Data Center Congestion Control – Where's the best fit in IETF/IRTF?

Paul Congdon (Tallac Networks)

IETF-106, Singapore, November 2019

IETF Note Well

https://www.ietf.org/about/note-well/

This is a reminder of IETF policies in effect on various topics such as patents or code of conduct. It is only meant to point you in the right direction. Exceptions may apply. The IETF's patent policy and the definition of an IETF "contribution" and "participation" are set forth in BCP 79; please read it carefully.

As a reminder:
By participating in the IETF, you agree to follow IETF processes and policies.

If you are aware that any IETF contribution is covered by patents or patent applications that are owned or controlled by you or your sponsor, you must disclose that fact, or not participate in the discussion.

As a participant in or attendee to any IETF activity you acknowledge that written, audio, video, and photographic records of meetings may be made public.

Personal information that you provide to IETF will be handled in accordance with the IETF Privacy Statement.

As a participant or attendee, you agree to work respectfully with other participants; please contact the ombudsteam (https://www.ietf.org/contact/ombudsteam/) if you have questions or concerns about this.

Definitive information is in the documents listed below and other IETF BCPs. For advice, please talk to WG chairs or ADs:

- BCP 9 (Internet Standards Process)
- BCP 25 (Working Group processes)
- BCP 25 (Anti-Harassment Procedures)
- BCP 54 (Code of Conduct)
- BCP 78 (Copyright)
- BCP 79 (Patents, Participation)
- https://www.ietf.org/privacy-policy/ (Privacy Policy)

Some History

- IETF-101
  - Introduced TSVWG and ICCRG to IEEE P802.1Qcz on Congestion Isolation
- IETF-103
  - Joint IETF / IEEE 802 workshop on Data Center Networking including topics on congestion control
- IETF-104
  - Side meeting on Hyperscale HPC/RDMA – 9 attendees – All discussion
- IETF-105
  - Side meeting on Large Scale Data Center HPC/RDMA – 35 attendees
    - Ideas explored/discussed for future research:
      - A new UDP based RDMA transport with a reliability/CC shim
      - Injecting more detailed feedback in packets from switches
      - Distinguishing in-network from incast congestion
      - Speeding up congestion notifications from the network
      - Local fast-response congestion mechanisms in switches
    - Drafts discussed;
      - https://tools.ietf.org/html/draft-zhh-tsvwg-open-architecture-00
      - https://tools.ietf.org/html/draft-yueven-tsvwg-dccm-requirements-00

Where to consider DCN CC Research/New-Work

- ICCRG Charter can be interpreted to include DCN
  - "…The ICCRG may also consider congestion and protocol performance problems in general IP networks, i.e., not only on the global Internet. One example of such IP networks are multi-tenant, heterogeneous datacenters,…"
- Congestion control work is on-going in TSVWG
  - However, nothing particularly DCN focused
- Perhaps a new IRTF group is appropriate
- Let's discuss this and status of contributions in our side-meeting

IETF-106 Questions on Congestion Control
in the HPC/RDMA/AI DataCenter Network

- What is needed from NICs for better CC?
  - An open framework to negotiate capabilities and algorithms – OpenCC
    - https://datatracker.ietf.org/doc/draft-zhuang-tsvwg-open-cc-architecture/
- How can the Network participate?
  - An AI model for parameter tuning
    - https://datatracker.ietf.org/doc/draft-zhuang-tsvwg-ai-ecn-for-dcn
  - Fast feedback from the network
    - https://tools.ietf.org/html/draft-even-iccrg-dc-fast-congestion-00
- Other interesting topics
  - Performance metrics for HPC/RDMA/AI networks – like the KPIs discussed by Neal Cardwell in ICCRG yesterday.

Join us for further discussion

- Non-WG IETF Mailing list rdma-cc-interest@ietf.org
  - Subscribe at: https://www.ietf.org/mailman/listinfo/rdma-cc-interest

- Side Meeting: Tuesday 8:30AM – 9:45AM – VIP-A
  - NOTE on side meetings:
    - Open to all
    - Meeting minutes will be posted to rdma-cc-interest@ietf.org
    - Not under NDA of any form

Agenda

- Welcome – Paul Congdon – 10 mins
- Fast Congestion management for Data Centers – Roni Even – 20 mins
  - https://tools.ietf.org/html/draft-even-iccrg-dc-fast-congestion-00
- An Open Congestion Control Architecture for high performance fabrics - Yan Zhuang – 15 mins
  - https://datatracker.ietf.org/doc/draft-zhuang-tsvwg-open-cc-architecture/
- Artificial Intelligence (AI) based ECN adaptive reconfiguration for datacenter networks - Yan Zhuang – 15 mins
  - https://datatracker.ietf.org/doc/draft-zhuang-tsvwg-ai-ecn-for-dcn
- The impact of mixing TCP and RoCEv2 – Yolanda Yu - 10 mins
- How to move forward – All - 5 mins

Fast Congestion management for Data Centers

draft-even-iccrg-dc-fast-congestion-00

Roni Even

Rachel Huang

DC congestion control

- The use case that we are looking at is congestion control for Data Centers, a controlled environment, for the definition see RFC8085 section 3.6.
- Datacenter applications demand high throughput(40Gbps and above) with ultra-low latency of less than 10 microsecond per hop from the network, with low CPU overhead.
- Alternatives for network congestion direction can be classifies as:
  - Based on estimation of network status: Traditional TCP, Timely, ….
  - Network provides limited information: DCQCN using only ECN, SCE, L4S , …
  - Network provides some information: HPCC, …
  - Network provides proactive control: RCP (Rate Control Protocol), …

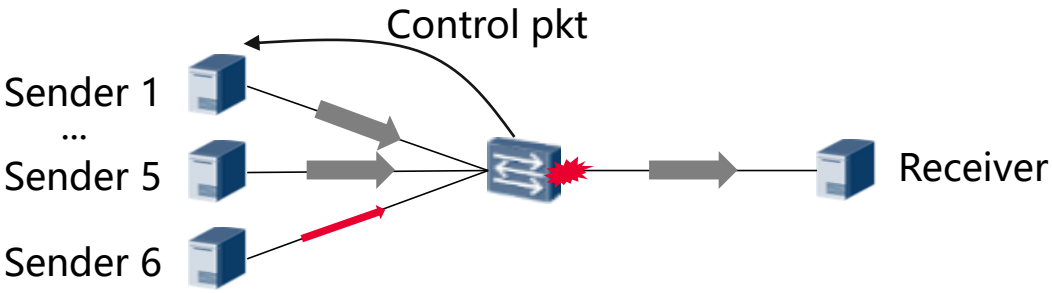
Proposed directions for DC congestion control

- Exploring these two directions
  - Reflect the network status more accurately – add metadata to the forward flow (e.g. using IOAM).
  - Notify the reaction point as soon as possible – report directly from the network to the sender (e.g. IOAM direct export)
- Issues to be addressed are discussed in the draft

Initial tests

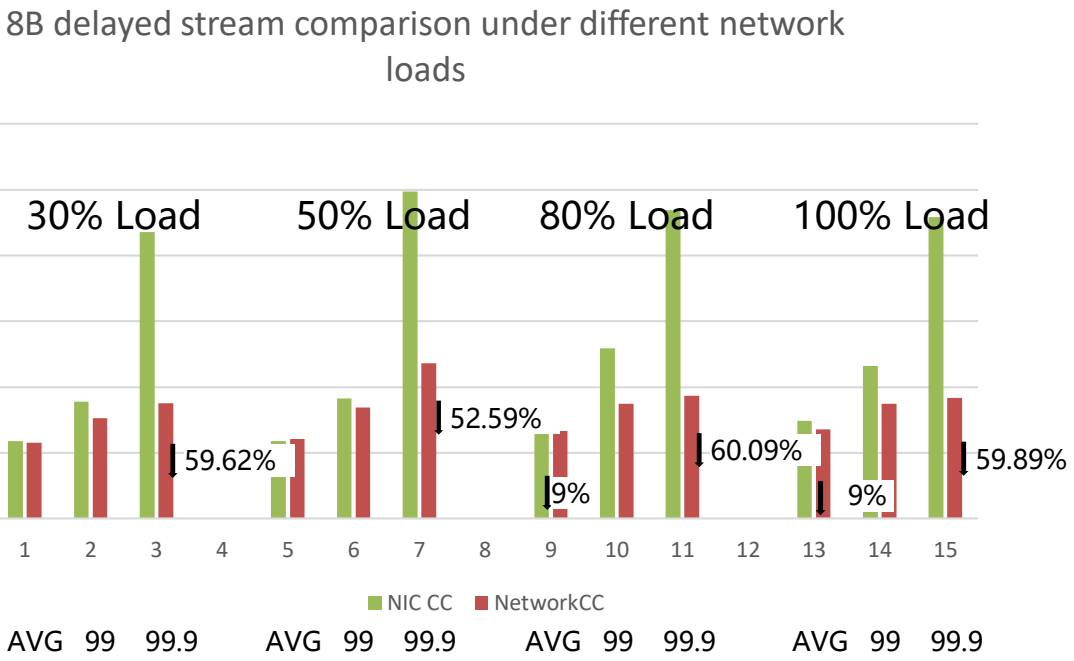
- Tested notification from the Network to the Sender
- For the network CC test used Huawei NICs and Switch running ROCEv2 and using a CNP like message from the Switch to the sender that included rate information.
- For the NIC2NIC ECN and DCQCN was used.

5 to 1 test environment

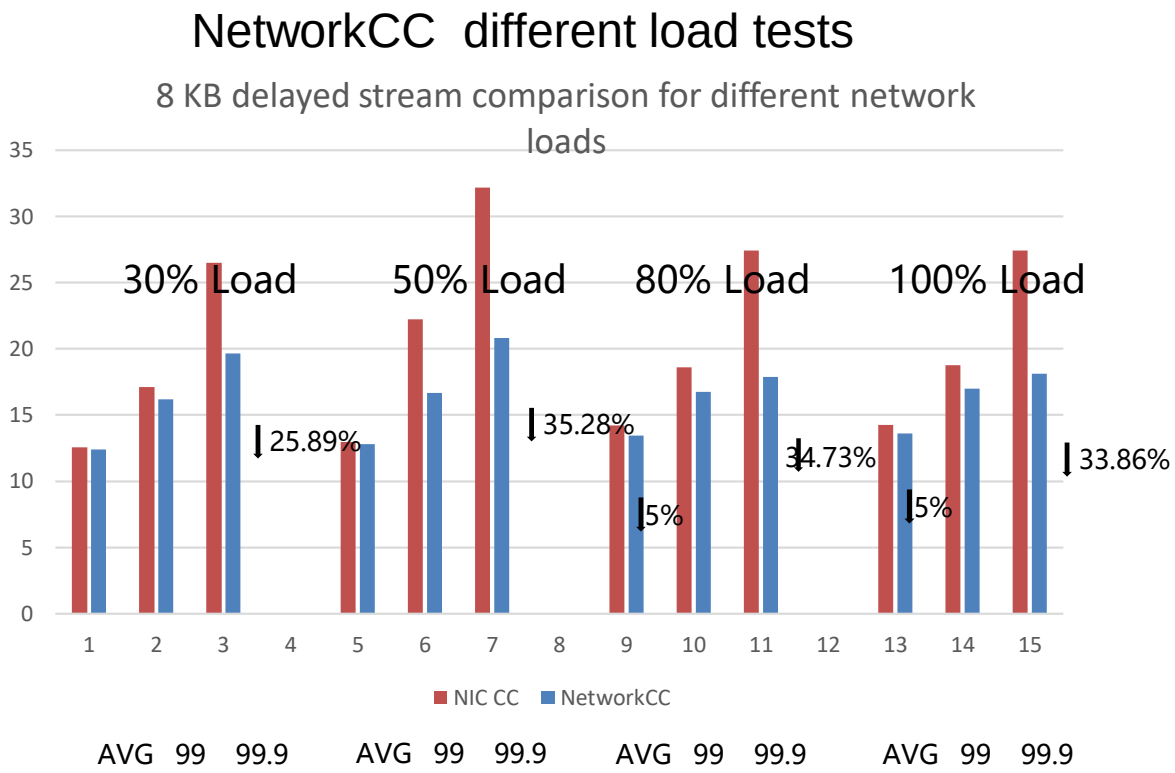


- The Sender 1~Sender 5 uses the ib_write_bw to send background long streams to the receiver to construct congestion.
- Sender 6 sends a delayed stream to the receiver through the ib_write_lat to test the network queuing delay.

NetworkCC different load tests



- The average bandwidth is similar for both cases.
- The average delay increases by 9%, and the 99.9% delay by 8B increases by 60.09% for the 80% load
  - avg lat is mainly related to bandwidth. The shallow switch queue has limited benefits to avg and is mainly reflected in high load.
  - 99.9 The lat is mainly related to the queuing time, and the benefits of the NetworkCC are obvious.
- Stability: The difference between the NetworkCC 99.9 delay and average delay is less than 1x, and the difference between the NIC CC99.9 delay and average delay is Max 3.12 x.



- The average bandwidth is similar in both cases.
- The average delay for 8 KB increases by 5.41%, and the 8 KB increases by 34.73% for the 80% load
- Stability:
  - For the NetworkCC, the background flow and delay flow are controlled large flows, and the control is stable. The difference between the 99.9 delay and the average delay is less than 1x.
  - For the NIC CC, the control of the large flow is unstable, and the difference between the 99.9 delay flow delay and the average delay is Max 1.51x.

Next Step

- Looking if there is interest in this direction.

An Open Congestion Control Architecture for high performance fabrics

draft-zhuang-tsvwg-open-cc-architecture-00

IETF 106, Singapore

Yan Zhuang, Wenhao Sun, Long Yan
Huawei Technologies

An Open Congestion Control Architecture for high performance fabrics

- **Scope**
  - Congestion control in datacenter networks
- **Motivation, requirements and use cases**
  - Support **CCA developers** to write their cc algorithms onto NICs while keeping the benefit of hardware offloading provided by NIC vendors.
  - Support **vendors** to optimize the NIC performance by hardware offloading while allow users to deploy and select new congestion control algorithms with the corresponding settings.
  - Support settings from **applications** to guarantee some QoS requirements.
  - Be **transport protocol independent**, for example can support TCP or RoCE et.al.
- **Objectives**
  - Define an open congestion architecture to enable more effective congestion control algorithms(CCA) deployment and configuration on smart NICs.

Architecture Overview

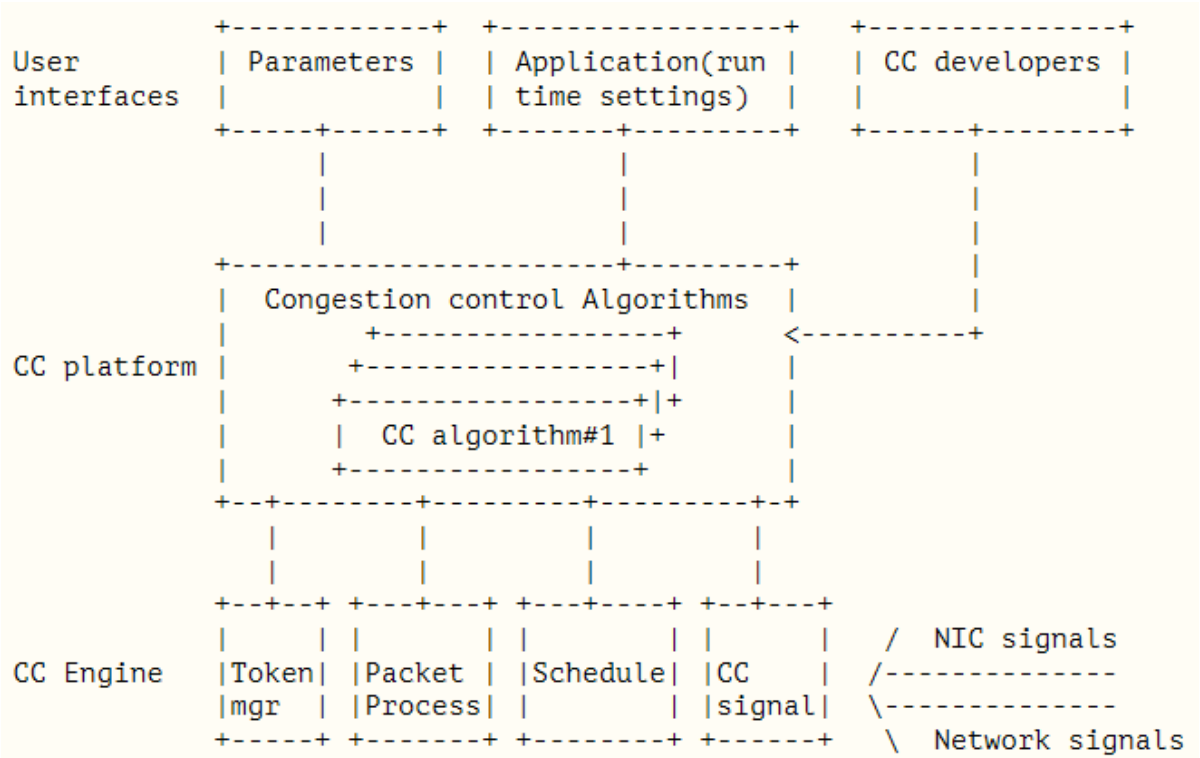


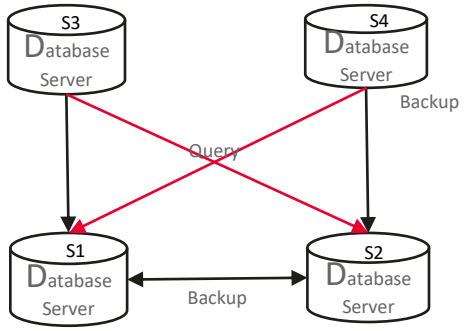
Figure 1. The architecture of open congestion control

Open Congestion Control (OpenCC)

- **Why?**
  - **More flexibility to deploy and select proper CCAs:** Traffic patterns may differ in CCA choices.
  - **Easy to deploy HW offloading**: New CC algorithms can use interfaces provided by vendors to take benefits of their hardware offloading optimization.
- **How?**
  - Decouple CCA related functions into two parts: (1) **a cc platform** is used to provide interfaces to users for extension and flexible configuration; (2) functions of packet transmission like scheduling is placed into **CC engine** which can be offloaded into hardware for efficiency.

Case Study: Traffic patterns may differ in CCA choices (1)

Each server with a 10Gbps NIC connected to a 10Gbps port on the switch. However, we limit all ports to 1Gbps to make congestion points. In the experiments, the database server S1 receives backup traffics from both S3 and S2 and one query traffic from S4. The server S2 gets back traffics from S1 and S4 and one query traffic from S3.



#1 experiment set: run one algorithm for both traffics.

	reno	cubic	bbr	dctcp
Throughput	64.92	65.97	75.25	70.06
Average latency	821.61	858.05	85.68	99.90
95% completion	894.65	911.23	231.75	273.92

As we can see, the average completion time of BBR and DCTCP is 10 times better than that of reno and cubic. BBR is the best to keep high throughput.

Conclusion

- Traffic patterns may differ in CCA choices, more flexible CCA selection and configuration based on traffic patterns will have benefits to users.
- New CCAs are being developed in the industry and it would be good to provide a module design to decouple the specific algorithm-based controls and common functions which can be offloaded into hardware to improve efficiency.

Feedbacks from the mailing list (I)

- Hardware offloading: existing pioneering work
  - Some work such as packet pacing and EDF scheduling pioneered by google, and algorithms like fq_codel and sch_cake are being pushed into the hardware.
- A modular NIC offload interface: why IETF/IRTF
  - Decoupling congestion controls on NICs are discussed in academic conferences.
  - Industry people are working on more new CCAs, like DCTCP, BBR, COPA, HPCC…
  - Hardware and software optimization for each CCAs costs.
  - We have both industry and academic people here to make a better solution.
- Whether it would be sufficient to slice the NIC and have different slices where all traffic is handled by one CC?
  - Agree, we are running experiments to use one CC while different configurations for these traffics to see how it works.

Feedbacks from the mailing list (II)

- The architectural idea to move away from in-band CC signaling from the network to the endpoints
  - The architecture is intended to support both signaling directly from network and signaling from receivers.
  - Each CC chooses their own ways of signaling and they might affect each other somehow if different CCs are used together while their signaling is different.
- Is it all about RoCE ?
  - Nope, the architecture is not binding to RoCE. The thought is to support several transport protocols including TCP. Current experiment results of different ccs are based on TCP (Reno, Cubi, bbr, dctcp), however we don't want to exclude RoCE either at this point.

Appreciate people for their considerations and comments~

Next Step

- Discussion points:
  - Common elements for congestion control engines that can be offloaded to hardware to improve efficiency.
  - CCA signal interface to the network environment for exiting and future extension.
  - Common interfaces to interact with upper layer users for different usage.
  - How CCAs cooperate with each other.
- More experiments of TCP CCAs and RoCE CCAs will be conduct.
- Welcome people to join the discussion and work.

Thank you!

Artificial Intelligence (AI) based ECN adaptive reconfiguration for datacenter networks

draft-zhuang-tsvwg-ai-ecn-for-dcn-00

IETF 106, Singapore

Yan Zhuang, Bai Zhang, Haotao Pan
Huawei Technologies

AI-based ECN adaptive reconfiguration for DCNs

- **Scope**
  - ECN adaptive reconfiguration for datacenter networks

- **Motivation, requirements and use cases**
  - Seek proper parameters of ECN adaptive reconfiguration by using artificial intelligence technologies to achieve self-tuning in a running data center network, so as to accommodate the changes of network resources to improve the network performance.

- **Objectives**
  - Provide a way to seek ECN adaptive reconfiguration (for AQM, such as RED/WRED) by using AI technologies in running data center network environment.

Why we want an adaptive ECN reconfiguration rather than static config?

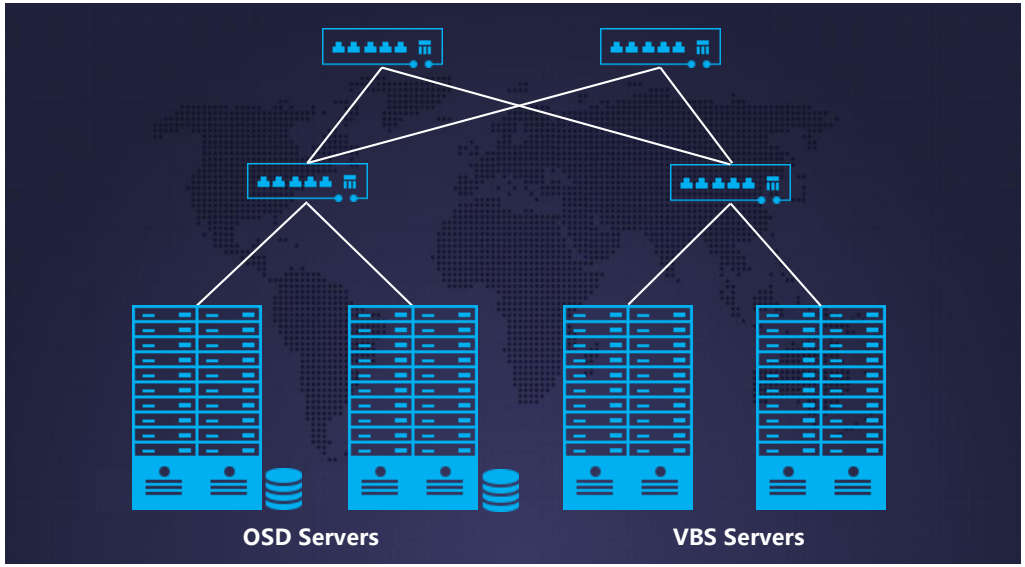
- **Network environment changes due to traffic in/out**

- **More dynamic to meet traffic-centric congestion control**
  - As stated in [RFC7567], with proper parameters, RED can be an effective algorithm. However, dynamically predicting the set of parameters (minimum threshold and maximum threshold) is difficult. As a result, its present use in the Internet is limited.
  - Other AQM algorithms have also been developed, while how to find proper parameters of algorithms for changeable application traffics is still difficult and affect the network performance.

Scene-based ECN Adaptive Reconfiguration

- **Phase 1：Scene Training**
  - **Step1**: construct typical scenes and generate a learning model to identify these scenes based on a set of network performance indicators.
  - **Step2**: provide proper ECN settings for these typical scenes based on human experience.
- **Phase 2: Adaptive reconfiguration (periodically running)**
  - **Snapshot**: Periodically, identify the "scene" of the current network based on the collected information over a period.
  - **Inducing**: identify current scene from one of the scenes that are collected and learned from datacenter networks running different traffics of various applications in training process.
  - **Reconfiguration**: ECN parameters of current network can be tuned to the settings of the identified scene.

Experiments in storage network

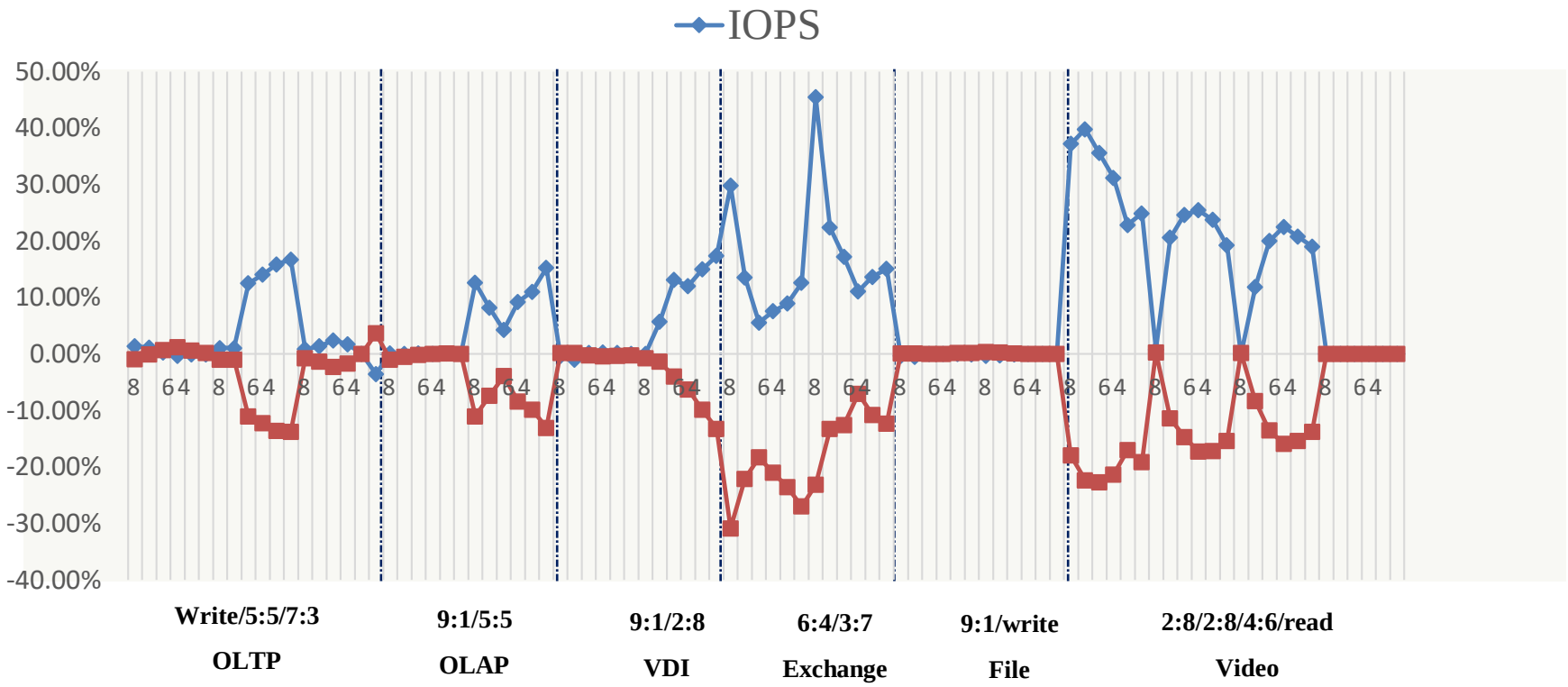


OSD：VBS = 1:3

Notes: current traffics under test are all RDAM traffics.

Traffic type		R:W	Block size	Num of concurrent processing
OLTP	OLTP-DATA	7:3/5:5	8K-64K	8-256
	OLTP-LOG	write	512B-64K	
OLAP		9:1/5:5	256K-4M	
VDI	Power-on	9:1	8K-16K	
	normal	2:8	1K-64K	
Exchange server		6:4/3:7	32K-512K	
File Server	Web File server	9:1	4K-64K	
	Web server log	write	8K	
Video	Video distribution	2:8	64K	
	Backup	2:8/4:6	16K-64K	
	VoD	Read	256K-4M	

Results of IOPS and Average latency



IOPS can improved by 50% at most, while the average latency can be reduced by 27% at most.

Feedbacks from rdma-cc mailing list

- Why RED/WRED, not fq-Codel for ECN?
  - Since the traffics under testing are rdma/rocev2 of storage applications, it requires no packet loss.
  - RED has already been implemented in asic, easy to deploy…

- What should be considered while RoCE is not a IETF work?
  - This work doesn't change RoCE protocol, while RoCE also uses ECN for its congestion notification.
  - This work asks for data/information collection and reconfiguration of ECN threshold which might be related to Netconf/Ops in IETF.

Conclusion and Next Step

- **Conclusion**
  - This work provides a way to provide adaptive configurations of network parameters based on learnt knowledge from human experiences and known scenarios.
- **Next Step**
  - Try TCP traffics to see how this method fits.
  - Try adaptive configurations for other AQMs.

Comments/discussions/participation are welcome :)

Thank you!

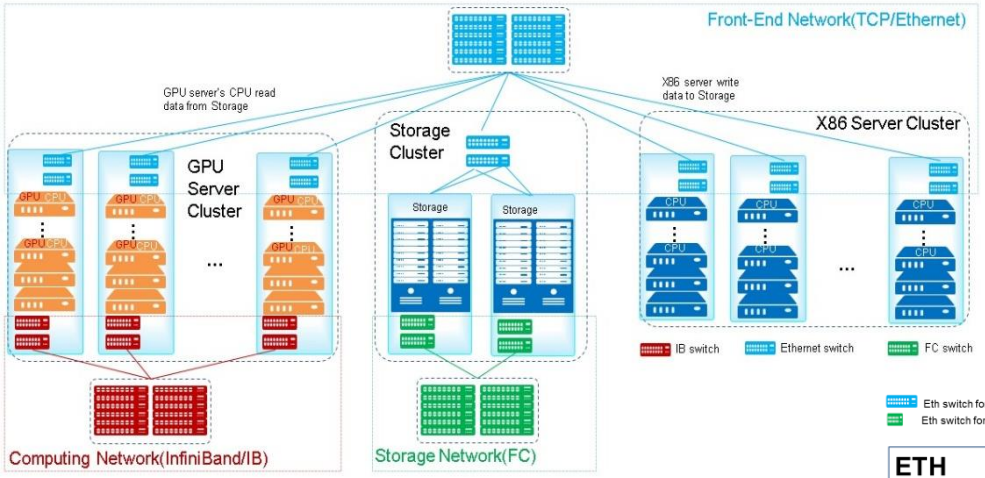
The Impact of Mixing TCP and RoCE

Yolanda Yu (yolanda.yu@huawei.com)
Marcus Sun (marcus.sun@huawei.com)

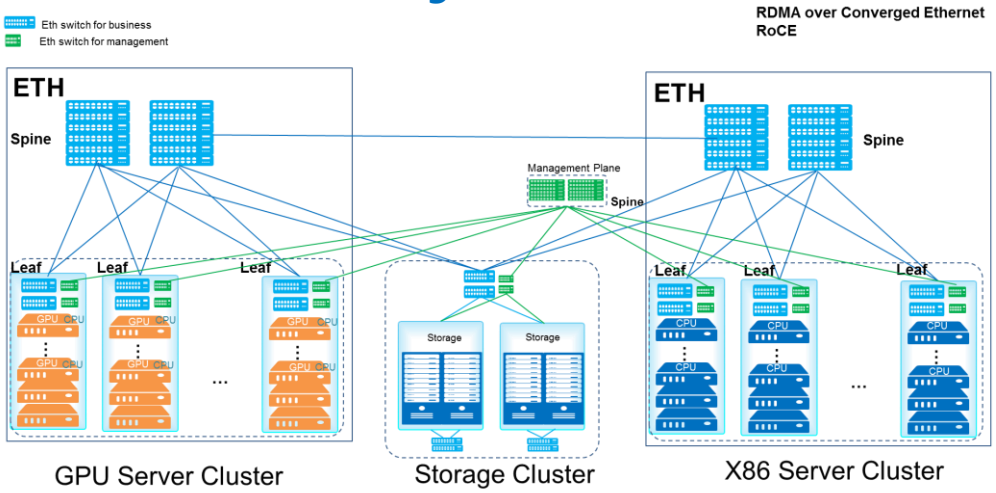
IETF-106, Singapore, November 2019

The Trend of Network Convergence in DC

Traditional DCN architecture: Separate InfiniBand (IB), Fiber Channel (FC), and Ethernet (Eth) networks

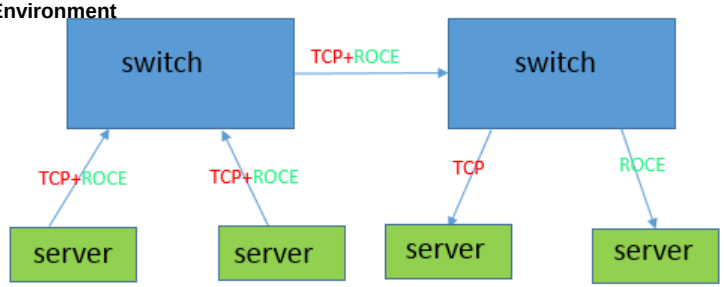


Single DCN



The Challenges when TCP+RoCE flows are mixed

1、Test Environment



2、Test setting：

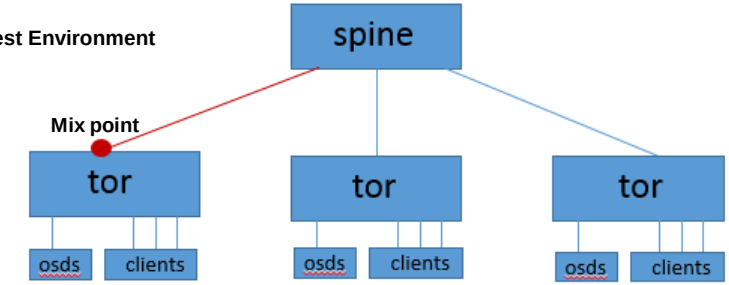
-- **Network：** tomahawk switch*2、100G
-- **Server：** ubuntu、mellanox CX5
-- **flow ：** tcp：iperf；roce：perftest

```
port link-type trunk
port trunk pvid vlan 30
undo port trunk allow-pass vlan 1
port trunk allow-pass vlan 2 to 30
lldp disable
qos drr 0 to 3
qos queue 3 drr weight 9
dcb pfc enable mode manual
transceiver 100GBASE-FIBER
```

3、Test

Expected TCP:RoCE	WDRR	Test Result
1:9	1:9	26.7:73.3
2:8	2:8	28.2:71.8
3:7	3:7	35.6:64.4
4:6	4:6	44.6:55.4
5:5	5:5	53.4:46.6
6:4	6:4	62:38:00
7:3	7:3	74.4:25.6
8:2	8:2	81.1:18.9
9:1	9:1	86.8:13.2

1、Test Environment



2、Test Seting：

-- **Network：** tomahawk switch * 4 、Speed up Ratio 1、Config pfc、ecn
-- **Server：** ubuntu、mellanox CX5
-- **flow：** roce：ceph -3 *osd，9*client，front –back end rdma，1M write；
tcp：iperf

3、Test

Expected TCP:RoCE	Test Result TCP :RoCE	Expec tOPS	IOPS Test	Decrea se
40:60	51:49	2188	1873	-14%
50:50	56:44	1877	1649	-12%

- TCP &ROCE mix，Qos of mix port can’t be assured；
- TCP &ROCE mix， the bandwidth assigned to ceph can’t be assured

Summary

- **QoS of traffic throughput mismatch when TCP+RoCE are mixed.**
- **More research need to be done to find the root cause and solution.**
- **Call for interest for more contribution.**

Thanks