IOWN GLOBAL FORUM

PoC Project name:

RDMA over Open APN between two DCI physical nodes

Classification: IOWN Global Forum Recognized PoC

Stage: SSF PoC Report

Confidentiality: Public

Version: 1.0

November 20, 2024

Table of Contents

List of Figures

List of Tables

1. Introduction

As a fundamental networking and computing infrastructure proposed by Innovative Optical and Wireless Network Global Forum (IOWN GF), APN (All-Photonic Network) and DCI (Data-Centric Infrastructure) play a crucial role in technology development. In the DCI Functional Architecture document [IOWNGF-DCI-FA], it describes the data plane acceleration is necessary for extreme use case, and comes up with frameworks. Specifically, RDMA over Open APN framework is proposed for long-range communication. To examine the idea, IOWN GF provides a RDMA over Open APN PoC Reference document [IOWNGF-RDMAoverAPN-PoC].

Aiming to gather real-world experience of RDMA over Open APN technology, we established a trial network infrastructure of Open APN with physical APN equipment. This Open APN connects two data centers hundreds of kilometers apart. On top of this, we planned a multi-phase PoC project to evaluate performance of RoCEv2 between two DCI physical nodes and discover applicable use cases mentioned in Section 5.

In phase 1, Open APN connects two distributed data centers located in northern and southern Taiwan, the communication distance is around 380 km. We describe the PoC system configuration in Section 6.1 and show the overall architecture in Figure 6.1-2.

According to the PoC Reference document [IOWNGF-RDMAoverAPN-PoC], we measured throughput and latency of RoCEv2 (RDMA over Converged Ethernet version 2) over Open APN in Section 6.3. Furthermore, we experimented to migrate VMs (Virtual Machine) across distributed data centers with RDMA over APN. We describe implementation and results of RDMA VM live migration in Section 6.4, which could be helpful to realize part of Services Infrastructure for Financial Industry Use Case [IOWNGF-SIFI-UC] proposed by IOWN GF.

To give feedback on phase 1 PoC to IOWN GF, we describe PoC's contribution in Section 7, and list suggested action items after conducting phase 1 PoC in Section 8.

For this multi-phase PoC project, we describe our next steps about phase 2 in Section 9, and make a conclusion of phase 1 in Section 10.

2. PoC Project Completion Status and Project Participants

This PoC is a multi-phase project. The phase 1 PoC system is described in Section 6.1. The targets of performance evaluation are described in Section 6.2. The measurement methods and results are described in Section 6.3 and Section 6.4. The technical findings are described in Section 6.5.

- PoC Project Name: RDMA over long-distance Open APN between two DCI physical nodes
- Overall PoC Project Completion Status: Phase 1 of the multi-phase PoC is completed
- PoC Stage Completion Status: Significant Step Forward (SSF)
- Project Participants:

Member Company	Name	E-mail
ChungHwa Telecom (CHT)	Jia-An Tsai **(contact person)**	tsaijiaan@cht.com.tw
	Shih-Che Chien	chiensc@cht.com.tw
	Chien-Hua Lee	leejenhwa@cht.com.tw
	Chun-Hao Chen	andychen122000@cht.com.tw

In phase 2 of the PoC project, we plan to scale out Open APN infrastructure to connect two data centers farther away than it is in phase 1. The same PoC system, targets and performance evaluation methods will be used.

3. Confirmation of PoC Demonstration

- Phase 1 of this PoC project had been conducted in Chunghwa Telecom's two operating data centers in Taiwan. The overall architecture is shown in Figure 3-1. Please find Figure Figure 6.1-2 for detailed configuration and see Section 6.1 for detailed information.

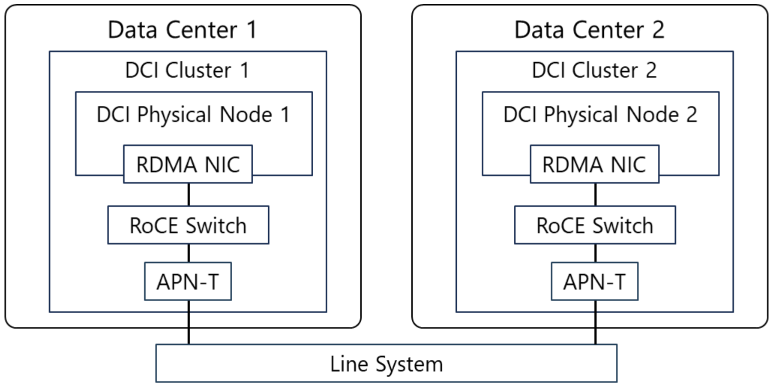

Figure 3-1: PoC architecture overview

- PoC Demonstration System Photos (Actual hardware photos of Figure 3-1 components):

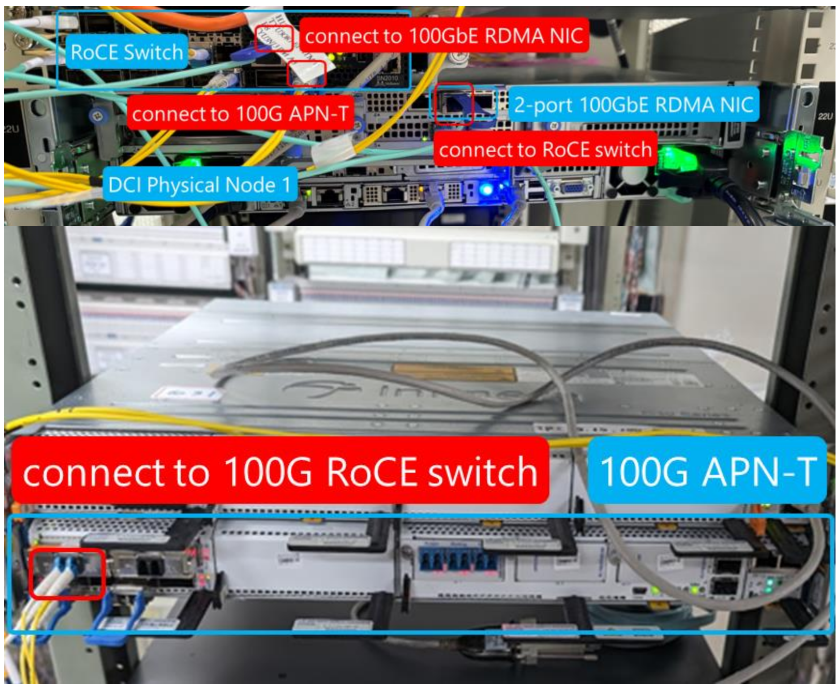

Figure 3-2: DCI Cluster 1 physical view of the PoC implemented system in phase 1

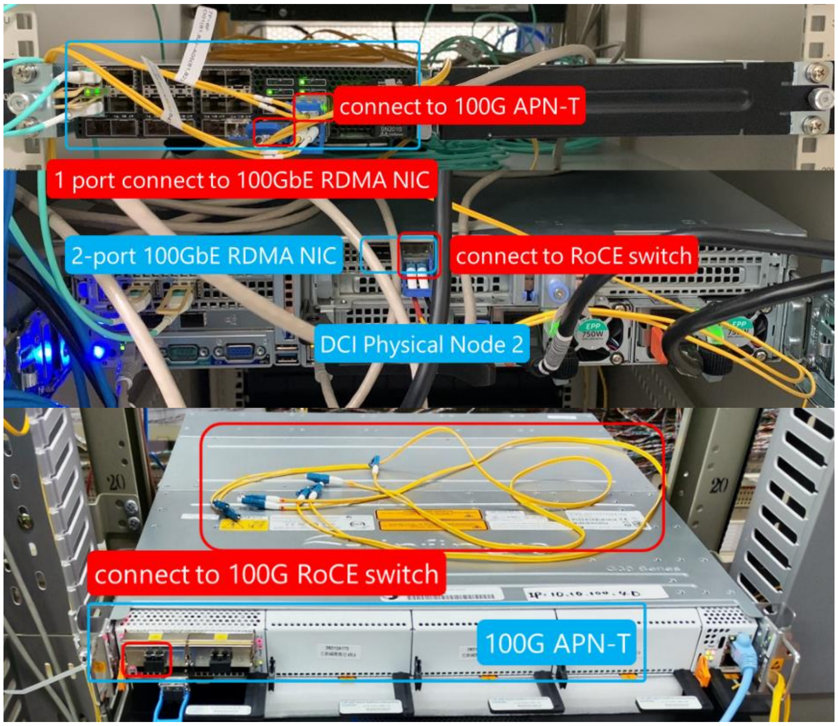

Figure 3-3: DCI Cluster 2 physical view of the PoC implemented system in phase 1

4. PoC Goals Status Report

In phase 1 of this PoC project, we executed RDMA over Open APN performance measurements of main memory to main memory, which is evaluation procedure step 1 in the PoC Reference document [IOWNGF-RDMAoverAPN-PoC].

Phase 1 of this PoC project has achieved PoC objectives as follows:

- PoC Project Goal #1: To gather experience by implementing the proposed technology (Goal Status: Demonstrated and met in phase 1)
    - We implemented the PoC system shown in Section 6.1 and did the performance evaluation in Section 6.3 and Section 6.4 to achieve the goal.
- PoC Project Goal #2: To discover practical algorithm(s) by tuning parameters to achieve highest performance (Goal Status: Demonstrated and met in phase 1)
    - We used different combinations of parameters to discover better performance as shown in Section 6.3. Also, we proposed high performance virtual machine migration with RDMA over Open APN in Section 6.4.
- PoC Project Goal #3: To discover bandwidth on certain distance of the RDMA-over-APN technology (Goal Status: Demonstrated and met in phase 1)
    - Please see Section 6.3.

We plan to achieve additional PoC objective(s) in phase 2 as follows:

- PoC Project Goal #4: To determine the dependence of bandwidth on distance of the RDMA-over-APN technology (Goal Status: Will demonstrate in phase 2)

5. Supposed Use Case

5.1 Relationship to Services Infrastructure for Financial Industry Use Case

In the Services Infrastructure for Financial Industry Use Case [IOWNGF-SIFI-UC] developed by IOWN GF, workload migration is expected to achieve operational resiliency and agility of financial services. According to the evaluation result of this PoC project, RDMA over Open APN technology can be taken to realize intra-regional and inter-regional VM migration or data backups. We describe the measurement methods and results in Section 6.4.

5.2 Use Case Scenario Supposed in the PoC

In today's implementation of the data-sharing platform, protocols such as NFS, CIFS/SMB, and SFTP are used. In the Data Hub Functional Architecture document [IOWNGF-IDH-FA], the gap analysis to realize the IOWN GF use cases is stated, and RDMA is proposed to accelerate data transfer built on APN and DCI, which brings about Linux-based NFS over RDMA, Windows-based SMB over RDMA, etc.

We have a performance evaluation plan for either of them in following phase of this PoC project. It helps determine whether the technology can be an accelerated data-sharing mechanism for IOWN GF use cases.

6. PoC Technical Report

6.1 Implemented System

The target scope of this PoC project is shown in Figure 6.1-1. Instead of using an emulator, we build a trial APN infrastructure, which connects two data centers, with physical equipment. RDMA data is transferred with the RoCEv2 (RDMA over Converged Ethernet version 2) protocol.

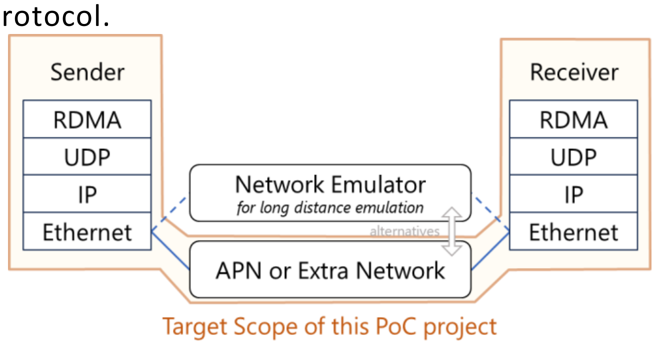

Figure 6.1-1: PoC scope of RDMA over Open APN with RoCEv2 protocol

The overall architecture of this PoC project and transfer data path is shown in Figure 6.1-2. We set up the same DCI Cluster layout in two connected data centers, there are a DCI Physical Node equipped with one 100GbE RDMA NIC, one 100GbE Ethernet switch that supports RoCE feature (RoCE Switch in Figure 6.1-2), and one 100G APN-T (Open APN Transceiver). Due to the regulation of CHT's data center, APN-T is in one server room, the others are in a rack of another server room. These two server rooms are in the same building. We prepared two 100GBASE-LR4 QSFP28 optical transceivers and a fiber cable for the connection between APN-T and RoCE switch. In the same rack where APN-T is, it connects to one ROADM (Reconfigurable Optical Add-Drop Multiplexer) node. Between these two data centers, there are five OLA (Optical Line Amplifier) sets. To connect data centers with Open APN, we assigned the central wavelength of APN-T's optical signals and set the channel bandwidth and frequency in ROADM nodes. In phase 1 of the PoC project, the distance between the two data centers is around 380 km and the network bandwidth is 100G.

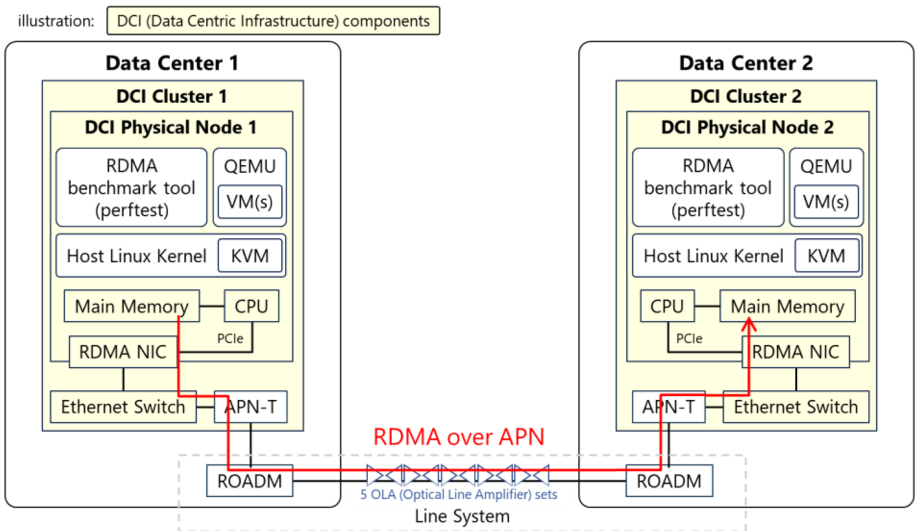

Figure 6.1-2: Implemented System Configuration of PoC project phase 1

6.1.1 DCI Cluster and Open APN Configuration

Table 6.1.1-1: DCI Cluster Configuration of PoC system

Item	DCI Cluster 1	DCI Cluster 2
Server Platform	Dell PowerEdge R750	Dell PowerEdge R740xd
CPU	Intel® Xeon®Gold 6342 CPU@2.80GHz (2 socket x 24 cores x 2 threads)	Intel® Xeon® Gold 6254 CPU@3.10GHz (2 socket x 18 cores x 2 threads)
Memory	32 x 64GB DDR4 3200MHz (SK Hynix HMAA8GR7CJR4N-XN) Total of 2048 GB	8 x 32GB DDR4 2933MHz (SK Hynix HMA84GR7JJR4N-WM) Total of 256 GB
BIOS	2.5.4	2.5.4
Operating System	Ubuntu 22.04 LTS	
	Linux kernel version: 5.15.0-97-generic	
100GbE RDMA Network Interface Card (NIC)	NVIDIA BlueField-2 integrated ConnectX-6 Dx network controller (MT42822) with 2 ports of 100Gb/s	
	Firmware version: 26.40.1000	
	Driver Version: mlx_core 24.01-0.3.3	
	Interface Maximum Transmission Unit (MTU): 4200 byte IBV MTU (RoCE active MTU for perftest): 4096 byte	
	PCI Express 4 (16 GT/s x8)	PCI Express 3 (8 GT/s x16)
Ethernet Switch	NVIDIA SN2010 25GbE/100GbE Switch (with RoCE Support)	
RoCE Switch Congestion Control Mechanism	ECN (Explicit Congestion Notification)	
APN-T	Infinera G30 (100G)	

Line System (functionally similar to APN-G & APN-I)	2 ROADM nodes (Infinera HiT 7300) and 5 OLA sets (Infinera)

BIOS Settings

Table 6.1.1-2: BOIS Settings of DCI Physical Node in the PoC system

Item	DCI Physical Node 1	DCI Physical Node 2
Hyper Threading	Disabled	Disabled
System Profile	Custom	Custom
CPU Power Management	OS DBPM	Maximum Performance
Memory Frequency	Maximum Performance	Maximum Performance
CPU C-state	Disabled	Enabled
CPU P-state	Enabled	Enabled
Turbo Boost	Enabled	Enabled
Energy Efficient Policy	Performance	Performance

6.1.2 RDMA Profile

Table 6.1.2-1: RDMA Profile used in the PoC system

Item	Description
Transport protocol stack	RoCEv2 (UDP/IP/Ethernet)
RDMA core library	linux-rdma/rdma-core: 2307mlnx47-1.2401033
RDMA benchmark tool	linux-rdma/perftest: perftest-24.01.0-0.38 (released on github.com)
RDMA service type	Reliable Connection (RC)
RDMA operation type	SEND, WRITE, and READ
Retransmission algorithm	Go-Back-N
Queue depth	8192
RDMA Message Size	from 2,048 (2K) bytes to 8,388,608 (8M) bytes.

6.1.3 VM Live Migration Configuration

Table 6.1.3-1: VM live migration configuration used in the PoC system

Item	Description
Hypervisor	QEMU [QEMU] v9.0.0 (libvirt [libvirt] v8.0.0)
C library for RDMA application	Libibverbs v39.0

6.2 Targets of Performance Evaluation

As defined in the PoC Reference document [IOWNGF-RDMAoverAPN-PoC], we focus on step 1 of the step-by-step procedures for RDMA over Open APN performance evaluation, which means communication type is main memory to main memory. The target benchmarks are "throughput for data transferring" and "latency between RDMA endpoints."

Additionally, we developed VM live migration with RDMA over APN during the implementation of the PoC. The target benchmark is the time spent to complete the VM live migration process, and the memory stress can be managed. The less time required to complete the VM live migration process, the better the performance. The more memory stress that can be managed, the better the performance.

We chose to impose memory stress rather than other types of stress for two reasons. First, while migrating a running VM from the source host to the destination host, this VM simultaneously generates dirty memory pages. We used stress tool [stress-tool] to simulate this behavior by continuously writing to memory, thereby creating a workload that generates dirty pages. Second, RDMA is known for allowing the transfer of memory data from one host to another without CPU involvement.

6.3 Measurement Methods and Results of RDMA over Open APN Performance

On top of the built Open APN infrastructure that connects two data centers, we use Linux-RDMA benchmark tool perftest [Linux-perftest] to transfer NIC-offloaded RoCE data with three types of operations (SEND, WRITE, and READ) and measure their performance. The perftest tool [Linux-perftest] designed many parameters for benchmarks as if there are various applications.

Besides APN, NIC configuration and how applications use NIC are the key factors of RDMA transmission under the RoCEv2 protocol. For one thing, to achieve better performance, we configured 8192 for the queue depth of NIC and set MTU as 4096 bytes in all conditions. For another thing, to discover how tx depth and queue pairs affect RDMA performance, we measured three RDMA operation types in different combinations of arguments listed in Table 6.3-1. To have better measurement accuracy, we run the tests twenty times and use the average value as the measurement results for each operation. The used measurement commands are shown in Appendix B.

Table 6.3-1: Performance evaluation with different combinations of arguments

Operation type	RDMA SEND					RDMA WRITE					RDMA READ			
Benchmark	Throughput			Latency		Throughput			Latency		Throughput		Latency	
Sequence	1	2	3	1	2	1	2	3	1	2	1	2	1	2
Tx depth (default: 128)	128	2048		128	2048	128	2048		128	2048	16	2048	128	2048
Queue Pair (default: 1)	1		4	1		1		4	1		16		1	1
Iteration (default:1000)	15000			15000		15000			15000		4096		4096	
Others	Default value													

The combinations include the default values of parameters, the arguments used in the IOWN GF recognized RDMA over Open APN PoC Report in June 2024 [IOWNGF-RDMAoverAPN-RecognizedPoC], and increased queue pairs (queue pair argument is invalid in latency measurement). The corresponding results are shown in Figure 6.3-1 to Figure 6.3-6. Please see detailed results in Appendix C.

Note: In Figure 6.3-4 to Figure 6.3-6, each of them has two almost fully overlapping lines. When tx depth is set to 128 and 2048 for WRITE, SEND and READ operations, the measured latency is close, which causes the overlaps. It might make PoC report readers think there is only one line in Figure 6.3-4 to Figure 6.3-6.

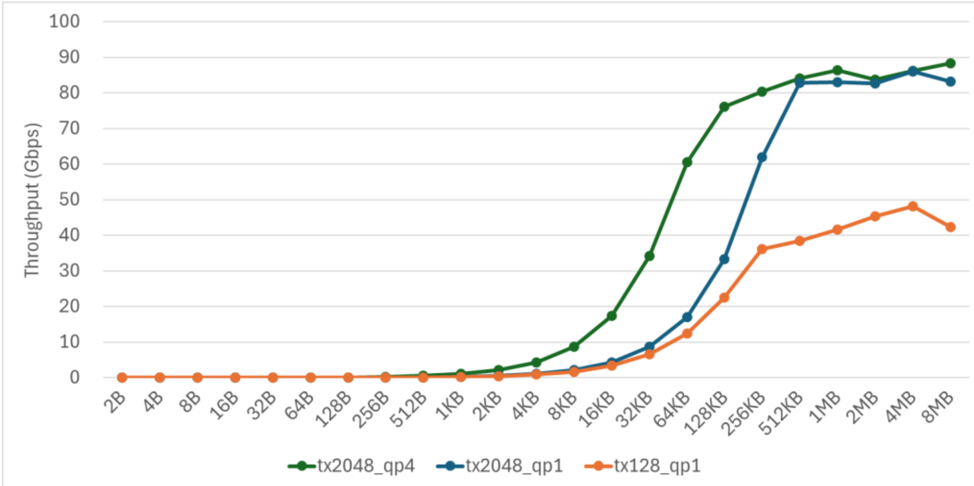

Figure 6.3-1: 100G RoCEv2 SEND throughput of main memory to main memory communication

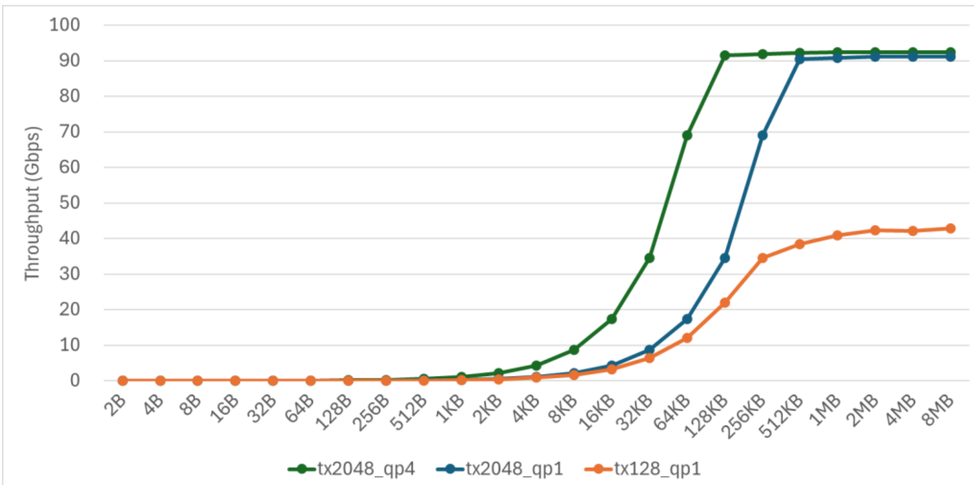

Figure 6.3-2: 100G RoCEv2 WRITE throughput of main memory to main memory communication

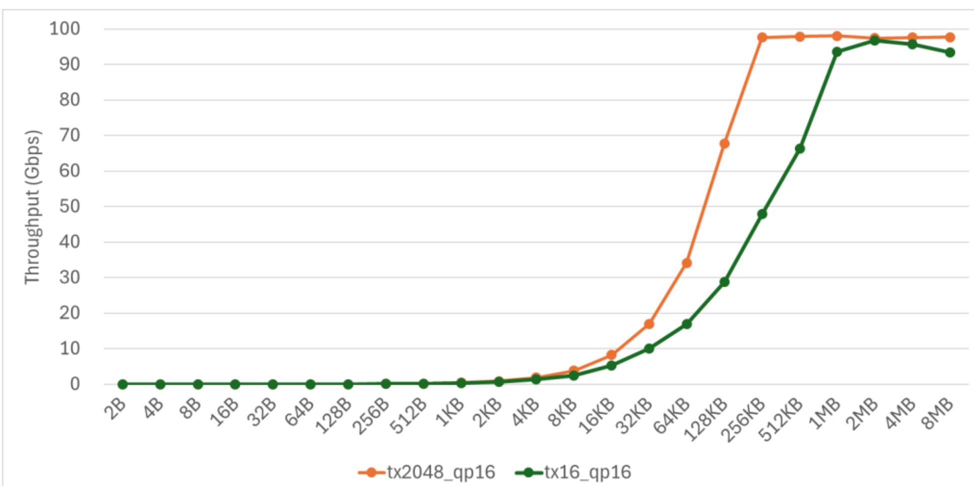

Figure 6.3-3: 100G RoCEv2 READ throughput of main memory to main memory communication

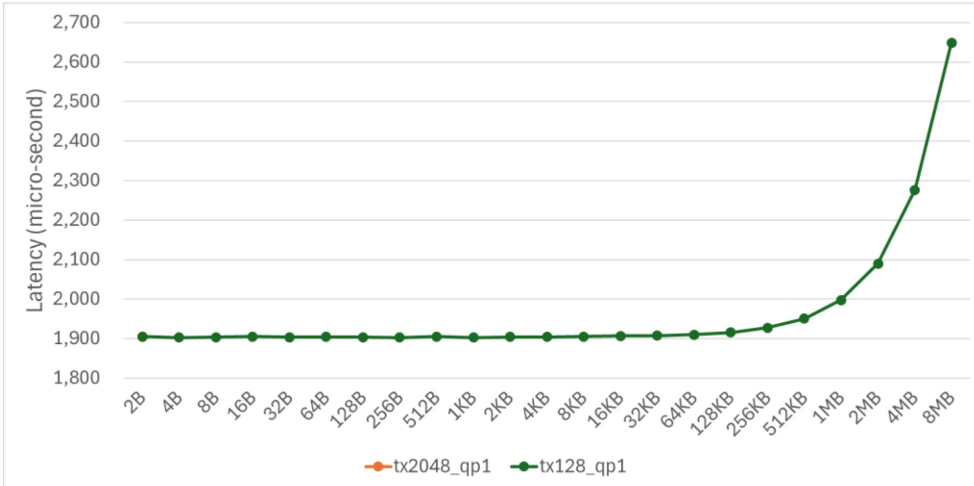

Figure 6.3-4: 100G RoCEv2 SEND latency of main memory to main memory communication

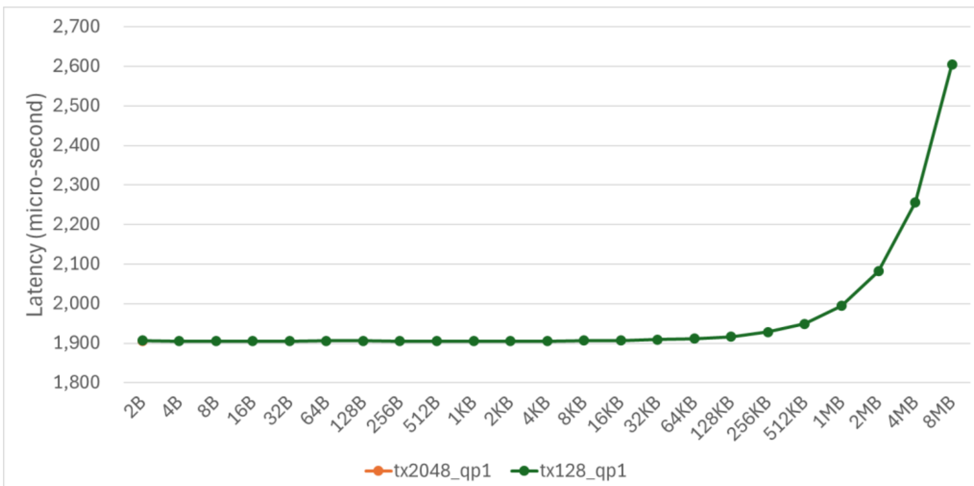

Figure 6.3-5: 100G RoCEv2 WRITE latency of main memory to main memory communication

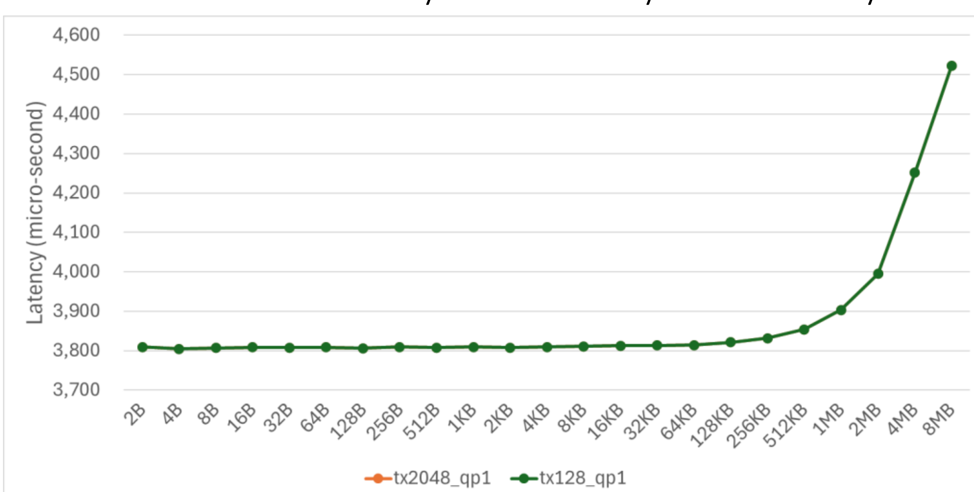

Figure 6.3-6: 100G RoCEv2 READ latency of main memory to main memory communication

6.4 Measurement Methods and Results of VM Live Migration Across Data Centers with RDMA over Open APN

We implemented VM live migration with RDMA over Open APN, the infrastructure and configuration are the same as what we used in Section 6.3. We evaluate the performance by migrating VMs of multiple specifications from DCI Physical Node 1 to DCI Physical Node 2. To evaluate the success and time spent for VM migration, memory stress levels of 0%, 20%, 40%, 60%, and 80% of the VM memory specifications were imposed within the VM. For example, we applied memory stress of 0 GB (0% of 8 GB), 2 GB (round up 20% of 8 GB), 4 GB (round up 40% of 8 GB), 5 GB (round up 60% of 8 GB), and 7 GB (round up 80% of 8 GB) to a VM with the specifications of 4 cores, 8 GB memory, and 100 GB storage. The used measurement commands for VM live migration are shown in Appendix F. To accentuate the availability and benefits of RDMA over Open APN for VM live migration, we also measure the results when TCP is used. For both RDMA and TCP, we run the tests four times and use the average values as the measurement results, as shown in Table 6.4-1 and Table 6.4-2. (Note: “m” means “minute”, “s” means “second” in Table 6.4-1 and Table 6.4-2). The success of the migration depends on the efficient and timely transfer of these dirty memory pages to ensure data consistency and minimize downtime. When a VM can’t be migrated to the destination successfully, we annotate it as "Did Not Finish" with "DNF." Nevertheless, the application(s) can still operate correctly in the original VM instance.

Table 6.4-1: Time spent to finish VM live migration across two data centers using RDMA over Open APN

VM Specification / Memory Stress			0%	20%	40%	60%	80%
CPU (Cores)	Memory (GB)	Storage (GB)					
4	8	100	1m4s	1m8s	1m4s	1m7s	1m8s
8	16	100	1m27s	1m32s	1m32s	1m33s	1m36s
16	32	100	1m37s	1m44s	**1m48s**	**1m55s**	**About 2 mins** **DNF 2 out of 4 times*
32	64	100	1m40s	**1m57s**	**About 2 mins** **DNF 2 out of 4 times*	DNF	DNF

Table 6.4-2: Time spent to finish VM live migration across two data centers using TCP over Open APN

VM Spec / Memory Stress			0%	20%	40%	60%	80%
CPU (Cores)	Memory (GB)	Storage (GB)					
4	8	100	1m16s	1m19s	1m18s	1m19s	1m22s
8	16	100	1m29s	1m34s	1m38s	1m39s	1m49s
16	32	100	1m40s	1m51s	DNF	DNF	DNF
32	64	100	1m41s	DNF	DNF	DNF	DNF

6.5 PoC Technical Finding

Table 6.5-1: Technical finding of RDMA performance evaluation at main memory to main memory

Objective Id	Main memory to main memory	
Description	We measured two benchmarks described in the PoC Reference document [IOWNGF-RDMAoverAPN-PoC]. Benchmark 1: throughput for data transferring Benchmark 2: latency between RDMA endpoints	
Pre-conditions	None	
Procedure	1	Measure long-distance RoCEv2 throughput in several combinations of arguments for WRITE, SEND and READ operations using the benchmark tool perftest [Linux-perftest]
	2	Measure long-distance RoCEv2 latency in several combinations of arguments for WRITE, SEND and READ operations using the benchmark tool perftest [Linux-perftest]
	3	Observe the measurement results
Finding Details	We demonstrated three RDMA operation types under RoCEv2 on the physical long-distance APN environment to provide real-word experience of the RDMA over Open APN technology. **The main findings about RDMA throughput include:** • The more tx depth and queue pairs are, the higher throughput RDMA transmission is. • Theoretically, the bigger transferred message size is, the better is the RDMA performance is. However, RDMA SEND/READ is not stable as expected. For example, when we set tx depth to 2048 and use 4 queue pairs, RDMA SEND throughput at 2 MB is lower than the one at 1 MB as shown in the green line of Figure 6.3-1. When we set tx depth to 16 and use 16 queue pairs for the measurement, RDMA READ throughput reaches the maximum at 2 MB, which is higher than the one at 4 MB and 8 MB as shown in the green line of Figure 6.3-3. **The main findings about RDMA latency include:** • When we use one queue pair and set "tx depth" to 128 and 2048 for WRITE, SEND and READ operations, RDMA latency does not change much as shown in Figure 6.3-4 to Figure 6.3-6. The exact values are shown in Appendix D. • When message size is small, RDMA WRITE/SEND latency is pretty close to the latency caused by distance.	

	Other findings: • Among some of the tests we ran twenty times, we found that when the message size is large, RDMA throughput decreases a lot frequently as shown in Appendix C. Go-Back-N retransmission is triggered as shown in Figure E-1. And congestion seemed to happen because the value of "rp_cnp_handled" counter increases shown in "rdma statistics show" command. To further check the congestion issue, we captured RDMA packets shown in Figure E-2, it turned out that the ECN codepoint is "10" which means no congestion experienced. • Even when the communication distance is fixed, for instance in phase 1 is 380 km, the measurement results of RDMA over Open APN could be different when a PoC is conducted with a different implemented system (the implemented system includes hardware and software, e.g. Open APN, server, NIC, switch, OS, BIOS, benchmark tool, arguments, etc.). Please see Appendix C and Appendix D for the examples of the difference.
Lessons Learnt & Recommendations	• Tx depth and queue pairs indeed affect RDMA throughput. How applications can utilize the parameters is the key to have better throughput performance. • By disabling "roce_adp_retrans" of the RDMA NIC, among the tests twenty times, we found that throughput decreased less frequently when the message size is large. (Note: The reason why the value of the "rp_cnp_handled" counter increases is still under investigation) • Real application traffic may be affected by different variables. Further research about long-distance RDMA over Open APN should be encouraged. For example, study tuning algorithms in different scenarios, discover the maximum distance on the premise that RDMA over Open APN technology can operate, etc.

Table 6.5-2: Technical finding of VM live migration across two data centers with RDMA over Open APN

Objective Id	VM live migration with RDMA over Open APN	
Description	We measured two benchmarks for VM live migration with RDMA over Open APN. Benchmark 1: Time spent to finish the VM live migration Benchmark 2: VM's memory stress RDMA can manage	
Pre-conditions	None	
Procedure	1	Prepare VMs in multiple specifications (CPU, memory and storage)
	2	Run stress tool [stress-tool] in the VM with a single-threaded CPU to impose memory stress, simulating a heavy workload in the VM.
	3	Manually execute the virsh [libvirt-virsh] command on the source host to migrate a running VM to the destination host. This running VM is one of the VMs prepared in the first procedure.
	4	Use Linux time command to measure time spent to complete migration along with virsh command.
	5	Collect the result and evaluate its performance.
Finding Details	• Under the same conditions of VM specification and memory stress, RDMA completes VM live migration that TCP cannot as shown in the blue entries in Table 6.4-1. We concluded that under the same CPU resource configuration (for RDMA and TCP), VM live migration with RDMA performs better than with TCP regardless of memory stress level. • Under the same condition of VM specification and memory stress, VM live migration with RDMA finishes VM live migration faster than with TCP as shown in Table 6.4-1. However, the larger the VM, the smaller the performance difference between RDMA and TCP. • As shown in the red entries in Table 6.4-1, when we imposed 26 GB of memory stress, we found that VM live migration with RDMA did not finish every time. We suppose this issue relates to what we observed in RDMA over Open APN throughput measurement and plan to confirm it in next phase. For more information related to this issue, please see "Other findings" in the row "Finding Details" of Table 6.5.1 and Appendix C.	
Lessons Learnt & Recommendations	• Without CPU involvement, VM live migration with RDMA manages memory stress better than with TCP. The benefits include the performance improvement, power saving and better CPU efficiency. The saved CPU resources can be used for running applications. • To clarify why the larger the VM, the smaller the performance difference between RDMA and TCP, further analysis of the distribution of time spent in VM live migration procedures is required. VM and block live migration procedures include pre-	

	migration, iterative pre-copy, stop and copy, commitment and remote activation. Currently, we suppose I/O-related processes might account for a large proportion because we used HDD in this phase of PoC. To improve the performance of VM live migration, the adoption of better storage devices should be considered. • When the VM specification is great and memory stress is high, RDMA over Open APN can't finish VM live migration. To have a better performance, we need to do further research on how QEMU implements RDMA and its dependence on the distance of RDMA over Open APN technology. • RDMA over Open APN technology can be used to realize VM live migration which is a common inter-DC use case and helpful for Services Infrastructure for Financial Industry Use Case [IOWNGF-SIFI-UC].

7. PoC's Contribution to IOWN GF

Table 7-1: PoC's contribution to IOWN GF

Contribution	WG/TF	Study/Work Item	Comments
RDMA over Open APN performance evaluation on the physical infrastructure	DCS TF	N/A	We established a trial APN infrastructure with two APN-Ts, two ROADM nodes, and five OLA sets. This APN connects two data centers 380 km apart. In these two data centers, we built one DCI Cluster respectively. On the above basis, we conducted throughput and latency measurement of long-distance RDMA over Open APN in phase 1.
VM live migration with RDMA over Open APN performance evaluation	DCS TF	N/A	We developed VM live migration with RDMA over Open APN, which is an additional scenario compared to the one described in the PoC Reference document [IOWNGF-RDMAoverAPN-PoC] and is helpful to promote RDMA over Open APN technology discussed in DCS TF.

	RIM TF	N/A	We verified the availability of VM live migration with RDMA over APN and completed performance evaluation by migrating VMs of multiple specifications. It could be helpful to realize intra-regional and inter-regional VM live migration or data backups for Services Infrastructure for Financial Industry Use Case [IOWNGF-SIFI-UC] in RIM TF.

We expect there will be more contributions such as providing investigation result of why "rp_cnp_handled" counter increases and congestion solution in the following phases of this PoC project.

8. PoC Suggested Action Items

8.1 Gaps identified in relevant standardization

We are researching the suitable configuration of implemented system, and plan to study related standardization status in the following phase of this PoC project.

8.2 PoC Suggested Action Items

- It is expected to explore tuning algorithms for RDMA over Open APN framework in different long-distances and scenarios.
- When a network fabric or switches is/are a part of RDMA over Open APN implementation, how to ensure network congestion should be considered.
- It is expected that IOWN GF would suggest distributed deployment for RDMA over Open APN technology.

9. Next Steps

Next in phase 2 of the PoC project, we will scale out Open APN infrastructure to connect two data centers farther away than 380 km (transmission distance in phase 1).
We plan to:

- Measure throughput and latency using the same arguments in phase 1.
- Further analyze how QEMU realized VM live migration and the distribution of time spent in VM live migration procedures.
- Check the availability of "VM live migration with RDMA over Open APN" implementation in phase 1 to determine the dependence of distance.
    - If it's available, we will use the same measurement methods in phase 1 to have the results.
    - If it's unavailable or its performance is a lot worse, we may apply related mechanisms to work it out.

In addition, the followings are also under consideration:

- Scale out the deployment scale of DCI cluster in the implemented system.
- Further study on the congestion/retransmission issue and discover at least one effective solution.
- Measure power consumption of RDMA transmission and its appliance on VM migration.
- Prepare GPUs to evaluate the performance of step 2 (XPU to XPU) suggested in the PoC Reference document [IOWNGF-RDMAoverAPN-PoC].
- Prepare NVMe devices to evaluate the performance of step 3 (main memory to NVMe Device) suggested in the PoC Reference document [IOWNGF-RDMAoverAPN-PoC]. In addition, these devices will be used for further study and evaluation of VM migration with RDMA to confirm the related technical findings mentioned in this PoC report. Different kinds of stress tests such as I/O or disk stress would be then performed.

10. Conclusion

In this PoC project, we implemented a system of physical APN and DCI infrastructure shown in Section 6.1. In phase 1, the transmission distance is around 380 km. This phase 1 PoC report shows performance evaluation results of RDMA over Open APN in Section 6.3. We also developed VM live migration with RDMA over APN and completed performance evaluation in Section 6.4. The corresponding contributions are listed in Table 7-1.

In the following phases of this PoC project, we will continue to tune the phase 1 implemented system, and evaluate RDMA over Open APN performance in longer distance and more scenarios. Moreover, with RDMA over Open APN, we plan to do more research on VM live migration and other possible use cases. We expect to provide feedback for further IOWN GF technology development such as Reference Implementation Models, use cases or PoC references.

Abbreviations and Acronyms

ABBREVIATION	DESCRIPTION
APN	All-Photonic Network
APN-G	Open APN Gateway
APN-I	Open APN Interchange
APN-T	Open APN Transceiver
CIFS	Common Internet File System
CPU	Central Processing Unit
IOWN	Innovative Optical and Wireless Network
IOWN GF	IOWN Global Forum
NFS	Network File System
NVMe	Non-Volatile Memory Express
OLA	Optical Line Amplifier
RDMA	Remote Direct Memory Access
RIM	Reference Implementation Model
ROADM	Reconfigurable Optical Add-Drop Multiplexer
RoCEv2	RDMA over Converged Ethernet version 2
SFTP	Secret File Transfer Protocol
SMB	Server Message Block
VM	Virtual Machine
ECN	Explicit Congestion Notification

References

REFERENCE	DESCRIPTION
[IOWNGF-DCI-FA]	IOWN Global Forum, “Data-Centric Infrastructure Functional Architecture,” Ver 2.0 (2023.03)
[IOWNGF-RDMAoverAPN-PoC]	IOWN Global Forum, “RDMA over Open APN PoC Reference,” Ver 1.0 (2022.07)
[IOWNGF-SIFI-UC]	IOWN Global Forum, “Services Infrastructure for Financial Industry Use Case,” Ver 1.0 (2024.07)
[IOWNGF-IDH-FA]	IOWN Global Forum, “Data Hub Functional Architecture,” Ver 2.0 (2023.07)
[Linux-perftest]	Linux-RDMA perftest: Infiniband Verbs Performance Tests https://github.com/linux-rdma/perftest
[IOWNGF-RDMAoverAPN-RecognizedPoC]	IOWN Global Forum, “RDMA over Open APN PoC Report,” Ver 1.0 (2024.06)
[QEMU]	QEMU: A generic and open source machine emulator and virtualizer https://www.qemu.org/, https://gitlab.com/qemu-project/qemu
[libvirt]	libvirt: The virtualization API https://libvirt.org, https://gitlab.com/libvirt/libvirt
[libvirt-virsh]	virsh: management user interface, https://libvirt.org/manpages/virsh.html
[stress-tool]	stress: tool to impose load on and stress test a computer system https://github.com/resurrecting-open-source-projects/stress

Appendix A. RDMA NIC configuration

Table A-1: RDMA NIC Configuration used in phase 1 of PoC project

<table>
<tr><th>Item</th><th>Description</th></tr>
<tr><td>IP address, interface MTU, and queue depth setting</td><td><u>100 GbE NIC on DCI Physical Node 1:</u>
#!/bin/bash
LINK="enp179s0f0np0"
IP="10.10.102.82/24"
MTU=4200

sudo ip link set $LINK up
sudo ip link set $LINK mtu $MTU
sudo ip addr add $IP dev $LINK

ethtool -G $LINK tx 8192
ethtool -G $LINK rx 8192

<u>100 GbE NIC on DCI Physical Node 2:</u>
LINK="enp175s0f0np0"
IP="10.10.102.92/24"
MTU=4200

sudo ip link set $LINK up
sudo ip link set $LINK mtu $MTU
sudo ip addr add $IP dev $LINK

ethtool -G $LINK tx 8192
ethtool -G $LINK rx 8192</td></tr>
<tr><td>ibv_devinfo</td><td><u>100 GbE NIC on DCI Physical Node 1:</u>
hca_id: mlx5_2
transport: InfiniBand (0)
fw_ver: 24.34.1002
node_guid: b83f:d203:00db:983c
sys_image_guid: b83f:d203:00db:983c
vendor_id: 0x02c9
vendor_part_id: 41686
hw_ver: 0x1
board_id: MT_0000000768
phys_port_cnt: 1
port: 1
state: PORT_ACTIVE (4)
max_mtu: 4096 (5)
active_mtu: 4096 (5)
sm_lid: 0
port_lid: 0
port_lmc: 0x00
link_layer: Ethernet
<u>100 GbE NIC on DCI Physical Node 2:</u>
hca_id: mlx5_2
transport: InfiniBand (0)
fw_ver: 24.34.1002
node_guid: b83f:d203:00db:96ac</td></tr>
</table>

	sys_image_guid: b83f:d203:00db:96ac vendor_id: 0x02c9 vendor_part_id: 41686 hw_ver: 0x1 board_id: MT_0000000768 phys_port_cnt: 1 port: 1 state: PORT_ACTIVE (4) max_mtu: 4096 (5) active_mtu: 4096 (5) sm_lid: 0 port_lid: 0 port_lmc: 0x00 link_layer: Ethernet
mlxconfig	**100 GbE NIC on DCI Physical Node 1 & 2:** (dump result is same) Device #1: ---------- Device type: BlueField2 Name: MBF2H536C-CECO_Ax_Bx Description: BlueField-2 P-Series DPU 100GbE Dual-Port QSFP56; integrated BMC; PCIe Gen4 x16; Secure Boot Enabled; Crypto Enabled; 32GB on-board DDR; 1GbE OOB management; FHHL Device: /dev/mst/mt41686_pciconf0 Configurations: Next Boot MEMIC_BAR_SIZE 0 MEMIC_SIZE_LIMIT _256KB(1) HOST_CHAINING_MODE DISABLED(0) HOST_CHAINING_CACHE_DISABLE False(0) HOST_CHAINING_DESCRIPTORS Array[0..7] HOST_CHAINING_TOTAL_BUFFER_SIZE Array[0..7] INTERNAL_CPU_MODEL EMBEDDED_CPU(1) INTERNAL_CPU_PAGE_SUPPLIER ECPF(0) INTERNAL_CPU_ESWITCH_MANAGER ECPF(0) INTERNAL_CPU_IB_VPORT0 ECPF(0) INTERNAL_CPU_OFFLOAD_ENGINE ENABLED(0) FLEX_PARSER_PROFILE_ENABLE 0 PROG_PARSE_GRAPH False(0) FLEX_IPV4_OVER_VXLAN_PORT 0 ROCE_NEXT_PROTOCOL 254 ESWITCH_HAIRPIN_DESCRIPTORS Array[0..7] ESWITCH_HAIRPIN_TOT_BUFFER_SIZE Array[0..7] PF_BAR2_SIZE 3 DPU_RESET_NOTIFICATION_ENABLED ENABLED(1) INTERNAL_CPU_RSHIM ENABLED(0) PF_NUM_OF_VF_VALID False(0) NON_PREFETCHABLE_PF_BAR False(0) VF_VPD_ENABLE False(0) PF_NUM_PF_MSIX_VALID False(0) PER_PF_NUM_SF False(0) STRICT_VF_MSIX_NUM False(0) VF_NODNIC_ENABLE False(0)

```
NUM_PF_MSIX_VALID                       True(1)
NUM_OF_VFS                              8
NUM_OF_PF                               2
PF_BAR2_ENABLE                          True(1)
HIDE_PORT2_PF                           False(0)
SRIOV_EN                                True(1)
PF_LOG_BAR_SIZE                         5
VF_LOG_BAR_SIZE                         0
NUM_PF_MSIX                             63
NUM_VF_MSIX                             11
INT_LOG_MAX_PAYLOAD_SIZE                AUTOMATIC(0)
PCIE_CREDIT_TOKEN_TIMEOUT               0
LAG_RESOURCE_ALLOCATION                 DEVICE_DEFAULT(0)
PHY_COUNT_LINK_UP_DELAY                 DELAY_NONE(0)
ACCURATE_TX_SCHEDULER                   False(0)
PARTIAL_RESET_EN                        False(0)
RESET_WITH_HOST_ON_ERRORS               False(0)
NVME_EMULATION_ENABLE                   False(0)
NVME_EMULATION_NUM_VF                   0
NVME_EMULATION_NUM_PF                   1
NVME_EMULATION_VENDOR_ID                5555
NVME_EMULATION_DEVICE_ID                24577
NVME_EMULATION_CLASS_CODE               67586
NVME_EMULATION_REVISION_ID              0
NVME_EMULATION_SUBSYSTEM_VENDOR_ID      0
NVME_EMULATION_SUBSYSTEM_ID             0
NVME_EMULATION_NUM_MSIX                 0
NVME_EMULATION_MAX_QUEUE_DEPTH          0
PCI_SWITCH_EMULATION_NUM_PORT           0
PCI_SWITCH_EMULATION_ENABLE             False(0)
VIRTIO_NET_EMULATION_ENABLE             False(0)
VIRTIO_NET_EMULATION_NUM_VF             0
VIRTIO_NET_EMULATION_NUM_PF             0
VIRTIO_NET_EMU_SUBSYSTEM_VENDOR_ID      6900
VIRTIO_NET_EMULATION_SUBSYSTEM_ID       1
VIRTIO_NET_EMULATION_NUM_MSIX           2
VIRTIO_BLK_EMULATION_ENABLE             False(0)
VIRTIO_BLK_EMULATION_NUM_VF             0
VIRTIO_BLK_EMULATION_NUM_PF             0
VIRTIO_BLK_EMU_SUBSYSTEM_VENDOR_ID      6900
VIRTIO_BLK_EMULATION_SUBSYSTEM_ID       2
VIRTIO_BLK_EMULATION_NUM_MSIX           2
PCI_DOWNSTREAM_PORT_OWNER               Array[0..15]
CQE_COMPRESSION                         BALANCED(0)
IP_OVER_VXLAN_EN                        False(0)
MKEY_BY_NAME                            False(0)
PRIO_TAG_REQUIRED_EN                    False(0)
UCTX_EN                                 True(1)
REAL_TIME_CLOCK_ENABLE                  False(0)
RDMA_SELECTIVE_REPEAT_EN                False(0)
PCI_ATOMIC_MODE
PCI_ATOMIC_DISABLED_EXT_ATOMIC_ENABLED(0)
TUNNEL_ECN_COPY_DISABLE                 False(0)
```

	LRO_LOG_TIMEOUT0 6 LRO_LOG_TIMEOUT1 7 LRO_LOG_TIMEOUT2 8 LRO_LOG_TIMEOUT3 13 LOG_TX_PSN_WINDOW 7 VF_MIGRATION_MODE DEVICE_DEFAULT(0) LOG_MAX_OUTSTANDING_WQE 7 ROCE_ADAPTIVE_ROUTING_EN False(0) TUNNEL_IP_PROTO_ENTROPY_DISABLE False(0) MULTI_PCI_RESOURCE_SHARING DEVICE_DEFAULT(0) ICM_CACHE_MODE DEVICE_DEFAULT(0) TLS_OPTIMIZE False(0) TX_SCHEDULER_BURST 0 ZERO_TOUCH_TUNING_ENABLE False(0) ROCE_CC_LEGACY_DCQCN True(1) LOG_MAX_QUEUE 17 LARGE_MTU_TWEAK_64 False(0) AES_XTS_TWEAK_INC_64 False(0) CRYPTO_POLICY UNRESTRICTED(1) LOG_DCR_HASH_TABLE_SIZE 11 MAX_PACKET_LIFETIME 0 DCR_LIFO_SIZE 16384 ROCE_CC_PRIO_MASK_P1 255 ROCE_CC_PRIO_MASK_P2 255 CLAMP_TGT_RATE_AFTER_TIME_INC_P1 True(1) CLAMP_TGT_RATE_P1 False(0) RPG_TIME_RESET_P1 300 RPG_BYTE_RESET_P1 32767 RPG_THRESHOLD_P1 1 RPG_MAX_RATE_P1 0 RPG_AI_RATE_P1 5 RPG_HAI_RATE_P1 50 RPG_GD_P1 11 RPG_MIN_DEC_FAC_P1 50 RPG_MIN_RATE_P1 1 RATE_TO_SET_ON_FIRST_CNP_P1 0 DCE_TCP_G_P1 1019 DCE_TCP_RTT_P1 1 RATE_REDUCE_MONITOR_PERIOD_P1 4 INITIAL_ALPHA_VALUE_P1 1023 MIN_TIME_BETWEEN_CNPS_P1 4 CNP_802P_PRIO_P1 6 CNP_DSCP_P1 48 CLAMP_TGT_RATE_AFTER_TIME_INC_P2 True(1) CLAMP_TGT_RATE_P2 False(0) RPG_TIME_RESET_P2 300 RPG_BYTE_RESET_P2 32767 RPG_THRESHOLD_P2 1 RPG_MAX_RATE_P2 0 RPG_AI_RATE_P2 5 RPG_HAI_RATE_P2 50 RPG_GD_P2 11 RPG_MIN_DEC_FAC_P2 50

	RPG_MIN_RATE_P2 1
	RATE_TO_SET_ON_FIRST_CNP_P2 0
	DCE_TCP_G_P2 1019
	DCE_TCP_RTT_P2 1
	RATE_REDUCE_MONITOR_PERIOD_P2 4
	INITIAL_ALPHA_VALUE_P2 1023
	MIN_TIME_BETWEEN_CNPS_P2 4
	CNP_802P_PRIO_P2 6
	CNP_DSCP_P2 48
	LLDP_NB_DCBX_P1 False(0)
	LLDP_NB_RX_MODE_P1 OFF(0)
	LLDP_NB_TX_MODE_P1 OFF(0)
	LLDP_NB_DCBX_P2 False(0)
	LLDP_NB_RX_MODE_P2 OFF(0)
	LLDP_NB_TX_MODE_P2 OFF(0)
	ROCE_RTT_RESP_DSCP_P1 0
	ROCE_RTT_RESP_DSCP_MODE_P1 DEVICE_DEFAULT(0)
	ROCE_RTT_RESP_DSCP_P2 0
	ROCE_RTT_RESP_DSCP_MODE_P2 DEVICE_DEFAULT(0)
	DCBX_IEEE_P1 True(1)
	DCBX_CEE_P1 True(1)
	DCBX_WILLING_P1 True(1)
	DCBX_IEEE_P2 True(1)
	DCBX_CEE_P2 True(1)
	DCBX_WILLING_P2 True(1)
	KEEP_ETH_LINK_UP_P1 True(1)
	KEEP_IB_LINK_UP_P1 False(0)
	KEEP_LINK_UP_ON_BOOT_P1 False(0)
	KEEP_LINK_UP_ON_STANDBY_P1 False(0)
	DO_NOT_CLEAR_PORT_STATS_P1 False(0)
	AUTO_POWER_SAVE_LINK_DOWN_P1 False(0)
	KEEP_ETH_LINK_UP_P2 True(1)
	KEEP_IB_LINK_UP_P2 False(0)
	KEEP_LINK_UP_ON_BOOT_P2 False(0)
	KEEP_LINK_UP_ON_STANDBY_P2 False(0)
	DO_NOT_CLEAR_PORT_STATS_P2 False(0)
	AUTO_POWER_SAVE_LINK_DOWN_P2 False(0)
	NUM_OF_VL_P1 _4_VLs(3)
	NUM_OF_TC_P1 _8_TCs(0)
	NUM_OF_PFC_P1 8
	VL15_BUFFER_SIZE_P1 0
	NUM_OF_VL_P2 _4_VLs(3)
	NUM_OF_TC_P2 _8_TCs(0)
	NUM_OF_PFC_P2 8
	VL15_BUFFER_SIZE_P2 0
	DUP_MAC_ACTION_P1 LAST_CFG(0)
	MPFS_MC_LOOPBACK_DISABLE_P1 False(0)
	MPFS_UC_LOOPBACK_DISABLE_P1 False(0)
	UNKNOWN_UPLINK_MAC_FLOOD_P1 False(0)
	SRIOV_IB_ROUTING_MODE_P1 LID(1)
	IB_ROUTING_MODE_P1 LID(1)
	DUP_MAC_ACTION_P2 LAST_CFG(0)
	MPFS_MC_LOOPBACK_DISABLE_P2 False(0)

	MPFS_UC_LOOPBACK_DISABLE_P2 False(0) UNKNOWN_UPLINK_MAC_FLOOD_P2 False(0) SRIOV_IB_ROUTING_MODE_P2 LID(1) IB_ROUTING_MODE_P2 LID(1) PHY_AUTO_NEG_P1 DEVICE_DEFAULT(0) PHY_RATE_MASK_OVERRIDE_P1 False(0) PHY_FEC_OVERRIDE_P1 DEVICE_DEFAULT(0) PHY_AUTO_NEG_P2 DEVICE_DEFAULT(0) PHY_RATE_MASK_OVERRIDE_P2 False(0) PHY_FEC_OVERRIDE_P2 DEVICE_DEFAULT(0) PF_TOTAL_SF 0 PF_SD_GROUP 0 PF_SF_BAR_SIZE 0 PF_NUM_PF_MSIX 63 ROCE_CONTROL ROCE_ENABLE(2) PCI_WR_ORDERING per_mkey(0) MULTI_PORT_VHCA_EN False(0) PORT_OWNER True(1) ALLOW_RD_COUNTERS True(1) RENEG_ON_CHANGE True(1) TRACER_ENABLE True(1) IP_VER IPv4(0) BOOT_UNDI_NETWORK_WAIT 0 UEFI_HII_EN True(1) BOOT_DBG_LOG False(0) UEFI_LOGS DISABLED(0) BOOT_VLAN 1 LEGACY_BOOT_PROTOCOL PXE(1) BOOT_INTERRUPT_DIS False(0) BOOT_LACP_DIS True(1) BOOT_VLAN_EN False(0) BOOT_PKEY 0 P2P_ORDERING_MODE DEVICE_DEFAULT(0) EXP_ROM_VIRTIO_NET_PXE_ENABLE True(1) EXP_ROM_VIRTIO_NET_UEFI_ARM_ENABLE True(1) EXP_ROM_VIRTIO_NET_UEFI_x86_ENABLE True(1) EXP_ROM_VIRTIO_BLK_UEFI_ARM_ENABLE True(1) EXP_ROM_VIRTIO_BLK_UEFI_x86_ENABLE True(1) EXP_ROM_NVME_UEFI_x86_ENABLE True(1) ATS_ENABLED False(0) DYNAMIC_VF_MSIX_TABLE False(0) EXP_ROM_UEFI_ARM_ENABLE True(1) EXP_ROM_UEFI_x86_ENABLE True(1) EXP_ROM_PXE_ENABLE True(1) ADVANCED_PCI_SETTINGS False(0) SAFE_MODE_THRESHOLD 10 SAFE_MODE_ENABLE True(1)

Appendix B. Measurement Commands used for RDMA over Open APN

Table B-1: Commands used for RDMA over Open APN performance evaluation

Benchmark & Operation type	perftest commands
Throughput in RDMA SEND	**Server on DCI Physical Node 2:** 1.ib_send_bw -d mlx5_2 -R -t 128 -a -n 15000 --report_gbits -F 2.ib_send_bw -d mlx5_2 -R **-t 2048** -a -n 15000 --report_gbits -F 3.ib_send_bw -d mlx5_2 -R -t 2048 **-q 4** -a -n 15000 --report_gbits -F **Client on DCI Physical Node 1:** 1.ib_send_bw -d mlx5_2 -R -t 128 -a 10.10.102.92 -n 15000 -F --report_gbits 2.ib_send_bw -d mlx5_2 -R **-t 2048** -a 10.10.102.92 -n 15000 -F --report_gbits 3.ib_send_bw -d mlx5_2 -R -t 2048 **-q 4** -a 10.10.102.92 -n 15000 -F --report_gbits
Latency in RDMA SEND	**Server on DCI Physical Node 2:** 1.ib_send_lat -d mlx5_2 -R -t 128 -a -n 15000 --report_gbits -F 2.ib_send_lat -d mlx5_2 -R **-t 2048** -a -n 15000 --report_gbits -F **Client on DCI Physical Node 1:** 1.ib_send_lat -d mlx5_2 -R -t 128 -a 10.10.102.92 -n 15000 -F --report_gbits 2.ib_send_lat -d mlx5_2 -R **-t 2048** -a 10.10.102.92 -n 15000 -F --report_gbits
Throughput in RDMA WRITE	**Server on DCI Physical Node 2:** 1.ib_write_bw -d mlx5_2 -R -t 128 -a -n 15000 --report_gbits -F 2.ib_write_bw -d mlx5_2 -R **-t 2048** -a -n 15000 --report_gbits -F 3.ib_write_bw -d mlx5_2 -R -t 2048 **-q 4** -a -n 15000 --report_gbits -F **Client on DCI Physical Node 1:** 1.ib_write_bw -d mlx5_2 -R -t 128 -a 10.10.102.92 -n 15000 -F --report_gbits 2.ib_write_bw -d mlx5_2 -R **-t 2048** -a 10.10.102.92 -n 15000 -F --report_gbits 3.ib_write_bw -d mlx5_2 -R -t 2048 **-q 4** -a 10.10.102.92 -n 15000 -F --report_gbits
Latency in RDMA WRITE	**Server on DCI Physical Node 2:** 1.ib_write_lat -d mlx5_2 -R -t 128 -a -n 15000 --report_gbits -F 2.ib_write_lat -d mlx5_2 -R **-t 2048** -a -n 15000 --report_gbits -F **Client on DCI Physical Node 1:** 1.ib_write_lat -d mlx5_2 -R -t 128 -a 10.10.102.92 -n 15000 -F --report_gbits 2.ib_write_lat -d mlx5_2 -R **-t 2048** -a 10.10.102.92 -n 15000 -F --report_gbits
Throughput in RDMA READ	**Server on DCI Physical Node 2:** 1.ib_read_bw -d mlx5_2 -R -t 16 -a -n 4096 -q 16 --report_gbits -F 2.ib_read_bw -d mlx5_2 -R -t **2048** -a -n 4096 -q 16 --report_gbits -F **Client on DCI Physical Node 1:** 1.ib_read_bw -d mlx5_2 -R -t 16 -a 10.10.102.92 -n 4096 -q 16 -F --report_gbits 2.ib_read_bw -d mlx5_2 -R -t **2048** -a 10.10.102.92 -n 4096 -q 16 -F --report_gbits
Latency in RDMA READ	**Server on DCI Physical Node 2:** 1.ib_read_lat -d mlx5_2 -R -t 2048 -a -n 4096 -F **Client on DCI Physical Node 1:** 1.ib_read_lat -d mlx5_2 -R -t 2048 -a 10.10.102.92 -n 4096 -F

Appendix C. 100G RDMA WRITE/SEND/READ over Open APN Throughput

We run the tests twenty times using perftest [Linux-perftest] tool with the commands listed in Table B-1, and organize the results in Table C-1. We use "Min", "Avg" and "Max" to record the minimum, average and maximum of the twenty-times results respectively. When the message size is increases, RDMA throughput should theoretically approach the bandwidth, which is 100 Gbps in this phase of PoC. However, in our tests, the performance is not as expected. We mark unexpected results using red text and mark ideal result using blue text. Please see "Finding Details" in Table 6.5-3 for the related information.

Table C-1: 100G RDMA WRITE/SEND/READ over Open APN Min/Average/Max Throughput

Msg_size	4 KB			8 KB			16 KB			32 KB		
Type	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max
WRITE throughput (Gbps)	4.35	4.4	4.40	17.22	17.3	17.56	17.22	17.3	17.56	32.11	34.2	35.09
SEND throughput (Gbps)	4.32	4.3	4.36	8.64	8.7	8.72	17.27	17.3	17.44	34.53	34.6	34.81
READ throughput (Gbps)	1.01	1.9	2.19	1.48	3.8	4.39	4.87	8.2	8.78	8.6	17.0	17.54
Msg_size	64 KB			128 KB			256 KB			512 KB		
Type	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max
WRITE throughput (Gbps)	51.72	60.6	69.91	56.13	76.1	97.68	56.30	80.3	97.98	67.92	84.0	98.12
SEND throughput (Gbps)	68.94	69.1	69.30	91.16	91.5	91.68	90.45	91.9	92.20	91.92	92.2	92.44
READ throughput (Gbps)	19.72	34.2	35.06	28.97	67.7	69.87	95.1	97.6	97.92	97.19	97.9	98.14
Msg_size	1 MB			2 MB			4 MB			8 MB		
Type	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max
WRITE throughput (Gbps)	66.78	86.3	98.20	66.94	83.8	98.23	70.12	86.2	98.25	71.44	88.4	98.26
SEND throughput (Gbps)	91.95	92.3	92.57	90.99	92.4	93.00	92.14	92.5	92.67	92.15	92.5	92.69
READ throughput (Gbps)	97.53	98.0	98.2	87.16	97.5	98.25	88.24	97.6	98.26	90.16	97.7	98.26

Appendix D. 100G RDMA WRITE/SEND/READ over Open APN Latency

Table D-1: RDMA over Open APN latency in microsecond (transmission distance=380 km)

Type / Arguments	RDMA WRITE		RDMA SEND		RDMA READ	
Tx depth / msg_size	128	2048	128	2048	128	2048
2 Byte	1907.171	1906.089	1905.173	1904.304	3810.567	3809.451
4 Byte	1905.625	1906.031	1903.242	1903.176	3805.553	3804.958
8 Byte	1905.625	1905.199	1903.908	1903.933	3807.153	3806.935
16 Byte	1904.937	1905.168	1905.16	1904.427	3807.743	3808.723
32 Byte	1905.281	1904.517	1903.687	1904.089	3808.268	3807.718
64 Byte	1906.135	1905.397	1904.488	1903.049	3809.492	3808.714
128 Byte	1906.019	1906.014	1903.384	1904.255	3805.189	3806.363
256 Byte	1905.363	1904.959	1903.064	1903.544	3808.462	3809.167
512 Byte	1905.261	1905.54	1905.678	1903.825	3806.583	3807.884
1 KB	1904.936	1906.271	1903.28	1903.911	3810.832	3809.696
2 KB	1905.576	1906.462	1904.969	1903.812	3808.446	3807.902
4 KB	1905.305	1905.978	1904.929	1905.699	3808.469	3809.269
8 KB	1906.467	1906.042	1905.199	1905.223	3810.236	3811.21
16 KB	1907.145	1907.183	1906.648	1906.817	3812.594	3812.833
32 KB	1909.271	1909.69	1907.846	1907.084	3813.834	3813.145
64 KB	1911.948	1911.322	1909.765	1910.497	3814.245	3814.009
128 KB	1916.769	1917.273	1915.616	1915.65	3821.531	3821.613
256 KB	1928.627	1929.169	1928.001	1927.072	3831.889	3831.779
512 KB	1949.191	1949.427	1950.849	1951.006	3853.81	3853.787
1 MB	1994.16	1993.59	1997.739	1997.77	3903.48	3903.127
2 MB	2082.11	2083.042	2090.115	2091.775	3994.13	3995.092
4 MB	2256.305	2256.722	2276.347	2276.155	4251.681	4251.45
8 MB	2605.536	2604.713	2649.389	2648.012	4523.742	4522.149

Appendix E. Congestion Issue Related Investigation

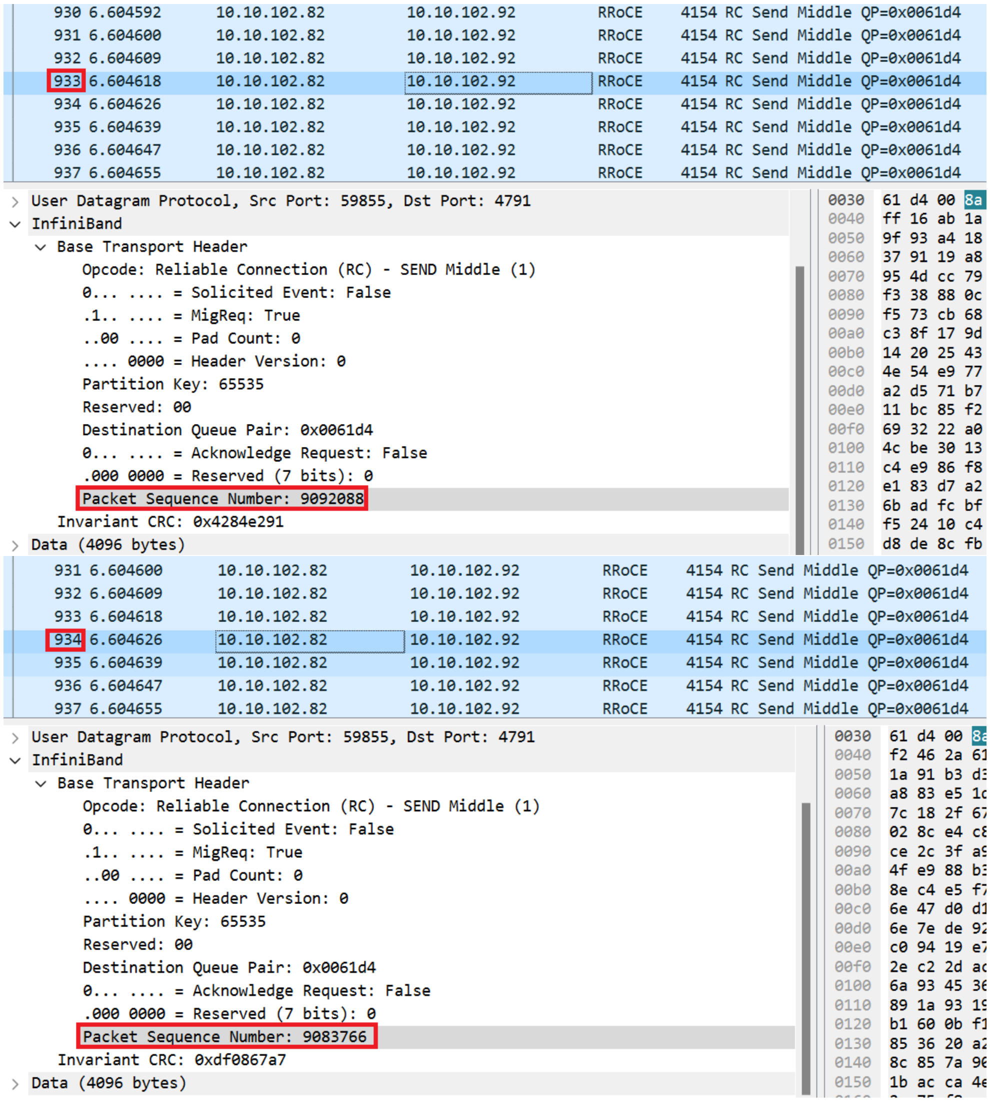

Figure E-1: Captured RDMA packets in a row showing that retransmission happened

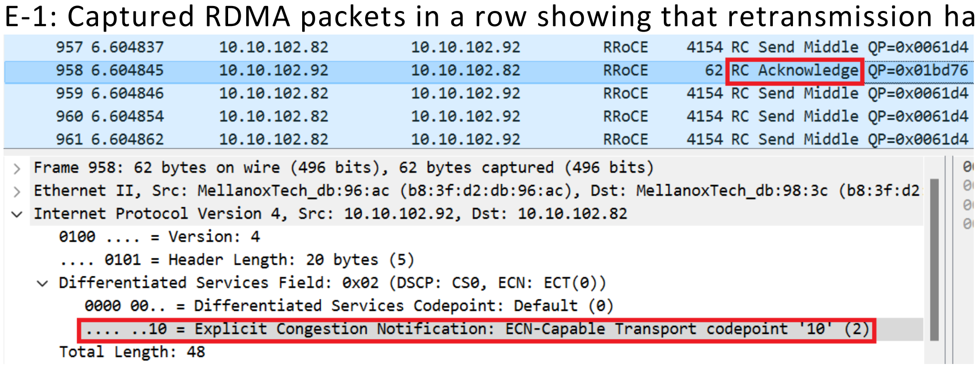

Figure E-2: Captured RDMA ACK packet showing no congestion experienced

Appendix F. Commands for VM Live Migration Performance Evaluation

Table F-1: Commands for VM Live Migration Performance Evaluation

Item	Description
VM live migration	**RDMA over Open APN:** time virsh migrate --live --rdma-ping-all --listen-address 10.10.102.92 --migrateuri rdma://10.10.102.92 vm1 qemu+tcp://10.10.102.92/system --copy-storage-all **TCP over Open APN:** time virsh migrate --live --listen-address 10.10.102.92 --migrateuri tcp://10.10.102.92 vm1 qemu+tcp://10.10.102.92/system --copy-storage-all **Note 1:** "time" is a Linux command to measures and report the duration of command execution. **Note 2:** virsh [libvirt-virsh] is a command line interface tool built on the libvirt [libvirt] management API. It enables users to manage virtual machines and the hypervisor.
Impose memory stress inside the VMs needs to be migrated	**For both RDMA and TCP over Open APN:** To impose memory stress in a VM, we use the stress tool [stress-tool] and its several parameters for the purpose as listed below: • We configured **--timeout** parameter as 600 to specify an execution time of 600 seconds. • We configured **--vm** parameter as 1 to use a single CPU thread for imposing stress. • We calculate the value of 20%, 40%, 60%, 80% of a VM's memory specification, and round up those values the nearest whole number, which is used to configure **--vm-byte** parameter for how much memory stress we want to impose. • We configured **--vm-hang** parameter as 1 to pause for 1 second when the memory stress, which we specified using --vm-byte parameter, is imposed completely. *Apply memory stress of 20%, 40%, 60%, 80% to a VM with the specifications of 4 cores, **8 GB memory**, and 100 GB storage:* **20%**: stress --vm 1 --vm-byte **2G** --vm-hang 1 --timeout 600 **40%**: stress --vm 1 --vm-byte **4G** --vm-hang 1 --timeout 600 **60%**: stress --vm 1 --vm-byte **5G** --vm-hang 1 --timeout 600 **80%**: stress --vm 1 --vm-byte **7G** --vm-hang 1 --timeout 600 *Apply memory stress of 20%, 40%, 60%, 80% to a VM with the specifications of 4 cores, **16 GB memory**, and 100 GB storage:* **20%**: stress --vm 1 --vm-byte **4G** --vm-hang 1 --timeout 600 **40%**: stress --vm 1 --vm-byte **7G** --vm-hang 1 --timeout 600 **60%**: stress --vm 1 --vm-byte **10G** --vm-hang 1 --timeout 600 **80%**: stress --vm 1 --vm-byte **13G** --vm-hang 1 --timeout 600 *Apply memory stress of 20%, 40%, 60%, 80% to a VM with the specifications of 4 cores, **32 GB memory**, and 100 GB storage:* **20%**: stress --vm 1 --vm-byte **7G** --vm-hang 1 --timeout 600 **40%**: stress --vm 1 --vm-byte **13G** --vm-hang 1 --timeout 600 **60%**: stress --vm 1 --vm-byte **20G** --vm-hang 1 --timeout 600 **80%**: stress --vm 1 --vm-byte **26G** --vm-hang 1 --timeout 600 *Apply memory stress of 20%, 40%, 60%, 80% to a VM with the specifications of 4 cores, **64 GB memory**, and 100 GB storage:* **20%**: stress --vm 1 --vm-byte **13G** --vm-hang 1 --timeout 600 **40%**: stress --vm 1 --vm-byte **26G** --vm-hang 1 --timeout 600 **60%**: stress --vm 1 --vm-byte **39G** --vm-hang 1 --timeout 600 **80%**: stress --vm 1 --vm-byte **52G** --vm-hang 1 --timeout 600

Document History

Version	Date	Author	Description of Change
0.1	July 1, 2024	Jia-An Tsai, CHT	Initial draft
0.2	July 22, 2024	Jia-An Tsai, CHT	• Reflecting comments from 1st round of DCS TF informal review • Add “VM live migration” related content • Restructure the document for “VM live migration” related content
0.3	Aug 5, 2024	Jia-An Tsai, CHT	• Add reference to Services Infrastructure for Financial Industry Use Case published in July 2024. • Reflecting comments from 2nd round of DCS TF informal review
1.0	Sep 26, 2024	Jia-An Tsai, CHT	• Add SSF PoC report cover sheet • Update “PoC Stage Completion Status” • Reflecting comments from TUCWG formal review