MULTIPATH RELIABLE CONNECTION (MRC) WHITE PAPER - HIGH PERFORMANCE SOLUTION FOR AI/ML WORKLOADS FOR AI SCALE-OUT NETWORKS USING AMD PENSANDO™ AI NICs

AUGUST 2026

DISCLAIMER

The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions, and typographical errors. The information contained herein is subject to change and may be rendered inaccurate for many reasons, including but not limited to product and roadmap changes, component and motherboard version changes, new model and/or product releases, product differences between differing manufacturers, software changes, BIOS flashes, firmware upgrades, or the like. Any computer system has risks of security vulnerabilities that cannot be completely prevented or mitigated. AMD assumes no obligation to update or otherwise correct or revise this information. However, AMD reserves the right to revise this information and to make changes from time to time to the content hereof without obligation of AMD to notify any person of such revisions or changes.

THIS INFORMATION IS PROVIDED 'AS IS." AMD MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND ASSUMES NO RESPONSIBILITY FOR ANY INACCURACIES, ERRORS, OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION. AMD SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AMD BE LIABLE TO ANY PERSON FOR ANY RELIANCE, DIRECT, INDIRECT, SPECIAL, OR OTHER CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF ANY INFORMATION CONTAINED HEREIN, EVEN IF AMD IS EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

AMD, the AMD Arrow logo, and combinations thereof are trademarks of Advanced Micro Devices, Inc. Other product names used in this publication are for identification purposes only and may be trademarks of their respective companies.

© 2022–2026 Advanced Micro Devices, Inc. All Rights Reserved.

REVISION HISTORY

VERSION	DESCRIPTION	DATE
1.0	Initial Version	April 17th, 2026
2.0	2nd Version	August 20th, 2026

CONTENTS

Revision History **3**
Contents **4**
Executive Summary **5**
Introduction **5**
Problem Statement. **6**
Under-utilization of Network and Hash Collisions **6**
Inefficient Retransmission Schemes **6**
Tuning Complexity and Low Throughput with Lossless Fabrics **6**
Solution **8**
Packet Spraying **9**
At the sender AI-NIC (multi-plane spray) 9
Mode1: At the network switches (DLB based spray) 10
Mode2: At the sender AI-NIC (EV based spray) 11
Choice of Packet Spraying Mode 12
Mode1: No EV-based packet spray on NIC, use DLB on switches 12
Mode2: EV-based packet spray on NIC, ECMP based LB hash on switches 12
Network Expectations **13**
Load-balancing 13
ECN marking 13
Packet-Trimming 13
MTU 13
No vendor lock-in 13
Packet-Types **14**
Congestion-Control using SACKs **15**
Congestion-Control using NACKs **16**
Probes **17**
Network Signal based Congestion Control (NSCC) **17**
EV states 17
Congestion Window Adjustments 18
QoS Considerations **18**
Performance Results and Benefits **19**
Up to 25% RCCL throughput improvement with MRC 19
Up to 50% faster model convergence in networks with 0.1% traffic loss 19
Considerations **20**
Configuration Best Practices **20**
Supported Scale **20**
Configuration on AI-NIC **20**
Enable the MRC profile 20
QoS configuration 21
QoS configuration on switch. 21
Mode1 Configuration: No packet spray on NIC, DLB enabled on switches 21
Mode2 Configuration: EV-based packet spray on NIC, ECMP hash-based LB on switches 21
Claims **22**
PEN-016 22
PEN-019 22
References **22**

EXECUTIVE SUMMARY

This paper introduces MRC (Multipath Reliable Connection), a next-generation transport solution designed to overcome bandwidth inefficiencies and congestion challenges in AI scale-out networks. It explains how MRC improves GPU cluster interconnect performance without relying on lossless fabrics or complex tuning.

KEY BENEFITS INCLUDE:

- Up to 25% higher RCCL (ROCm Communication Collectives Library) throughput compared to standard RoCEv2.
- Up to 50% faster model convergence and loss recovery.
- Flexible deployment options, including EV-based NIC spraying and switch-based dynamic load balancing.
- Seamless integration with existing AI compute and networking infrastructure and ease of deployment for both lossy and lossless networks.

MRC combines adaptive packet spraying, intelligent congestion control, and selective retransmissions to maximize network utilization and accelerate AI/ML workload job completion times. The following sections detail the challenges with traditional data center designs, the architecture and operational modes of MRC, and configuration best practices for optimal performance.

INTRODUCTION

Current data center network designs face significant challenges when supporting high-performance AI scale-out environments. This white paper talks about the problems faced by traditional data center (DC) designs and networks when they are deployed to be used in connecting high-performance back-end AI scale-out networks to interconnect GPU clusters. It explains the motivation for introducing new protocols / standards such as MRC that address these challenges and renders a high-performance scale-out network to interconnect several GPU nodes/clusters. MRC achieves this by defining and implementing innovative features designed to optimize performance and reliability:

- Adaptive/intelligent packet-spray
- Out-of-order packet handling and in-order delivery
- Selective (SACK) and negative (NACK) acknowledgements that lead to faster re-transmissions and improved congestion recovery
- New probing techniques for faster path-failure detection and recovery

The overall goal is to increase the throughput of the scale-out network, reduce end-to-end latency, reduce the amount of time spent waiting for networking, avoid congestion whenever possible, detect/manage and react to congestion faster. In summary, an efficient network leads to GPUs not idling out waiting for the network and this solution helps utilize GPUs (the most expensive resource in AI clusters) to the fullest.

In this paper, the features of MRC are explained at a high level with the intent to help in understanding the need, its advantages/benefits, terminology, and some data on performance/scale for active deployments. A few modes of deployments are also recommended, and it is up to the customer to decide which mode/design best suits their needs. A few configuration snippets and operational monitoring procedures are also mentioned for reference.

Before presenting the details of MRC, this paper first examines the limitations of traditional network designs and nature of RoCEv2 traffic to then establish the motivation behind the features implemented as part of MRC.

PROBLEM STATEMENT

AI model training and inference workload demand tens of thousands of GPUs working collaboratively in a distributed computing environment across several compute nodes. To get to this scale, we need hundreds or thousands of inter-connected nodes (each with a limited number of GPUs, typically 8) The jobs running on these nodes exchange the state through RDMA for a distributed computing job using collective communication go libraries (CCLs), within the cluster. The nature of the AI/ML workloads and CCLs involve frequent exchange and synchronization of raw and processed model data and computations across several nodes which traverse a network. This network between large scale GPU nodes or a cluster of nodes is called the "scale-out" network. The term "back-end" network or "east-west" network is also used inter-changeably with the term scale-out network.

UNDER-UTILIZATION OF NETWORK AND HASH COLLISIONS

For Ethernet based AI clusters, customers can use typical CLOS DC 2-tier (leaf,spine) or 3-tier (leaf, spine, super-spine) architectures to inter-connect several GPU hosts. The type of traffic between GPUs is mostly RDMA / RoCEv2 data traffic. When a pair of GPUs communicate with each other, they communicate over a transmit-receive queue-pair (QP) connection. Networks in the CLOS fabric typically load-balance based on 5-tuple packet parameters also called 5-tuple ECMP hash. Now given a QP connection, all packets of a given QP or a flow, use the same path in the CLOS fabric. Even if the fabric is built with multi-path links comprising say 100G/200G/400G/800G links, with standard RoCEv2 deployment, a given QP always ends up using the same path or set of links across the fabric as it hashes the same way for all packets in that flow. Elephant flows and long-lived flows can hog paths for a long duration. Flows from several other QPs can also hash to the same set of links leading to congestion easily. So effectively, even if there are several other non-colliding paths available, we can end up in scenarios where some links/paths are congested whereas others are underutilized. This results in an overall reduction in network throughput for applications using GPUs.

INEFFICIENT RETRANSMISSION SCHEMES

Congestion leads to packet drops, and packet drops lead to re-transmissions. RoCEv2 uses what is called a "Go-Back-N" re-transmission mechanism. For a packet sequence, 1, 2...N... M, when packet N is lost, the sender re-transmits packet N and all subsequent packets till "M". This is very inefficient, and it increases the overall data-delivery time and reduces the effective overall bandwidth (BW) consumed by unwanted re-transmissions. Also, the re-transmitted packets are more likely to use the same congested path as the original packet and may get dropped again to worsen the situation.

TUNING COMPLEXITY AND LOW THROUGHPUT WITH LOSSLESS FABRICS

RoCEv2 deployments recommend using lossless fabric and use ECN-marking, PFC, DCQCN congestion control to achieve this, but customers view these protocols / standards as **complex features** requiring a lot of tuning of parameters based on their environment and applications[1]. PFC works well in providing a lossless fabric, but the problem with PFC is that it can slow down all senders, especially slow-down senders not contributing to congestion as well since the protocol is class-aware but not path-aware. Reduction in sender rates across the fabric again reduces the overall network throughput, and this has considerable impact on GPU job completion times.

Also, even if newer load-balancing schemes like per-packet based DLB (dynamic load balancing) are used in fabric, the RoCEv2 traffic can get out-of-order. RoCEv2 is UDP based so there is no in-built in-order delivery feature like TCP. Re-sequencing of out-of-order packets at the receiver is a very compute intensive operation which can further reduce overall GPU job completion times.

The following figure illustrates the key challenges associated with traditional network designs and protocols while supporting RDMA over scale-out networks:

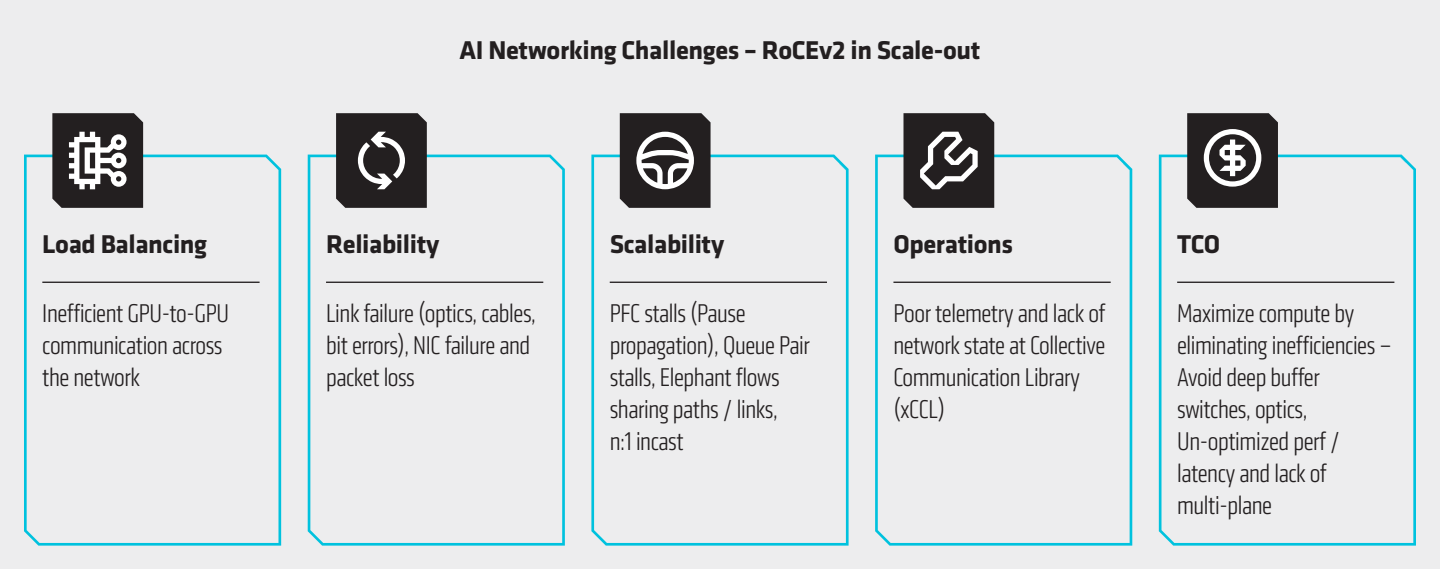


Figure 1: RoCEv2 in Scale-out

The networking solution needed here needs to use all the bandwidth of the fabric if available, explore efficient congestion avoidance schemes, and utilize congestion detection and congestion management algorithms. Also, this solution needs to improve and make re-transmissions more efficient, fasten failure detection and recovery, and handling out-of-order packet reception while doing in-order completions to GPU memory / application.

In the next section, we describe how MRC defines and solves the various challenges presented by nature of today's AI and HPC workloads while providing a significant boost in overall performance and reliability.

SOLUTION

In this section, we explain the features of MRC and the benefits it brings in. The main goal of MRC is to define a reliable transport protocol for RDMA traffic between GPUs over the scale-out network thereby improving the scale and performance characteristics of AI/ML workloads and leading to faster job completion times for AI training and inference workloads, reducing the time to train models and ultimately reduce the overall TCO for a deployment. The solution helps in utilizing these GPUs more, therefore leading to faster job completion times.

The following are the key features and enhancements provided by MRC over traditional networks:

- Packet spraying using EV (entropy value) variation by source AI NIC
- In-order delivery by direct placement of out-of-order (OOO) packets by receiver AI NIC into destination GPU's memory
- Congestion-control detection using feedback from ECN, RTT measurements
- Selective-acknowledgement (SACK) for efficient loss retransmission
- Negative-acknowledgment (NACK) for quick retransmission of dropped packets (this assumes switches trimming support)
- Fast failure / congestion detection using EV probes
- No dependency on lossless networks and PFC – works on lossy networks enabling highly scalable scale-out fabric buildouts.

As GPU clusters expand beyond one data center, it requires inter-connecting GPUs for distributed training across DCs as well. We call this type of network "scale-across" network. AI NICs can be deployed at the edge of the DC and at the terminating point of DCI and can solve similar challenges seen in scale-out networks as well. All features of MRC, using different AI transport configuration parameters, can be used on AI NIC in scale-across networks as well to give similar performance benefits.

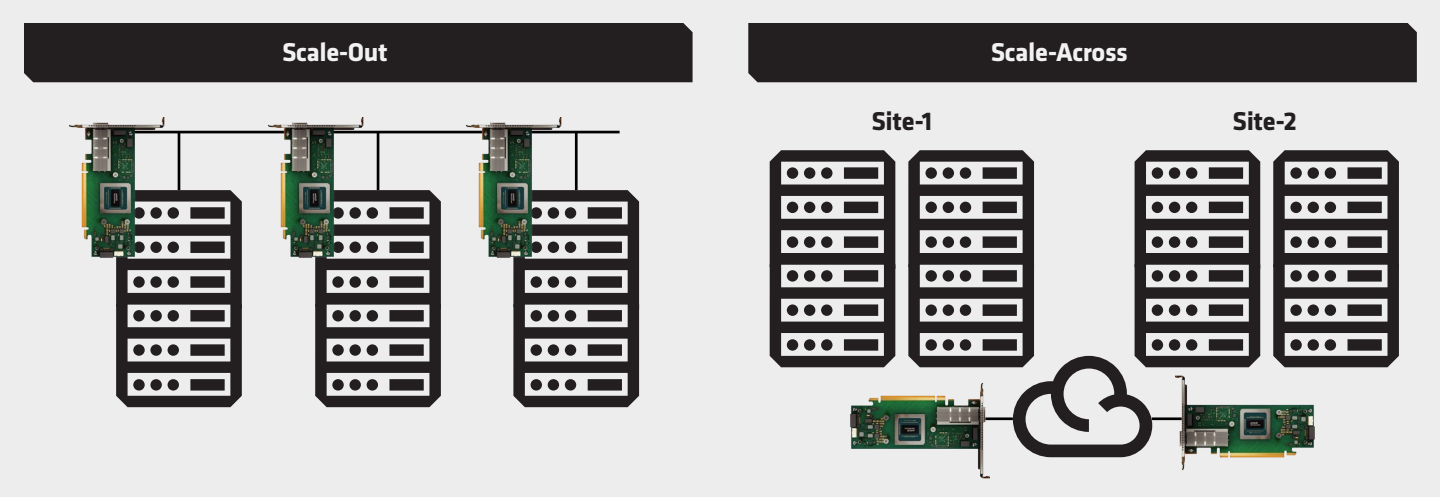


Figure 2: Scale-out vs Scale-across

In the subsequent sections, we explain each of the MRC features in more detail.

PACKET SPRAYING

Packet spraying is defined as the ability of a device (NIC/switch) to load-balance packets of the same flow (or RDMA QP) or the same IP 5-tuple across multiple equal-cost paths / links in the fabric. The goal of packet spraying is to utilize the full available network bisectional bandwidth, thereby maximizing the end-to-end throughput resulting in faster GPU job completion times. To add to this, the packet spraying as defined in MRC is adaptive, intelligent, and dynamic in nature as it is path aware as the spraying is adjusted in a way that congested paths in the network are avoided to alleviate further congestion. In other words, adaptive packet-spray accomplishes path-aware congestion control. Although packet-spray is applied to both RDMA Write and WriteIMM operations in forward direction and to transport ACKs in the reverse direction, it is largely effective in alleviating congestion issues for data packets as opposed to ACK packets as ACK packets are generally smaller in size and not subject to as much congestion as data packets.

Packet spray itself can occur at multiple points in the network:

AT THE SENDER AI-NIC (MULTI-PLANE SPRAY)

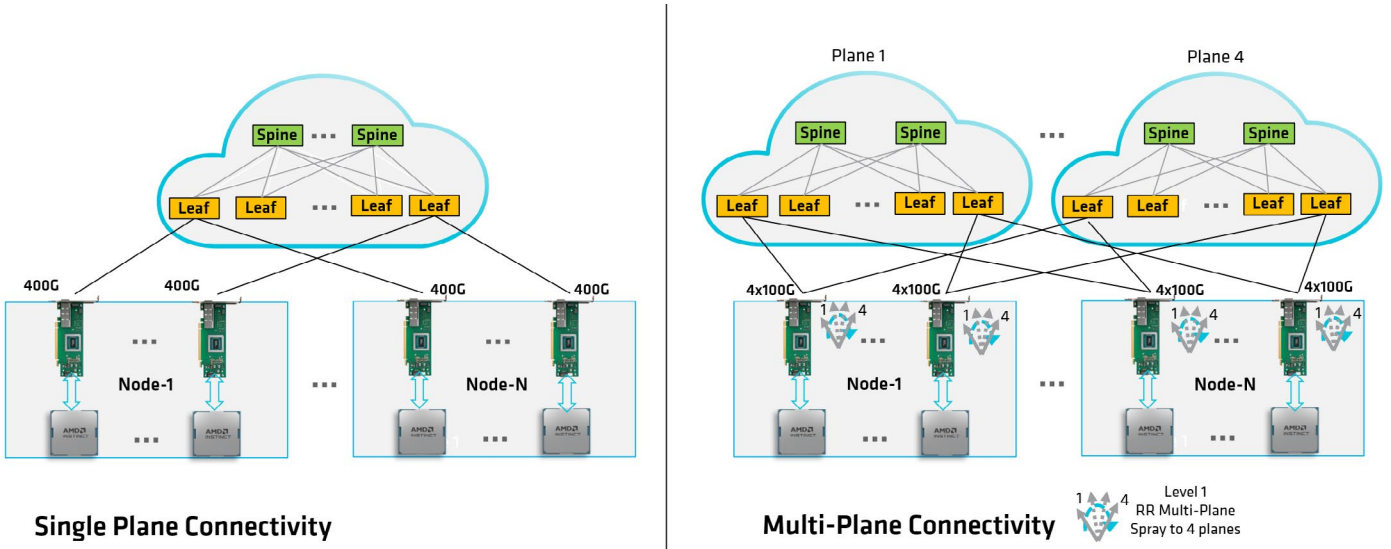


Figure 3: Multi-plane spray

If the AI NIC's uplink port (say 400G port) is connected as-is to a single 400G port on the uplink leaf switch, it is called a single-plane uplink connection (as shown in the figure of the left).

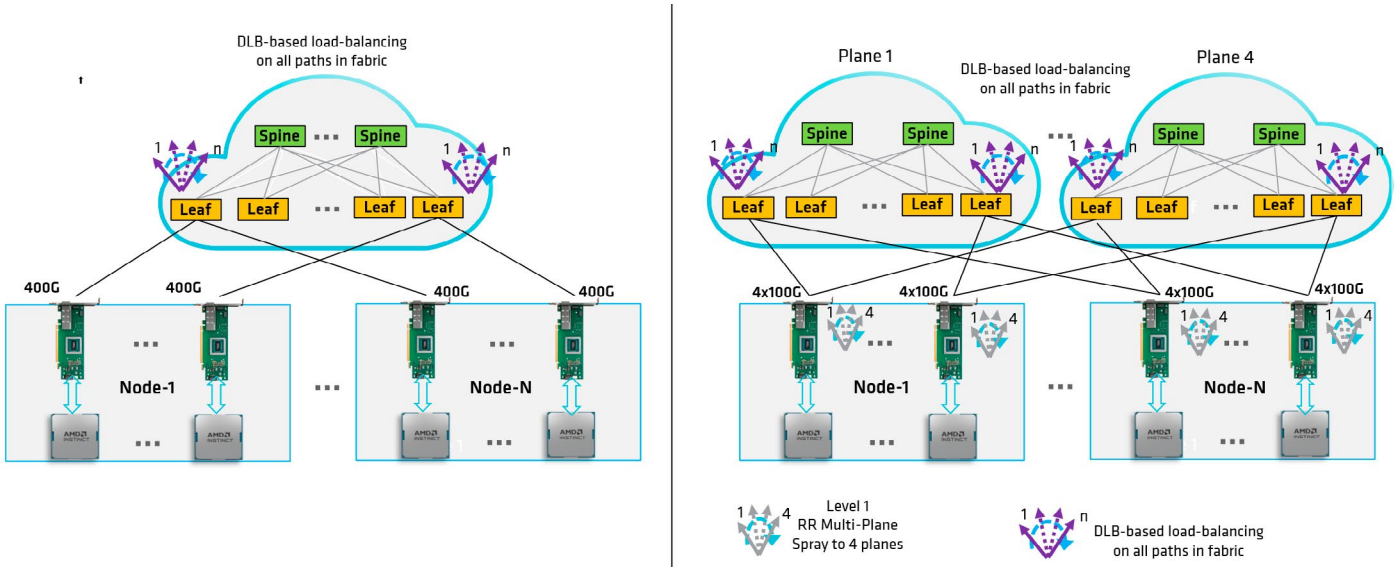
If the AI NIC's uplink port (say 400G port) is broken out into 4x100G ports and each of the 4x100G ports is connected to a different leaf switch (1 leaf per uplink) in a different plane, this is called a multi-plane uplink connection (as shown in the figure on the right). Connecting each individual split NIC port to a separate leaf switch is defined as part of the rail connectivity architecture. Multi-plane architectures are proven to have an overall lower TCO , better fault-isolation, and plane redundancy. A failure in one given plane does not affect the other. From the source AI NIC port to the destination AI NIC port, the packets exclusively travel within the same plane through a set of unique leaf-spine switch fabric.

With multi-plane connections, packets of a given flow/QP can be sprayed across the 4 broken out uplink ports in a pure per-packet round-robin fashion. This is the 1st level of packet-spraying that is enforced at the NIC itself applicable in multi-plane topologies.

Now the next level of packet-spray either happens in the switching fabric (mode 1) or at the AI NIC itself (mode2).

MODE1: AT THE NETWORK SWITCHES (DLB BASED SPRAY)

Some vendor switches support DLB (dynamic load balancing) where the load-balancing is not based on packet-hash or ECMP but it's based on egress link utilization BW. The switch sends the subsequent packets on the link that is least utilized. This avoids congestion at least until the point that all egress links are loaded. In this mode, the EV based packet-spraying at the NIC needs to be disabled and this is done by setting the EV-MAX-PATHS to 1. DLB is enabled on all switches in fabric. All packets belonging to the same QP/flow will use the same UDP source port.



Single Plane Connectivity with DLB enabled fabric

Multi-Plane Connectivity with DLB enabled fabric

Figure 4: DLB-based spray

MODE2: AT THE SENDER AI-NIC (EV BASED SPRAY)

Now the sender AI NIC irrespective of being connected to the switch fabric in a single-plane or a multi-plane fashion can modify the RoCEv2 packet (which is an UDP packet) in a way to induce entropy into the packet for better hashing / load-balancing in the switch fabric. This is accomplished by either changing the UDP source-port field or IPv6 flow-label field or a combination of both fields in some cases. The value used in these fields is called the entropy-value (EV). The number of unique values of EV is user configurable. The number of EV values is typically 1:1 mapped to set of unique physical paths between the source AI NIC and the destination AI NIC in each plane. For example, if there are 128 unique paths between source AI NIC and destination, then the maximum number of EVs needs to be set to 128 as well, for the packet-spray logic to potentially use all the 128 physical paths. Because of source UDP port variation, these packets belonging to the same QP get hashed and load-balanced across several paths in the fabric resulting in maximizing the throughput.

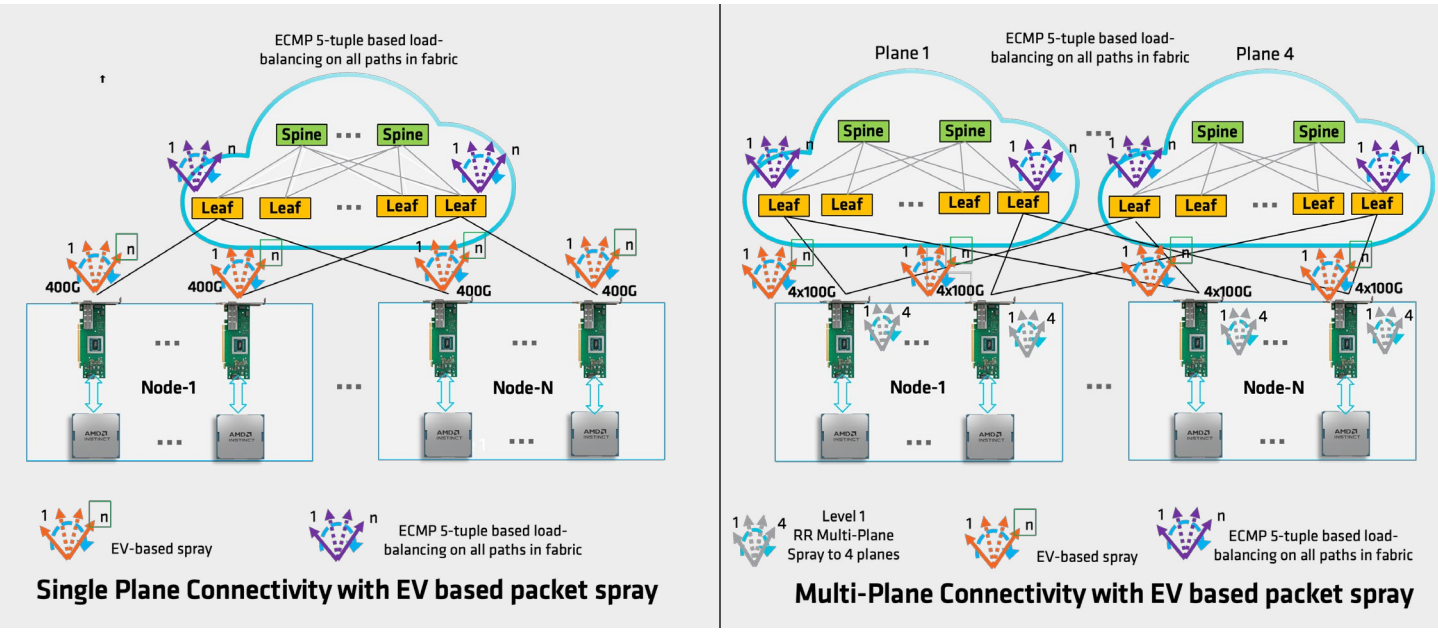


Figure 5: EV-based spray

Irrespective of the spray mode configured, whenever packets get sprayed by the NIC/switches they have the possibility to arrive at the destination AI NIC out-of-order. The destination AI NIC has the ability to deliver these OOO (out-of-order) packets by directly writing to the destination GPU memory, as the virtual-address is carried by the packet and the virtual-physical address mapping is known to the AI NIC. This added functionality on the AI NIC eliminates the need of compute intensive packet re-ordering processing on the AI NIC.

CHOICE OF PACKET SPRAYING MODE

The choice of the packet spray mode is left at the discretion of the customer. It could also depend on the nature of the deployed environment, RTT values, fabric-latency etc. The AI NIC offers flexible scale-out options by offering 2 different modes of spraying. The recommendation is to use mode 2, which is EV-based packet spray at the NIC and ECMP based LB hash on network switches, but we give the pros and cons of each approach here:

MODE1: NO EV-BASED PACKET SPRAY ON NIC, USE DLB ON SWITCHES

PROS:	CONS:
• The reaction-time to use and cut-over to under-utilized non-congested paths will be faster than the method described in mode2. In mode2, the reaction time is a function of RTT, where the destination AI NIC needs to receive an ECN marked packet and respond with a SACK followed by the sender AI NIC changing the subsequent packet-path by using a different EV. In DLB mode, the decision to switch paths will be more immediate and instantaneous and can be applied to very next packet in queue. Alternative ways Mode2 avoids this issue of path polarization is by using larger number of paths, thereby improving the spray across varied set of end to end paths.	• Not all switch vendors may support DLB (less common hashing scheme). • Troubleshooting can be challenging as the behavior of picking a link is dynamic • If ECN thresholds are not set correctly, ECN marking can start occurring on a packet even before DLB gets enforced. As a result of which sender AI NIC can slow down by reducing its congestion window pre-maturely not effectively utilizing un-used available bandwidth in the fabric.

MODE2: EV-BASED PACKET SPRAY ON NIC, ECMP BASED LB HASH ON SWITCHES

PROS:	CONS:
• No dependency on switch vendors to support DLB. • More control from the AI NIC side (server-admin) to control the load-balancing across fabric. Tuning control happens on NIC. • With adaptive, intelligent, and path-aware congestion control, bad congested paths (bad EV values) are not used for subsequent data packets and re-transmission.	• The reaction-time to switch to un-congested under-utilized links can be more compared to mode1, which is more instantaneous. In this mode, the sender AI NIC needs to learn about congested paths from SACK/NACK packets and then react. So there is increased latency in switching over to alternate paths.

NOTE: Users should NOT mix both modes, i.e. using DLB on switches (mode1) and using NIC-based spray as well (mode2) on the same topology/fabric. Using DLB along with NIC-based EV spray will make the source AI NIC lose all path-aware congestion state, since EVs will not map to physical paths anymore.

NETWORK EXPECTATIONS

In this section, we describe the expectations from the AI scale-out network.

PHYSICAL NETWORK DESIGN:

The AI-scale out network is typically a 2-tier (leaf, spine) or a 3-tier (leaf, spine, super-spine) CLOS network. With multi-plane support on the AI NIC, the AI scale-out network can be implemented with a 2-tier rail or a 3-tier rail architecture as well for better TCO and fault-isolation.

Overall expectation from switches in the network are as follows:

LOAD-BALANCING

Switches need to support one of the following load-balancing modes:
- **ECMP 5-tuple based hash** – this is the most typical supported by all switch vendors
- **DLB (dynamic load balancing)** – available only on select switch vendors

ECN MARKING

Switches in the fabric need the ability to mark the ECN bit in the packet at times of congestion. The ECN min, max threshold needs to be configured at a per-queue level.

For example, an ECN config of min threshold 20% and max threshold of 80% would lead to the following behavior:
- All packets will get ECN marked with queue-depth > 80%
- Packets will get ECN marked with probability (P) if depth > 20% and < 80%
- No ECN marking if queue-depth < 20%

PACKET-TRIMMING

Packet-trimming is defined as part of the UEC spec UETTRIM. Packet-trimming is NOT a hard requirement. It is optional. At times of heavy congestion, instead of tail dropping the packet, the packet can be trimmed, and only headers can be transmitted (payload dropped). These trimmed packets are classified into a higher priority queue to get priority treatment within the switch fabric. The UETTRIM spec handles trimmed packets separately than ECN marked packets when it comes to acknowledgements. When the destination AI NIC receives trimmed packets from the network, a NACK is generated, and congestion window is lowered as per the algorithm specification.

MTU

To maximize throughput in the scale-out network it is recommended that all scale-out network links are configured with jumbo MTU (9000 bytes).

NO VENDOR LOCK-IN

There is no vendor lock-in here. Any vendor switch that has one of the LB schemes and ECN support is a candidate for use within this AI scale-out network.

PACKET-TYPES

This section describes the different packet types that we are expected to see with MRC enabled AI NICs. Some of these are part of regular RDMA / RoCEv2 and some of these are added as part of MRC specs.

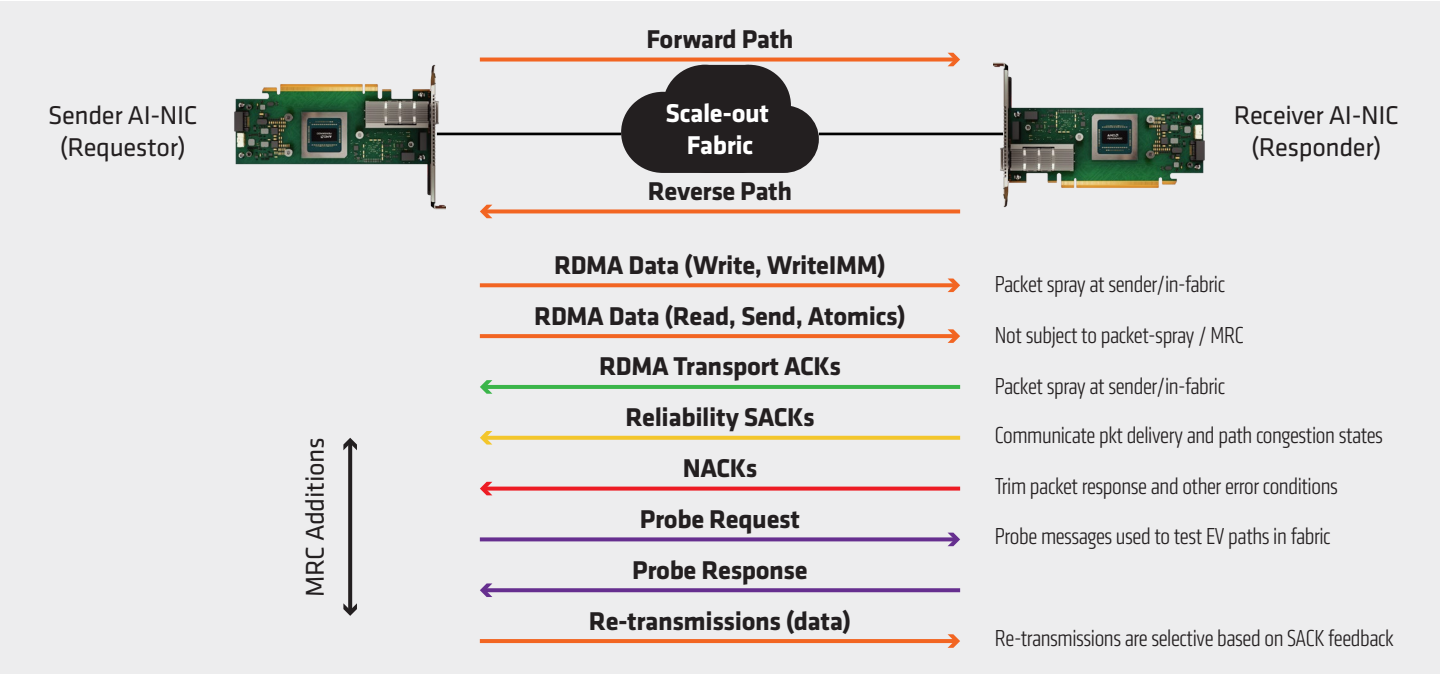


Figure 6: RDMA Packet Types

The above picture shows the common types of packets that are exchanged between the sender AI NIC and the receiver AI NIC across the scale-out switch fabric.

The direction of data transfer is referred to as the "forward" path, and the direction of acknowledgements is referred to as the "reverse" path.

The following packet types already exist as part of RDMA (RoCEv2):

- **RDMA Data (Write, WriteIMM)** – These type of data packets are subject to the MRC packet-spray by the AI NIC.
- **RDMA Data (Read, Send, Atomics)** – These type of data packets are NOT subject to MRC packet-spray by the AI NIC.
- **RDMA Transport ACKs** – These are cumulative ACKs sent by the receiver AI NIC. These packets are also subject to packet-spray.

Now, the following are the NEW packet types added specifically for MRC:

- **Reliability SACKs** – Sent by receiver AI NIC in response to a received ECN marked packet.
- **NACKs** – Sent by receiver AI NIC in response to a trimmed packet received from the network.
- **EV probe request** – Used by source AI NIC to perform EV path-health checks.
- **EV probe responses** – Sent by receiver AI NIC in response to EV probe requests.
- **Selective Re-Transmissions (data)** – This is sent by the source AI NIC on receiving the SACK packet from the receiver AI NIC.

In the next section, we describe the newly defined packet types in more detail, the key fields in each packet and what purpose they serve in the overall operation of MRC.

CONGESTION-CONTROL USING SACKs

In this section, we describe how congestion is detected in the network and how the congestion is reported, managed, and potentially avoided in MRC. This not only avoids congested paths, but also makes sure the available un-used bandwidth is always used first before attempting to slow down the sender by congestion window management and flow-control.

In addition to RDMA transport ACKs, which are cumulative ACKs, MRC defines and uses a new type of packet called the SACK (selective acknowledgement) packet to communicate data packet delivery state and network congestion state of paths.

SACKs are generated by the responder AI NIC on receiving valid, non-trimmed ECN-marked packets OR re-transmitted packets OR probes. SACKs may also be generated by the receiver upon receiving a specified number of data packets even when there is no congestion experienced.

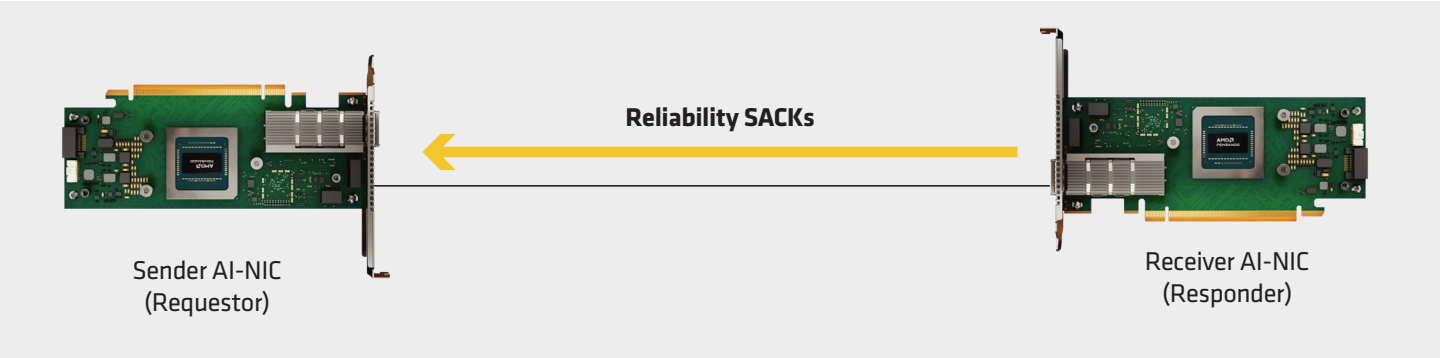


Figure 7: Congestion Control Using SACKs

The SACK packet contains the following key information:

PACKET FIELD	DESCRIPTION
"m" field set	Indicates that the original data packet resulting in this SACK was ECN marked. Value is set to "01" for congested ECN marked paths.
Entropy	This is the EV value that is copied from the corresponding data-packet that was ECN marked.
SACK bitmap	64-bit bitmap used to communicate to the source AI NIC about which packets with corresponding PSN have been received and which packets with corresponding PSN have not been received.

Table 1: SACK Packet Fields

The following actions happen on the source AI-NIC on reception of SACK:

- Update EV state: The source AI NIC will set the EV state to "skip" if m-bit has value of 0x1 and set EV state to "bad" if m-field has value set to 0x2.
- The source AI NIC derives the required selective re-transmission list from the received SACK bitmap.
- The source AI NIC will selectively re-transmit the data based on the sack-bitmap using a different paths/EV to avoid the packet drop.
- ECN marking and computed RTT used in decision regarding adjusting sender congestion window as described in section **Congestion Window Adjustments**.

CONGESTION-CONTROL USING NACKs

In this section, we describe the conditions that result in generation of NACKs (negative acknowledgements) from the receiver, NACK packet format and its key fields and how the AI NIC sender responds to NACKs to manage/control congestion.

NACKs are generated by the receiver AI NIC under the following conditions:

- Receiver AI NIC received trimmed data packets. Some switches in the network fabric support packet-trimming. Packet trimming occurs under extreme congestion where instead of tail-dropping the packet, the network switch sends trimmed packets (header only, no payload) over high-priority queues, so that trimmed packets can make it to the receiver, serving as a signal for a severe congestion.
- PSN errors or receiving packets out of PSN range.
- Packet drops at receiver AI NIC due to receiver resource issues like buffer/bitmap resource overflow.
- Other receiver related errors.

Figure 8: Congestion Control using NACKs

The NACK packet has the following key information:

PACKET FIELD	DESCRIPTION
"m" field set	Indicates that the trimmed data packet resulting in this NACK was ECN marked. Value is set to "01" for congested ECN marked paths.
nack_reason field	This is set to "TRIM" or "LAST_HOP_TRIM: if NACK is generated as a result of receiving a trimmed data packet.
entropy	This is the EV value that is copied from the corresponding trimmed data-packet.
nack_psn	This is the PSN of the original trimmed data packet that caused the NACK. This is used by sender AI NIC for selective fast-retransmission.

Table 2: NACK Packet Fields

The following actions are taken by the sender AI-NIC on receiving the NACK:

- From the "nack_psn" field, the sender AI NIC knows which packet needs to be re-transmitted.
- From the entropy field, the EV value is marked as "skip_once" if m-bit is 0x1 and marked as "bad" if m-bit is set to 0x2.
- A different EV value is used for re-transmitting the packet.

PROBES

In this section, we describe how probes are used in MRC by the sender AI NIC to test the health of the EV paths. The sender AI NIC sends probe packets for all the EVs that are in BAD state. An EV is moved to BAD state by the sender AI NIC when it determines that the given EV maps to a path causing packet drop. This information is derived from the SACK bitmap inside the SACK packet containing zeros for those PSNs, corresponding to the packets that were not received by the receiver AI NIC at the time of sending the SACK. The receiver AI NIC responds to EV probes by sending a Probe Response packet back to the sender AI NIC.

The Probe packet has the following key information:

PACKET FIELD	DESCRIPTION
entropy	Probe packets use this to test the health of this EV value.

Table 3: Probe Packet Fields

The state of EV is updated after Probe Responses are received for a given EV. This keeps the EV state current on the source AI NIC.

NETWORK SIGNAL BASED CONGESTION CONTROL

The MRC NSCC (network signal based congestion control) performs the following 2 key tasks:

- Load-balance across multiple-paths in the switch fabric by using adaptive EV packet spray.
- Flow-controls the sender AI NIC by adjusting the cwnd (congestion window).

The goal and intent is to use all available bandwidth in the switching fabric, by intelligently using all under-utilized paths and also adjust the congestion window upon seeing any congestion on any path, skipping that path once, thereby removing congestion on that path. This makes sure that at a per-QP level the throughput is maximized.

The NSCC algorithm uses feedback from all of the following to achieve a highly efficient path-aware congestion control mechanism:

- ECN marking information (from SACK) - detect congested paths
- RTT delay calculations
- Trimmed packet feedback (from NACK) - detect super-congested paths
- EV probes – actively test health of paths and expose healthy paths to data packets

EV STATES

The following table describes the EV states:

EV STATE	DESCRIPTION
Good	Default starting state. Only EVs in this state are picked by the round-robin EV selection logic.
Skip	EVs are moved to "Skip" state when the sender AI NIC receives either a SACK with M flag set to skip_once (ECN marked) OR on receiving a NACK received with TRIM
Denied	This is administratively marking a path as Bad.
Assumed Bad	EVs are moved to "Bad" state when the sender AI NIC receives a SACK with M flag set to "always_skip".

Table 4: EV States

CONGESTION WINDOW ADJUSTMENTS

The sender AI NIC uses feedback both from ECN marking of a given EV value and also the RTT measurements it makes for a given EV value.

The actions taken by sender AI NIC to adjust and flow-control the sender is based on combination of ECN marking and RTT calculations as described in the following table:

ECN	PACKET DELAY	INFERENCE	CC ACTION (CWND MANIPULATIONS)
0	RTT < target_queue_delay	NW is NOT congested	proportional_increase ()
0	RTT > target_queue_delay	NW was congested but congestion has reduced now	fair_increase()
1	RTT < target_queue_delay	NW becoming congested	No increase ()/ decrease ()
1	RTT > target_queue_delay	NW congested	Multiplicative_decrease ()

Table 5: Congestion Window Actions

The target_queue_delay is either set by the administrator for a given queue-pair or the algorithm can use base_rtt (configurable or determined based on unloaded network) on a per QP basis.

QoS CONSIDERATIONS

The MRC specification does NOT mandate any specific QoS settings on the NIC or the switch. As a recommendation, it is advisable to disable PFC end-to-end on all devices in the fabric, on both NICs and switches. Having PFC on will have an adverse effect on RTT measurement (and reaction to higher RTT in PFC enabled network), and end-to-end throughout and performance as it slows down sender AI NIC pre-maturely and more aggressively to a point where the bandwidth available to fabric is not utilized fully. The advantages of adaptive packet-spray will be nullified by PFC.

The following table summarizes one of the possible QoS configurations that can be used on a 4-queue system.

The AI NIC supports 4 queues. Example here, give separate queues for RDMA data, RDMA re-transmissions / trimmed data packets, SACK/NACK/transport ACK, and control packets:

QUEUE#	CLASSIFICATION	PFC	SCHEDULING	PURPOSE
q3	DSCP (26)	disabled	SPQ	xCCL CTS
q2	DSCP (48)	disabled	DWRR	SACK, NACK, transport ACK
q1	DSCP (33)	disabled	DWRR	RDMA re-transmissions, trimmed data packets
q0	DSCP (24)	disabled	DWRR	RDMA data

Table 6: Example AI NIC Queues

PERFORMANCE RESULTS AND BENEFITS

This section illustrates the key performance benefits of enabling the MRC features on AI NIC. Significant throughput benefits compared to standard RoCEv2, faster job completion times when training LLMs (large-language-models) and improved model convergence in large clusters with congestion and faster loss recovery times. Adaptive packet-spray, in-order delivery, selective re-transmissions, fast error recovery, probing, path-aware congestion-control, all the MRC features collectively contribute to this improved performance. The findings are summarized here.

UP TO 25% RCCL THROUGHPUT IMPROVEMENT WITH MRC

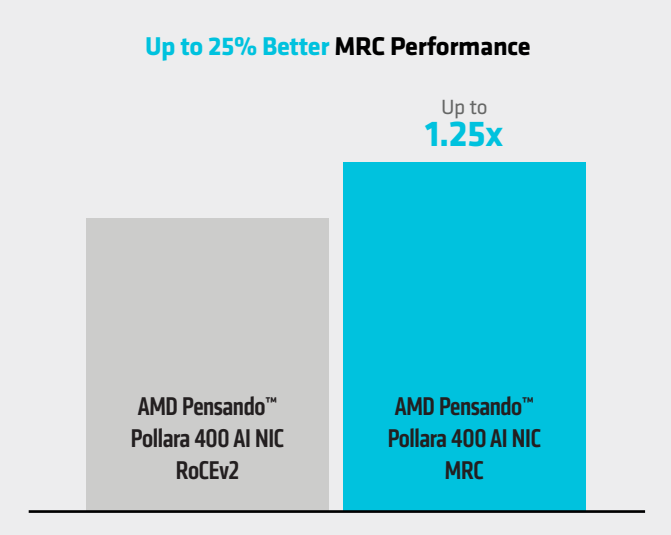


Figure 9: RCCL Throughput Improvement

Overall RCCL throughput was up to 25% more with MRC features than compared to standard RoCEv2 on the Pensando Pollara 400 AI NIC.

Check PEN-016 for more details.

UP TO 50% FASTER MODEL CONVERGENCE IN NETWORKS WITH 0.1% TRAFFIC LOSS

The following bar graph shows 50% faster loss recovery with MRC features enabled as compared to standard RoCEv2 turned on when packet loss was introduced at a rate of 0.1% as a measure to simulate 0.1% drops due to congestion in the network.

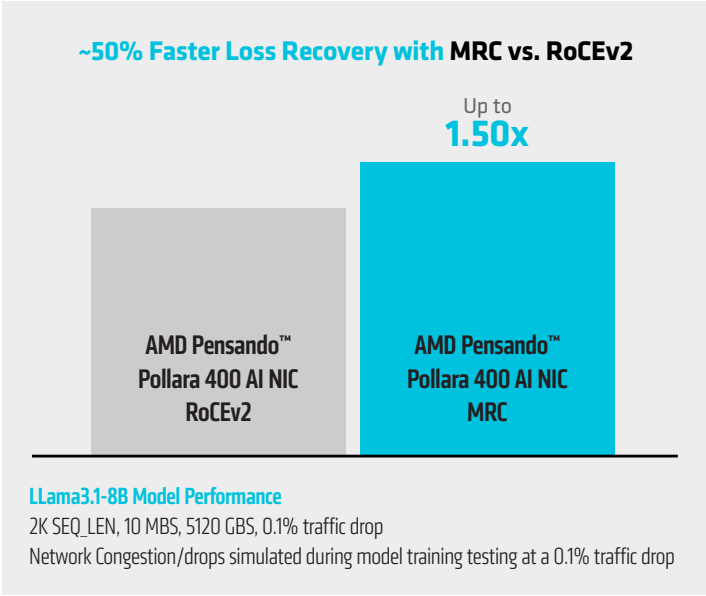


Figure 10: Model Convergence with MRC

See PEN-019: for more details on this test.

Collectively what we can infer from this is that MRC significantly improved the overall throughput and maximized the bandwidth utilization in the network. It does significantly better in scenarios of congestion and loss recovery and real-time model convergence is also observed to be faster.

CONSIDERATIONS

The following are the considerations to be noted for MRC:

- Only Write and WriteIMM operations are performance enhanced by MRC features. The enhancements do not apply to Read, Send and Atomics.
- Number of inflight WriteIMM operations is limited by responder WriteIMM tracking resources.
- Does not support RNR NACK transport semantics.

CONFIGURATION BEST PRACTICES

- Disable PFC end-to-end on all devices: on all AI NICs and all switches in fabric
- Configure jumbo MTU on all network interfaces in fabric.

SUPPORTED SCALE

The following table captures the scale of different parameters in MRC:

Max QP (single-plane profile)	2048
Max QP (multi-plane profile)	1024
Max EV per QP (single-plane profile)	256
Max EV per QP (multi-plane profile)	512 (128 per plane)

Table 7: RDMA Scale Parameters

CONFIGURATION ON AI-NIC

This section provides a snippet for bare minimum configuration needed to enable the MRC feature. This needs to be done after a successful installation procedure of AI NIC. For customers already using AI NIC for RoCEv2, transitioning to MRC is a simple and smooth process.

The first step is to enable the pf1_puec profile which enables the MRC mode.

ENABLE THE MRC PROFILE

The MRC feature set is supported under a new profile called the “pf1_puec” profile for single-plane connectivity or under profile “pf1_multi_plane_puec” for multi-plane connectivity.

For single-plane connectivity use the following CLI:

```
nicctl update card profile -p pf1_puec -i  <image_file>
```

For multi-plane connectivity use the following CLI:

```
nicctl update card profile -p pf1_multi_plane_puec -i <image-path>
```

Under this profile, 4x100G uplink ports will be available to connect to 4 different planes.

QOS CONFIGURATION

As described in the QoS considerations section, a typical QoS configuration for MRC could use all 4 supported queues on AI NIC. Separate queues for RDMA data-traffic, RDMA re-transmissions/trimmed data packets, SACK/NACK/transport-ACK, control-traffic (xCCL CTS) could be used. DSCP can be used as the classification type. PFC is recommended to be disabled on all queues. SPQ to be used for xCCL CTS traffic and all other queues for data, re-transmissions and control-traffic can be DWRR queues.

The following are example QoS commands that can be used:

```
sudo nicctl update qos --classification-type dscp
sudo nicctl update qos dscp-to-priority --dscp 24 --priority 0
sudo nicctl update qos dscp-to-purpose --dscp 24 --purpose data
sudo nicctl update qos dscp-to-priority --dscp 26 --priority 3
sudo nicctl update qos dscp-to-purpose --dscp 26 --purpose xccl-
cts sudo nicctl update qos dscp-to-priority --dscp 48 --priority 2
sudo nicctl update qos dscp-to-purpose --dscp 48 --purpose rdma-ack
sudo nicctl update qos dscp-to-priority --dscp 33 --priority 1
sudo nicctl update qos dscp-to-purpose --dscp 33 --purpose rexmit-
data
```

The above commands classify packets with DSCP 24,33,48 and 26 to queues 0,1,2 and 3 respectively. RDMA data traffic is marked with DSCP 24, RDMA re-transmitted data is marked with DSCP 33, RDMA-ACK is marked with DSCP 48 and CTS is marked with DSCP 26 using the respective purpose commands.

The implementation implicitly classifies SACK, NACK and trimmed data packets on a separate queue.

QOS CONFIGURATION ON SWITCH

A matching switch QoS configuration needs to be done on the switch with appropriate config for ECN marking. Refer to vendor switch QoS configuration guides for this.

MODE1 CONFIGURATION: NO PACKET SPRAY ON NIC, DLB ENABLED ON SWITCHES

In this mode, there is no EV-based packet spray on the AI NIC. The switches in the fabric are enabled with DLB configuration. EV-based packet spray is disabled using the option IONIC_RCQ_NUM_PATHS=1 in the application. DLB MUST be enabled on all switches in fabric for packet-spray within the fabric. The switches in fabric MUST also be configured with appropriate QoS config including configuration required for ECN marking to mark packets during congestion.

Sample config for DLB on Cisco switches:

```
hardware profile dlb
 mode per-packet
 dlb-interface all
```

The “dlb-interface all” command requires a switch reload to take effect.

OOO (out-of-order) delivery is always enabled implicitly at the receiver AI NIC by default, so that all arriving out-of-order frames are placed directly in destination DPU memory.

MODE2 CONFIGURATION: EV-BASED PACKET SPRAY ON NIC, ECMP HASH-BASED LB ON SWITCHES

In this mode, the EV-based packet spray is enabled on the sender AI NIC. The maximum number of paths is specified by the application using the attribute IONIC_RCQ_NUM_PATHS=128 for example.

The switching fabric is configured with ECMP 5-tuple hash-based load-balancing scheme which is the default in most vendor switches. Matching QoS config and ECN config on all switches in fabric required. OOO (out-of-order) delivery is enabled implicitly on the receiver AI NIC and OOO packets are directly delivered to destination GPU memory on reception.

NOTE:

- *Multi-plane configuration can be enabled in both modes “mode1” and “mode2”.*
- *The default congestion control and congestion window tuning parameters should work for most of the deployments. If not, it is possible to tune these parameters and separate CLI commands to tune these parameters can be provided as well.*

CLAIMS

PEN-016

Testing conducted by AMD Performance Labs as of [28th April 2025] on the [AMD Pensando™ Pollara 400 AI NIC], on a production system comprising of: 2 Nodes of 8xMI300X AMD GPUs (16 GPUs): Broadcom Tomahawk-4 based leaf switch (64x400G) from MICAS network; CLOS Topology; AMD Pensando Pollara AI NIC – 16 NICs; CPU Model in each of the 2 nodes - Dual socket 5th gen Intel® Xeon® 8568 - 48 core CPU with PCIe® Gen-5 BIOS version 1.3.6 ; Mitigation - Off (default)System profile setting - Performance (default) SMT-enabled (default); Operating System Ubuntu 22.04.5 LTS, Kernel 5.15.0-139-generic. Following operation were measured: Allreduce Average 25% for All-Reduce operations with 4QP and using MRC vs the RoCEv2 for multiple different message size samples (512MB, 1GB, 2GB, 4GB, 8GB, 16GB). The results are based on the average at least 8 test runs.

PEN-019

Testing conducted by AMD Performance Labs as of [15 September 2025] on the AMD Pensando Pollara AI NIC, on a test system comprising of SMC-300X server for GPU-GPU communication: 2x AMD Pensando Pollara AI NIC, 2P AMD EPYC 9454 48-Core -2P Processor, 8x AMD Instinct MI300X GPU, Ubuntu 22.04.5 LTS, kernel 5.15.0-139-generic, ROCm 6.4.1.0-69b59e5. Testing running Llama-3.1-8B, Model Configuration: SEQ_LEN=2048, TP=1, PP=1, CP=1,FP8=1, MBS=10, GBS = 5120. Iteration = 2, No. of paths/QP : 128. Results may vary based on factors including but not limited to system configuration and software settings.

REFERENCES

1. https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/lossless.pdf

2. https://yibozhu.com/doc/ecndelay-conext16.pdf