Priority-based RDMA Transmission Control Method for Low-CPU CPS Data Collection under Mixed-Deadline Requirements

1st Ryouhei Tsugami
Access Network Service Systems Laboratories NTT, Inc.
Tokyo, Japan
ryouhei.tsugami@ntt.com

2nd Tatsuya Fukui
Access Network Service Systems Laboratories NTT, Inc.
Tokyo, Japan
tatsuya.fukui@ntt.com

3nd Satoshi Narikawa
Access Network Service Systems Laboratories NTT, Inc.
Tokyo, Japan
satoshi.narikawa@ntt.com

***Abstract*—Cyber-physical systems (CPSs) collect heterogeneous sensor streams over wide area networks (WANs), combining diverse latency targets—from sub-100 ms alerts to 1 s bulk telemetry—with high aggregate volumes. Beyond bandwidth limits, the collection server's CPU overhead becomes a material bottleneck, motivating a latency-aware, low-CPU collection architecture. Remote Direct Memory Access (RDMA) lowers CPU cost, yet it degrades in WANs where data center mechanisms such as Priority Flow Control (PFC) are impractical. Our prior RDMA transmission control method (RDMA-TCM) enables WAN RDMA without conventional congestion control but ignores traffic priority. We present Priority-based RDMA-TCM (P-RDMA-TCM), a priority-aware, per-endpoint scheduler that enqueues send requests by priority and serves higher priority first, protecting latency-critical traffic while sustaining throughput for delay-tolerant flows. We implemented P-RDMA-TCM on real hardware and evaluated it under mixed workloads with budgets of $\leq$ 100 ms (critical) and $\leq$ 1 s (tolerant). Up to 66% offered load relative to system capacity, P-RDMA-TCM keeps both classes within deadline and cuts the critical flow end-to-end P95 latency by >50% versus RDMA-TCM, without sacrificing tolerant-flow throughput. These results suggest that, even without PFC, WAN RDMA can support low-CPU-overhead CPS data collection while meeting diverse latency requirements.**

***Index Terms*—RDMA, RoCE, CPS, Data collection, Priority control, CPU load reduction**

I. Introduction

Modern CPS integrate large numbers of heterogeneous sensors—e.g., cameras and lidars—and stream latency-sensitive data over WANs. In the area-management use case of the Innovative Optical and Wireless Network (IOWN), up to 100,000 cameras are envisioned within a 1 km^2 area (Fig. 1), increasing both per-device capability and overall sensor density. These streams exhibit diverse latency targets, from sub-100 ms alerts for intrusion or wildlife detection to 1 s delay-tolerant telemetry.

At this density, the aggregate ingress at the collector becomes substantial. As a result, not only do WAN capacity constraints matter, but the CPU cost of receive-side processing also emerges as a bottleneck. Meeting these requirements across geographically distributed endpoints calls for a latency-aware, low-CPU-overhead collection architecture.

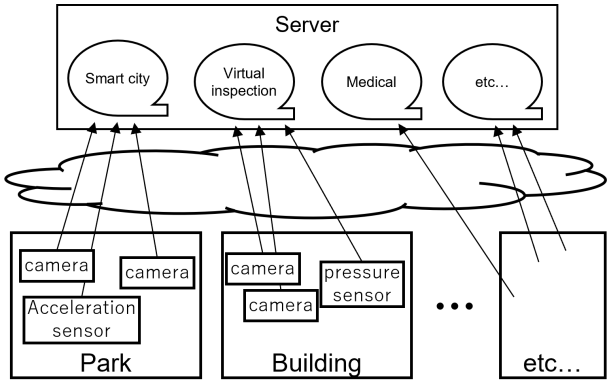

Fig. 1: Overview of Cyber-Physical Systems

RDMA is attractive because it reduces CPU overhead by bypassing the kernel and avoiding redundant copies. To further improve responsiveness, network devices can apply priority control—e.g., Strict Priority Queuing (SPQ)—to preferentially handle latency-critical traffic. However, RDMA typically depends on PFC or Explicit Congestion Notification (ECN) to provide lossless or near-lossless behavior. In WANs these mechanisms are often unavailable or impractical, so relying only on network-level priority can lead to congestion losses and severe performance degradation.

To enable efficient RDMA communication over WANs without PFC/ECN, our prior work proposed the RDMA-TCM [2], in which a centralized controller coordinates send timing to achieve low CPU load and high network efficiency. A limitation of RDMA-TCM is that it serves transmission requests strictly in arrival order (FIFO), preventing latency-aware collection; urgent data can be blocked behind low-priority traffic.

This paper proposes the P-RDMA-TCM. P-RDMA-TCM assigns transmission priorities according to application latency requirements and schedules send requests accordingly,

prioritizing latency-sensitive data used for real-time analysis over latency-tolerant data such as log collection. Our goal is to satisfy heterogeneous latency targets while preserving the low CPU overhead of RDMA in WAN environments without PFC/ECN.

We evaluate P-RDMA-TCM on real hardware in a WAN-emulated setting. Our evaluation focuses on: (1) packet loss and CPU utilization rate, (2) end-to-end latency from data generation to transmission completion, and (3) the proportion of data that meets its latency target The contributions are:

1) Design: We present P-RDMA-TCM, a priority-aware RDMA transmission framework for latency-aware CPS data collection over WANs.
2) Effectiveness: We show that P-RDMA-TCM satisfies diverse latency requirements, handling both latency-sensitive and latency-tolerant traffic.
3) Efficiency: We demonstrate stable throughput and low CPU overhead in WAN environments without relying on PFC or ECN.

II. Related work

A. Ingesting high volumes from many devices in CPS/WAN

The Data Plane Development Kit (DPDK) [3] delivers high-throughput TCP/UDP I/O via busy polling [4], but dedicating cores to polling increases receive-side CPU utilization and becomes a bottleneck under extreme fan-in. Similarly, kernel-optimized alternatives such as io_uring x and AF_XDP x have been investigated. While io_uring can block on eventfd/io_uring_enter() and AF_XDP can rely on interrupt- or NAPI-driven operation at low load, at high ingress both converge to tight completion/submission (CQ/SQ) or ring polling, driving receive-side CPU utilization toward busy-polling levels.

B. RDMA for large-fan-in reception inside data centers

Within data center (DC) networks, RDMA has been widely applied to lower communication overhead and improve system and application performance—for example, in streaming frameworks (e.g., Apache Storm [5]), in containerized deployments through placement and network optimization [7], and in blockchain transaction processing [8]. RDMA-aware scheduling further couples big data job placement with a high-performance transport, increasing aggregate throughput [11]. RDMA has also been used to offload metric collection in monitoring systems [6].

C. RDMA beyond the DC for WAN operation

Extending RDMA over Converged Ethernet (RoCE) across WAN segments is complicated by PFC-induced deadlocks under ring or tree topologies [10] [11] [12]. DCQCN [13] mitigates congestion via ECN but retains PFC and offers no deadlock guarantees for WAN topologies. To avoid PFC on lossy networks, iWARP [14] (RDMA over TCP) and IRN [15] (a PFC-free redesign of RoCE) have been proposed; however, shifting reliability and ordering into the NIC increases NIC-side processing costs and complexity, which may limit sustaining line-rate processing under extreme fan-in.

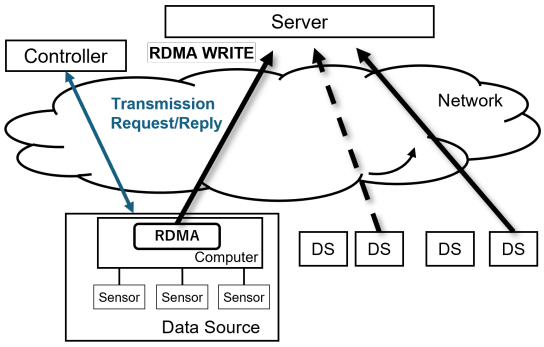

Fig. 2: Overview of RDMA-TCM

D. Summary

A. indicates that DPDK and similar technologies are insufficient at CPS/WAN scale and motivates considering RDMA; B. demonstrates RDMA ' s DC-side benefits; and C. outlines the challenges of making RDMA WAN-ready, motivating a PFC-free, WAN-oriented design focus.

III. Proposed Method

A. RDMA-TCM

As shown in Fig. 2, the proposed RDMA-TCM [2] consists of a Data Source (DS) that senses data, a data collection server that collects sensing data from the DS, and a controller that manages the currently available bandwidth and responds to a request from the DS for transmission. The controller manages the bandwidth available in the entire system, refers to the bandwidth information managed by itself, and returns a response to the DS after reducing the bandwidth from the bandwidth information managed if there is any remaining bandwidth that can accept a transmission request.

Data to be transmitted by RDMA-TCM is assumed to be transmitted when moving object detection, abnormality detection, or a certain amount of log data is accumulated.

By this operation, data transmission by RoCE can be realized in a finite bandwidth, and as a result, CPU resources required for data reception can be reduced.

B. Priority-based RDMA-TCM

Fig. 3 shows the priority control function of the RDMA-TCM controller. In this function, data requiring data transmission at a preset low latency is allocated to a high-priority allocation standby queue, and data requiring no problem even if the latency increases, such as simple collection of log data, is allocated to a low-priority allocation standby queue. By this priority allocation control, data transmission from a specific data source can be started with priority, and it is possible to reduce the latency of latency-sensitive data such as analysis without increasing the latency of the whole system.

In P-RDMA-TCM, transmission priority is controlled at the flow level rather than the packet level. Once a transmission permission is granted, the data source sends the entire frame using RDMA WRITE without interruption. This design follows the nature of RDMA ' s zero-copy mechanism, where

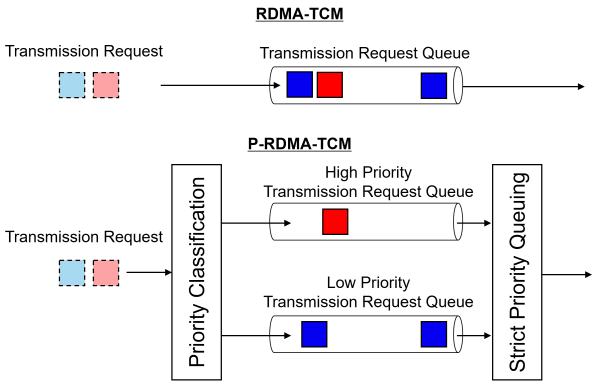

Fig. 3: Priority control function in controller

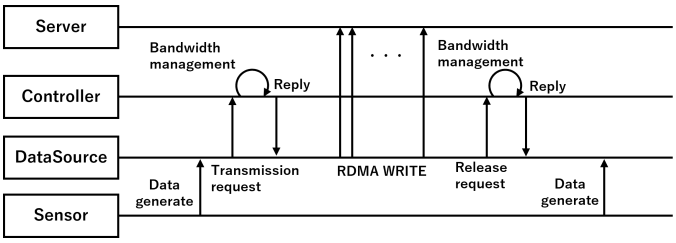

Fig. 4: Data transmission flow

mid-transfer preemption would break memory consistency and increase CPU overhead. Therefore, flow-level non-preemptive scheduling is adopted to maintain RDMA ' s low-latency and low-overhead characteristics.

Fig. 4 shows that the transmission procedure follows a controller-mediated RDMA WRITE sequence, in which each data source requests transmission permission, receives bandwidth allocation, transmits data, and releases resources upon completion.

IV. Experiment

A. Experimental environment

In this experiment, as shown in Fig. 5, 1 data collection server and controller and 12 DSs are used. RoCE communication is realized by PCIe connection of ConnectX of Mellanox to each device. In addition, to create congestion in the network, a 20G shaper was configured on the port to which the server was connected.

In this experiment, event-driven data transmissions such as motion or anomaly detection and log accumulation were emulated using a Poisson process. Among 12 data sources (DSs), one generated high-priority traffic and the remaining eleven low-priority traffic. High-priority events (e.g., dangerous object detection) transmitted a single 4K uncompressed frame ($\simeq$ 25 MB) per event, while low-priority ones transmitted 15 aggregated frames to emulate tolerant workloads. Each request was classified into high and low priority queues based on DS ID, and the controller scheduled transmissions using Priority Queuing (PQ) to serve high-priority requests

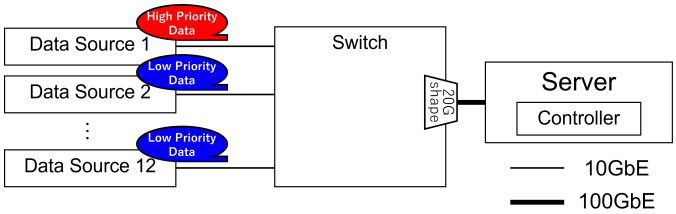

Fig. 5: Experimental system

first. RoCE communication was implemented using librdmacm [16].

As for the amount of data in the experiment, the value of λ (Average number of events per second) in the Poisson distribution of high-priority DS and low-priority DS was changed so that the occurrence rate of high-priority data was about 10%. Also, the amount of data generated by all DSs for the available bandwidth of 20Gbps is defined as the offered load, and the change in the amount of data transmitted within the latency requirement is evaluated when the offered load is increased.

In this experiment, each DS requests 10Gbps bandwidth to the controller. The controller returns a response to the DS with a reduction of 10Gbps if the total amount managed at the time of the request is greater than or equal to 10Gbps. If it is less than 10Gbps, the request is queued. Upon receiving the response, the DS transmits the data to the server by RDMA WRITE, and requests the controller to release the resource after the data transmission is completed.

B. Evaluation item and method

In this paper, we evaluate the number of packet losses, the delay, and the amount of data completed within the delay requirements when data is transmitted from the DS to the server. The measurement methods for each endpoint are described below.

1) Packet Loss: The number of packet losses is determined by using the counter value of the port to which the server of the switch used in this experimental system is connected. Specifically, the "TX discard packets" value of the L2 switch to which the server is connected was used. The counter value is acquired before and after the test and evaluated as the number of packet losses by taking the difference.

2) Latency: In the latency measurement in this experiment, the latency was measured as a time difference from the start of transmission of frame data to the completion of transmission at the DS side. In this case, if a retransmission occurs due to a packet collision or the like, the time until the retransmission is completed becomes a latency. The reason for measuring the latency on the sending side is that if the latency is measured by the time from the start of data transmission on the sending side to the completion of data reception on the receiving side, This is because the latency error increases depending on the accuracy of time synchronization.

3) Accepted Load: Accepted Load denotes the fraction of generated bytes delivered within each latency target: 100 ms

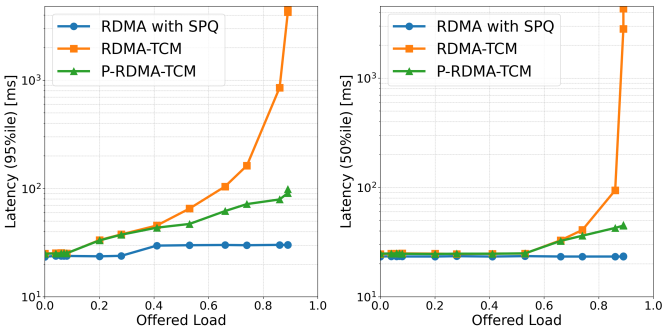

Fig. 6: Latency of high-priority data
(95 percentile value, left; Median, right)

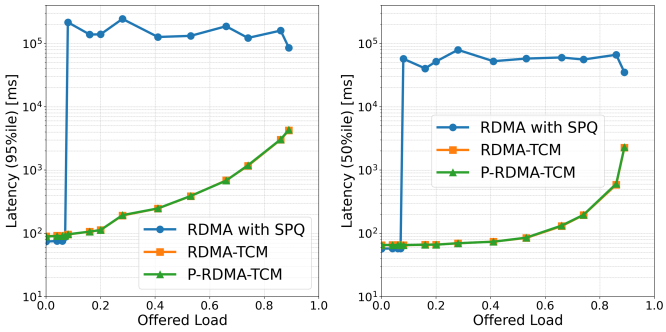

Fig. 7: Latency of low-priority data
(95 percentile value, left; Median, right)

[17] for urgent and 1 s for tolerant data. It quantitatively represents the ratio of successfully collected data satisfying latency requirements beyond simple throughput.

V. Evaluation

In this experiment, we evaluate the number of packet losses, the latency from data generation to transmission completion, and the percentage of data that can be transmitted within a specified time by using three methods, RDMA with SPQ, RDMA-TCM, and P-RDMA-TCM.

A. Packet Loss and CPU Utilization Rate

Table I shows the number of packet-loss events under a light offered load (Offered Load = 0.09), well below the link capacity. As shown in the table, packet loss occurs in RDMA with SPQ. In WANs without mechanisms such as PFC or ECN, traffic classes that lack priority control contend at bottlenecks and are dropped, leading to loss. By contrast, RDMA-TCM exhibits no packet loss: by controlling transmission timing so that bursts do not exceed the available bandwidth, it avoids contention and thereby prevents packet loss. Turning to CPU usage, both methods show consistently low utilization regardless of the Offered-Load level, because RDMA bypasses the kernel and eliminates OS-level memory copies.

On the other hand, in the RDMA-TCM which performs priority control with the RDMA-TCM, packet loss has never occurred. In RDMA-TCM, by controlling the transmission timing so that burst traffic exceeding the available bandwidth does not occur, traffic collision can be avoided and packet loss can be prevented.

B. Latency

For the three methods—RDMA with SPQ, RDMA-TCM, and P-RDMA-TCM—Fig. 6 shows the median and 95th-percentile latency values of the high-priority DS, and Fig. 7 shows those of the low-priority DS. In these graphs, the horizontal axis represents the Offered Load, and the vertical axis represents the latency from data generation to transmission completion.

Focusing on the high-priority DS, the latency ranks as RDMA with SPQ $<$ P-RDMA-TCM $<$ RDMA-TCM in both median and 95th-percentile values. This is because RDMA with SPQ can transmit data without additional overhead by using the switch ' s built-in priority control. P-RDMA-TCM, on the other hand, incurs an overhead for allocating resources during transmission. At an Offered Load of about 0.9, the latency is approximately 45 ms, about 20 ms higher than RDMA with SPQ, but it is still significantly lower than that of RDMA-TCM. This improvement occurs because the controller processes high-priority allocations before low-priority requests, reducing queueing time in the controller.

For the low-priority DS, both the median and 95th-percentile latency values are nearly the same for RDMA-TCM and P-RDMA-TCM. This is because the high-priority DS generates data less frequently than the low-priority DS, so the impact of prioritizing high-priority traffic is limited. In contrast, RDMA with SPQ exhibits large latency due to the strict-priority control of SPQ and resulting packet loss.

C. Accepted Load

Fig.8 shows the accepted load of high-priority data and low-priority data. If we focus on the high-priority Accepted Load, both RDMA with SPQ and the proposed P-RDMA-TCM achieve high values. In contrast, RDMA-TCM shows a sharp drop in the range where the Offered Load is high. In other words, RDMA with SPQ and P-RDMA-TCM can achieve data collection with a high deadline satisfaction rate regardless of the amount of data generated, due to their function of preferentially transmitting urgent data. However, in RDMA-TCM, data transmission that satisfies the latency requirement becomes difficult as the amount of generated data increases, because data is transmitted in the order of generation.

Focusing on the low-priority Accepted Load, RDMA-TCM and the proposed P-RDMA-TCM remain high, while RDMA with SPQ is low. In RDMA with SPQ, priority is given to urgent data, which limits the bandwidth available for low-priority data. Frequent packet collisions and retransmissions exceed the retransmission limit of `librdmacm` [16], causing the operation to fail. On the other hand, in RDMA-TCM and P-RDMA-TCM, since transmission timing is controlled by the controller, packet collisions do not occur, and data can be transmitted without operational failures. As a result,

TABLE I: Number of Packet loss and CPU utilization rate

	Number of packet loss (Offered Load = 0.09)	CPU utilization rate (Offered Load = 0.09)	CPU utilization rate (Offered Load = 0.66)
RDMA with SPQ	1359405504	0.029	0.049
RDMA-TCM	0	0.027	0.222
P-RDMA-TCM	0	0.027	0.234

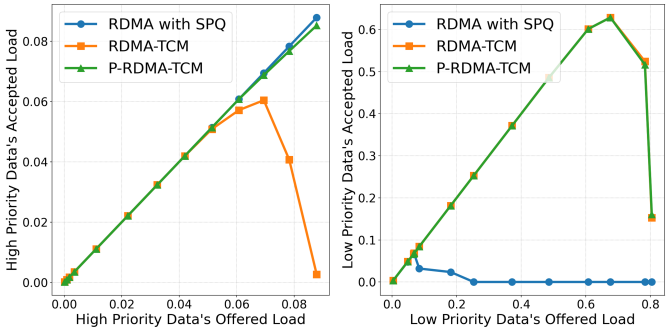

Fig. 8: Accepted Load of high-priority data(left) and low-priority data(right)

the Accepted Load is higher than that of RDMA with SPQ. However, when the Offered Load is about 0.66 and the amount of generated data exceeds approximately 13.2 Gbps in this experimental system, it becomes difficult to satisfy the 1-second latency requirement. This is because the overhead and data volume of RDMA-TCM increase, causing a large number of transmission requests to arrive at the controller and lengthening the response time.

These results show that the P-RDMA-TCM can be used to achieve data collection that satisfies the latency requirement even for low-priority data when the Offered Load is up to approximately 0.66.

VI. Conclusion

In this paper, we propose P-RDMA-TCM, which applies priority control to transmission timing control in a controller, as a data collection method considering various latency requirements in WAN environments. By using this scheme, the end-to-end P95 latency can be reduced by about 50% compared with the conventional RDMA-TCM, and the degree of achievement of the latency requirement can be increased regardless of the priority if the load is up to 66% relative to the system capacity. Thus, it is possible to realize efficient data collection with low CPU overhead even in an environment where urgent data such as detection of dangerous animals and abnormality detection and data tolerant to latency such as a log monitor are mixed.

References

[1] Supplement to InfiniBand Architecture Specification Volume 1 Release 1.2.1 Annex A17: RoCEv2, InfiniBand Trade Association, 2014.

[2] T. Ryouhei et al. (in press), "RDMA Transmission Control Method: Using Network Resource Allocation For Wide-Area Data Collection," in Proc. Consumer Communications & Networking Conference, Jan. 2024.

[3] Intel, "Data Plane Development Kit," http://dpdk.org, 2017.

[4] D. Géhberger et al., "Performance evaluation of low latency communication alternatives in a containerized cloud environment," in 2018 IEEE 11th International Conference on Cloud Computing, Jul. 2018, pp. 9-16.

[5] Zhang, Z., Liu, Z., Jiang, Q. et al. RDMA-Based Apache Storm for High-Performance Stream Data Processing. Int J Parallel Prog 49, 671–684 (2021). https://doi.org/10.1007/s10766-021-00696-0

[6] Z. Wang et al., "Zero overhead monitoring for cloud-native infrastructure using RDMA," in 2022 USENIX Annual Technical Conference, Jul. 2022, pp. 639–654.

[7] U. Abbasi, E. H. Bourhim, M. Dieye and H. Elbiaze, "A Performance Comparison of Container Networking Alternatives," in IEEE Network, vol. 33, no. 4, pp. 178-185, July/August 2019, doi: 10.1109/MNET.2019.1800141.

[8] Huang, Y., et., (2019). BoR: Toward High-Performance Permissioned Blockchain in RDMA-enabled Network. IEEE Transactions on Services Computing. PP. 1-1. 10.1109/TSC.2019.2948009.

[9] S. Wu, H. Chen, Y. Wang and H. Jin, "Argus: Efficient Job Scheduling in RDMA-assisted Big Data Processing," 2021 IEEE International Parallel and Distributed Processing Symposium (IPDPS), Portland, OR, USA, 2021, pp. 827-836, doi: 10.1109/IPDPS49936.2021.00092.

[10] M. Alizadeh et al., "Deconstructing datacenter packet transport," in Proc. 11th ACM Workshop on Hot Topics in Networks, Oct. 2012, pp. 133-138.

[11] A. Shpiner et al., "Unlocking credit loop deadlocks," in Proc. 11th ACM Workshop on Hot Topics in Networks, Nov. 2016, pp. 85-91.

[12] A. Dixit et al., "On the impact of packet spraying in data center networks," in Proc. IEEE INFOCOM, Apr. 2013, pp. 2130–2138.

[13] Y. Zhu et al., "Congestion control for large-scale RDMA deployments," ACM SIGCOMM Computer Communication Review, vol. 45, no. 4, pp. 523-536, Oct. 2015.

[14] R. Recio et al., A remote direct memory access protocol specification, document RFC5040, 2007. [Online]. Available: https://www.rfc-editor.org/rfc/rfc5040

[15] R. Mittal et al., " Revisiting network support for RDMA, " in Proc. ACM SIGCOMM, Aug. 2018, pp. 313–326.

[16] Linux RDMA community, rdmacm, GitHub, https://github.com/linux-rdma/rdma-core

[17] Cyber-Physical System Use Case Release-1, "IOWN Global Forum" 2021.