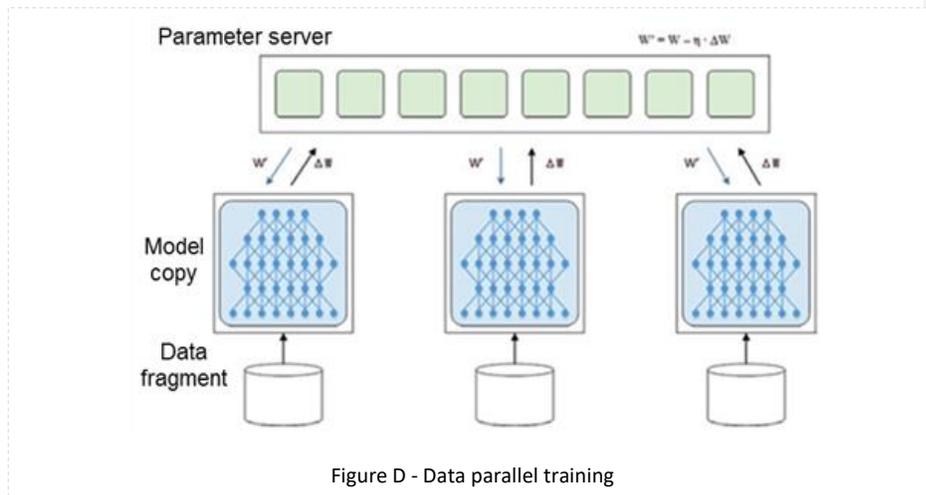
- The algorithm distributes the data between various cores.
- Each core independently tries to estimate the same parameter(s)
- Cores then exchange their estimate(s) with each other to come up with the right estimate for the step.

In the model-parallel approach:

- The algorithm sends the same data to all the cores.
- Each core is responsible for estimating different parameter(s)
- Cores then exchange their estimate(s) with each other to come up with the right estimate for all the parameters.

Data-parallel approach is useful when there are smaller number of nodes in the cluster and the number of parameters to be estimated is small whereas model-parallel approach is useful in the opposite condition.

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 [8].



Evolving technologies

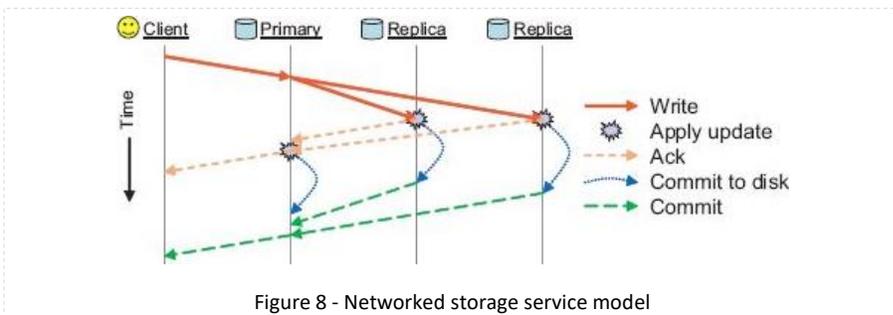
<< Here we describe some key new technologies that are evolving to meet the requirements. The main pieces should be fast storage (SSDs), GPUs, Smart Nics, Protocols like RDMA. Have a small sub-section on each of the above technology areas. End with what it means to the network to support these new technologies >>

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

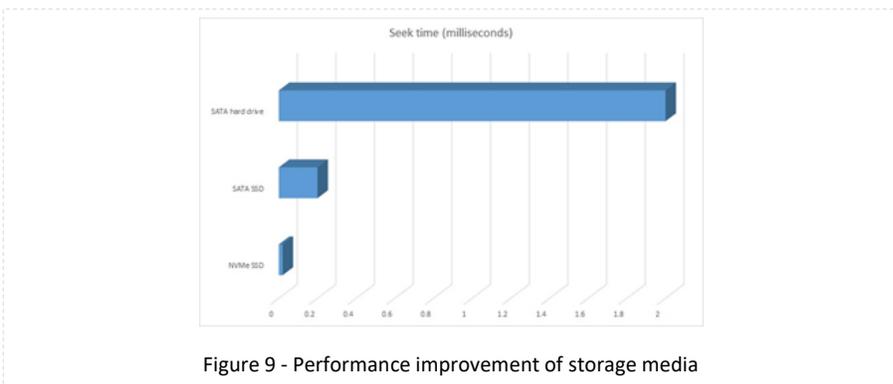
In networked storage, a file is distributed to multiple storage servers for IO acceleration and redundancy. When a data center application reads a file, it will concurrently access different parts of data from different servers, and the data will be aggregated through a data center switch at nearly the same time. When a data center application writes a file, the data can trigger a series of storage transactions between distributed and redundant storage nodes. Figure 8 shows an example of data center communication triggered by the networked storage service model.

When an application (i.e. Client in Figure 8) 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 8). 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 8). 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. << Consider making a comment about the impact of network latency here >>.



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. Figure 9 shows the difference in seek time between the technologies; HDD = 2-5 ms, SATA SSD = 0.2 ms, and NVMe SSD = 0.02 ms. Shorter overall average seek times are better, but performance of drives in each category can vary [9].

When NVMe is used for networked storage, the much faster access speed of the medium can result in network bottlenecks. Figure 10 shows a classical networked storage traffic model. In this traffic



model, when data is aggregated each time, incast (many-to-one) easily occurs. As storage performance continues to increase, pressure on the network increases, affecting distributed storage IO throughput.

Incast is a network traffic pathology that affects many-to-one communication patterns in datacenters. Incast increases application latency with the queuing delay of flows and decreases application throughput to something well below the link bandwidth [10]. The problem especially affects computing paradigms, such as AI training, where distributed processing cannot continue until all parallel threads in a stage complete.

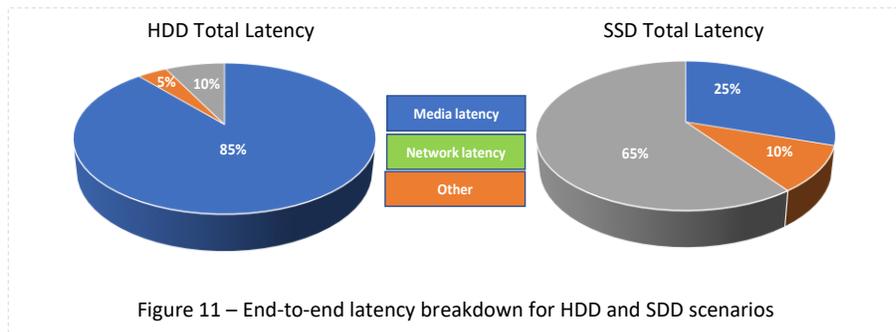


Figure 11 – End-to-end latency breakdown for HDD and SSD scenarios

Since incast increases application latency, the concurrency of the networked storage system will be affected. Therefore, the performance of distributed IOPS is limited by network latency. With newer, faster storage technologies, the impact of network latency becomes more significant. Figure 11 shows that network latency is the primary bottleneck in networked SSD storage, whereas network latency was negligible with networked HDD storage. Looking to the future, with NVMe over fabrics (i.e. networked NVMe storage), to attain the maximum IOPS performance, the network latency problem must be resolved first.

To analysis network latency further, it can be classified into 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 forwarding chip and the transmission distance. It usually has a fixed specification. Figure X says that static latency is generally at ns (10^{-9} second) or sub-us (10^{-6} level) in the industry, and accounts for less than 1% of the total network delay.

The dynamic latency greatly affects the network performance. The dynamic latency ratio is greater than 99%. The dynamic latency includes the internal queuing latency and packet retransmission latency, which are caused by network congestion and packet loss. In the AI era, traffic conflicts become more and more severe on networks. Packet queuing or packet loss often occurs, causing the latency within sub-seconds. Therefore, the key of the low-latency network is the low dynamic latency.

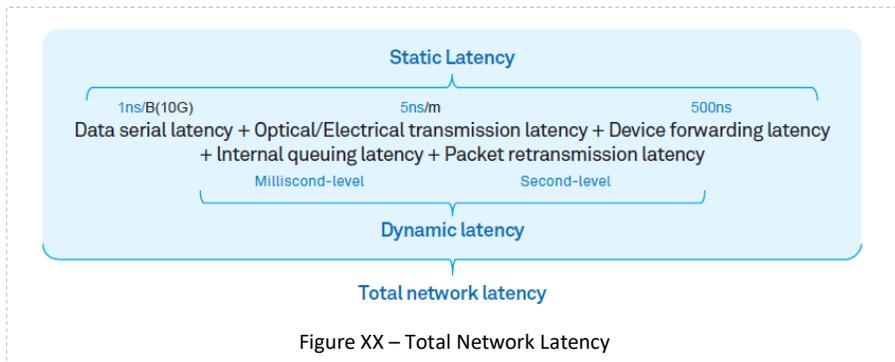


Figure XX – Total Network Latency

Most existing network solutions focus on reducing the static latency caused by network device forwarding, while the dynamic latency caused by packet loss during network congestion has proven to have a more severe impact on applications. In most existing systems the impact from latency comes primarily from dynamic latency which occurs across the network during packet loss of congestion management. Figure X below shows a typical network latency distribution.

<< NOTE: We should define dynamic latency and its components. Highlight the issue of packet loss. That has not been discussed yet. >>

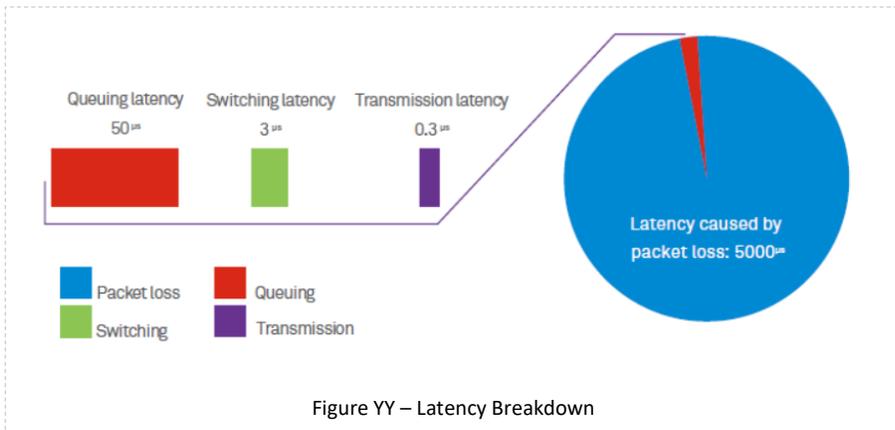
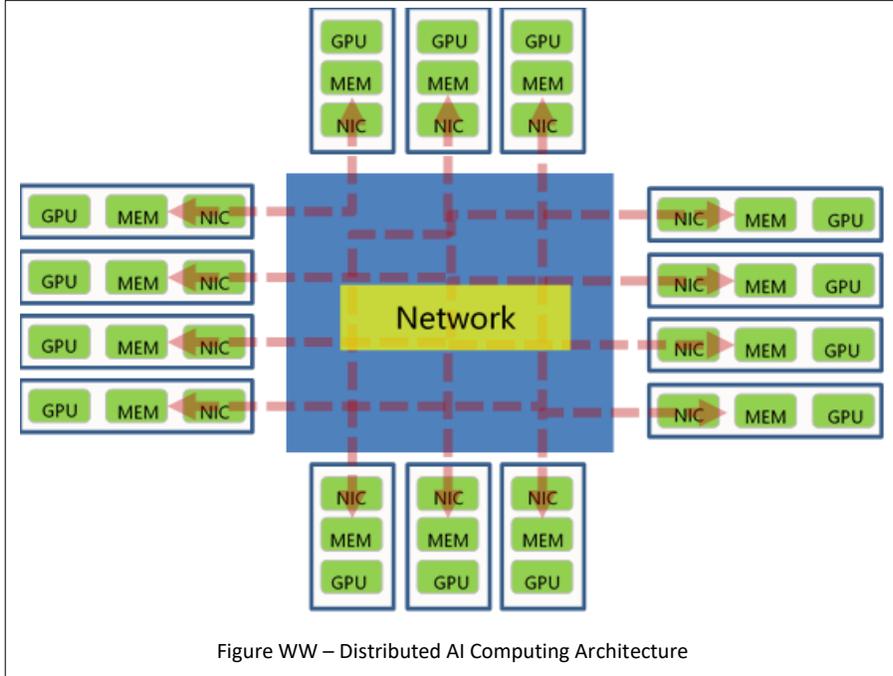


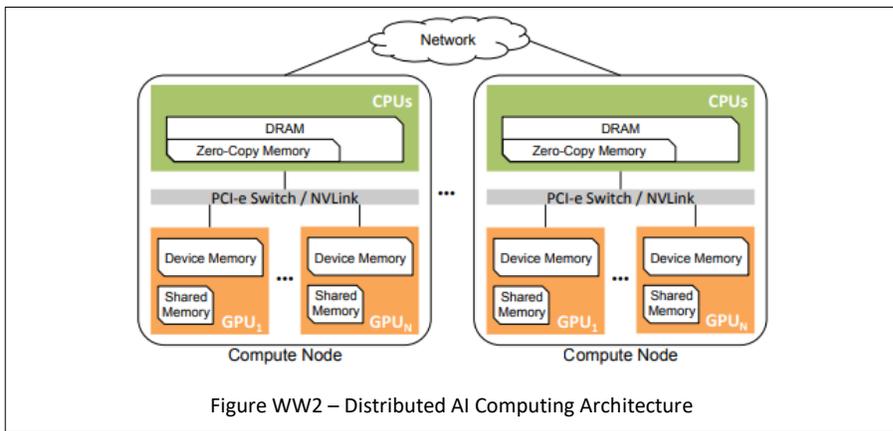
Figure YY – Latency Breakdown

GPUs: Ultra-low latency network for parallel computing

As the number of AI algorithms and AI applications continue to increase, and the distributed AI computing architecture emerges, AI computing is implemented on a large scale. GPUs have ignited a worldwide AI boom. They have become a key part of modern supercomputing. They've been woven into a sprawling new hyperscale data centers. Still prized by gamers, they have become accelerators speeding up all sorts of tasks from encryption to networking to AI. GPUs provide much



higher memory bandwidth than today’s CPU architectures. Nodes with multiple GPUs are now ubiquitous in high-performance computing because of their power efficiency and hardware parallelism. Figure X 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 [11].



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. 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.

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 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 12 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 parameter servers at nearly the same time. When the computing time is improved by deploying faster GPUs, the pressure on the network and resulting incast increases.

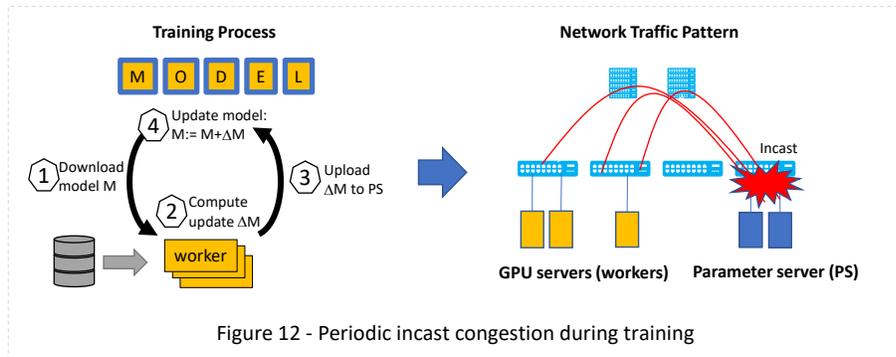
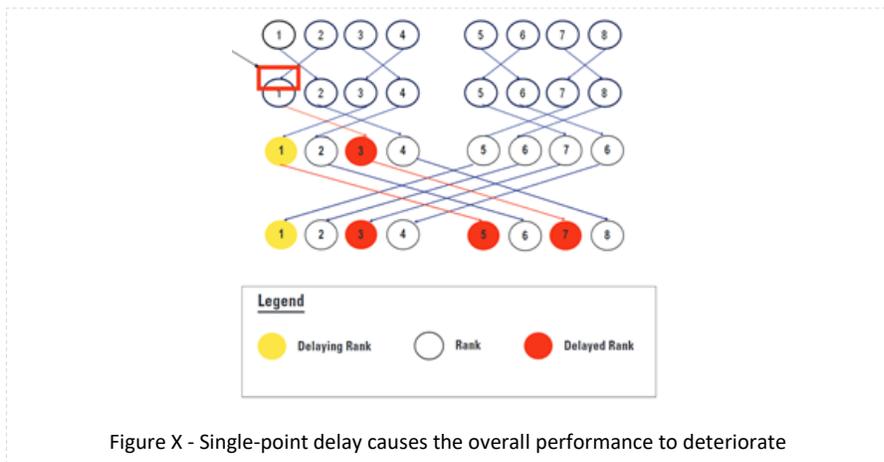


Figure 12 - Periodic incast congestion during training

The high-bandwidth and low-latency DCN with only physical links cannot meet requirements of large-scale and highly concurrent AI/HPC applications. In the iteration process of distributed AI computing, a large amount of burst traffic is generated within milliseconds. In addition, because a parameter server (PS) architecture is used to update parameter weights of the new model for data parallelization, the incast traffic model at a fixed time is easily formed. In this case, packet loss, congestion, and load imbalance occur on the network. As a result, the Flow Completion Time (FCT) of some data flows is too long. Distributed AI computing is synchronous. If few flows are delayed, more computing processes are affected. Consequently, the completion time of the entire application is delayed. This is what we call the 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. It is very critical to the whole distributed computing system. Figure X shows how tail latency injures the whole system performance.

Consequently, in order to minimize the FCT to complete the entire computing task, we need to reduce the tail delay as much as possible. Because the microbursts in data center network are within milliseconds, the tail delay needs to be controlled within milliseconds to ensure optimal system performance. Therefore, For HPC services, in order to have an ultra-low latency lossless network, the data center network should first solve the tail delay problem.



SmartNICs

- ✓ SmartNIC become the computer in front of computer
 - SmartNIC is a NIC with all NIC functions regardless CPU/FPGA. Host CPU only request to install NIC driver.
 - SmartNIC is a computer in front of computer. SmartNIC has independent OS and is able to run some applications independently.
 - SmartNIC can be used to accelerate application
 - Accelerate computing, storage...
 - SmartNIC can be used to offload host CPU to run specific application more efficient
 - SmartNIC is part of computing resource. Participate the application computing together with host CPU and GPU.
 - Complement of CPU and GPU computing resource
 - SmartNIC is not the replacement of CPU and GPU, major applications still run on CPU/GPU
 - SmartNIC can be the independent domain than host domain and protect the host domain
 - Offload OVS to SmartNIC to isolate the data classification from hypervisor
 - SmartNIC can be emulated to other PCIe devices to support more advanced application
 - NVMe emulation

- SmartNIC is programmable and easy use
 - Open source software, major Linux
 - Easy to program, no special request for programmer
- SmartNIC is not proprietary NIC, one NIC fits many applications, easy for user to program

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. By transferring telegrams directly into the storage space of the other computer through the network, data can be quickly transferred from one system to the storage space of another system, reducing or eliminating the need for multiple copies of data telegrams during transmission, thus freeing up memory bandwidth and CPU cycles and greatly improving system performance. Figure E shows the principle of RDMA protocol.

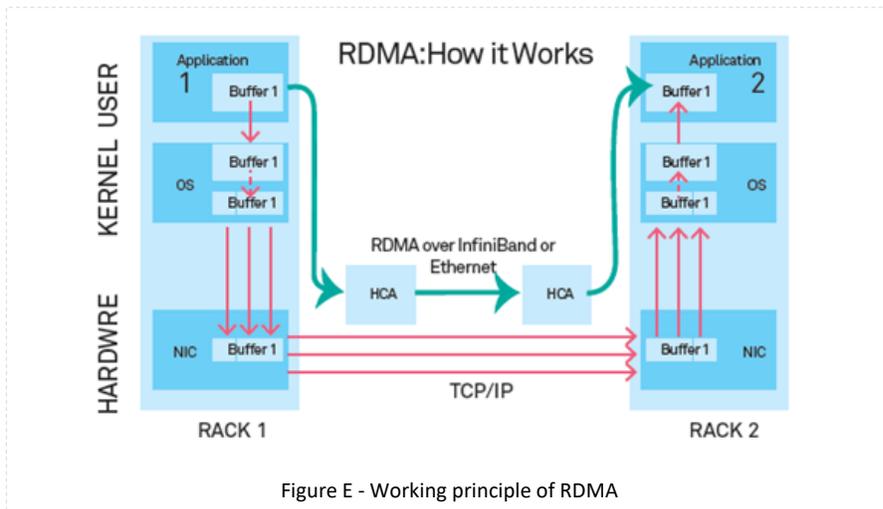


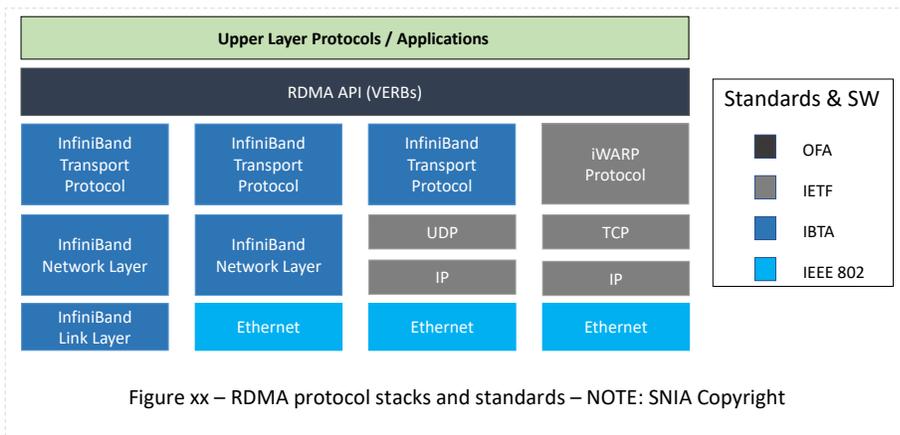
Figure E - Working principle of RDMA

RDMA's development in the transport layer/network layer currently goes through 3 technologies, Infiniband, iWarp and RoCEv1/RoCEv2.

Infiniband

In 2000, the IBTA (InfiniBand Trade Association) released the first RDMA technology, Infiniband, which is a customized network technology for RDMA multi-layered, new design from the hardware perspective to ensure the reliability of data transmission. The InfiniBand technology uses RDMA technology to provide direct read and write access to remote nodes. RDMA used InfiniBand as the transport layer in its early days, so it must use InfiniBand switches and InfiniBand network cards to implement.

iWarp



Internet wide area RDMA protocol, also known as RDMA over TCP protocol, is the IEEE/IETF proposed RDMA technology. It uses the TCP protocol to host the RDMA protocol. This allows RDMA to be used in a standard Ethernet environment (switch) and the network card requirement is an iWARP enabled network card. In fact iWARP can be implemented in software, but this takes away the performance advantage of RDMA.

RoCE (RDMA over Converged Ethernet)

In April 2010, the IBTA released RoCEv1, which was released as an add-on to the Infiniband Architecture Specification, so it is also known as IBoE (InfiniBand over Ethernet). The RoCE standard replaces the TCP/IP network layer with an IB network layer on top of the Ethernet link layer and does not support IP routing. The Ethernet type is 0x8915. In RoCE, the link layer header of the infiniband is removed and the GUID used to represent the address is converted to an Ethernet MAC. infiniband relies on lossless physical transport, and RoCE relies on lossless Ethernet transport.

RoCEv2

Since the RoCEv1 data frame does not have an IP header, it can only communicate within a 2-tier network. To solve this problem, in 2014 IBTA proposed RoCE v2, which extends RoCEv1 by replacing GRH (Global Routing Header) with a UDP header + IP header. Because RoCE v2 packets are routable at Layer 3, they are sometimes referred to as "Routable RoCE" or "RRoCE" for short. As shown in the figure below.

RoCE technology can be implemented through a common Ethernet switch, but the server needs to support RoCE network cards. Since RoCEv2 is a UDP protocol, although the UDP protocol is relatively high efficiency, but unlike the TCP protocol, there is a retransmission mechanism to ensure reliable

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 X - Compares RDMA Network Technologies

transmission, once there is a packet loss, must rely on the upper layer of the application found and then do retransmission, which will greatly reduce the transmission efficiency of RDMA. So in order to play out the true effect of RoCE, it is necessary to build a lossless network environment for RDMA without losing packets.

RoCE can run in both lossless and compromised network environments, called Resilient RoCE if running in a compromised network environment, and Lossless RoCE if running in a compromised network environment.

RDMA is more and more widely used in market, especially in OTT companies. There have been tens of thousands of servers supporting RDMA, carrying our databases, cloud storage, data analysis systems, HPC and machine learning applications in production. Applications have reported impressive improvements by adopting RDMA [13]. 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.

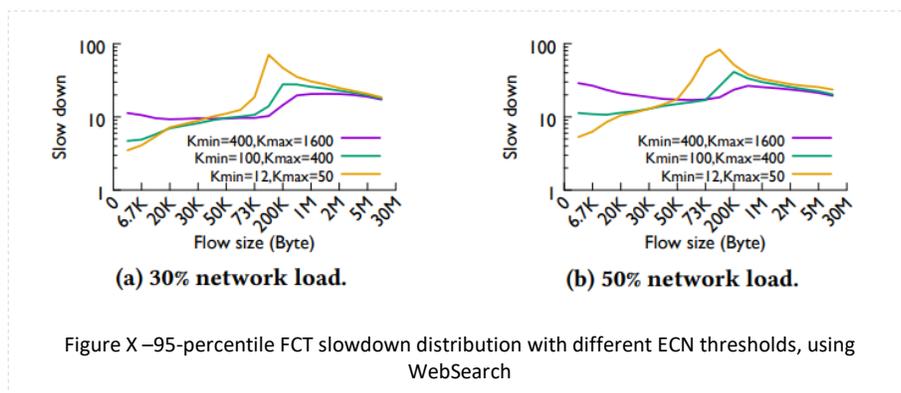
4 Challenges with today's data center network

High bandwidth and low latency tradeoff

When we talk about data center network performance, low latency and high bandwidth are always the ultimate goals of data center network users. In the previous chapter, we introduced the features of the RDMA protocol. The zero copy characteristic help applications can perform data transfers without the involvement of the network software stack. Data is sent and received directly to the

buffers without being copied between the network layers. In addition, the RDMA applications can perform data transfers directly from user-space without kernel involvement, which is called kernel bypass. Though “high bandwidth and low latency” has become a “catchphrase” of RDMA, we find it is practically hard to achieve them simultaneously in one DCQCN configuration. This is because for consistently low latency the network needs to maintain steadily small queues in buffers (which means low ECN marking thresholds), while senders will be too conservative to increase flow rates if ECN marking thresholds are low.

Experimentation shows the tradeoff still exists after varying algorithms, parameters, traffic patterns and link loads [13]. Figure X approximately shows the issue. It shows the FCT slowdown with different ECN marking thresholds (Kmin, Kmax) in switches and WebSearch as input traffic loads. Figure Xa shows that when we use low ECN thresholds, small flows which are latency-sensitive have lower FCT, while big flows which are bandwidth-sensitive suffer from larger FCT. The trend is more obvious when the network load is higher (Figure Xb when the average link load is 50%).



For instance, the 95th-percentile RTT is about $150\mu s - 30 (\text{slowdown}) \times 5\mu s$ (baseline RTT) — when $K_{min} = 400KB$, $K_{max} = 1600KB$, which is a lot worse than the ML application’s requirement ($<50\mu s$). We have tried out different DCQCN parameters, different average link loads and different traffic traces, and the trade-off between bandwidth and latency remains.

- ✓ It’s difficult to achieve high bandwidth and low latency simultaneously
- ✓ Experimentation shows the tradeoff still exists after varying algorithms, parameters, traffic patterns and link loads
- ✓ Reason explanation about why tradeoff exists

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 must be no packet loss due to buffer overflow at the switches [14]. The RoCE protocol is based on UDP and requires Priority-based Flow Control (IEEE Std 802.1Q-2018, Clause 36 [15]) to ensure that no packet loss occurs in the entire data center network. Packet loss in the network severely affects the RoCE service performance. As shown in Figure X, the RoCE service throughput decreases rapidly with an increase in the packet loss rate. As

little as one thousandth of packet lost in the network, decreases RoCE service performance by roughly 30%.

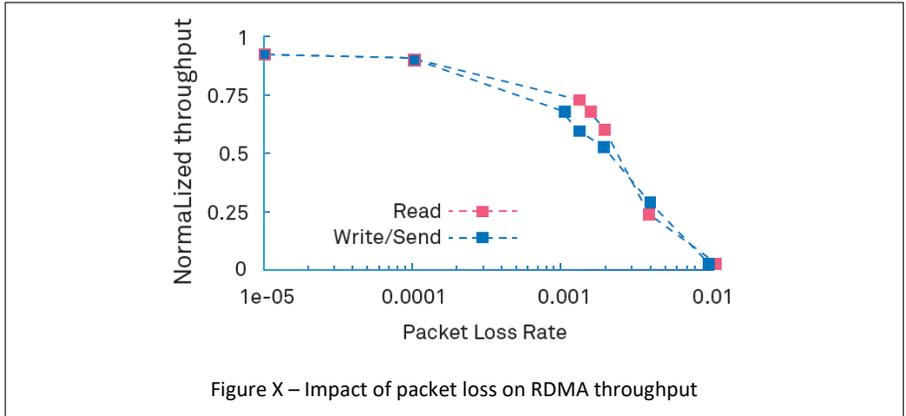


Figure X – Impact of packet loss on RDMA throughput

Priority-based Flow Control (PFC) prevents packet loss due to buffer overflow by pausing the upstream sending device when 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.

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 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. These paths naturally contain loops in the physical topology. A PFC deadlock in the network can completely halt network traffic.

Commented [PC2]: We should include a number of references that describe the issue at this point.

Commented [PC3R2]: ❖Tagger: Practical PFC Deadlock Prevention in Data Center Networks ❖David Lee Paper and other related references to deadlocks

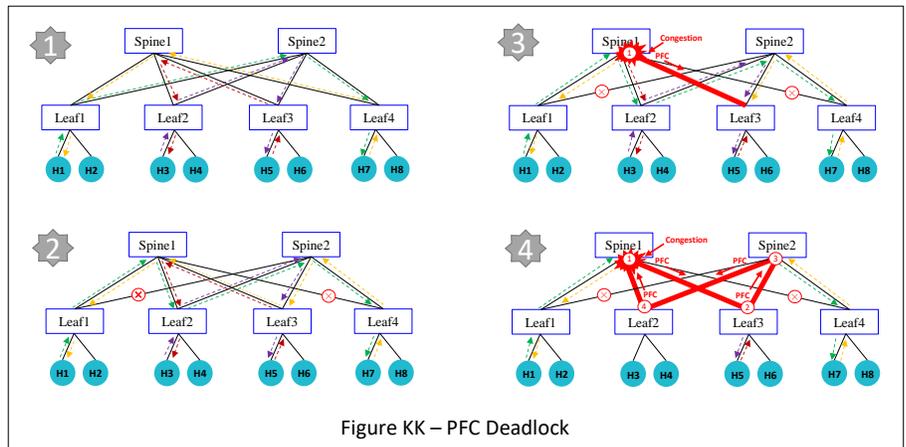


Figure KK – PFC Deadlock

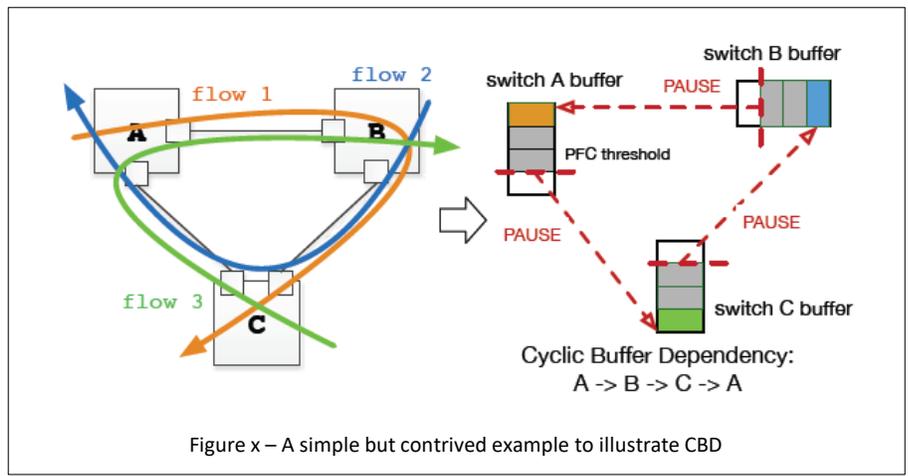
Consider the example in Figure KK. The figure shows four phases of PFC deadlock creation. In phase 1, four flows are equally load balanced across the Clos fabric and the network is running smoothly. In phase 2, the red cross indicates a transient or permanent fault in the topology, such as link failure, 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 exponentially. Achieving larger scale and optimal performance is a key objective of the intelligent lossless data center network of the future.

A number of techniques have been explored to address the PFC deadlock problem [16]. <<Finish solution discussion>>

<< Consider including the following additional text that explains the CDB issue if necessary. The text above Figure KK tries to summarize CDB as concisely as possible. If we want more detailed description of the CDB issue, consider inserting this text somewhere above Figure KK and its associated paragraph >>

Before introducing detail information about PFC deadlock problem, a new term should be described first. It is called Cyclic Buffer Dependency (CBD) [17]. CBD means the occupied buffers are waiting for each other in a loop. A simple example is shown in Figure x.



Congestion control issues in large-scale data center networks

With the popularization of RDMA applications, networks face a new challenges: How to implement large-scale RDMA applications? When the RDMA technology is initially used by customers, customers are conservative, so they use RDMA only in the POD. In this way, the application scenarios of RDMA are limited to dedicated scenarios, such as computing or storage networks.

However, with the application of RDMA, its performance advantages are obvious. The customer began to think about a new question: How to use the RDMA network across PODs? Figure X is a schematic diagram of a large-scale RoCE network application. As shown in the picture, the customer's entire data center network is deployed based on the Ethernet. The computing cluster and storage cluster use the RDMA protocol while the X86 server cluster use traditional TCP/IP protocol.

In the large-scale data center network scenario shown in figure x above, a new scenario is inevitable. That is TCP and RoCE traffic is transmitted mixed together in network. Currently, there are several traffic hybrid scenarios.

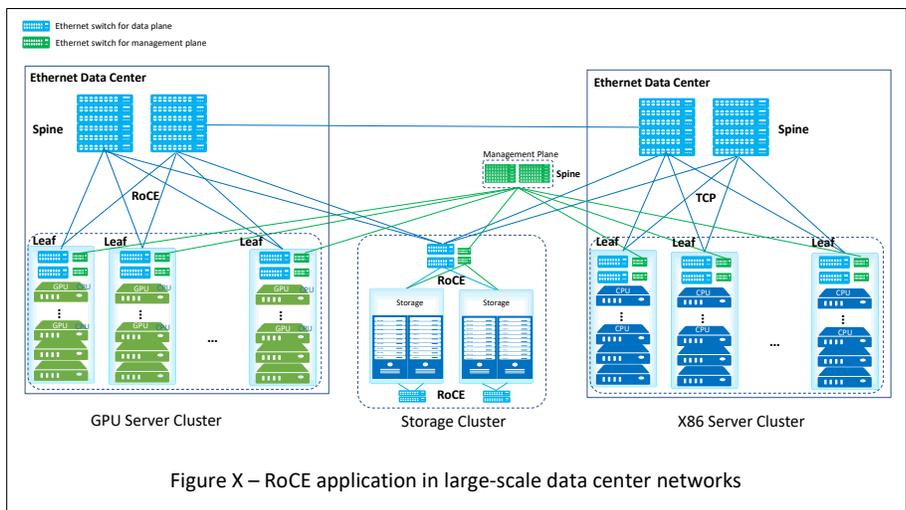


Figure X – RoCE application in large-scale data center networks

Scenario 1: For example, when an end user needs to invoke some stored data from cloud, user needs to submit a request to the web service through TCP. Then the web service cluster will invoke the storage node through TCP protocol. When storage cluster receives the request of data reading, it uses RoCE protocol to handle reading work in order to make sure the network has very high IOPS. After obtaining data user needed, storage cluster will send data back to web server and to end user finally, using TCP.

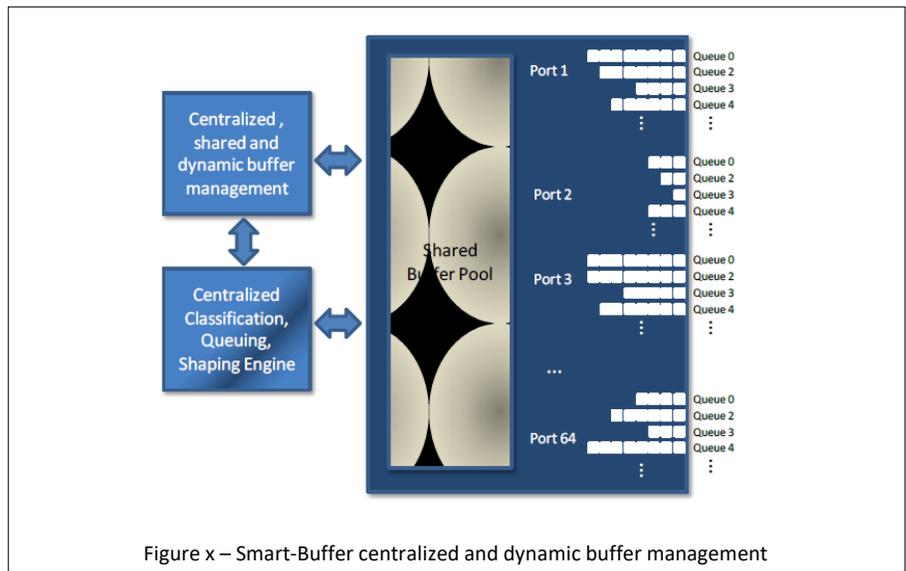
Scenario 2: The service network of the computing/storage cluster uses the RoCE network, while the management traffic (from the management switch in green in figure x or from the SDN controller) in the DC is basically based on the TCP protocol. Therefore, the two types of traffic may run together in the cluster.

Scenario 3: Although RoCE has been gradually used in large scale computing and storage network. However there are still TCP-based storage and computing data centers. Therefore, in large-scale data center applications, there may be multiple combinations of TCP or RoCE between computing and computing, between storage and storage, and between computing and storage.

A large amount of TCP and RoCE traffic exists on the network. Therefore, the switch needs to use different queues to schedule different types of traffic. The IEEE 802.1P and IEEE802.1Qaz (ETS) mechanisms are used for scheduling. These mechanisms need to be implemented on the switch chip. The switch chip of the current data center encounters the Switch packet buffer performance and cost tradeoff problem. Microburst is a common problem in current data center network. To prevent packet loss caused by microbursts, each queue on each port of the switch should be configured with enough buffers to absorb the burst. On the other hand, for the chip implementability, the cost is too high for a switch to implement purely static per-port buffer allocation schemes.

Based on the multi-queue technology, mainstream switching chip vendors use a smart buffer mechanism to solve the tradeoff problem. One of the core ideas of smart buffer is to put forward a concept of dynamic buffer. Smart-buffer mechanisms attempt to optimize buffer utilization and burst absorption, Dynamic sharing and self-tuning is transparently enabled across all ports [18].

The Smart-Buffer architecture, as shown in Figure 1, takes into consideration that congestion in a typical data center environment is localized to a subset of egress ports at any given point in time and realistically never happens on all ports simultaneously. This enables its centralized on-chip buffer to be right-sized for overall cost and power; at the same time, the buffer is dynamically shareable and weighted towards congested ports exactly when needed using self-tuning thresholds. Dynamic sharing and self-tuning is transparently enabled across all ports, regardless of the processing pipeline or switching core to which the ports belong.



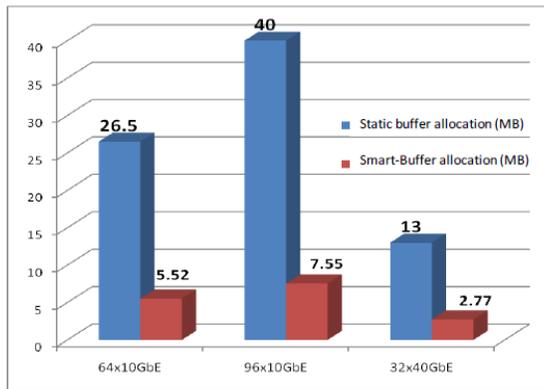


Figure X – Smart-Buffer delivers up to 5x better packet buffer utilization

Switch Chip Company did some experiments to compare the effects of static and dynamic buffers. Figure X shows the buffer utilization comparison.

Contrasted with static per-port buffer allocation schemes found in other switch device architectures, Smart-Buffer significantly improves buffer utilization and enables maximum performance per unit of packet buffer for data center applications. In data center traffic tests and

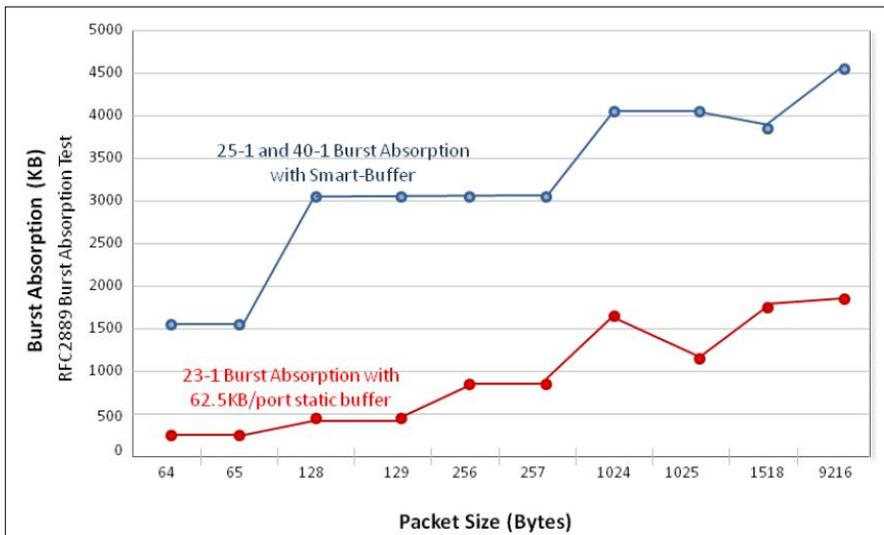


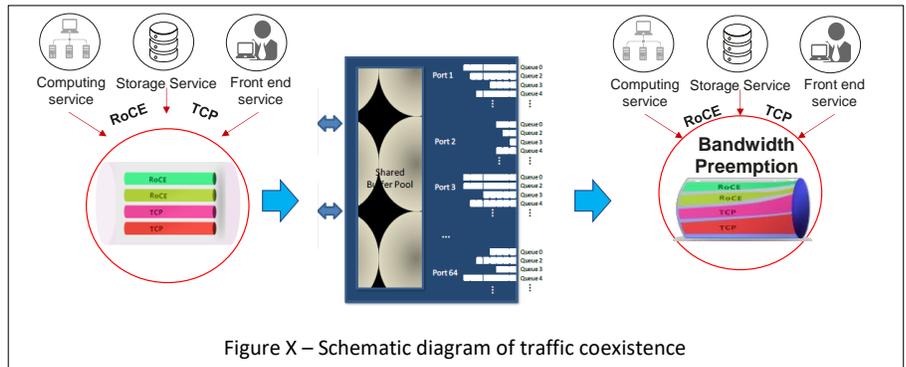
Figure X – Burst Absorption Test: 3 to 6x worse burst absorption capacity in 10GbE switch with per-port static buffer architecture as compared to a Smart-Buffer based switch

simulations, results showed, on average, five times better packet buffer utilization with Smart-Buffer.

For burst problem, the smart buffer mechanism improve 3 to 6 times better than the static buffer. In Figure x, the red line denotes the industry-standard RFC 2889 burst absorption performance for a 10GbE switch architecture with per-port static buffering. The data is produced under a 23-to-1 burst condition (23 ingress ports bursting to a single egress destination).

In summary, smart-buffer delivers optimal buffer utilization and burst absorption for data center workloads by taking a holistic approach to buffer management – using real-life data center traffic scenarios to maximize overall throughput and lossless behavior.

However, for the TCP and RoCE flow mix scenario, the smart buffer mechanism of switch chip may cause an unexpected problem. That is SLA (Service Level Agreement) cannot be guarantee when TCP and RoCE traffic coexists. Because of different congestion control mechanism, the TCP flows and preempt the bandwidth of RoCE flows, even when using separate traffic classes. Figure X shows the schematic diagram of the problem. At the beginning, network bandwidth is allocated to different flows based on service requirements in initialization phase. But due to shared buffer architecture and different congestion control methods, TCP preempts RoCE bandwidth. The RoCE flow delay increased by 100 times (ms level).



ODCC conducted several tests to verify the problem of traffic coexistence. Table X is the test specification on flow mix scenario.

<< Test tables omitted – too much detail for this paper, need to reference test results, papers, published ODCC documents. Summarize the test findings instead of referencing specific details. >>

The data in red in the preceding test results is abnormal performance data. From test results that we can see:

In a scenario where both RoCE and TCP flows are transmitted, TCP flows preempt the bandwidth of RoCE flows. In other words, a traffic bandwidth proportion of the two at the receiver is different from that is set at the sender. Especially when the TCP traffic proportion increases (ROCE:TCP=9:1, message size = 64 KB), the effect of RoCE preemption is especially obvious, and the traffic proportion deviation reaches more than 50%. As described above, the smart buffer mechanism of the chip enables the TCP and RoCE to share the shared buffer. In addition, the Congestion Control

mechanisms for TCP and RoCE traffic are different. The TCP packet loss threshold is higher than the CNP threshold of RoCE. As a result, TCP preempts more buffers. These two reasons cause it easier for TCP to preempt the bandwidth. Therefore, TCP congestion control is more aggressive than RoCE. To ensure the QoS of RoCE traffic, a proper ECN threshold for RoCE need to be configure.

As the number of QPs on the data center RDMA network increases, when the proportion of RoCE is small (e.g. TCP:RoCE = 9:1), the latency increases sharply (from microsecond level to millisecond level).In this case, all packets are marked with ECN, the queue is still overstocked, and the ECN becomes invalid. When the number of QP flows is large (for example, 4096 in the test), the bandwidth allocated to each flow is small. As a result, the packet interval of each flow (that is, the minimum interval of CNP packets that can be obtained by the flow) is greater than the rate increase interval. In this case, the rate of the flow is still increased even network is already congested. As a result, the rate control fails and the delay increases sharply. In the test result, when TCP:RoCE is 9:1 and the number of QPs is 4096, the maximum network latency is over 9 ms. This problem is easy to occur in a large-scale data center. For example, if there are 64 servers in a data center, at least 63 QPs are required for connecting one server to other 63 servers. At least $63 \times 64 = 4032$ QPs are required for 64 servers.

- ✓ How large scale today's data center is?
- ✓ Use cases for TCP and RoCE flows mixture
- ✓ Smart-buffer mechanisms in mainstream switch chips
- ✓ SLAs cannot be guarantee when TCP and RoCE traffic coexists

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

5

New technologies to address new data center problems

Approaches to PFC storm elimination

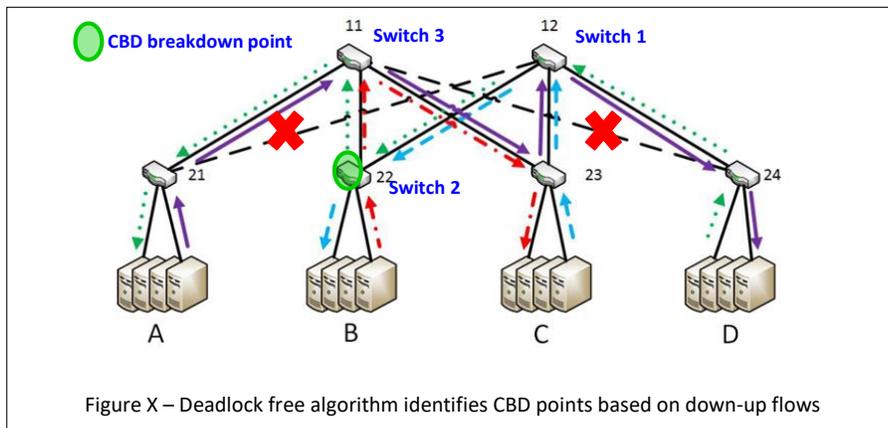
Although traffic on the Clos network is up-down and loop-free, rerouting occurs when a transient or permanent link fault occurs, down-up traffic may be generated. (In large-scale data centers, down-up traffic is common. According to Microsoft's paper, the proportion of down-up traffic is about 10-5 [17].) The reroute probability is around 10-5. Though 10-5 is not a big number, given the large traffic volume and the large scale data center networks, the deadlocks due to packet reroute do not just exist in paper designs. They are real! Especially with the scale of the data center

RDMA network increases, RDMA starts to be deployed across PODs in large data centers. The larger the scale, the higher the probability of PFC deadlock, and the lower the service availability.

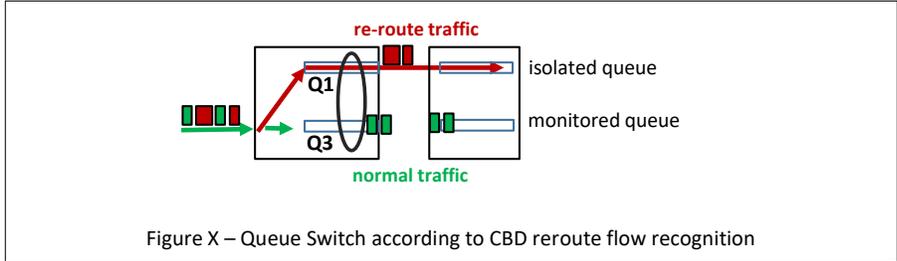
ODCC proposes a mechanism to prevent the deadlock problem. The deadlock-free technology and algorithm is a PFC-based deadlock prevention technology, which prevents deadlocks and eliminates PFC-based deadlocks in data center network. According to the preceding description, an important prerequisite for PFC deadlock is the occurrence of CBD loops. Therefore, to prevent PFC deadlock, it is a very important to discover and avoid CBD loops.

The core idea of the deadlock-free algorithm is to break the CBD deadlock. An innovative distributed topology role auto-discovery protocol is used to identify network locations and roles and identify abnormal traffic. The deadlock free algorithm mechanism consists of two algorithms: control plane algorithm and forwarding plane algorithm.

The following figure X shows the working principle. For example, in the figure, after the link from 24 to 11 goes down, green traffic is rerouted from server D to TOR24, to switch 1, to switch 2, to switch 3, to TOR switch 21, and finally to server A. This is the situation of down-up flow as we describe above. So the switch 2 can recognize itself as a CBD point. We propose a method to implement



distributed self-learning of switches and obtain the switch level and port uplink/downlink attribute by using the extended LLDP protocol and algorithm. All switches learn the level and uplink/downlink attributes of their locations, and maintain and update the attributes automatically. This technology now is standardizing in the IEEE 802.1Qcz. More detail information will be discussed in chapter 6.



After recognizing the CBD point, the forwarding plane is responsible for breaking the CBD. According to our description in the preceding chapter, when a traffic loop occurs and a CBD is formed, flows from each switches that in this loop are all in the same queue. Therefore, to break the CBD point, packets need to be switched to another queue. Figure X illustrates the process of queue switch. We know that switch 2 is the CBD point so the flow queue need to be switched. Assume there are two flows in the queue, the red one is re-route traffic that is determined by down-up reroute path. The algorithm will switch the red traffic into another queue. The flow that change to another queue will lead the elimination of PFC deadlock. Different flows can pass by through different queues.

To verify the effect of this algorithm, ODCC conduct a test to demonstrate the performance of the deadlock free algorithm. CAICT, Baidu, Meituan, China Telecom, China Mobile, Huawei, Cisco, Mellanox, H3C, Centec, Keysight companies participate in this test. We still use the same test topology in the Table X and enable the switch's deadlock-free feature.

Figure X shows the test result of deadlock free. Without deadlock-free mechanism, the flows that cause CBD loop are all in the same queue (queue 3, shown in figure ddd). This is the main reason why PFC deadlock occurs. After PFC deadlock free feature is configured, the traffic causing deadlock switch from queue 3 to queue 4. This is due to the algorithm we described above, which implements automatic switching of the deadlock traffic queue.

```

<Leaf3> dis q q s int 100g 1/0/2
Queue  CIR/PIR      Passed      Pass Rate      Dropped      Drop Rate      Drop Time
(% or kbps) (Packets/Bytes) (pps/bps) (Packets/Bytes) (pps/bps)
-----
0      0              0           0              0             0              -
100000000
-----
1      0              0           0              0             0              -
100000000
-----
2      0              0           0              0             0              -
100000000
-----
3      0              73844526   1447931        0             0              -
100000000      102404595868  16063466016
-----
4      0              55720164   1092552        0             0              -
100000000      86852595976  13623936616
-----
5      0              0           0              0             0              -
100000000
-----
6      0              60         1              0             0              -
100000000      4840         736
-----
7      0              0           0              0             0              -
100000000
-----

```

Figure X – traffic causing deadlock switch from queue 3 to queue 4 to avoid deadlock

In addition to the queue automatic switch, the test result shows that no deadlock occurs on the network. We obey the test step again according to the test specification in Table X. The test output shows that no PFC deadlock occurs on the network after the PFC deadlock free function is enabled.

```
<Leaf2> dis int b | i up
PHY: Physical
*down: administratively down
^down: standby
(l): loopback
(s): spoofing
(b): BFD down
(e): ETHOAM down
(d): Dampening Suppressed
(p): port alarm down
(dl): DLDLP down
(c): CFM down
(sd): STP instance discarding
InUti/OutUti: input utility rate/output utility rate
Interface      PHY      Protocol  InUti  OutUti  inErrors  outErrors
100GE1/0/1     up       up        0.01%  46.76%  0         0
100GE1/0/2     up       up        46.76% 0.01%  0         0
25GE1/0/1      up       up        99.98% 99.97%  0         0
LoopBack0      up       up(s)     0%     0%     0         0
MEth0/0/0      up       up        0.01%  0.01%  0         0
NULL0          up       up(s)     0%     0%     0         0
Vlanif20      up       up        --     --     0         0
```

Figure X – No Deadlock occurs, and the network traffic transmission is normal

- ✓ 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

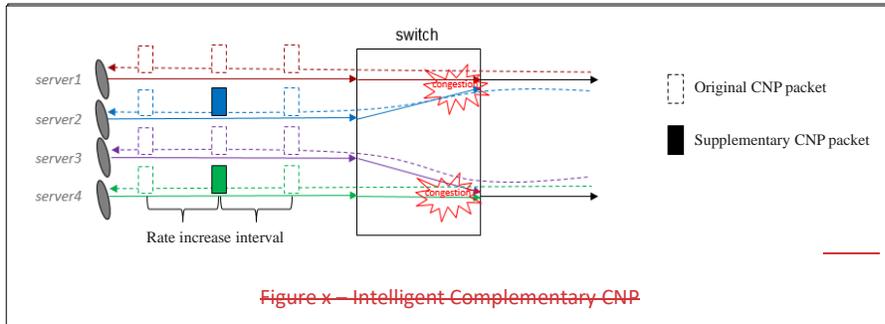
Improving Congestion Notification

According to the preceding test in table X, in large-scale data center when the RoCE traffic proportion is small in flow coexistence transmission scenario, the latency increases sharply (from dozens of microseconds to milliseconds) and the bandwidth QoS cannot be guaranteed. In addition, more QP connections of RoCE flows will aggravate the problem.

If these problems occur when traffic coexist, the switch sends a large number of PFC pause packets to the upstream device. As a result, packets are stacked on the outbound interface of the switch, causing a long delay.

After analyzing the causes of the high latency problem, we find that when the network is congested, even if all RoCE packets are marked with ECN, the queue is still overstocked, and the ECN becomes invalid. The root cause is that when network is congested, the sender cannot receive sufficient CNP notification packets. As a result, the sender still sends data, causing queue stacking and latency increasing.

To ensure the performance of the RoCE network, PFC is widely used in data centers to ensure no packet loss and DCQCN is used as the congestion control algorithm. As in Figure X, the DCQCN algorithm consists of three parts: a reaction point (RP), a switch (CP, congestion point), and a notification point (NP).



CP, Congestion Point

On the switch side, if the number of packets in an egress queue exceeds the threshold, the packets are marked with ECN. The ECN marking probability is determined by the queue length. When the queue length is less than K_{min} , traffic is not marked. When the queue depth is greater than K_{max} , all packets passing through the queue are marked. When the queue depth is between K_{min} and K_{max} , the marking probability increases according to queue depth.

NP, Notification Point

When a data packet with an ECN flag arrives at a receiver, it indicates that congestion occurs in a network. Receiver needs to send a CNP packet back to the transmit end to transfer congestion information. If a data flow packet with an ECN flag arrives and no CNP has been sent in the past N microseconds, in this case, the receiver immediately sends a CNP packet. With enhancement of a current data center network adapter chip capability, N may be set to 0, that is, a receive end may respond to each packet with an ECN flag, and send a CNP packet.

RP, Reaction Point

When the transmit end receives a CNP, it decreases the current rate and updates the rate decrease factor. If the transmit end does not receive any CNP within a specified period, it increases the transmit rate based on a certain algorithm.

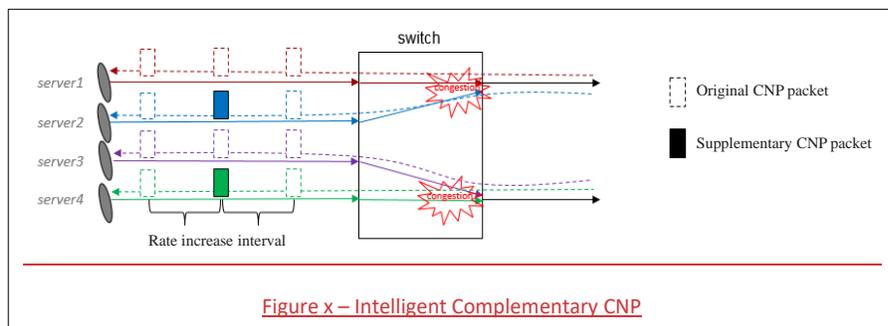
With the increase of the data center network scale, when the flow scale is large, the average bandwidth allocated to each flow is small. For the flow that is congested, the packet interval of each flow (that is, the minimum interval for the flow to obtain CNP packets) may be greater than the rate increase interval. So that the rate of the flow that is still in the congested state is increased, as a result, the speed control convergence fails.

For example, if the port rate of the switch is 25 Gbit/s and the number of RoCE flows is 300, the average rate of each RoCE flow is as follows: 25 Gbit/s (port bandwidth) /300 (number of flows) = 80 Mbit/s. In this case, a 4 KB packet interval (MTU for NIC) is generated every 400 us, if the speed-up interval time of NIC is 300us (default value in commercial NIC), so the time of packet interval is greater than the rate increase/decrease interval, the speed control fails when the switch port is congested.

To solve this problem, ODCC provides a solution to intelligently supplement the rate of the CNP packet based on the congestion level of the port, interval of the received CNP packet, and interval of the DCQCN rate increase on the network side.

This algorithm proposes a method for network congestion awareness and network congestion control.

After receiving CE packets, the CP determines whether a queue is congested based on the frequency of CE packets and determines whether a flow is congested based on the sequence number of CE packets. If a queue is congested and a flow is congested, the CP proactively supplements CNP packets based on the intervals for CE packets and congestion control rate increase. The problem that the sending rate increases when congestion occurs on the switch port is solved. The CNP packet is supplemented to cause the conditions are not met for triggering rate increase. It can prevent unexpected rate increase when the port is congested, thereby solving the problem of the rate control failure. The CNP operation is performed only when the port is severely congested and no CNP is performed for a long time. Therefore, the rate increase and throughput are not affected when the DCQCN is in normal state. The solution is shown in Figure x.



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

- ✓ Improved Explicit Congestion Notification
- ✓ Enhanced version of Quantized Congestion Notification (originally IEEE 802.1Qau)
- ✓ Intelligent Methods of improving QoS support in mixed traffic environments
- ✓ Test verification (ODCC lossless DCN test specification and result)

Configuration complexity of congestion control algorithms

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.

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 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.

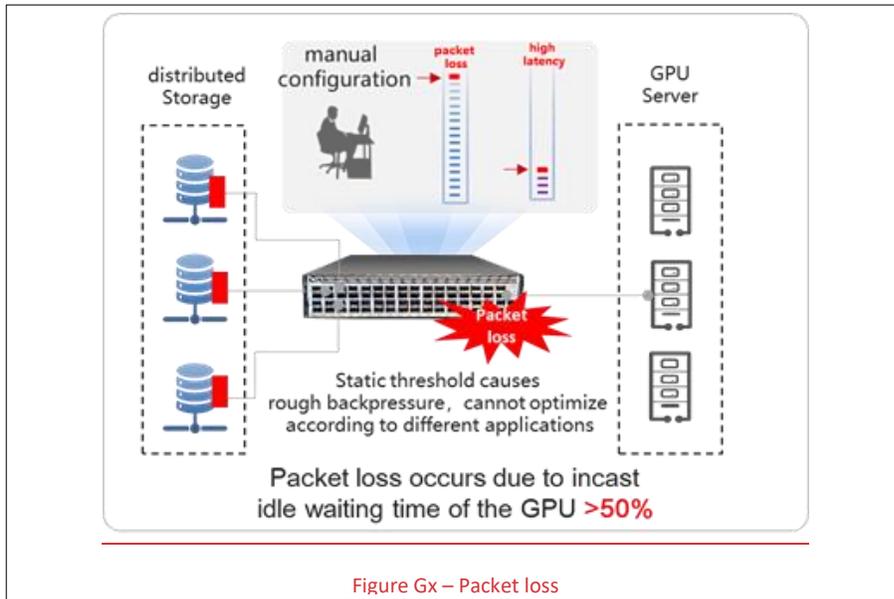
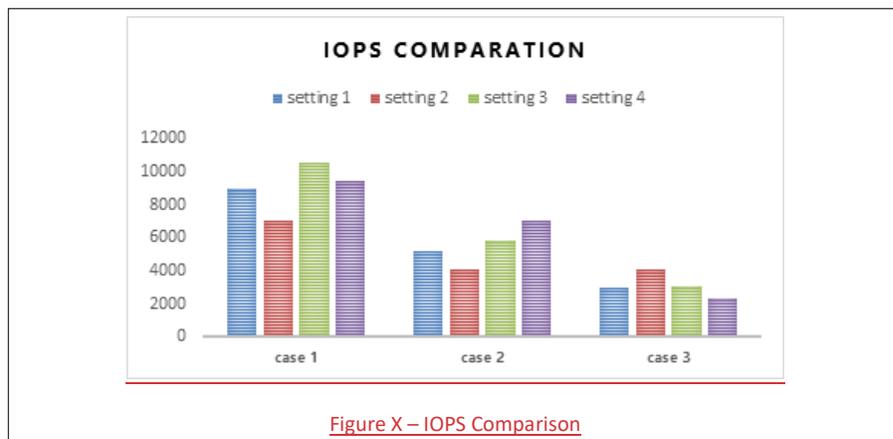


Figure Gx – Packet loss

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.



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.

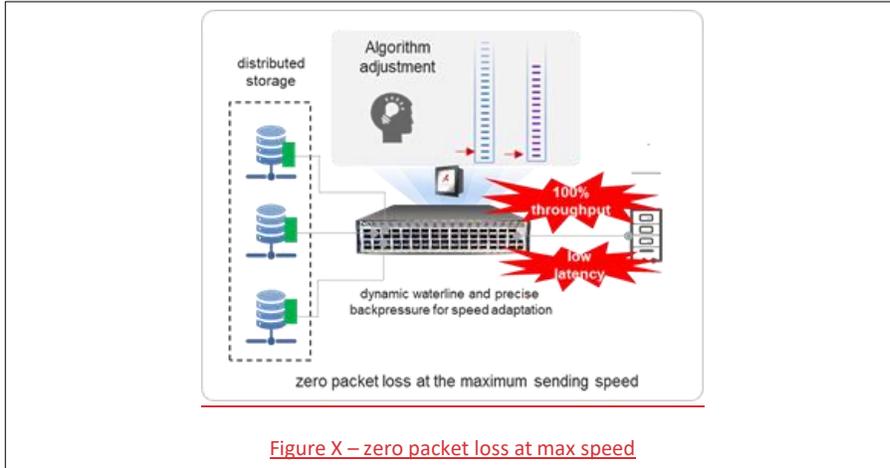


Figure X – zero packet loss at max speed

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%.

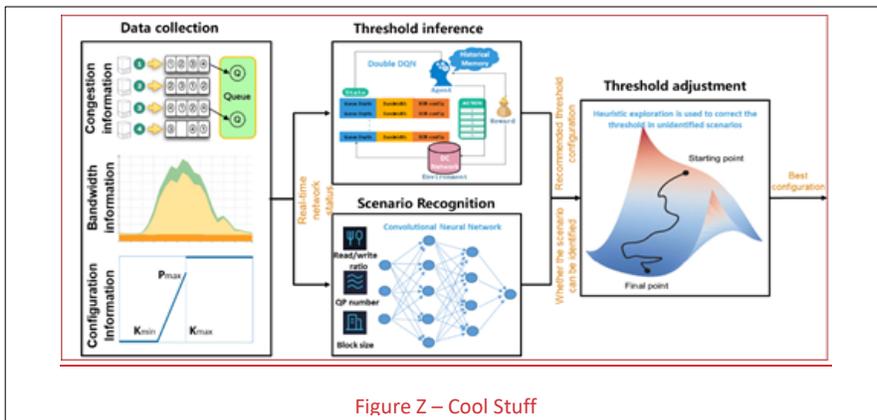


Figure Z – Cool Stuff

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

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 [19].

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 [15]). 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 [14].

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)

Buffer optimization of lossless queues

- ~~✓ Intelligent headroom calculation~~
- ✓ Self-adaptive headroom configuration

6

Standardization Considerations

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



ords...

Conclusion



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> [Accessed 14 05 2020]. Formatted: Position: Horizontal: Left, Relative to: Column, Vertical: 0", Relative to: Paragraph, Horizontal: 0.13", Wrap Around
- [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]. Formatted Table
- [3] J. Wiles, "Mobilize Every Function in the Organization for Digitalization," Gartner, 03 December 2018. Available: <https://www.gartner.com/smarterwithgartner/mobilize-every-function-in-the-organization-for-digitalization/>. [Accessed 10 June 2020]. Formatted: Position: Horizontal: Left, Relative to: Column, Vertical: 0", Relative to: Paragraph, Horizontal: 0.13", Wrap Around
- [4] 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]. Formatted: Position: Horizontal: Left, Relative to: Column, Vertical: 0", Relative to: Paragraph, Horizontal: 0.13", Wrap Around
- [5] 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]. Formatted: Position: Horizontal: Left, Relative to: Column, Vertical: 0", Relative to: Paragraph, Horizontal: 0.13", Wrap Around
- [6] Huawei, "AI, This Is the Intelligent and Lossless Data Center Network You Want!," 13 March 2019. Available: <https://www.cio.com/article/3347337/ai-this-is-the-intelligent-and-lossless-data-center-network-you-want.html>. [Accessed 14 05 2020]. Formatted: Position: Horizontal: Left, Relative to: Column, Vertical: 0", Relative to: Paragraph, Horizontal: 0.13", Wrap Around
- [7] E. K. Karupiah, "Real World Problem Simplification Using Deep Learning / AI," 2 November 2017. Available: https://www.fujitsu.com/sg/Images/8.3.2%20FAC2017Track3_EttikanKarupiah_RealWorldProblemSimplificationUsingDeepLearningAI%20.pdf. [Accessed 14 05 2020]. Formatted: Position: Horizontal: Left, Relative to: Column, Vertical: 0", Relative to: Paragraph, Horizontal: 0.13", Wrap Around
- [8] O. Cardona, "Towards Hyperscale High Performance Computing with RDMA," 12 June 2019. Available: https://pc.nanog.org/static/published/meetings/NANOG76/1999/20190612_Cardona_Towards_Hyperscale_High_Performance_Computing_with_RDMA_High_v1.pdf. [Accessed 14 05 2020]. Formatted: Position: Horizontal: Left, Relative to: Column, Vertical: 0", Relative to: Paragraph, Horizontal: 0.13", Wrap Around

- [9] J. L. Jacobi, "NVMe SSDs: Everything you need to know about this insanely fast storage," 10-Ma [Online]. Available: <https://www.pcworld.com/article/2899351/everything-you-need-to-know-nvme.html>. [Accessed 14 05 2020]. **Formatted:** Position: Horizontal: Left, Relative to: Column, Vertical: 0", Relative to: Paragraph, Horizontal: 0.13", Wrap Around
- [10] M. Alipio, N. M. Tiglaio, 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. **Formatted:** Position: Horizontal: Left, Relative to: Column, Vertical: 0", Relative to: Paragraph, Horizontal: 0.13", Wrap Around
- [11] Z. Jai, Y. Kwon, G. Shipman, P. McCormick, M. Erez and A. Aiken, "A distributed multi-GPU system for processing," in *VLDB Endowment*, 2017. **Formatted:** Position: Horizontal: Left, Relative to: Column, Vertical: 0", Relative to: Paragraph, Horizontal: 0.13", Wrap Around
- [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 05 2020]. **Formatted:** Position: Horizontal: Left, Relative to: Column, Vertical: 0", Relative to: Paragraph, Horizontal: 0.13", Wrap Around
- [13] Y. Li, R. Miao, H. H. Liu, Y. Zhuang, F. Feng, L. Tang, Z. Cao, M. Zhang, F. Kelly, M. Alizadeh and M. Yu, "high precision congestion control," in *Proceedings of the ACM Special Interest Group on Data Communication (SIGCOMM '19)*, New York, NY, USA, 2019. **Formatted:** Position: Horizontal: Left, Relative to: Column, Vertical: 0", Relative to: Paragraph, Horizontal: 0.13", Wrap Around
- [14] C. Guo, H. Wu, Z. Deng, G. Soni, J. Ye, J. Padhye and M. Lipshteyn, "RDMA over Commodity Ethernet in *In Proceedings of the 2016 ACM SIGCOMM Conference (SIGCOMM '16)*, 2016. **Formatted:** Position: Horizontal: Left, Relative to: Column, Vertical: 0", Relative to: Paragraph, Horizontal: 0.13", Wrap Around
- [15] IEEE, IEEE Std 802.1Q-2018, IEEE Standard for Local and Metropolitan Area Networks — Bridges and Networks, IEEE, 2018. **Formatted:** Position: Horizontal: Left, Relative to: Column, Vertical: 0", Relative to: Paragraph, Horizontal: 0.13", Wrap Around
- [16] S. Hu, Y. Zhu, P. Cheng, C. Guo, K. Yan, J. Padhye and K. Chen, "Deadlocks in datacenter networks: Why form, and how to avoid them," in *Proceedings of the 15th ACM Workshop on Hot Topics in Networks*. **Formatted:** Position: Horizontal: Left, Relative to: Column, Vertical: 0", Relative to: Paragraph, Horizontal: 0.13", Wrap Around
- [17] 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 Networks, Experiments and Technologies (CoNEXT '17)*, 2017. **Formatted:** Position: Horizontal: Left, Relative to: Column, Vertical: 0", Relative to: Paragraph, Horizontal: 0.13", Wrap Around
- [18] 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/doc/and-downloads/collateral/etp/SBT-ETP100.pdf>. [Accessed 24 June 2020]. **Formatted:** Position: Horizontal: Left, Relative to: Column, Vertical: 0", Relative to: Paragraph, Horizontal: 0.13", Wrap Around
- [19] 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].