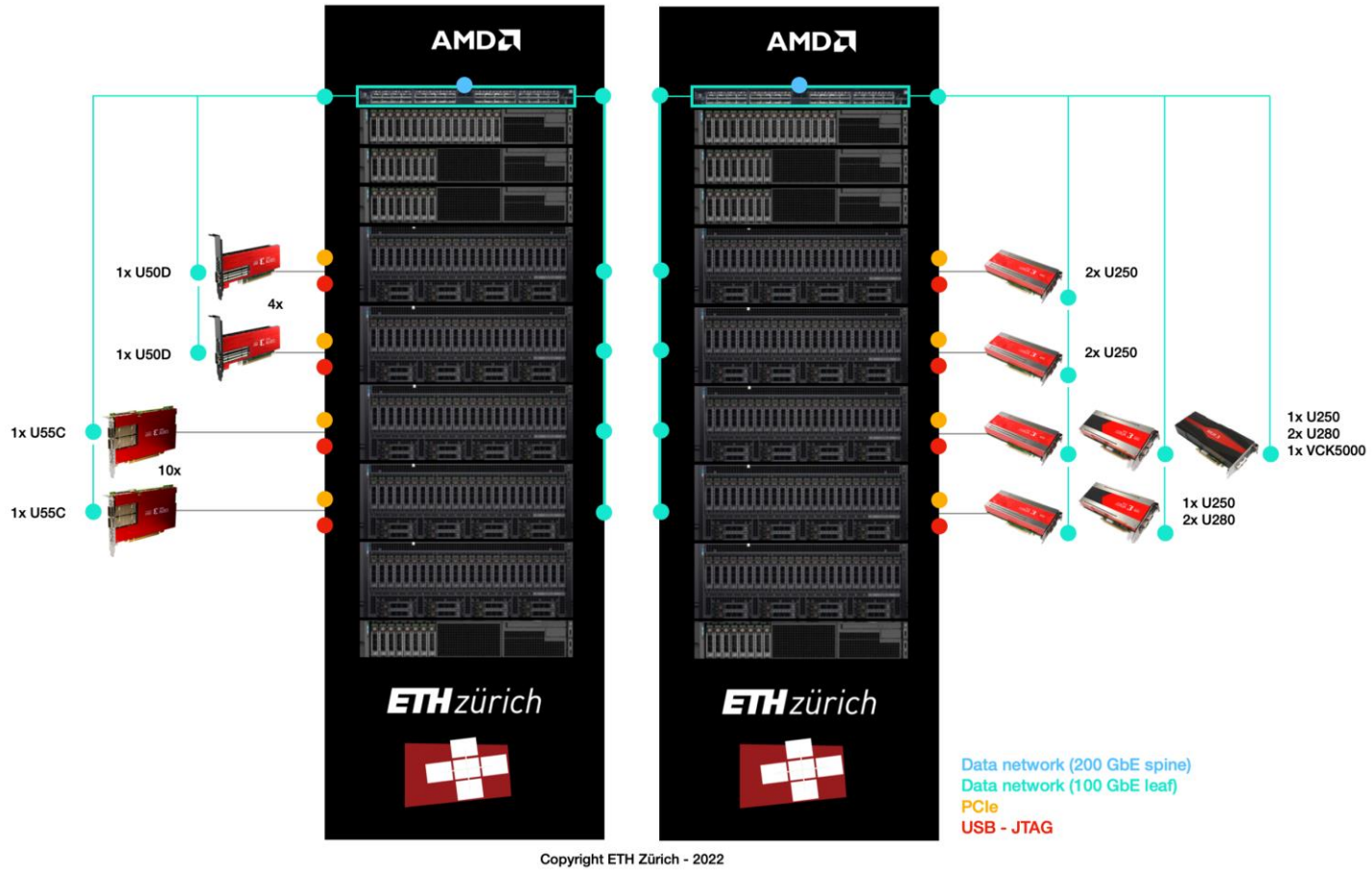
Vitis Network Example (VNx)
A Lightweight UDP Network Layer for FPGA

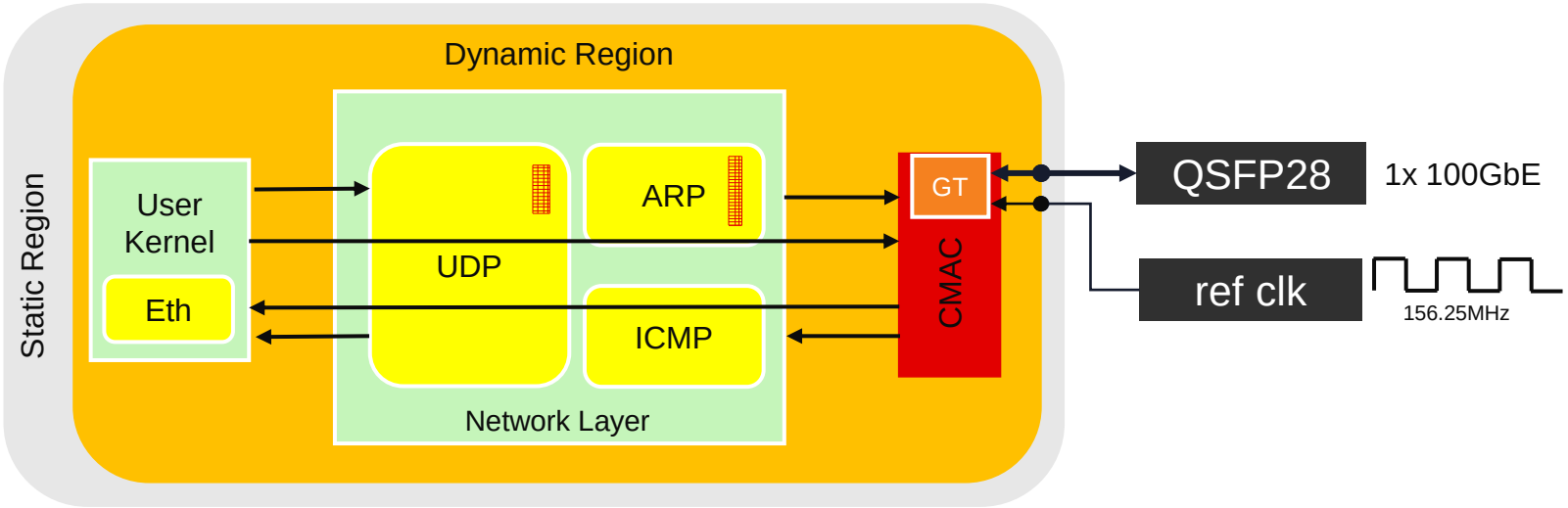
Lucian Petrica

VNx Goals

- Provide networking on Vitis platforms for Alveo cards
  - Fast stream transport over routed fabrics
  - Low resource overhead
  - Easy to link with existing designs that use streaming kernels
  - Portable across Alveo range
- Easy integration & ease of use
  - Works out-of-the-box at HACC
  - PYNQ-based Python driver
  - C++ driver (community contributed)



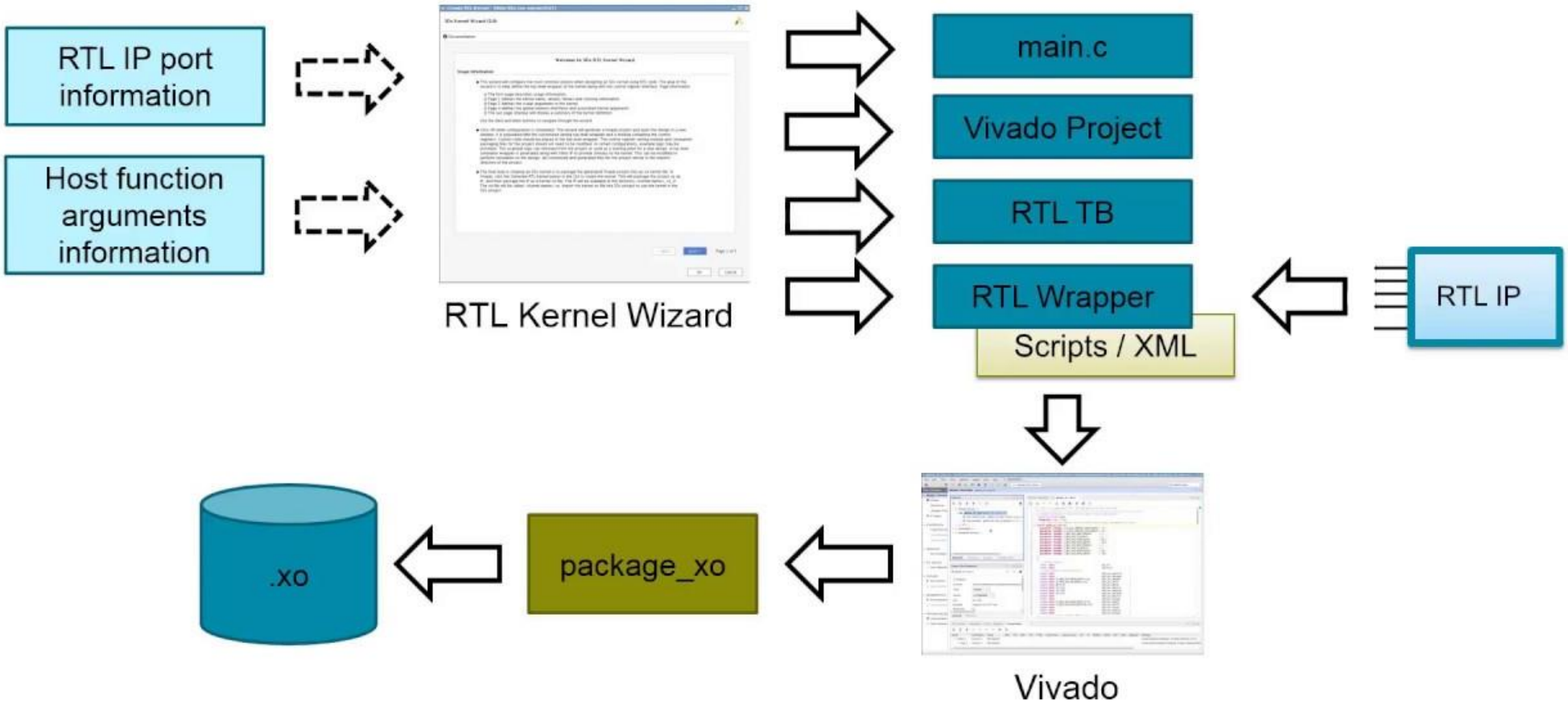
VNx Architecture



- Support for multiple interfaces and cards
  - Any Vitis Alveo platform with GT access
  - U50, U55C, U200, U250 and U280
- Layered architecture
  - Easily adaptable to user needs

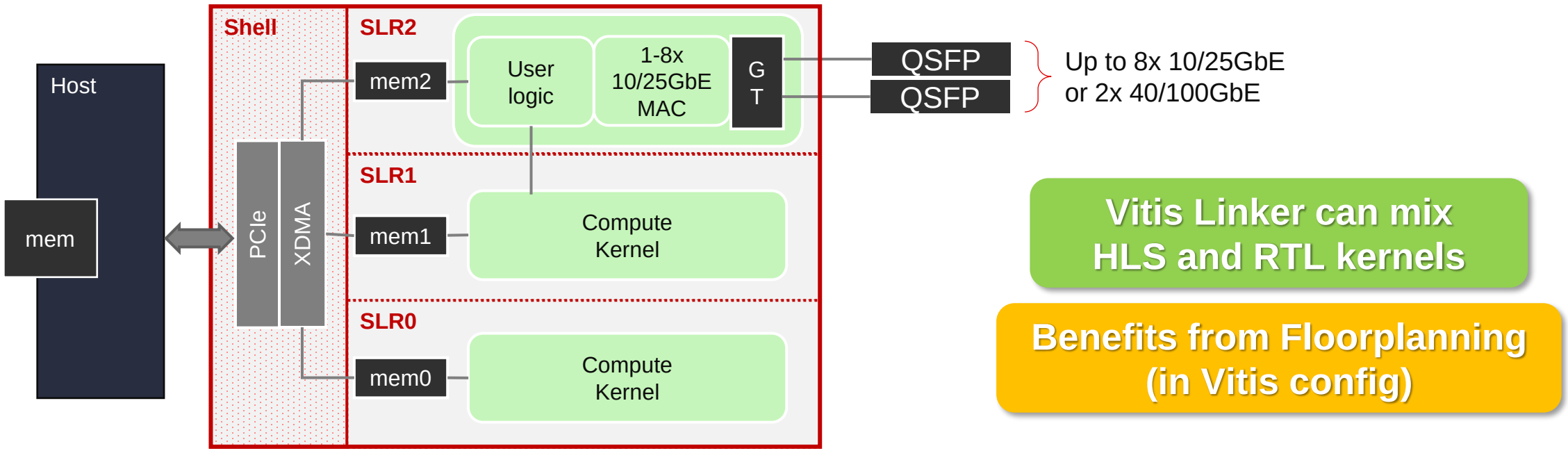
- CMAC kernel
  - Configuration is specific to Alveo card and network interface (if card is multi-port)
- Network Layer
  - UDP as transport protocol
  - Single channel application interface

Refresher: Vitis RTL Kernels



Vitis GT Kernels – Networking Flexibility

- RTL kernels which connect to Gigabit Transceiver (GT) pins
- Can implement your choice of MAC
- Documentation here



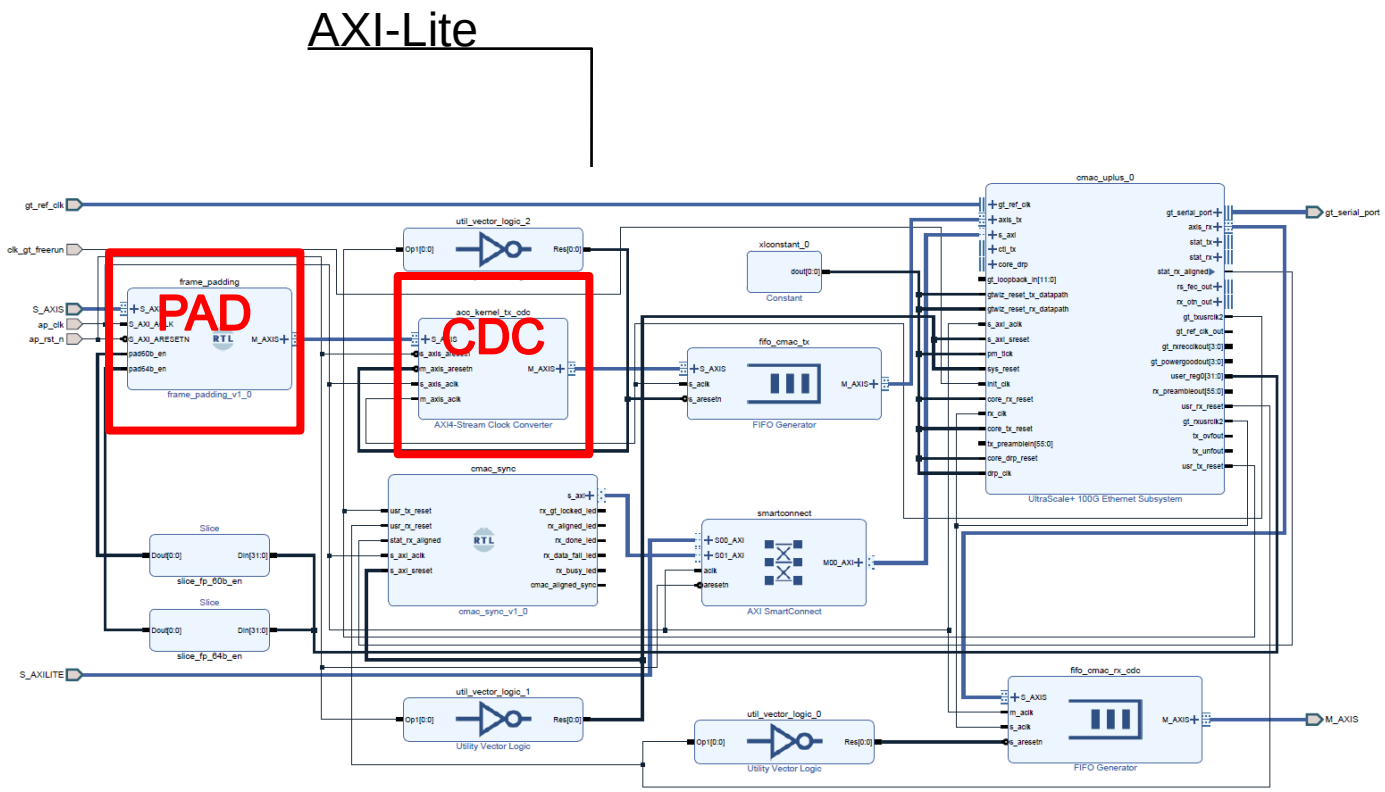
Vitis Linker can mix HLS and RTL kernels

Benefits from Floorplanning (in Vitis config)

Production support for all Alveo cards

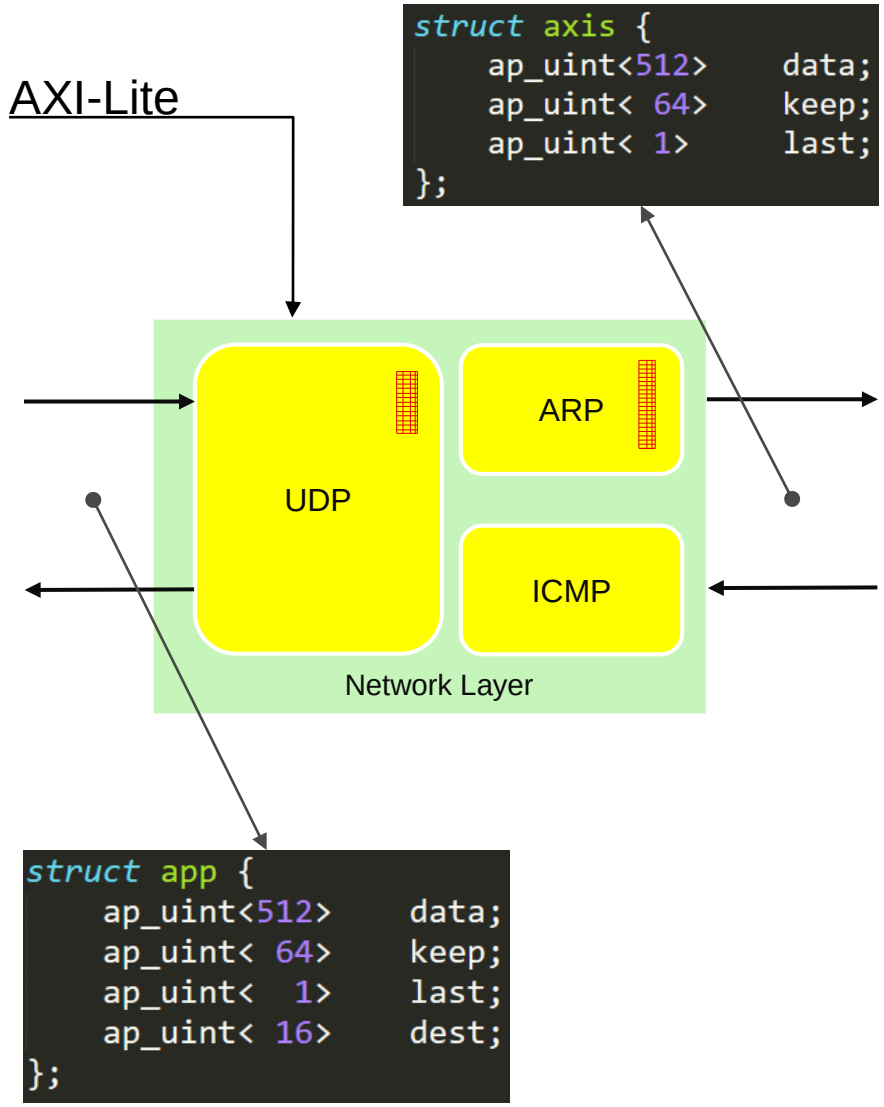
CMAC Kernel Implementation

- **Vitis-compatible**
  - Standard AXI-Lite and AXI-Stream interfaces + GT pins
  - Single external kernel clock
  - All configuration via AXI-Lite - access to the CMAC IP register map
- **Statistics** gathering and readout via AXI-Lite
- **RS-FEC** capability included in the CMAC IP
  - optionally enabled from host code
- **Portable** across UltraScale+ Alveo cards
  - CDC included to internal clock domains
  - No fixed clocks required, due to internal CDC
- Optional **frame padding** to 60 or 64 bytes

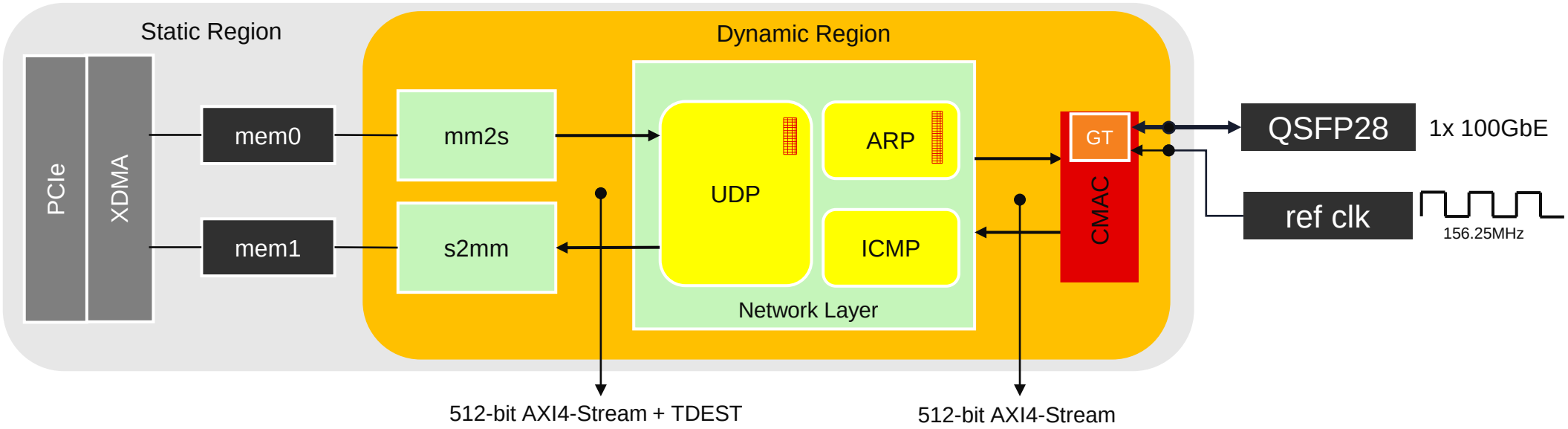


Network Layer

- Based on Limago
- ARP
  - Translation between MAC and IP addresses
  - ARP table is accessible from host
- ICMP
  - ping functionality
- UDP
  - Transport protocol
  - Host-configured connection/socket table (16 entries)
  - TDEST in application interface specify connection ID
- AXI-Stream Interfaces
  - 64-byte data (with associated TKEEP)
  - 16-bit TDEST carries index in connection table

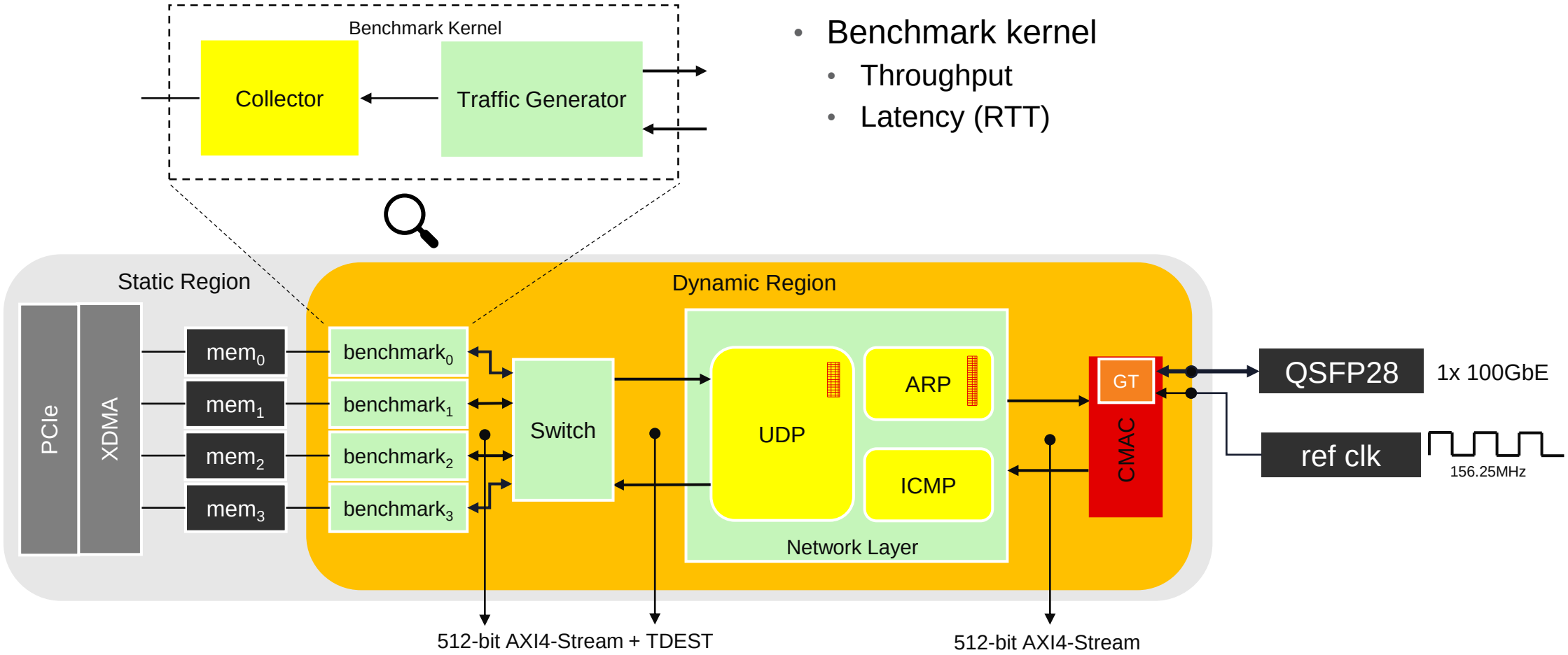


Basic Example



- Two user kernels
- Memory mapped to stream (mm2s)
  - Push data from host to network
- Stream to memory mapped (s2mm)
  - Pull data from network to host
- Works for 1 or 2 interfaces, depending on the board

Benchmark Example

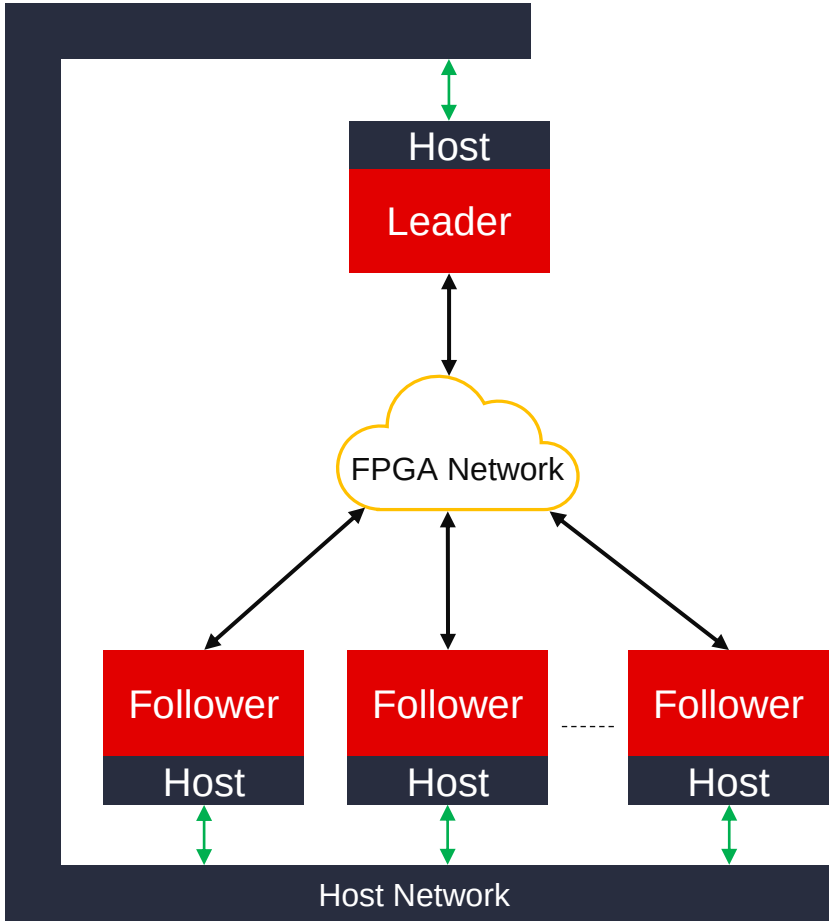


- Benchmark kernel
  - Throughput
  - Latency (RTT)

Host Code Execution on Heterogeneous Systems

- Steps for application execution
  1. Configure the devices
     - Download FPGA configuration file
     - Configure IP addresses
     - ARP Discovery
     - Populate UDP table

     } setup
  2. Allocating buffers
  3. Writing the buffers to FPGA memory
  4. Running the accelerators
  5. Reading the buffers from FPGA memory
- How to execute these steps remotely?

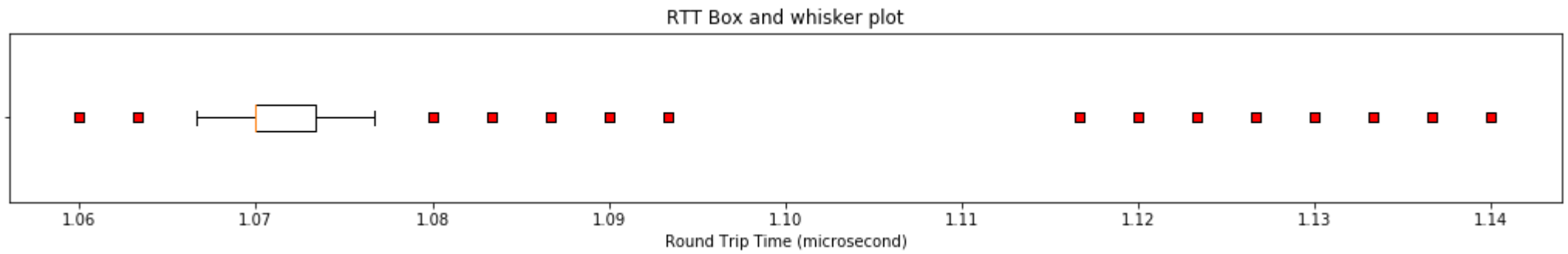
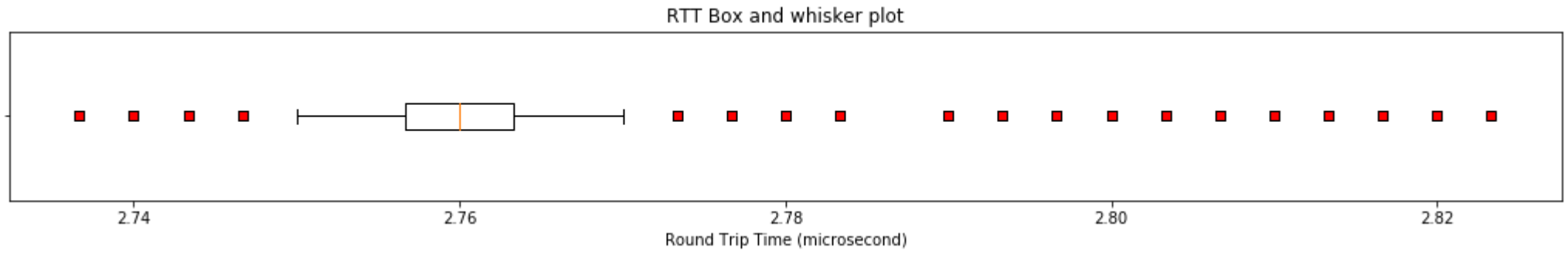


Dask-on-PYNQ

- Dask is a flexible library for parallel computing in Python
- PYNQ-on-Alveo as host code
  - Notebooks on Jupyter Lab
- Dask class wrapper
  - Distributed configuration
  - Remote buffer allocation
  - Remote task execution
  - All the same functionality as PYNQ but distributed!
- Not specific of VNx

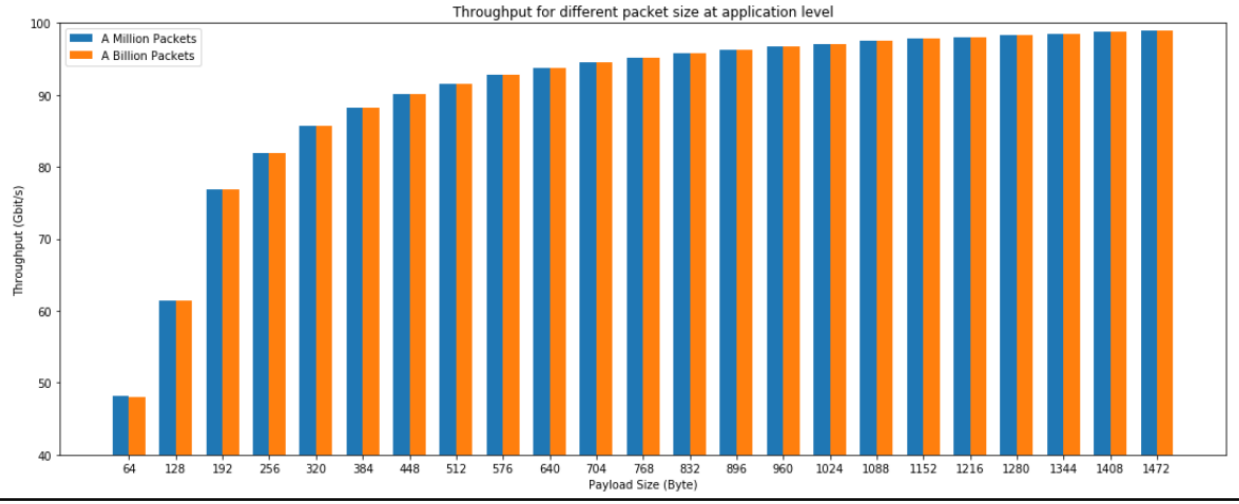
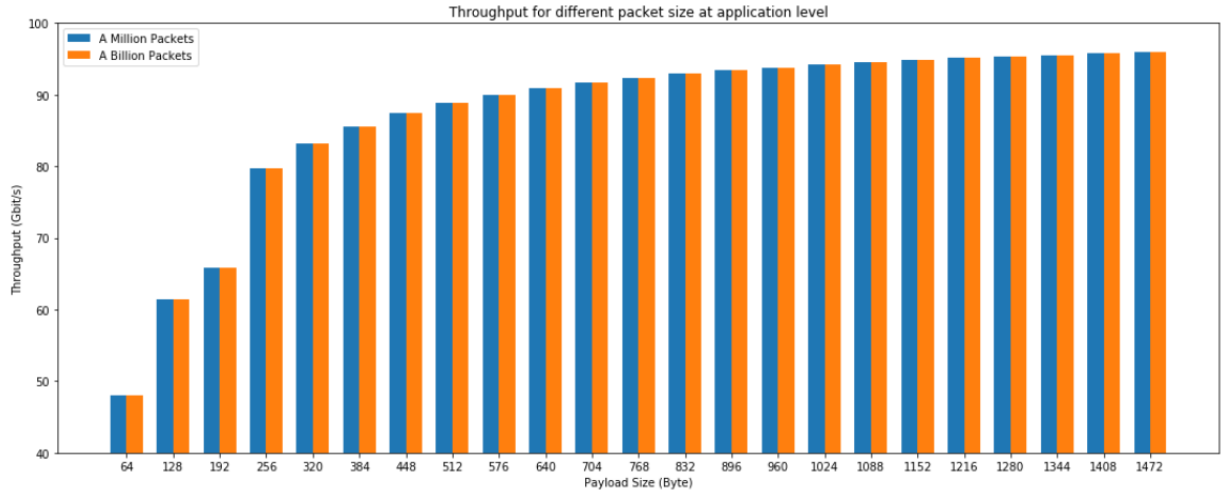

```
daskdev_w0 = DaskDevice(client, workers[0])
daskdev_w1 = DaskDevice(client, workers[1])

xclbin = 'vnx_benchmark_if3.xclbin'
ol_w0 = pynq.Overlay(xclbin, device=daskdev_w0)
ol_w1 = pynq.Overlay(xclbin, device=daskdev_w1)
```

Results: Benchmark Application Latency

Scenario	RTT @ app
Point to Point	
100GbE Switch	

*Results are included in the Notebooks

Results: Benchmark Application Throughput

Scenario	Throughput @ app
Point to Point	
100GbE Switch	

*Results are included in the Notebooks

VNx Summary

- 100 Gb/s UDP networking on Alveo cards
- Lightweight but unreliable
  - Ideal for point to point or controlled environment
- Simple interface
  - Easy to integrate (compute) kernels
- Two examples designs with companion PYNQ notebooks
- Dask class for distributed configuration
  - Not specific of VNx
- Python driver on top of PYNQ
- C++ driver, community contributed
- Open source
  - https://github.com/Xilinx/xup_vitis_network_example

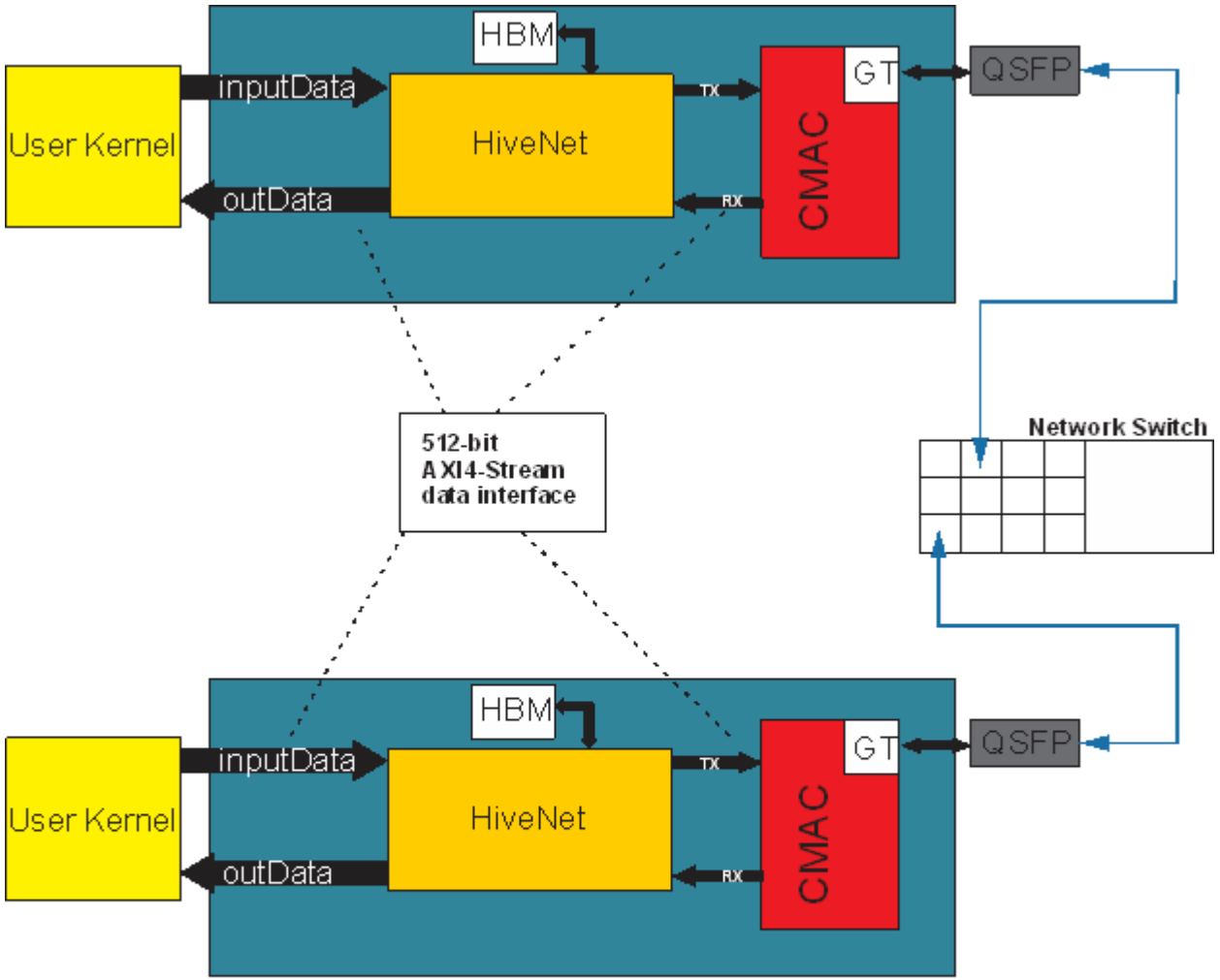
	VNx
Platform(s)	U50/U250/U280
Host code	pynq (+ dask or MPI)
Transport protocol	UDP (unreliable)
Packet reorder	No
#100G Interfaces	Up to 2, depends on card
#Memory Banks	none
#Connections	16
Application Interface	Single AXI-Stream bus
Throughput	Max (@ 256B+/packet)
RTT	2.75 µs
LUT*	22,773 (1.75%)
FF*	48,914 (1.88%)
BRAM*	58.5 (2.9%)

* Resource consumption for Network Layer only, U280 (xilinx-u280-xdma-201920.3)

Preview: HiveNet ROCE Kernel

https://github.com/Xilinx/AlveoLink

- ROCE protocol enables reliable transport over UDP with retransmission
- HiveNet is a Vitis-compatible ROCE kernel currently under validation
- User interface similar to VNx (AXI-Streams with 13-bit TDEST for connection ID)
- Can use same CMAC kernel as VNx
- Not as portable currently, utilizes fixed clocks only available on select platforms e.g. U55C
- Higher resource utilization compared to VNx, requirement for HBM access



Disclaimer & Attribution

Timelines, roadmaps, and/or product release dates shown in these slides are plans only and subject to change.

The information contained herein is for informational purposes only and is subject to change without notice. While every precaution has been taken in the preparation of this document, it may contain technical inaccuracies, omissions and typographical errors, and AMD is under no obligation to update or otherwise correct this. Advanced Micro Devices, Inc. makes no representations or warranties with respect to the accuracy or completeness of the contents of this document, and assumes no liability of any kind, including the implied warranties of noninfringement, merchantability or fitness for particular purposes, with respect to the operation or use of AMD hardware, software or other products described herein. No license, including implied or arising by estoppel, to any intellectual property rights is granted by this document. Terms and limitations applicable to the purchase or use of AMD's products are as set forth in a signed agreement between the parties or in AMD's Standard Terms and Conditions of Sale.


AMD