Istituto Nazionale di Fisica Nucleare
Sezione di Padova

Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Legnaro

FEROCE

Front-End Rdma Over Converged Ethernet

Area di ricerca: Rivelatori, Elettronica, Calcolo

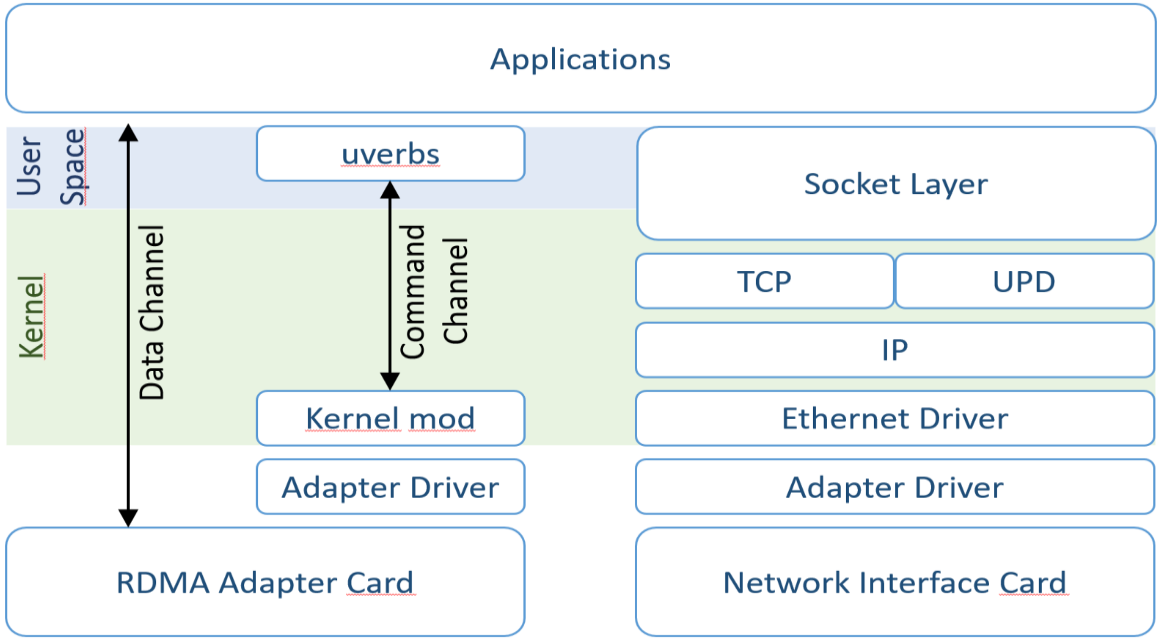
Responsabile nazionale: Andrea Triossi (Unipd – INFN Padova)

Unita partecipanti: INFN sezione di Padova
Laboratori Nazionali di Legnaro

26 Giugno 2025

Objectives

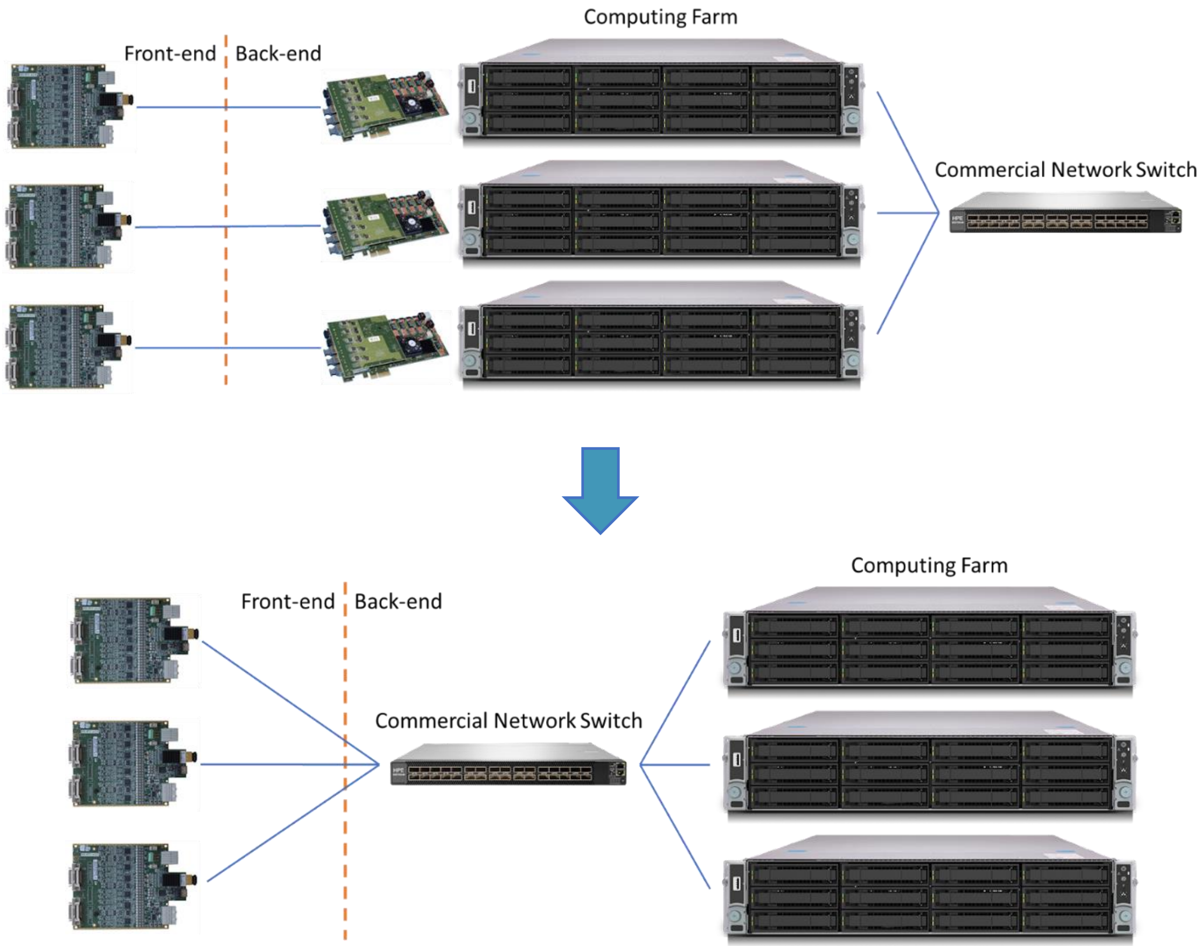
- Processing power is important as an efficient data movement
- In a DAQ system a large fraction of CPU is engaged in networking
  - Data manipulation (several copies)
  - Latency increase and throughput reduction
- Zero-copy is obtained by adding RDMA layer to the network stack



- FEROCE wants to move the adoption of the network protocol to the data producer
  - Front-end initiates the RDMA transfer
  - No point-to-point connection between front-end and back-end
  - Dynamical switching routing according to node availability

Methodology

- Several network stacks implementing RDMA
  - InfiniBand, RoCE, iWARP…
- RoCE (RDMA over Converged Ethernet)
  - Based on Ethernet networks
  - Industry-standard
  - Multi-vendor ecosystem
  - RoCEv2 packet switching (layer 2 and 3)



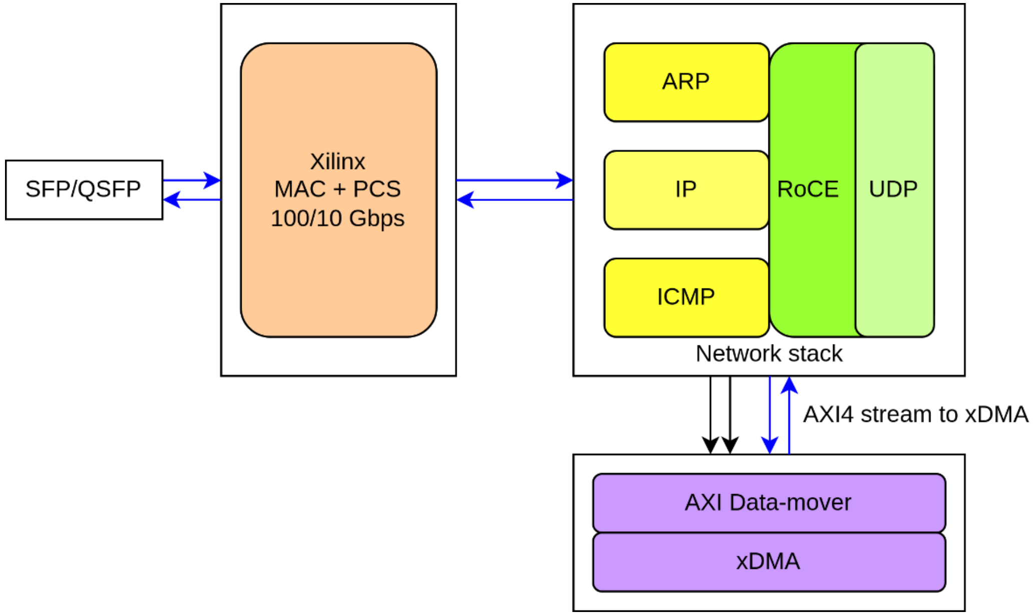
Context and trends

- Efficacy of RDMA already be proven @INFN: NaNet, APEnet…
- FPGA commonly adopted for implementing high throughput RDMA network stacks in datacenter even for physics experiments (e.g. ATLAS)

- DRD7 WP5: options for future readout links
  - Increasing complexity in Front-End with datalink following Ethernet bandwidth trend
  - R&D on 100G SiPh
  - Several investigations ongoing, from the no-backend to the smart-switch (concentrator of synchronous FE links to asynchronous Ethernet)

HLS experience

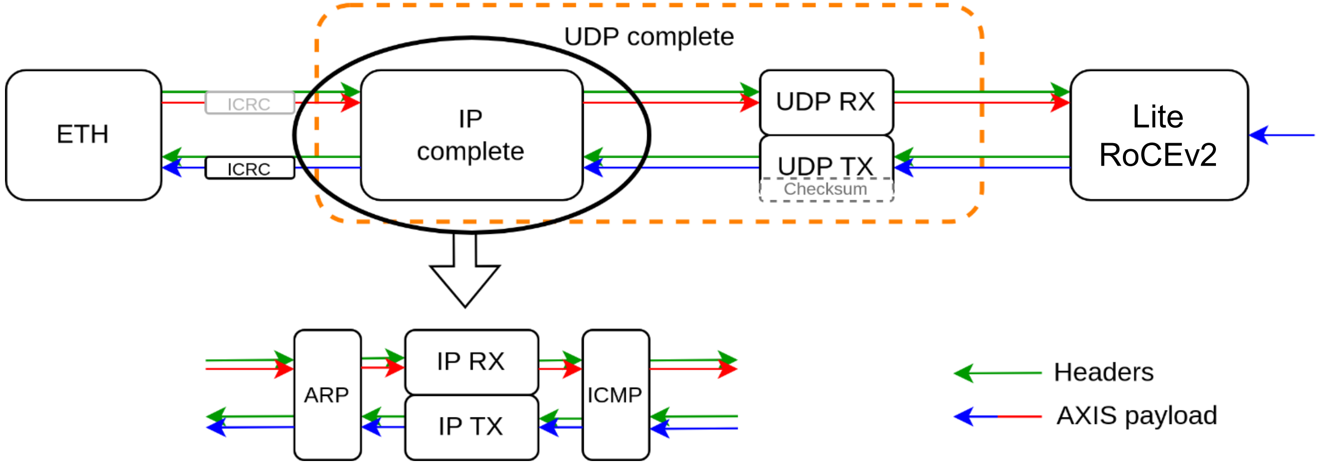
- ETH Zurich Network Stack
  - Entirely written in HLS
  - 10/100G via AMD 10G and 100G MAC IPs
  - TCP/IP support for out of band communication
  - DDR4 memory and recently HBM support
  - Used in many projects, wide community



- Reads/writes data from/to the host machine's memory through the xDMA
- Cannot stream from FPGA logic directly
- Bound to AMD ecosystem
- Quite resource hungry
- Not so easy to decouple TX and RX
- Very useful starting point for getting familiar with the protocol and the verbs but not suited for our needs
- It was decided to rewrite the stack at RTL in Verilog

Lite-RoCEv2 Module

- Based on Alex Forencich UDP/IP 1G/10G/25G open-source network stack for low level layers, with minor modifications
- Completely written at RTL (Verilog)
- Only RDMA SEND and WRITE with immediate for completion of the transfer
- Slow control based on UDP (setting QP, etc.)
- Main GitHub code
  - Supported speeds 10G/25G/100G
  - FAST CRC32
  - Support scripts for packet verification
- Presented at CHEP 2024



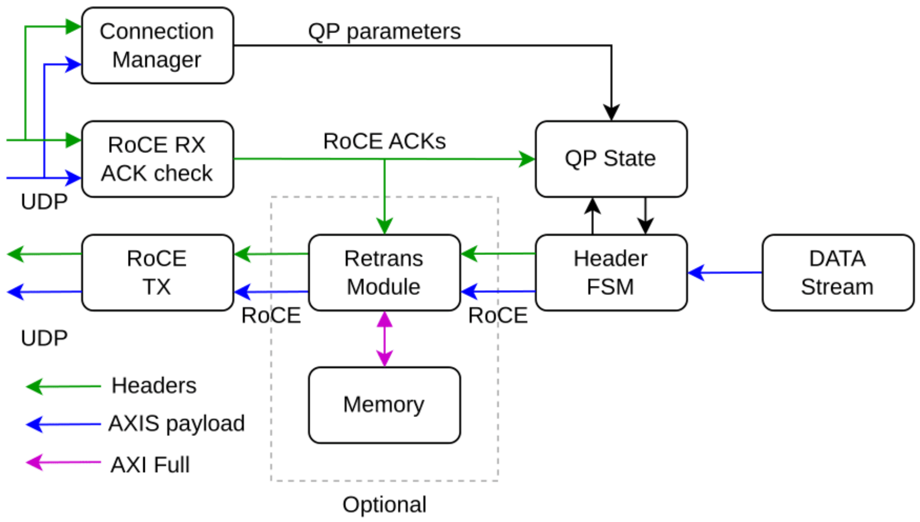
Lite-RoCEv2 Module

- Data is transferred using AXI4-Stream interface
  - Headers and data transmitted separately
- Added 100G datapath
- UDP checksum completely disabled
- Added ICRC insertion module
  - For 10/25G CRC32 is computed for 64b data words using a modified version of FCS
  - For 100G completely new module computes CRC parallelly on 512b data words
- Added ICMP echo reply server
- Added RoCEv2 TX
  - Supports for RDMA SEND, WRITE, with of without IMMEDIATE operations
  - FSM to splits AXI data stream in RDMA WRITE ONLY, FIRST, MIDDLE or LAST based on the selected MTU
- Added RoCEv2 RX
  - Only RDMA ACK packets are decoded
  - Used for latency and throughput measurements

Speed	Datapath	CLK Freq
10G	64b	156.250 MHz
25G	64b	390.625 MHz
100G	512b	322.266 MHz

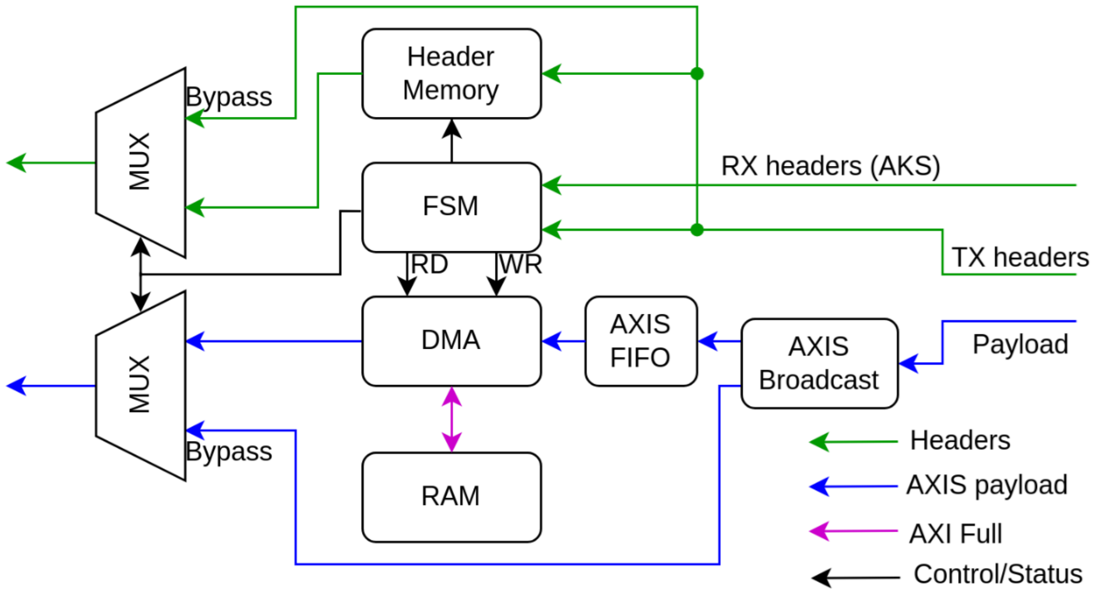
Lite-RoCEv2 TX

- Generate the Queue Pair parameter from the receiver side and allocate memory
  - Remote QP number
  - Remote Virtual Address
  - Registration key (to access the memory)
  - Remote Packet Sequence Number (PSN) needed to check stream's structure
- Send such parameter to FPGA on a sideband channel (UDP in our case)
- Stream data through the data AXI stream port
- Notify somehow the receiver that the transfer is completed (Immediate message/sideband)
- Latency and Throughput measured with the PSN of the sent packet and received ACK
- Without re-transmission, if a packet is not received properly the connection must be closed. Re-transmission needed for lossless networks



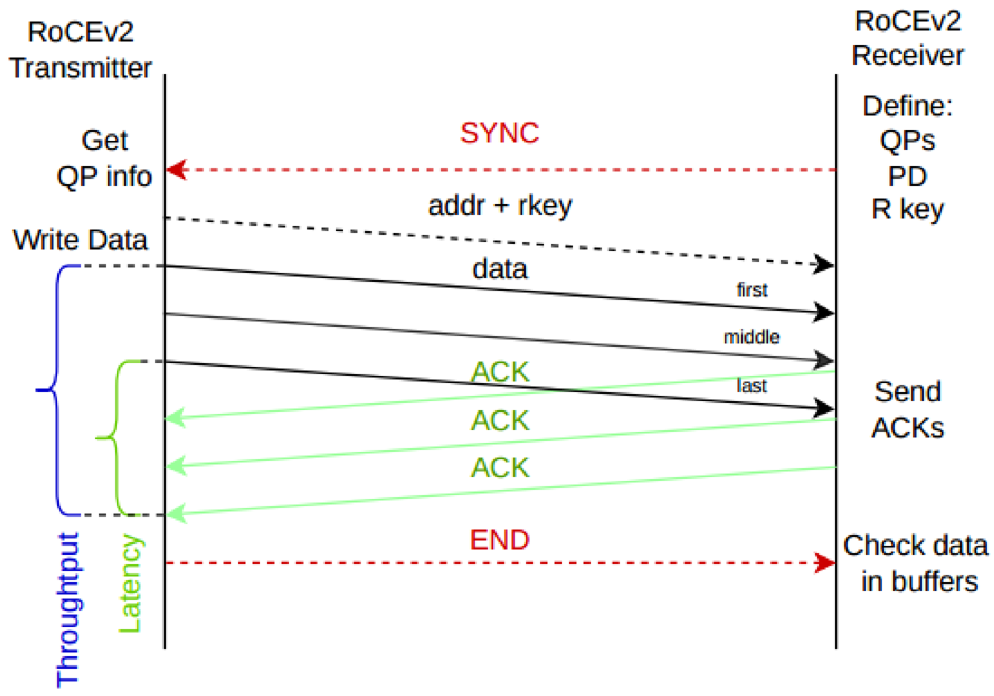
Lite-RoCEv2 Re-transmission

- Data continuously streamed and written in memory, PSN used as address
- RAM can be any kind of memory supporting AXI MM interface
- Good ACK reception resets the timeout counter
- Timeout or NAK reception trigger a retransmit
- Retransmit starts from the last ACKed PSN + 1 or the PSN in the NAK
- Upon reaching the last buffered PSN the normal data stream is resumed
- Timeout and retry values are set as parameters
- Only single QP supported



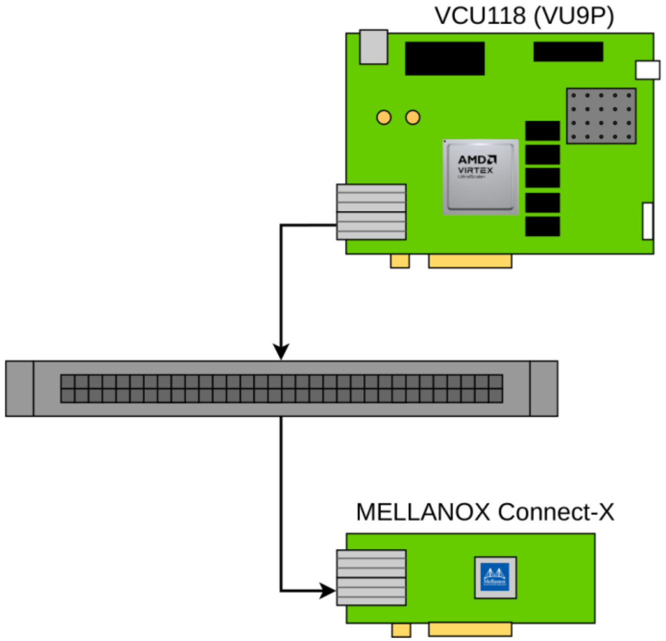
Latency and throughput

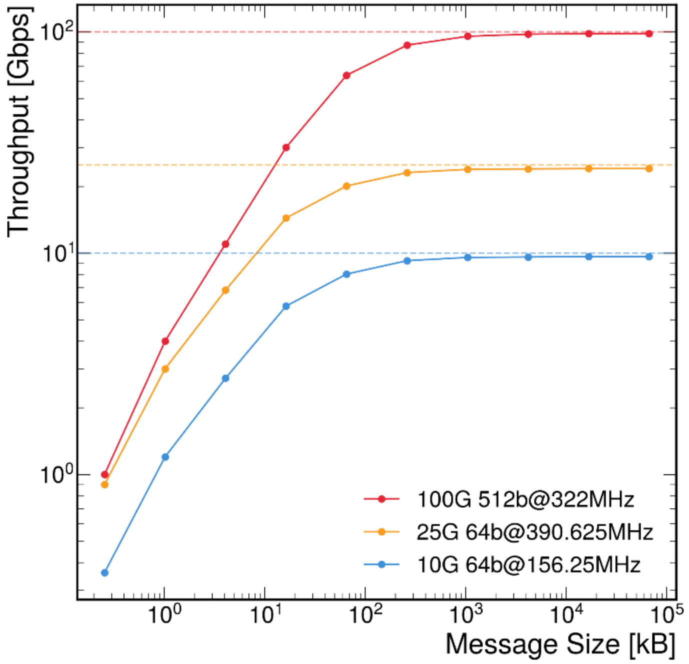
- Latency is computed averaging the time from each packet’s transfer start and its acknowledge (round trip)
- Throughput is computed considering the payload size and the time from first packet sent and last acknowledge received
- First transfer in the QP must be small (warm-up), it’s needed to for caching the QP info (NIC side), otherwise packet loss/retransmission is observed
- After that, no packet loss/retransmission is observed
- SYNC and END messages are outside the RoCEv2 protocol, but RDMA WRITE IMMEDIATE packet can trigger a completion message upon finishing



Point-to-point test

- Implementation on VCU118 (Virtex Ultrascale+)
- NVIDIA SN2100 16x100G network switch with cumulus linux
- Test performed at 10/25/100G with fixed MTU 4096B but variable message size



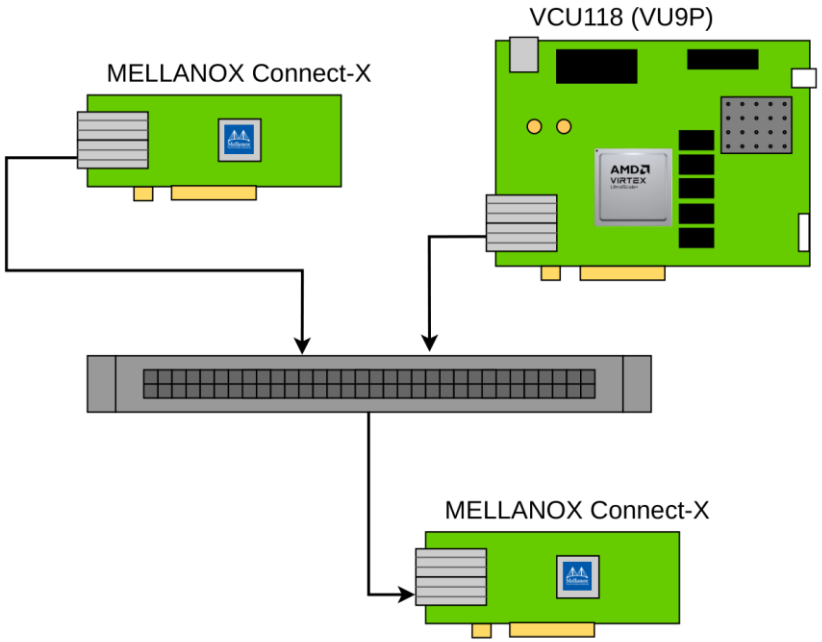


	Latency	Throughput
10G	4.6 µs	9.6 Gbps
25G	5 µs	24.2 Gbps
100G	3.2 µs	98.1 Gbps

Message size of 1 MB

Congestion test

- Two senders – one receiver
- All participants set at the same speed, forcing congestion on the receiver NIC
- No high-level congestion avoidance protocol only ethernet flow control
- Pause frames sent to stall the TX stream, latency increase
- Total throughput similar to point-to-point (93%)
- Slight throughput asymmetry in transmission probably do to unbalanced use of priorities

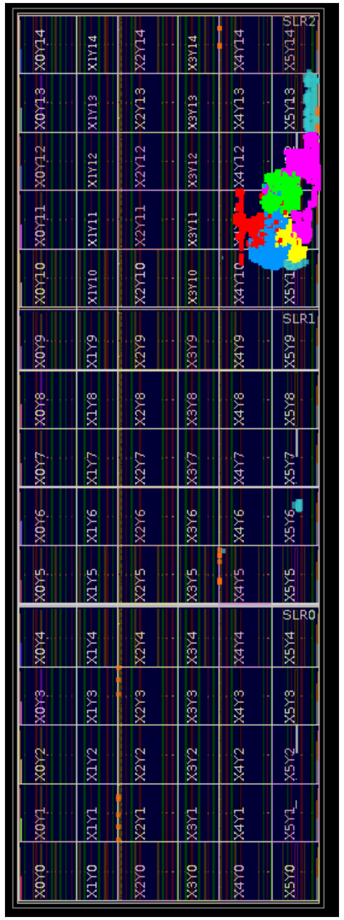


	Latency	Throughput
10G	15 µs	8.8 Gbps
25G	20 µs	22.2 Gbps
100G	25 µs	96.5 Gbps

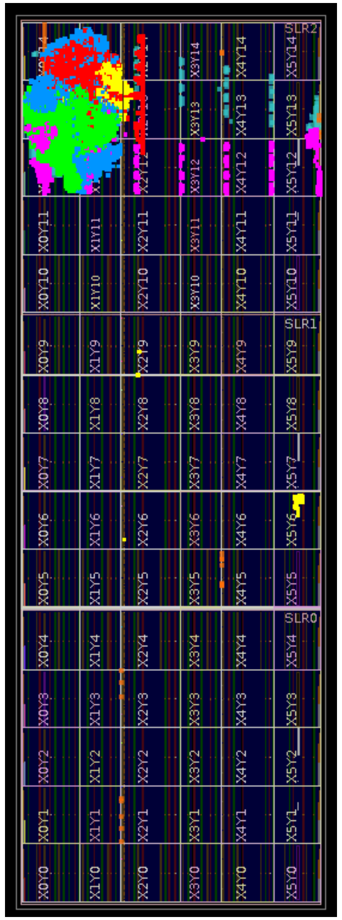
Message size of 1 MB

Resources

10/25G Floorplan



100G Floorplan



	Speed	LUTs (k)	REGs (k)	BRAM	URAM
Without Retrans	10G	13	16	11.5	0
	25G	13	16	11.5	0
	100G	30	43	24.5	0
With Retrans	10G	16	19	13.5	8
	25G	16	19	13.5	18
	100G	37	52	24.5	64

- Example of floorplan for 10/25/100G implementations

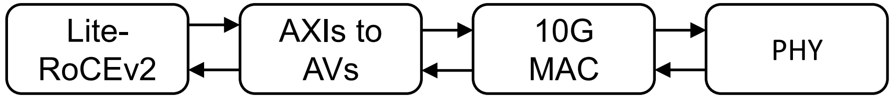
  - MAC/PHY
  - UDP/IP
  - Lite-RoCEv2
  - Retrans
  - Debugger

- ILAs used for debugging and monitoring
- Re-transmission memory buffer set to about 160 µs worth of data. 2MB for 100G, 0.5 MB for 25G and 0.26MB for 10G

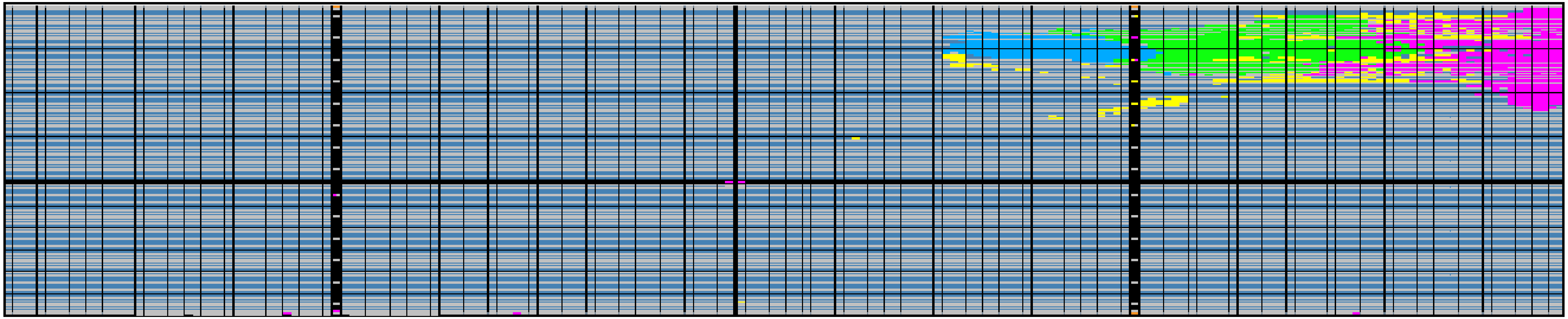
Porting to flash-based FPGA

- Tested on the evaluation kit of Microchip Polarfire
- Interfaced with Microchip 10G MAC and PHY
- Fw occupancy very similar to AMD FPGA

LUT (k)	DFF (k)	LSRAM	uSRAM
13	14	24	108



Chip Planner



Prospective applications

- CMS (CSN1)
  - L1T Scouting is a project aiming at acquiring the L1 primitives at the full bunch crossing rate
  - It is meant for HL-LHC but a demonstrator based on commercial electronic is already deployed
  - At present as DAQ link it adopts a lite version of the TCP/IP protocol at 100G ⟶ move to lite-RoCEv2
  - Next Generation Trigger is designing new electronics for scaling to 400G
- Known user of Lite-RoCEv2
  - HEIG-VD institute (CH) evaluating to use it in SKAO

Prospective developments

- Porting the stack to Versal
  - 100G $\longrightarrow$ MRMAC, 200G/400G $\longrightarrow$ DCMAC
  - Might be useful to use the Versal PS. Connection manager? QP management?
  - Software development to read this huge data stream, 200G and 400G for sure are non-trivial tasks
    - Ring buffer
    - Backpressure
- Unburdening CPU by serving data directly to GPU memory
  - RDMA to GPU (GPUDirect)
  - Enabling CUDA-based applications on front-end data

Project organization

- Two working package
  - WP1 Front-end firmware core
  - WP2 Application layer and emulation
- Four milestones

100% DONE • ~~31 Dec 2023~~ M1.1 Setting up of a global-oriented simulation environment for a small appliance of the RoCE stack (WP1)

100% DONE • ~~31 Dec 2023~~ M1.2 Testing a RoCE network based on COTS products (WP2)

100% DONE • ~~31 Dec 2024~~ M2 Complete test of the scalable RoCE firmware stack for front-end (WP1+WP2)

ON TRACK • 31 Dec 2025 M3 Complete test of the developed firmware on a radiation tolerant front-end FPGA (WP1+WP2)

		2023			2024			2025		
		Q1	Q2	Q3	Q1	Q2	Q3	Q1	Q2	Q3
WP1	**Front-end firmware core**	Development light RoCE		Simulation light RoCE	Implementing different data paths		Test FE RoCE on network	Porting to flash technology		Test on flash FPGA
WP2	**Application layer and emulation**	Soft RoCE	Setting up RoCE	Test of RoCE network	Acquisition system		Test FE RoCE on network	GPUdirect	Test of GPUdirect	Test on flash FPGA

Contributions

- Talks
  - G. Bortolato, Front-End RDMA Over Converged Ethernet, real-time firmware simulation, Topical Workshop on Electronics for Particle Physics (TWEPP), Geremeas, Italy, 2023
  - G. Bortolato, Front-End RDMA Over Converged Ethernet, real-time firmware simulation, Technology & Instrumentation in Particle Physics (TIPP), Cape Town, South Africa, 2023
  - G. Bortolato, Front-End RDMA Over Converged Ethernet, lightweight RoCE endpoint, Conference on Computing in High Energy and Nuclear Physics (CHEP), Kraków, Poland, 2024
  - A. Triossi, FEROCE and the journey of data from the detector to the computing farm, Workshop on Electronics for Physics Experiments and Applications @INFN, Torino, Italy, 2025
- Publications
  - G. Bortolato et al., Front-end RDMA over Converged Ethernet, real-time firmware simulation, Journal Of Instrumentation, 10.1088/1748-0221/19/03/C03038
  - G. Bortolato et al., FEROCE: Front-End RDMA Over Converged Ethernet, a lightweight RoCE endpoint, submitted to EPJ WoC