The Network Management Unit (NMU): Securing Network Access For Direct-Connected FPGAs

Daniel Rozhko and Paul Chow

High-Performance Reconfigurable Computing Group · University of Toronto

February 26th, 2019

FPGAs in Datacenters

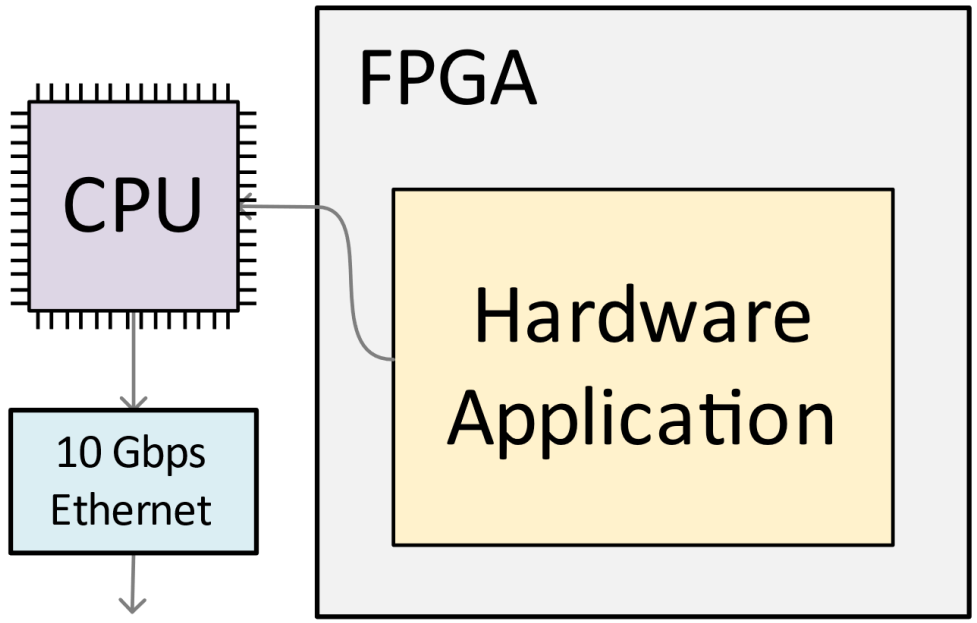
- FPGAs are increasingly being deployed in datacenter and cloud environments

FPGAs in Datacenters

- FPGAs are increasingly being deployed in datacenter and cloud environments
- Major deployments are available by many vendors:

FPGA Connectivity Models

- Traditional FPGA Connectivity Model – Accelerator



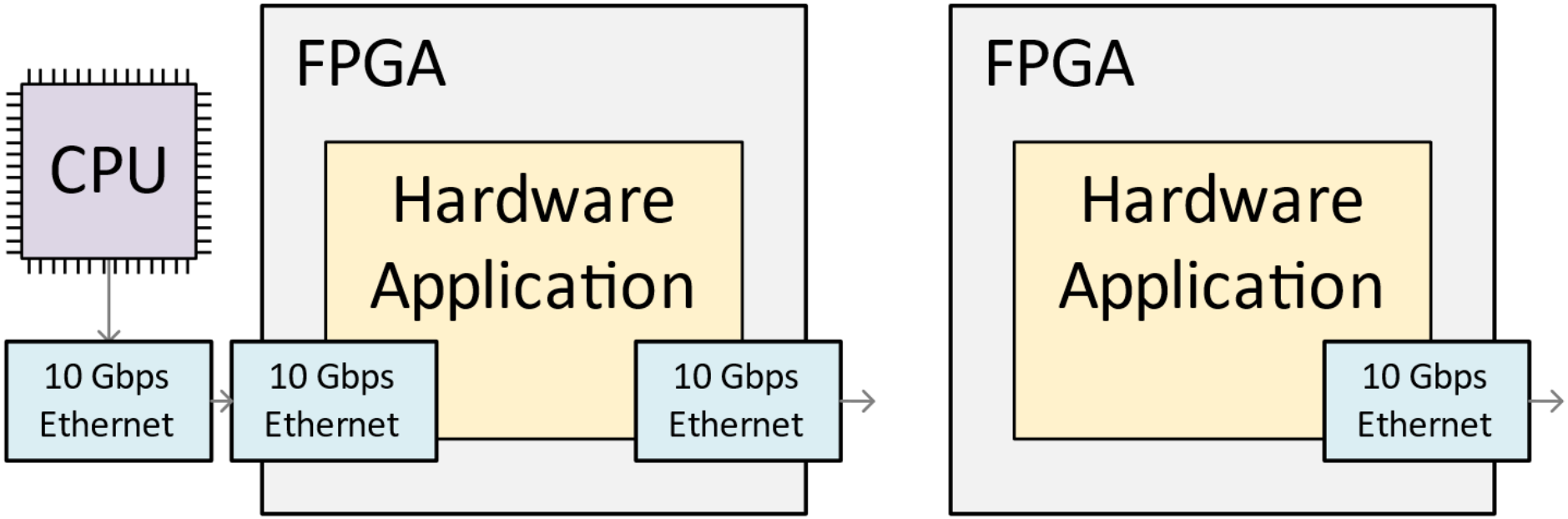
FPGA Connectivity Models

- Traditional FPGA Connectivity Model – Accelerator

Network connectivity can be secured in software

FPGA Connectivity Models

- Increasingly Deployed Model – Direct-Connected FPGA



FPGA Connectivity Models

- Increasingly Deployed Model – Direct-Connected FPGA

Network connectivity must be explicitly secured in hardware

Securing Network Access for FPGAs

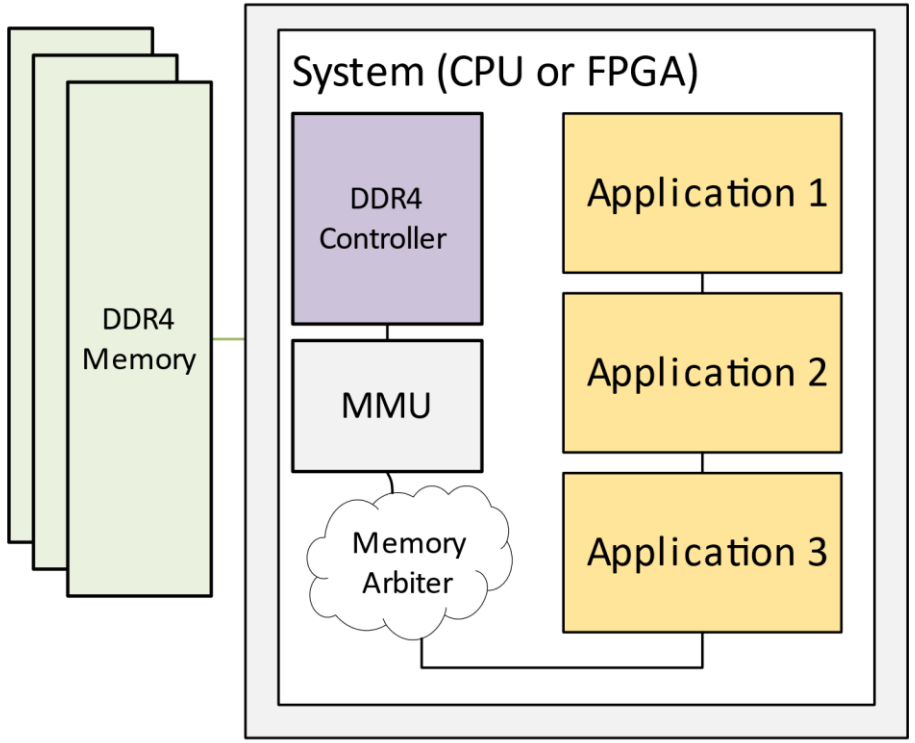
- Why do we need to secure network connectivity?

Securing Network Access for FPGAs

- Why do we need to secure network connectivity?
- Multi-user or multi-tenant environments
  - Multiple applications can affect/observe network behaviour
- Un-trusted users (i.e. in cloud-like deployments)
  - Network (potentially) exposed to errant or malicious behaviour

Analogue – Memory Management Unit (MMU)

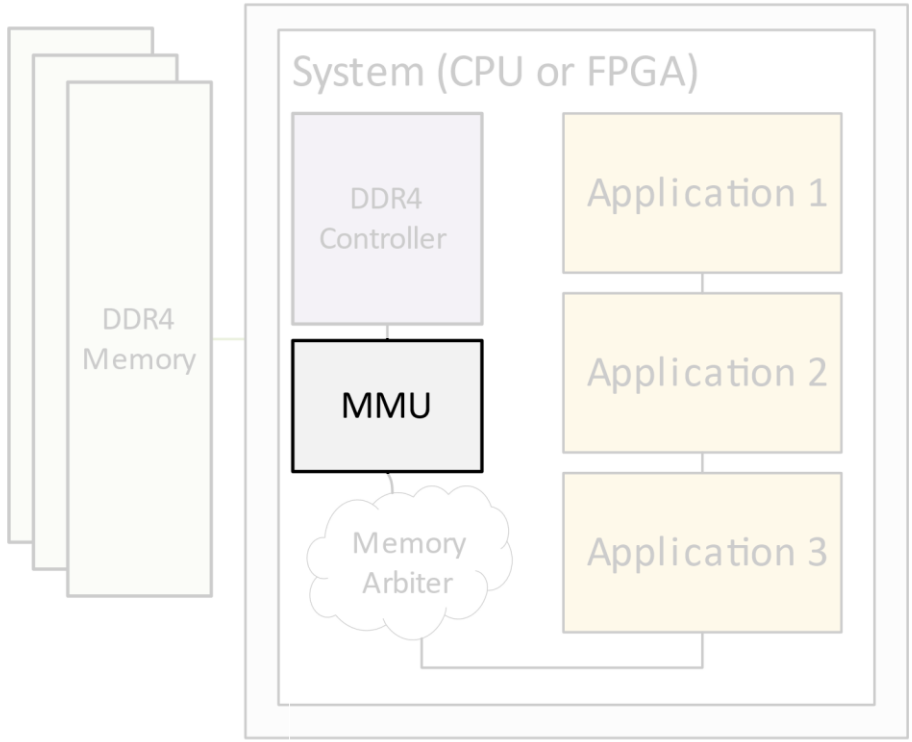
- An analogous shared resource – memory



Analogue – Memory Management Unit (MMU)

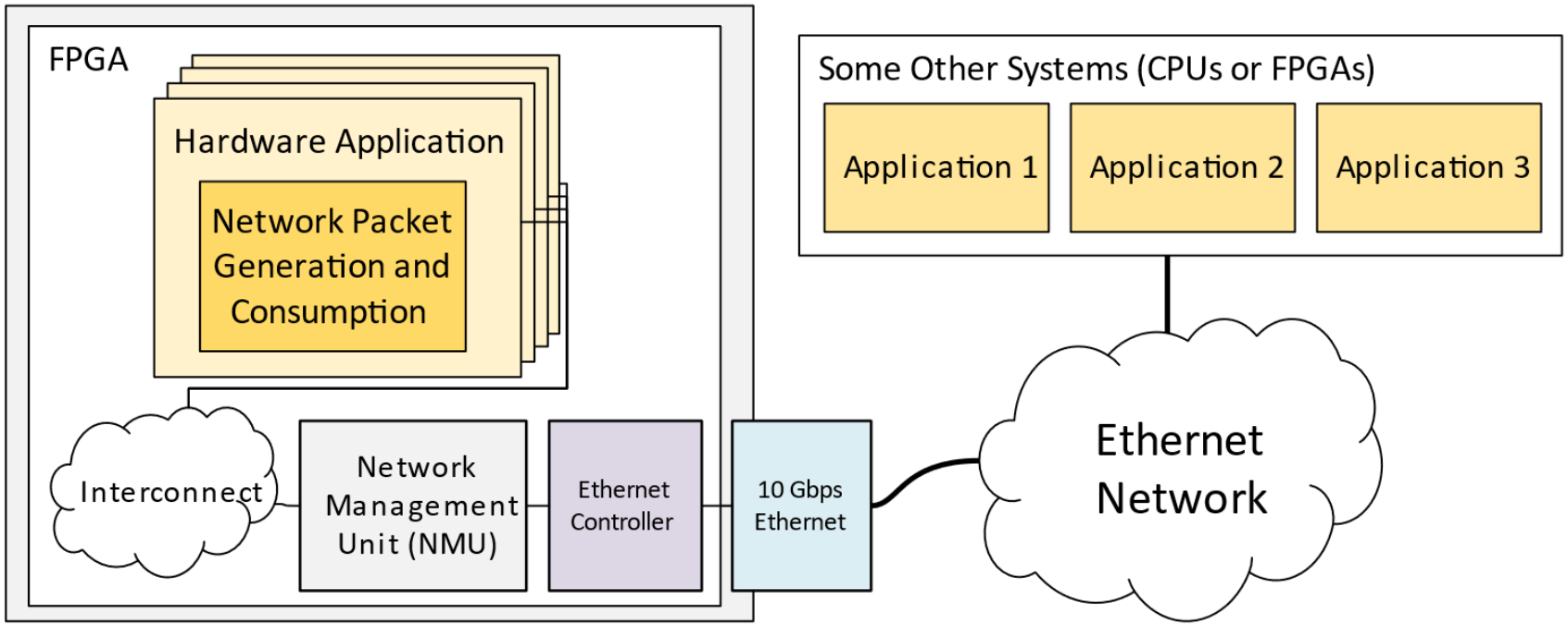
- An analogous shared resource – memory

MMU provides for each application:
- isolation to specific parts of memory
- rejection of invalid requests



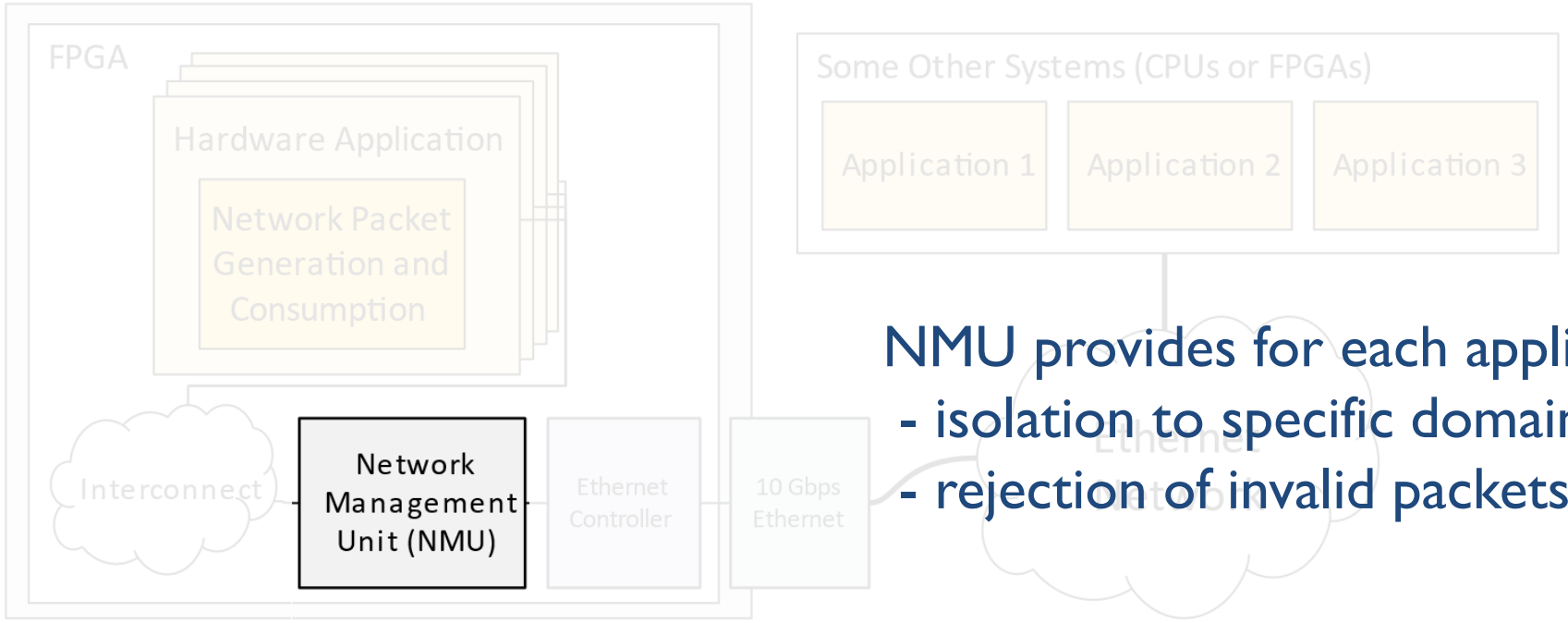
The Network Management Unit (NMU)

- Introducing the NMU



The Network Management Unit (NMU)

- Introducing the NMU – securing network connectivity



NMU provides for each application:
- isolation to specific domains of network
- rejection of invalid packets

Outline

- Motivation for NMU
- NMU Architecture Types
- Our Hardware Implementation
- Evaluation of NMU Types
- Conclusions

Outline

- Motivation for NMU
- NMU Architecture Types
- Our Hardware Implementation
- Evaluation of NMU Types
- Conclusions

Previous Work – Hardware

- Network security schemes from previous FPGA works
  - Packet encapsulation
  - MAC source address replacement
  - Full network switch in soft-logic
    - e.g. OpenFlow switch on FPGA

Previous Work – Hardware

- Network security schemes from previous FPGA work
  - Packet encapsulation (1)
  - MAC source address replacement (2)
  - Full network switch in soft-logic (3)
    - e.g. OpenFlow switch on FPGA

- Either very simplistic (1,2) or high utilization (3)

Previous Work – Software

- Firewalls
  - Network Access Control Lists (NACL)
  - Both Source and Destination Address ACLs
- Virtualization
  - VLAN (tag-based) ,VXLAN, NVGRE (encapsulation-based)
- Hairpinning
  - Pushing securitization to another switch or appliance

NMU Architecture Types

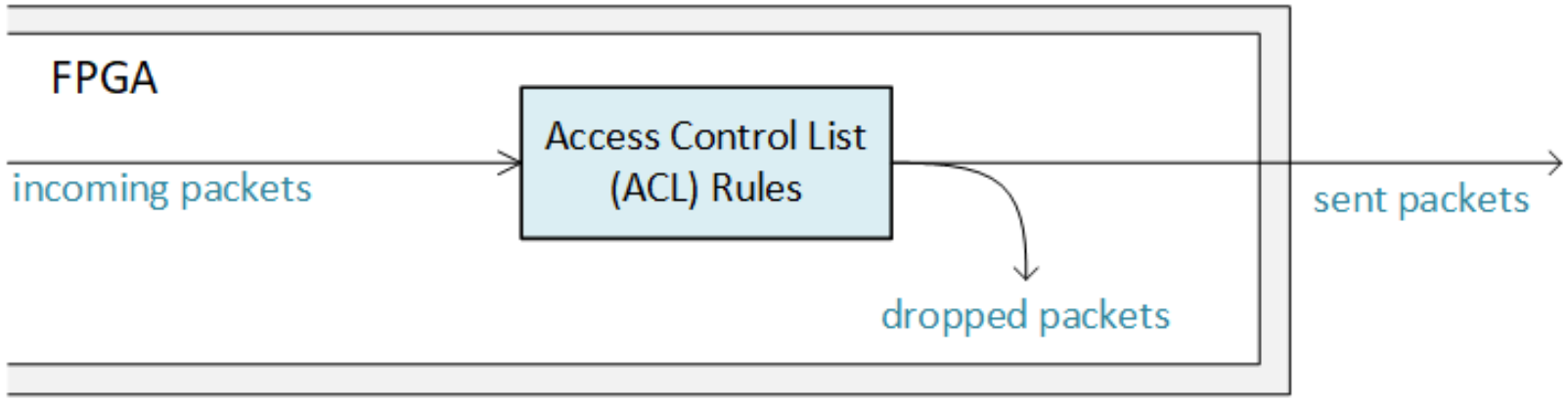
- Four main considerations identified for NMU design

1) Access Controls Implemented
2) Support for Internal Routing
3) Virtual Networking Functionality
4) Network Layer of Operation

NMU Architecture Types

- Four main considerations identified for NMU design

  1) Access Controls Implemented



Access Controls

- ACLs can be implemented in the NMU, or in the downstream physical switch

Access Controls

- ACLs can be implemented in the NMU, or in the downstream physical switch
- Our classification of NMU Types:

  Type A → no ACLs implemented in NMU (802.1Qbg, 802.1pr)

Access Controls

- ACLs can be implemented in the NMU, or in the downstream physical switch
- Our classification of NMU Types:

  Type A → no ACLs implemented in NMU (802.1Qbg, 802.1pr)

  Type B → Sender Address ACLs only in NMU

Access Controls

- ACLs can be implemented in the NMU, or in the downstream physical switch
- Our classification of NMU Types:

  Type A → no ACLs implemented in NMU (802.1Qbg, 802.1pr)

  Type B → Sender Address ACLs only in NMU

  Type C → Sender and Destination Address ACLs in NMU

Access Controls

- ACLs can be implemented in the NMU, or in the downstream physical switch
- Our classification of NMU Types:

  Type A → no ACLs implemented in NMU (802.1Qbg, 802.1pr)

  Type B → Sender Address ACLs only in NMU

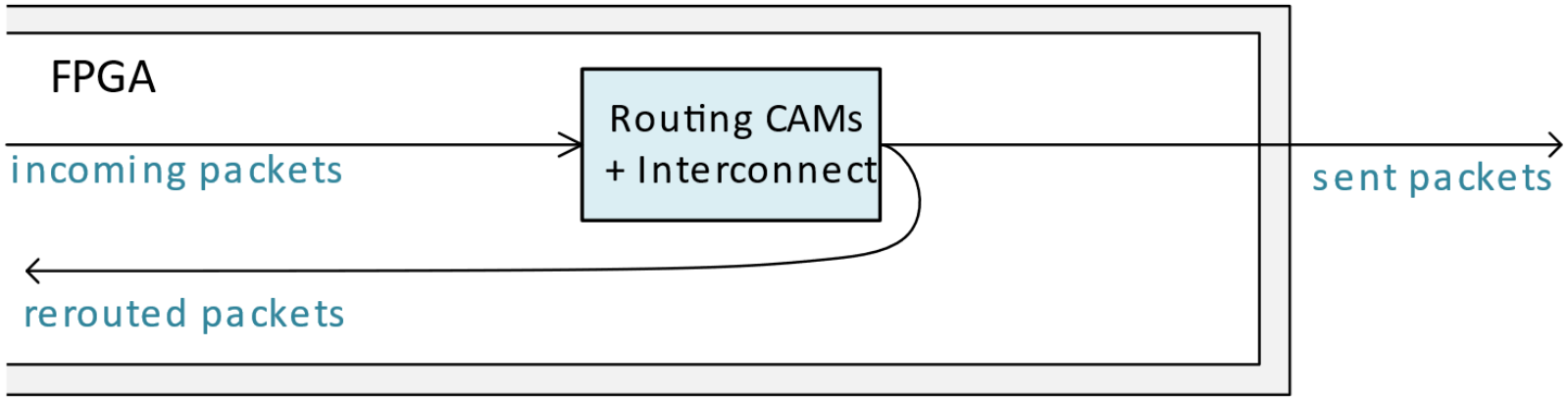
  Type C → Sender and Destination Address ACLs in NMU

  Type E → Encapsulation, no ACLs necessary

NMU Architecture Types

- Four main considerations identified for NMU design

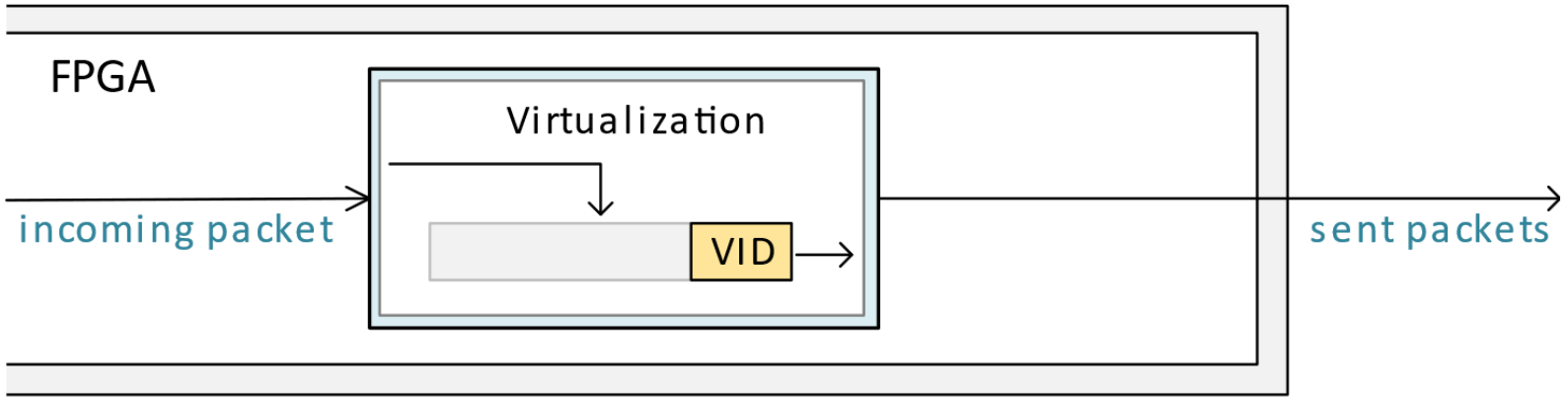
2) Support for Internal Routing



NMU Architecture Types

- Four main considerations identified for NMU design

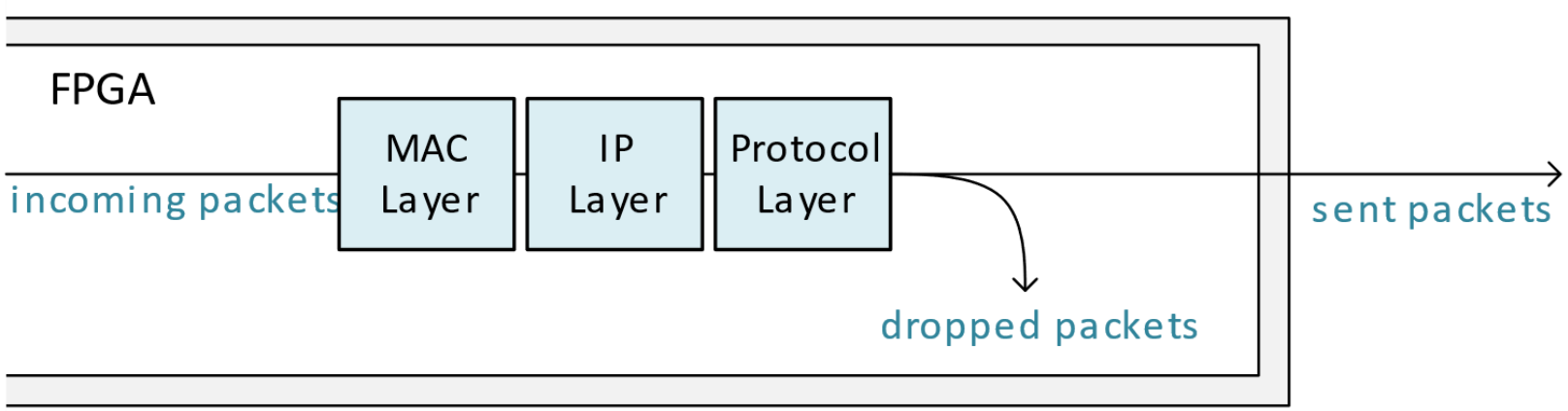
3) Virtual Networking Functionality



NMU Architecture Types

- Four main considerations identified for NMU design

4) Network Layer of Operation



Outline

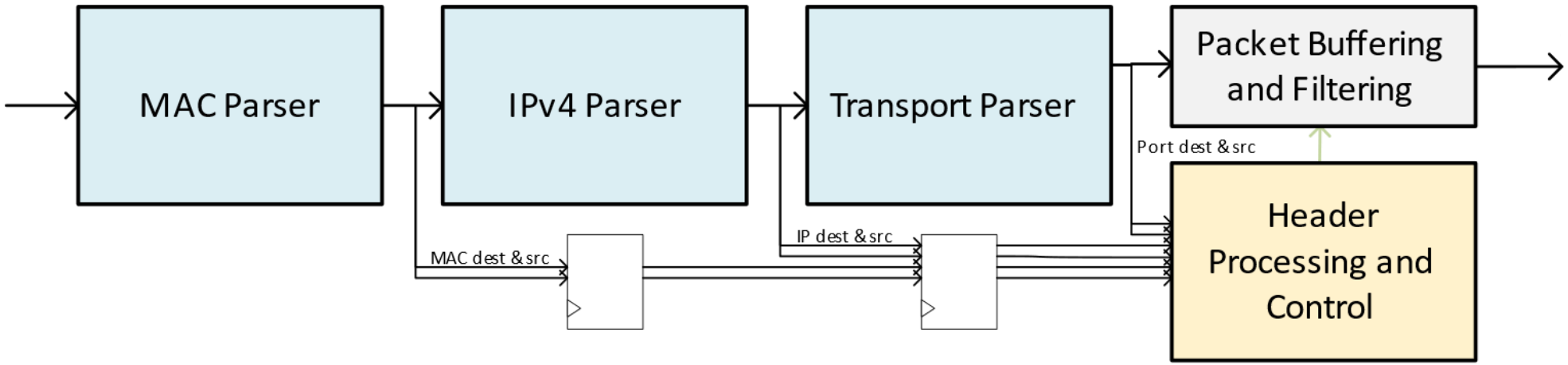
- Motivation for NMU
- NMU Architecture Types
- Our Hardware Implementation
- Evaluation of NMU Types
- Conclusions

Principal Hardware Sub-Components

- Three main reusable sub-components
  a) Packet Parsers
  b) Encapsulator/Tagger
  c) De-Encapsulator/De-Tagger

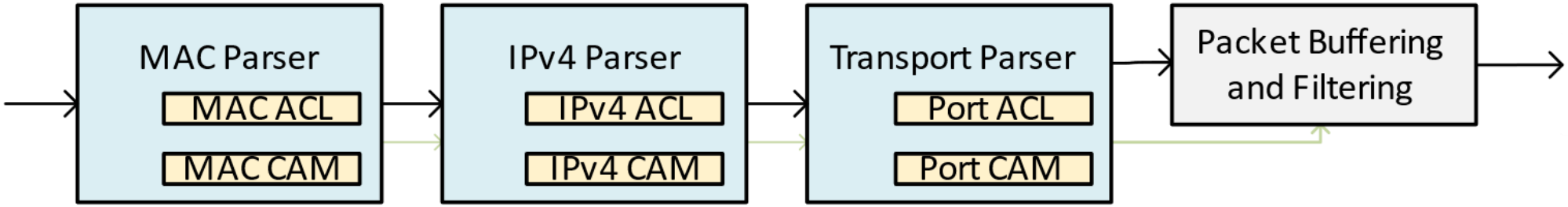
Traditional Packet Parsers

- Traditional packet parser system:



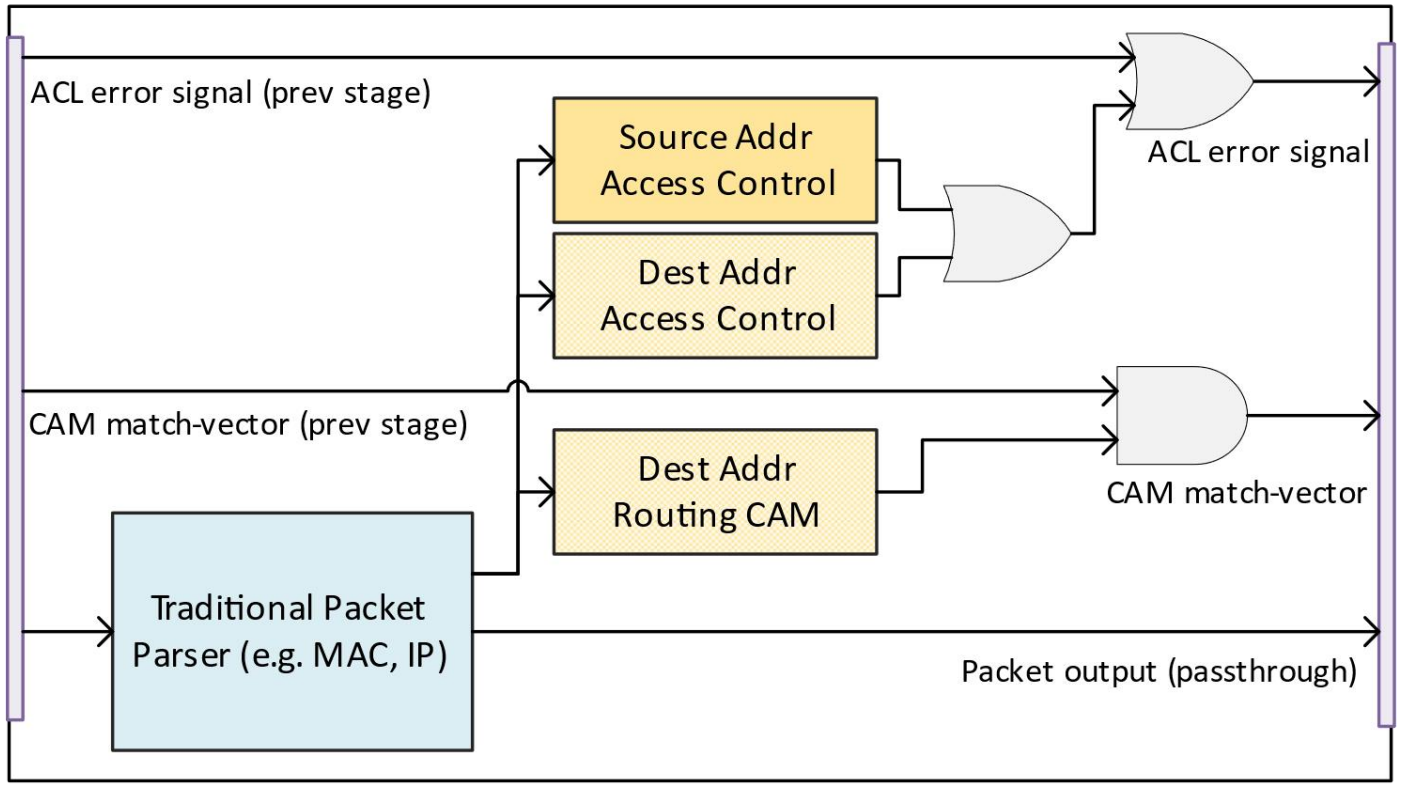
Our Packet Parsers

- Traditional packet parser, but with processing done in flight



Our Packet Parsers

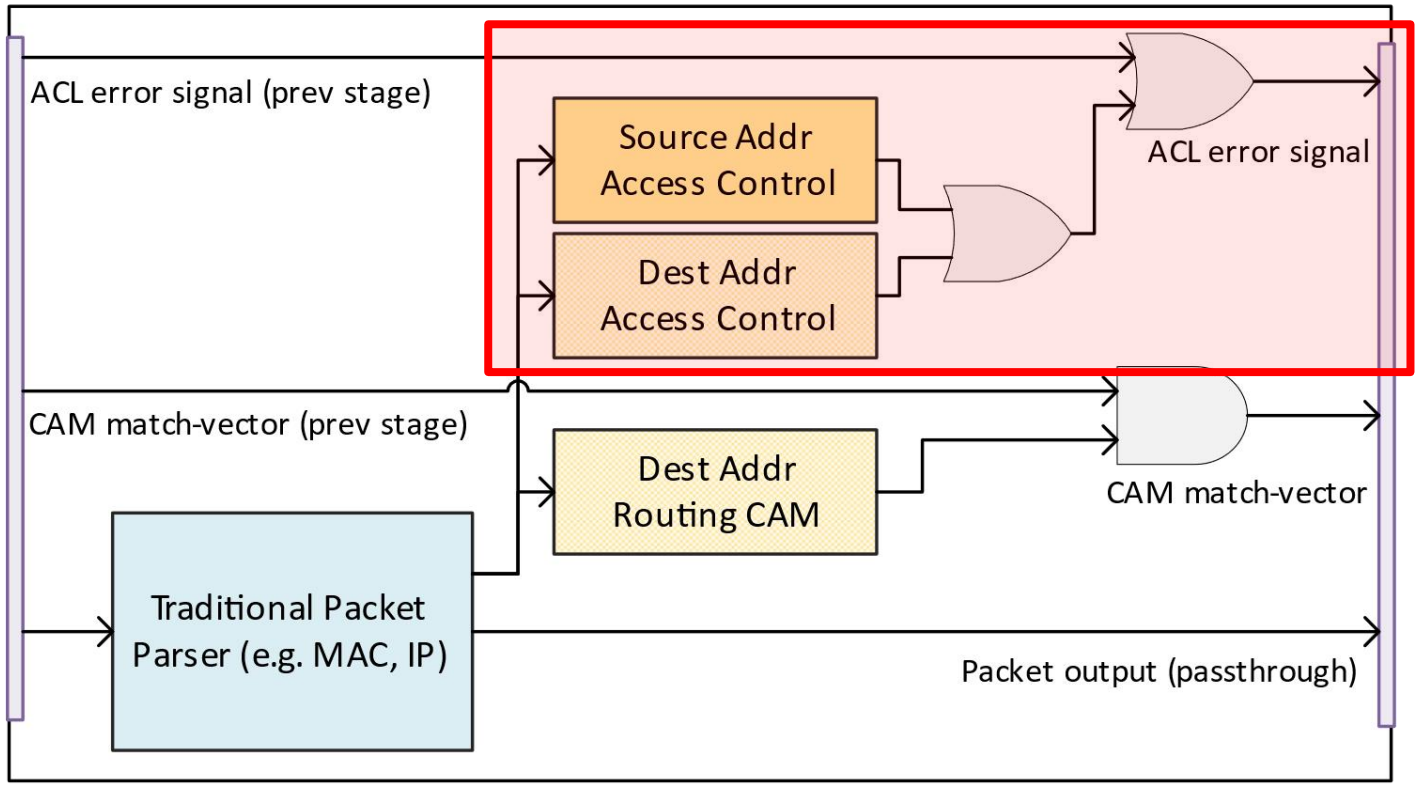
- Traditional packet parser, but with processing done in flight



Our Packet Parsers

- Traditional packet parser, but with processing done in flight

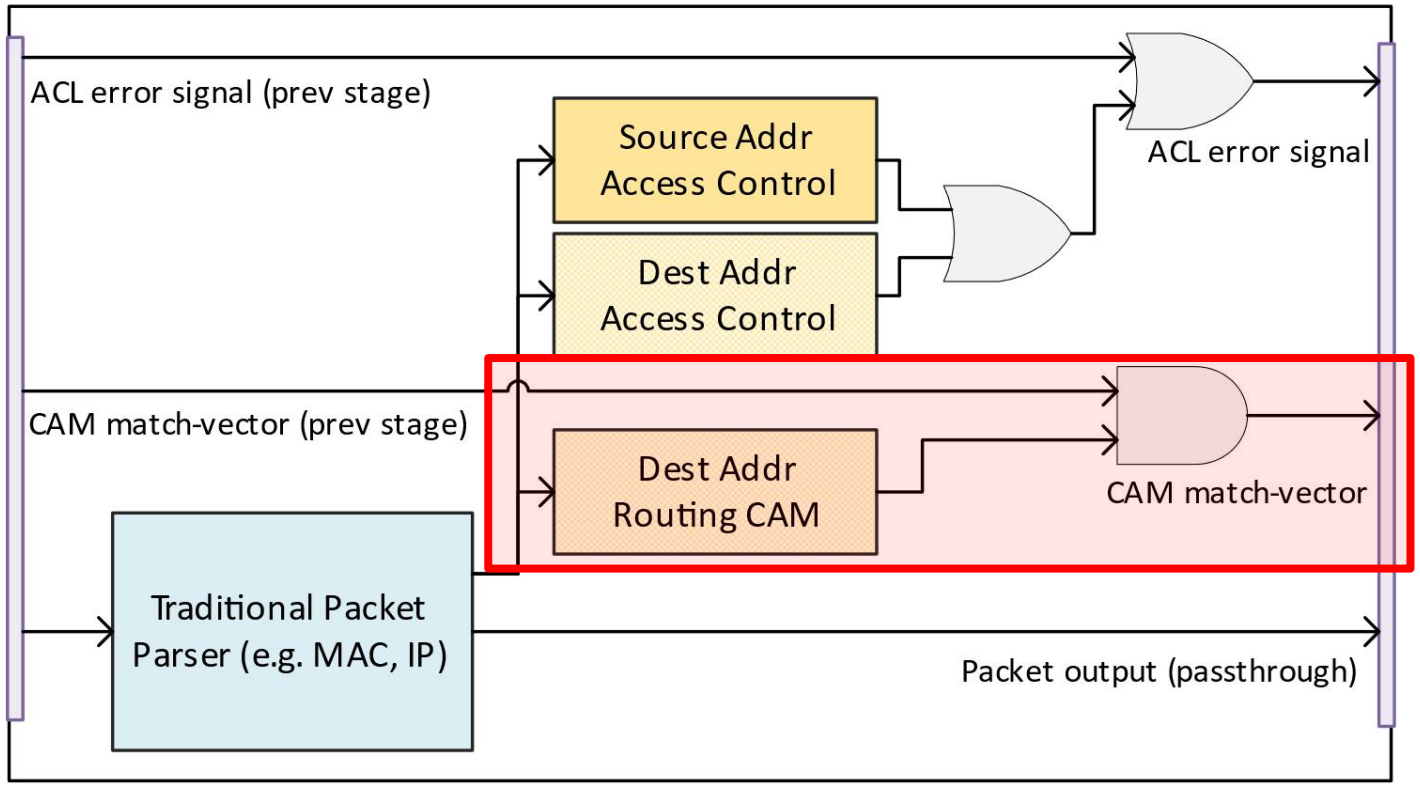
ACL error signal propagated



Our Packet Parsers

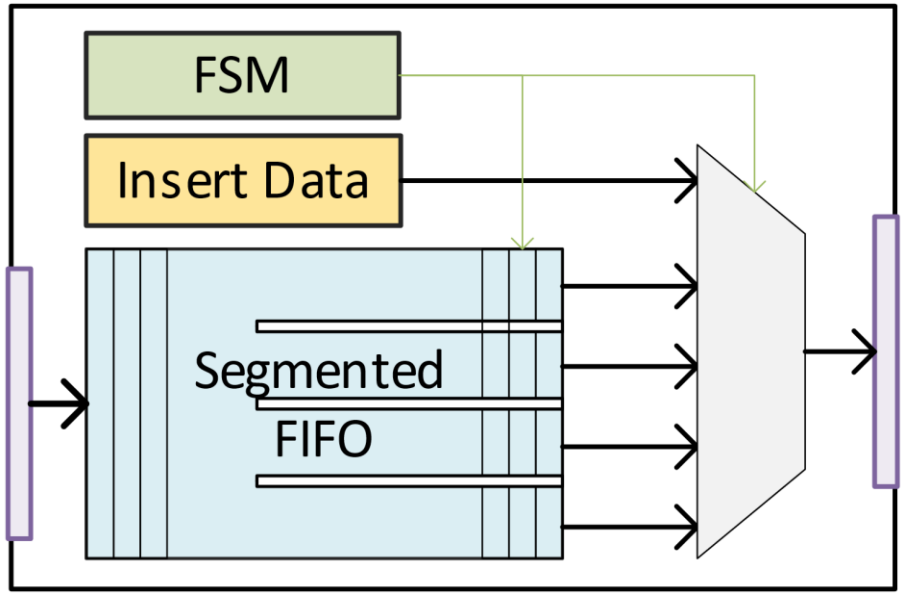
- Traditional packet parser, but with processing done in flight

Cumulative CAM match-vector propagated



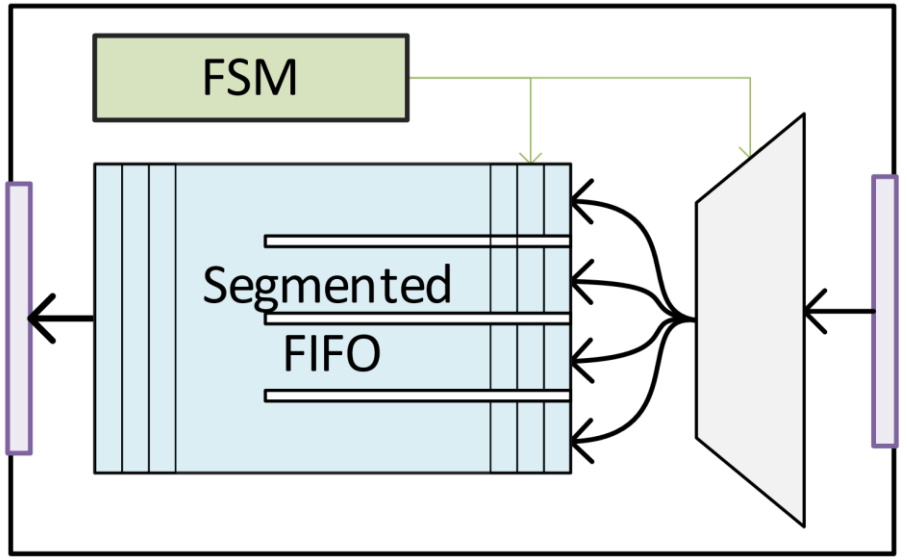
Encapsulators/Taggers

- Packet split into segment FIFOs, read out with inserted data



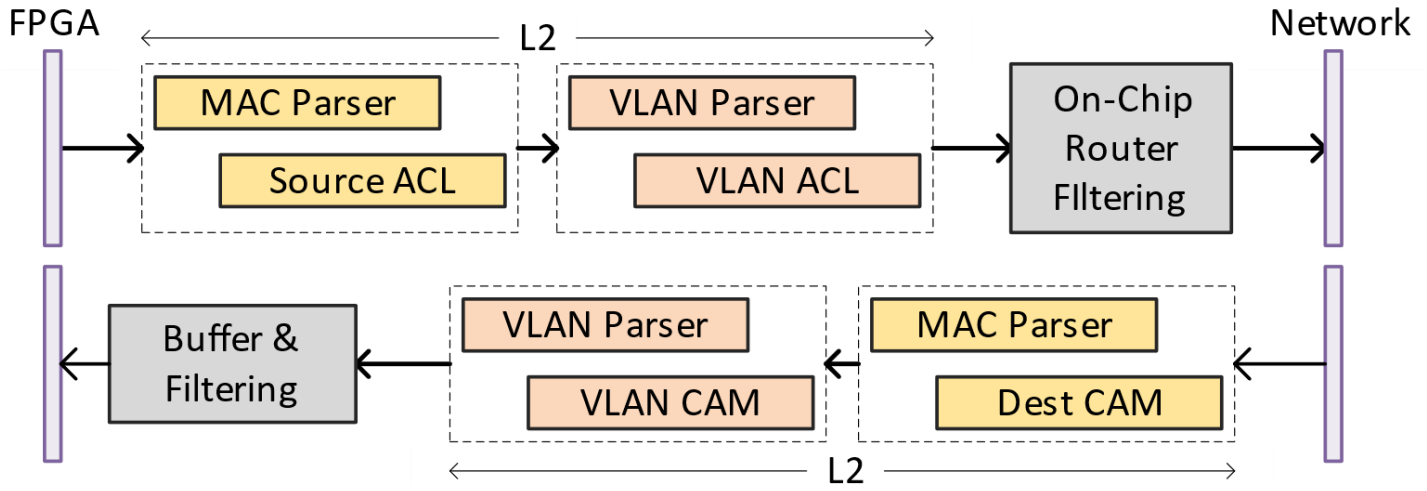
De-Encapsulators/De-Taggers

- Data to be removed from packet never inserted into FIFOs



Putting It Together

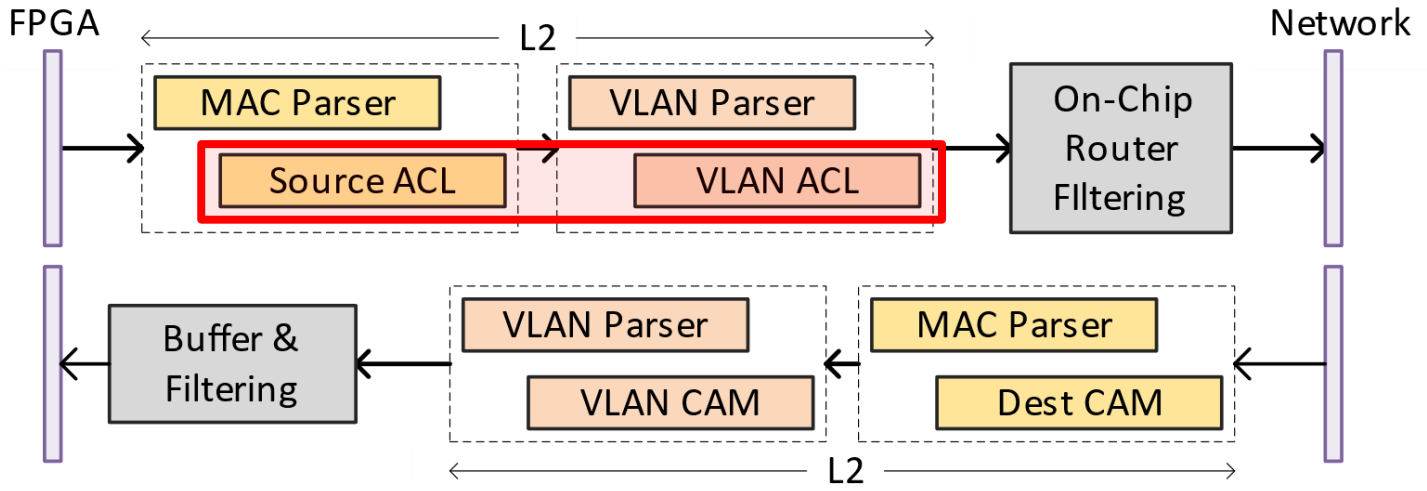
- Type B-L2 NMU (source ACLs, MAC layer processing)



Putting It Together

- Type B-L2 NMU (source ACLs, MAC layer processing)

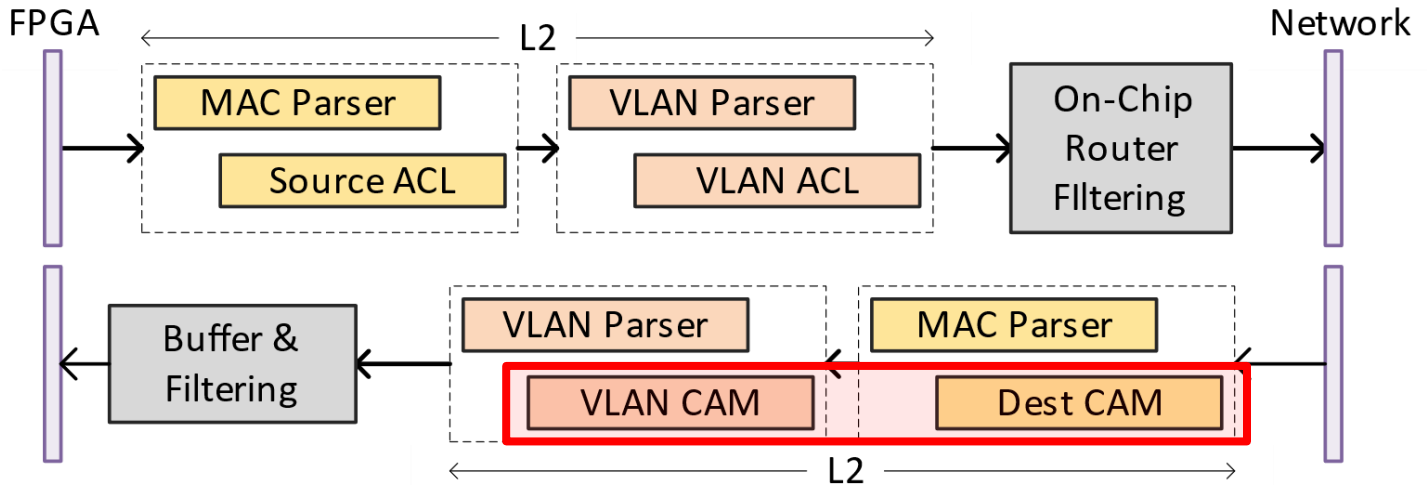
ACLs for source fields in Parsers



Putting It Together

- Type B-L2 NMU (source ACLs, MAC layer processing)

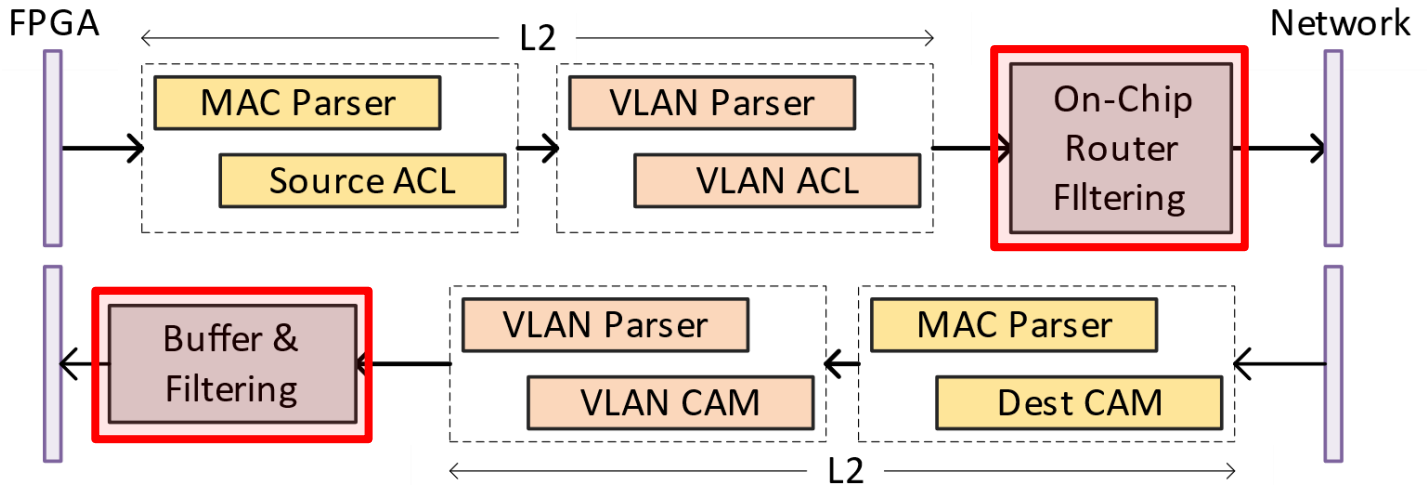
CAMs in ingress path for routing



Putting It Together

- Type B-L2 NMU (source ACLs, MAC layer processing)

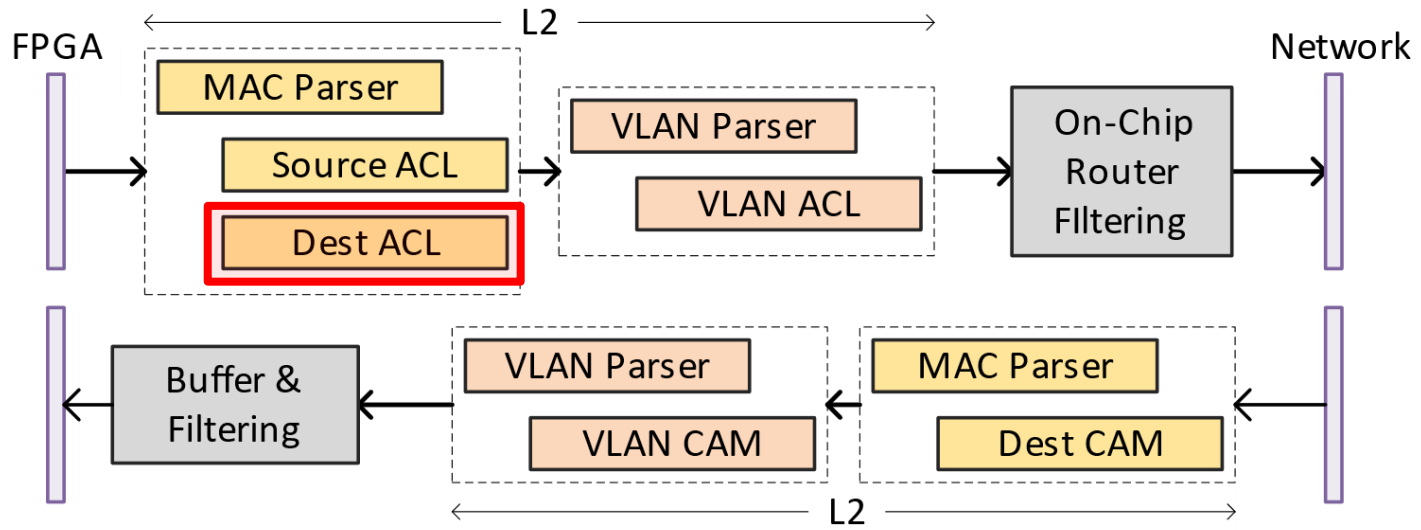
Filter packets with ACL error or no valid dest



Putting It Together

- Type C-L2 NMU (source & dest ACLs)

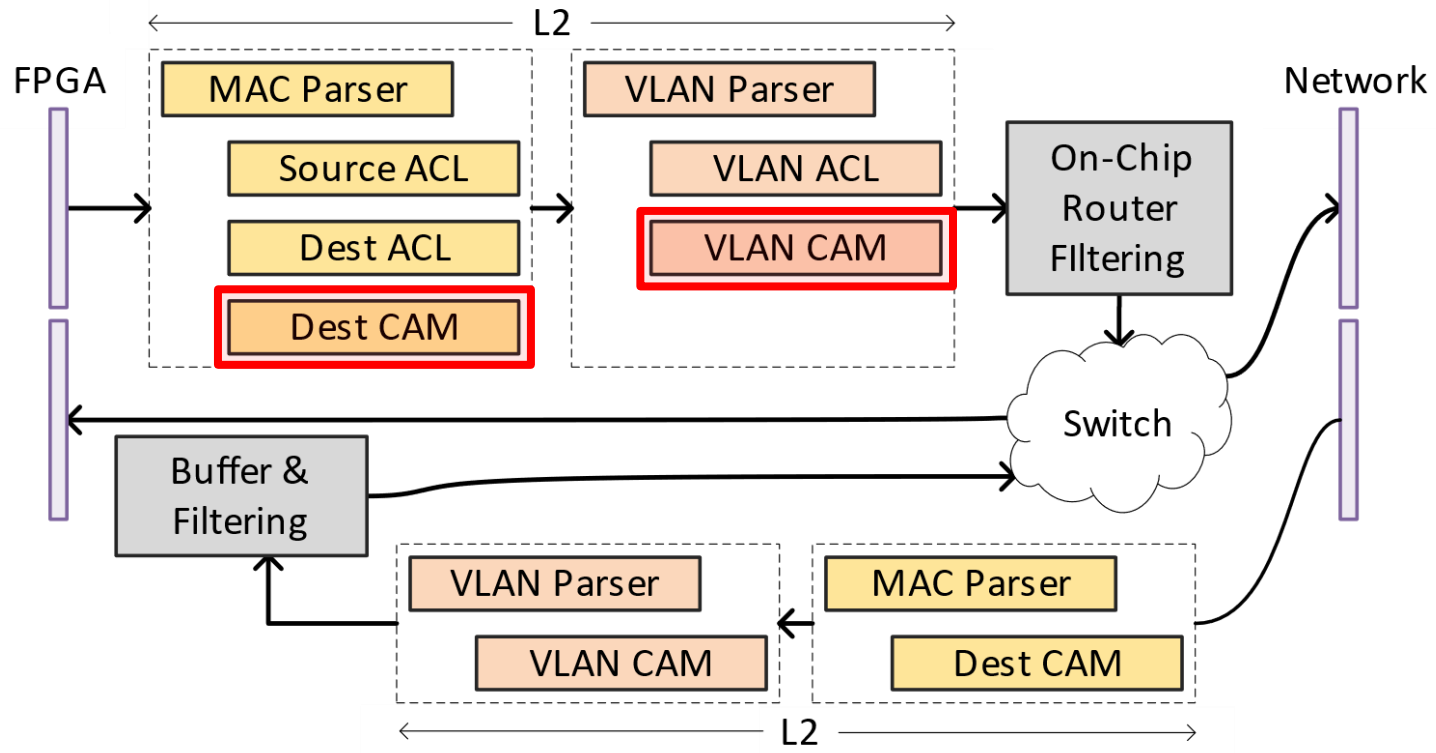
ACLs for dest. fields added



Putting It Together

- Type CR-L2 NMU (adding internal routing)

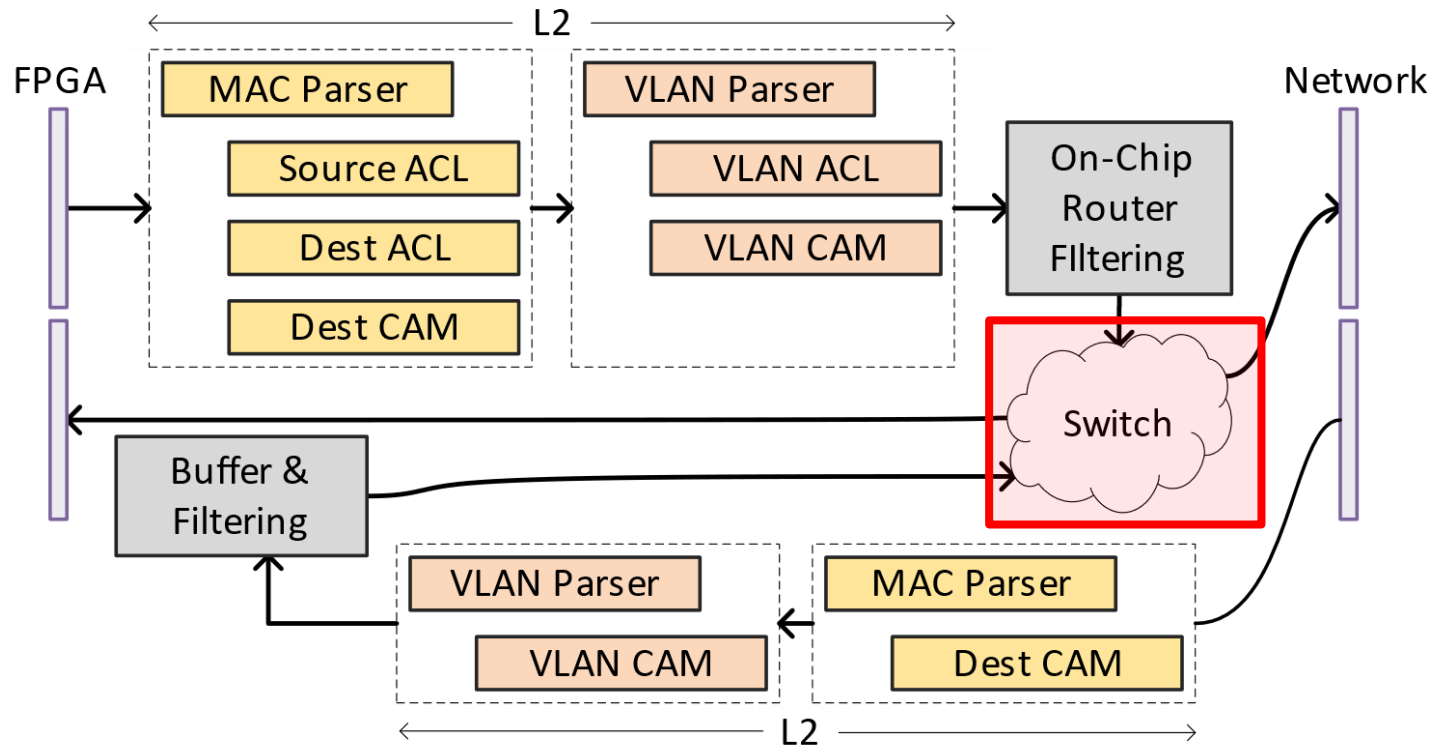
CAMs in egress path for internal routing



Putting It Together

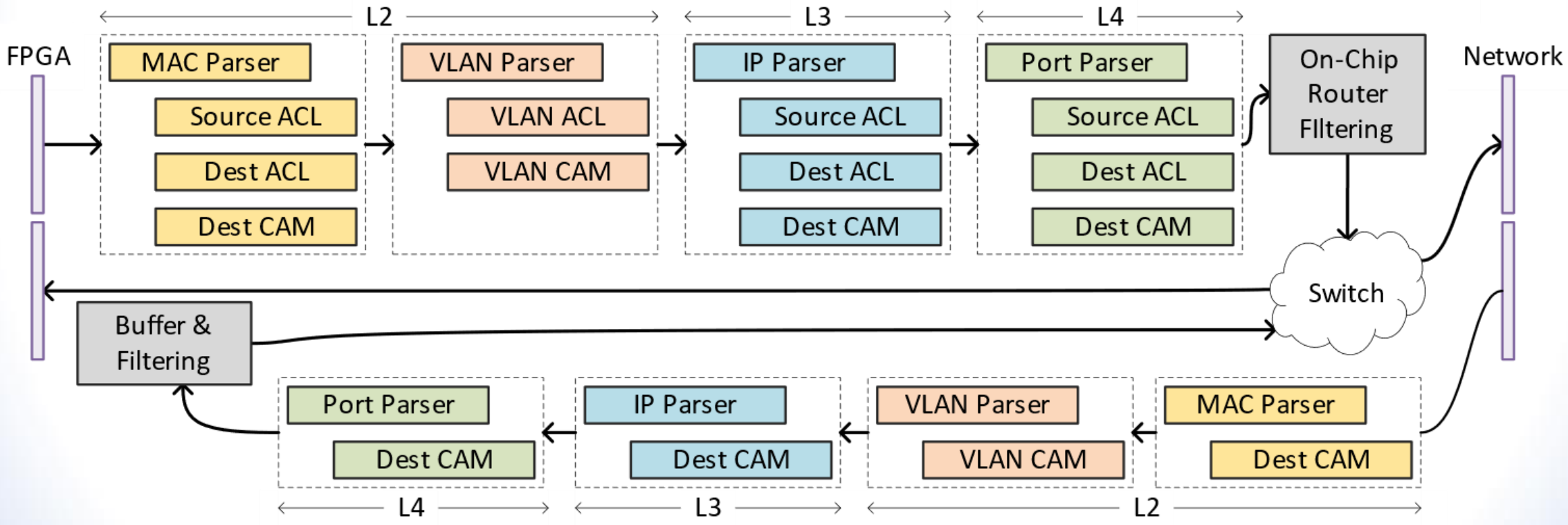
- Type CR-L2 NMU (adding internal routing)

AXI-Stream interconnect



Putting It Together

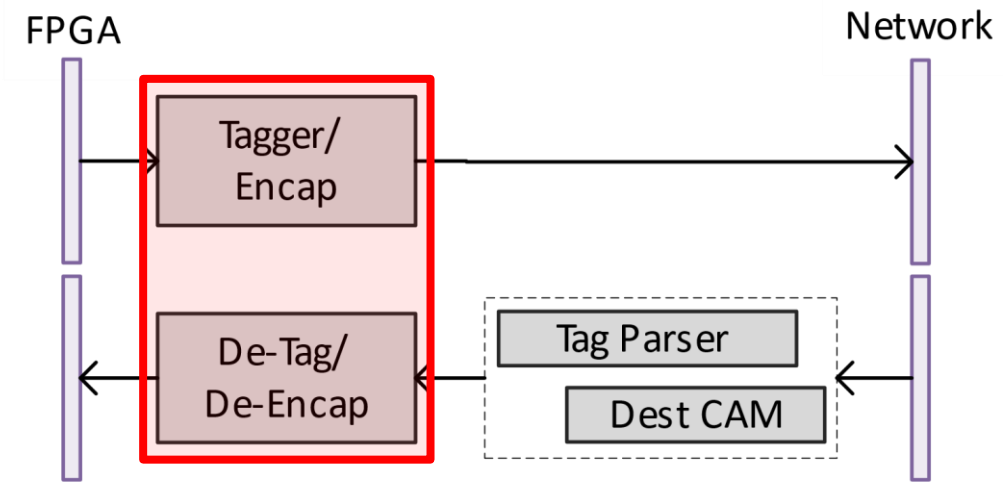
- Type CR-L4 NMU (expanding to layer 4 packet processing)



Putting It Together

- Type A (tagging) and Type E (encapsulation)

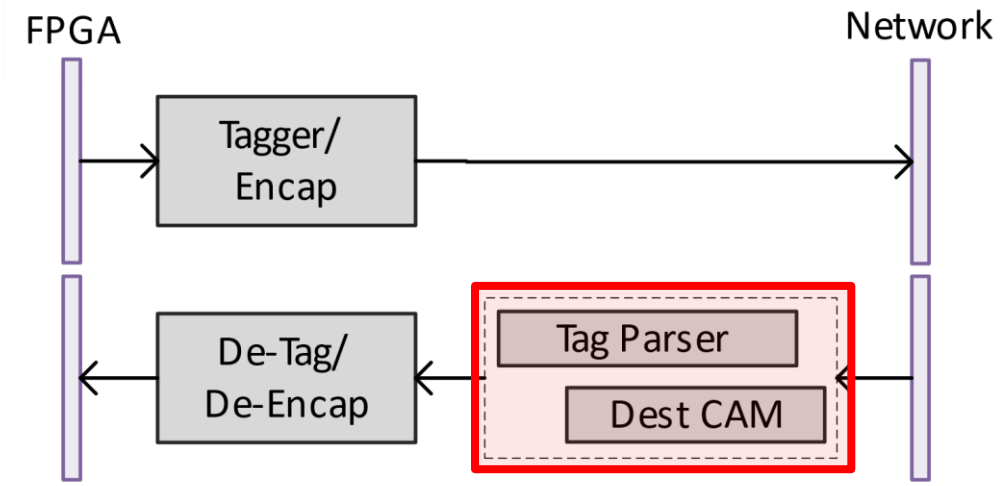
tag/encap on egress
de-tag/de-encap on ingress



Putting It Together

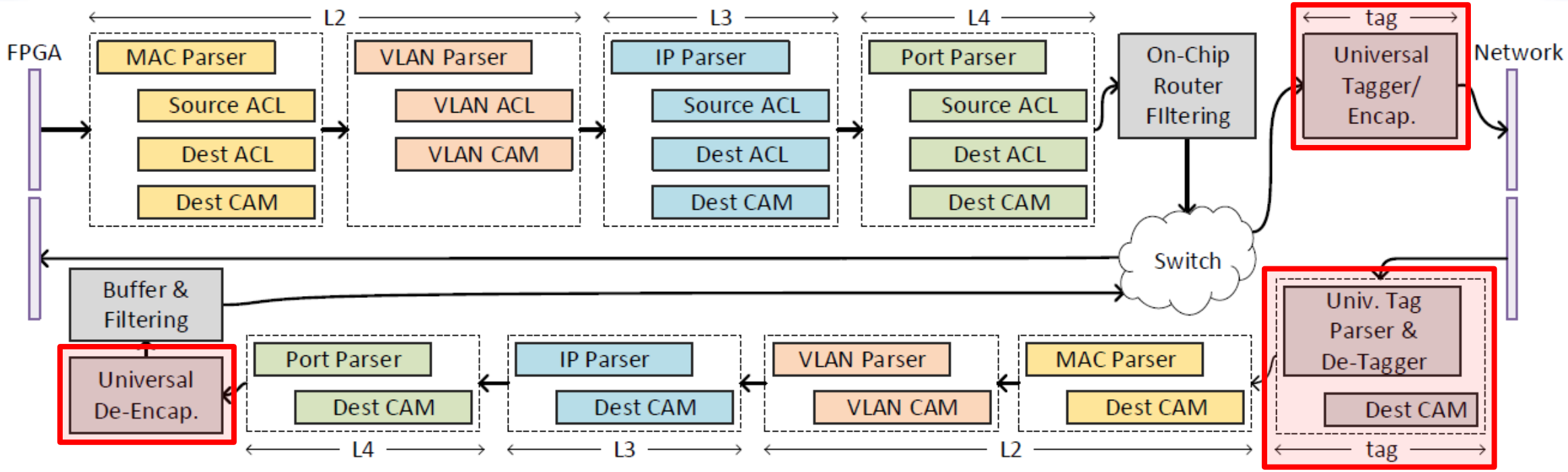
- Type A (tagging) and Type E (encapsulation)

CAMs in ingress path for routing



Putting It Together

- The Universal NMU



Add encap/de-encap components to L4 NMU architecture

Multi-Tenant Considerations

- Can have multiple applications on one FPGA
  - NMU needs to secure multiple logical connections separately
  - We implement NMUs with 32 logical connections

Outline

- Motivation for NMU
- NMU Architecture Types
- Our Hardware Implementation
- Evaluation of NMU Types
- Conclusions

Evaluation Setup

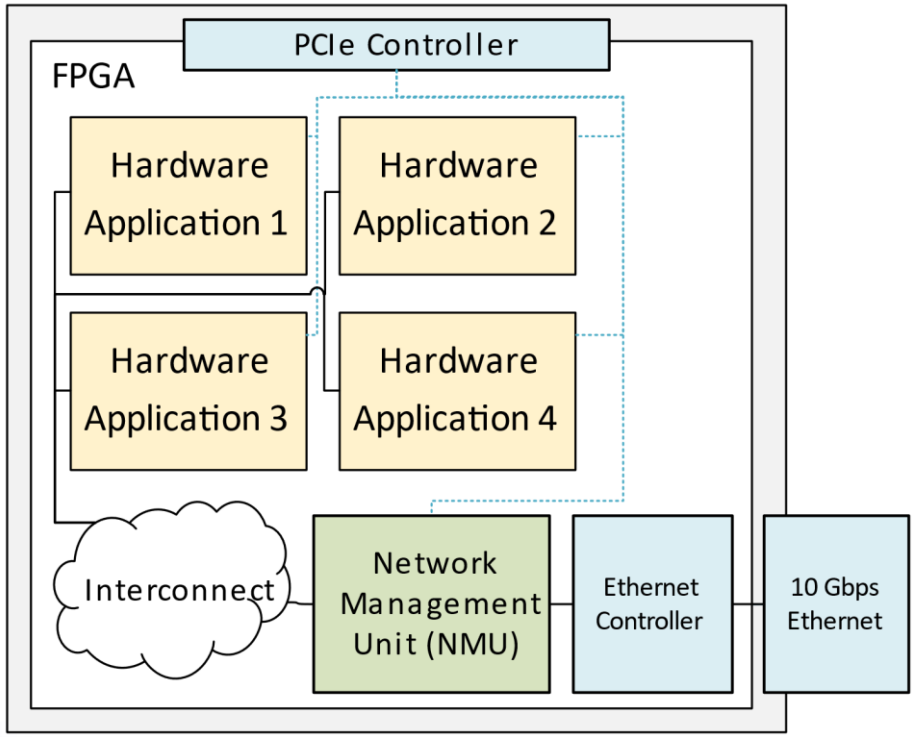
- What qualities of NMUs characterize its performance?

Evaluation Setup

- What qualities of NMUs characterize its performance?
  - ❖ Throughput (10Gbps line-rate, no need to measure)
  - ❖ Area (need to measure LUT/FF utilization)
  - ❖ Latency (need to measure cycles added in ingress/egress path)
    - freq. =156.25 MHz (freq. of Ethernet controller)

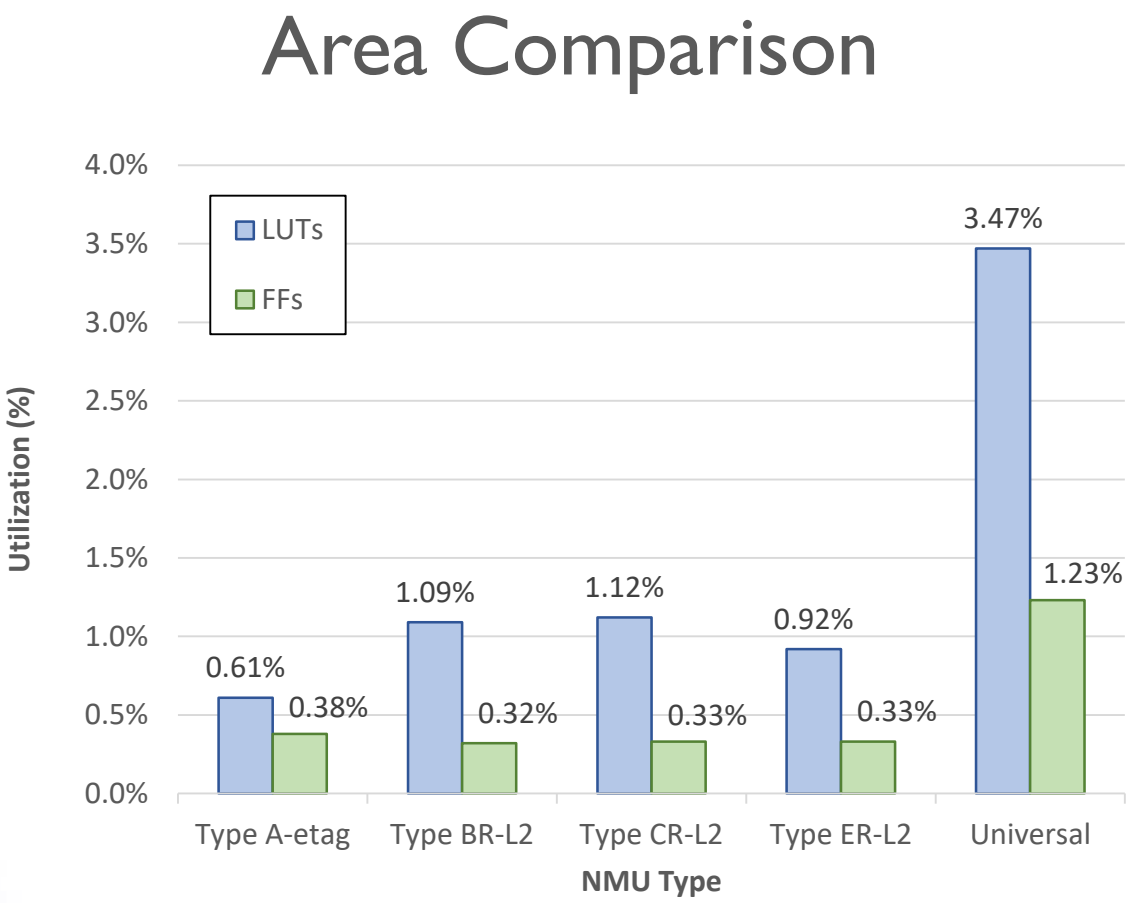
Evaluation Setup

- Four simple hardware applications on one FPGA

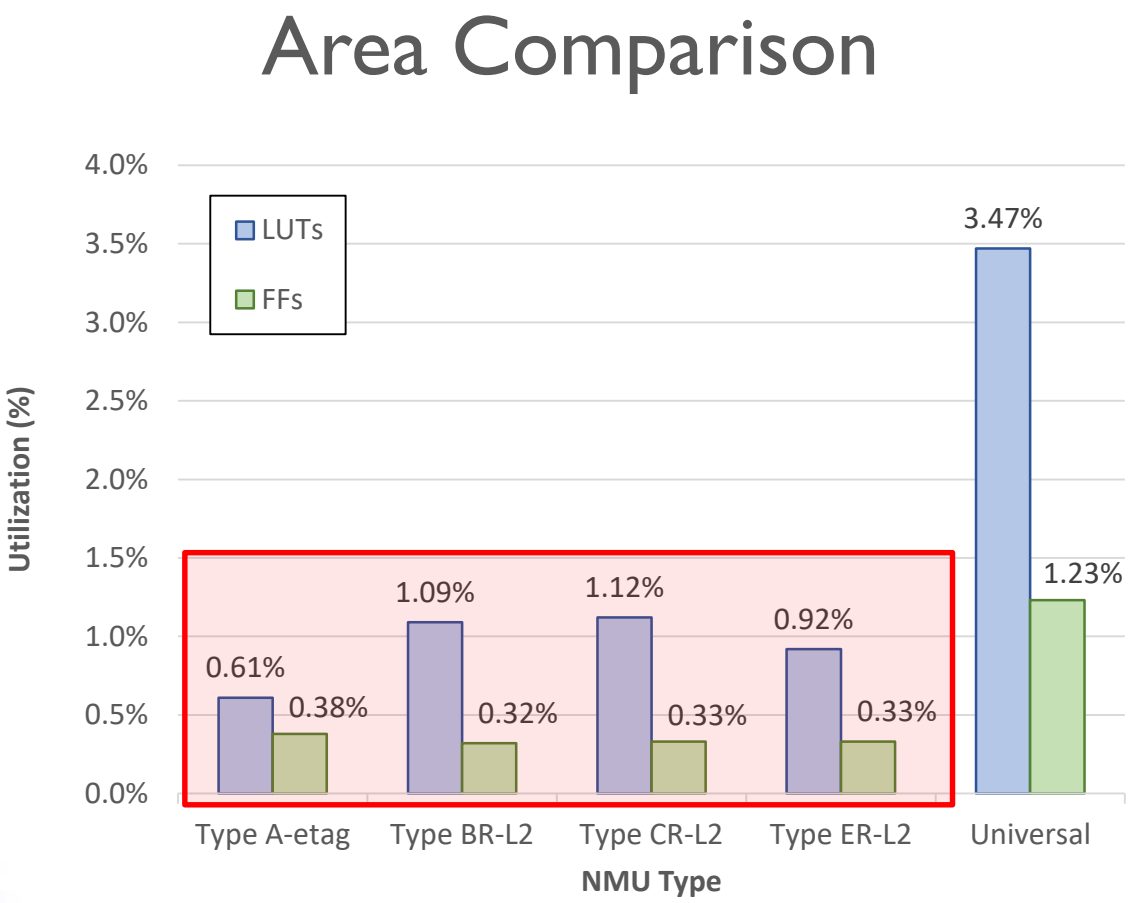


- Configured over PCIe
- 32 logical connections
  - 4 applications x 8 connections
- Kintex Ultrascale XCKU115

NMU Evaluation – Access Control Type

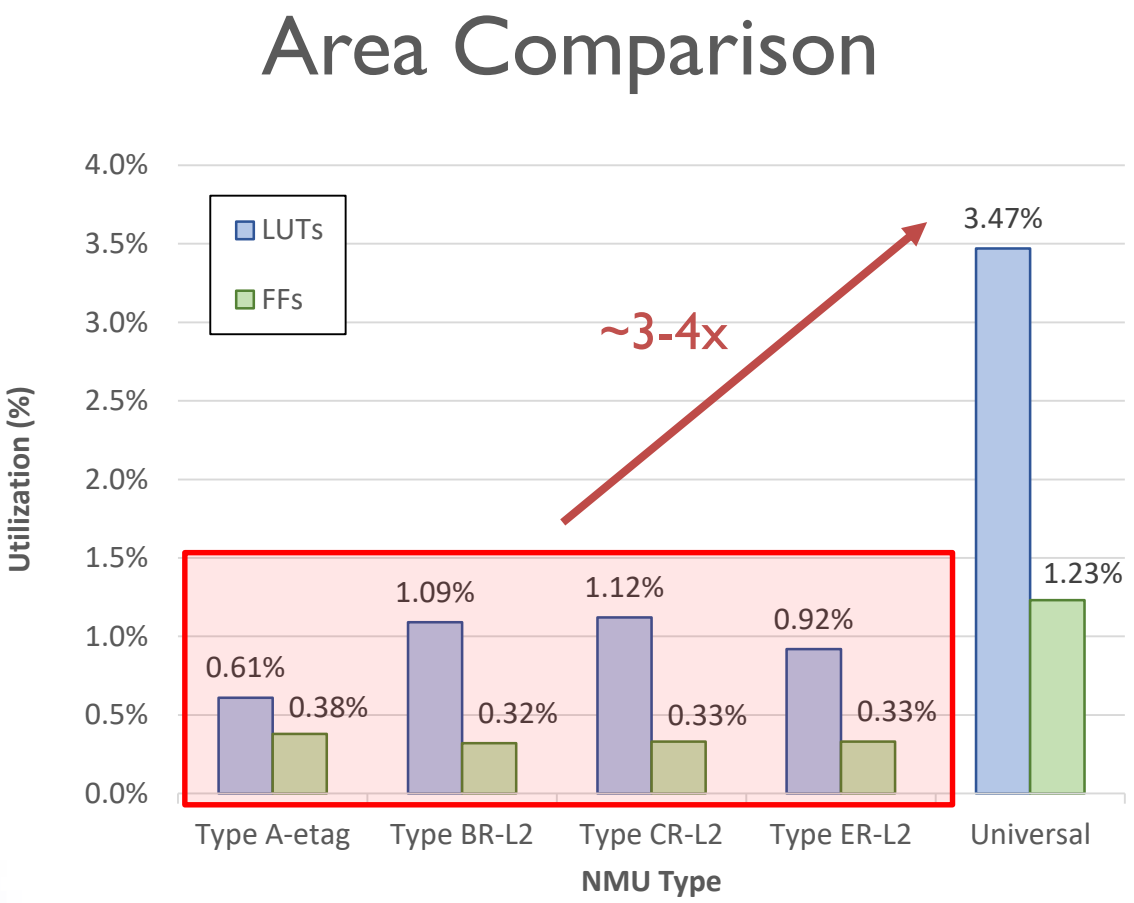


NMU Evaluation – Access Control Type



Not much difference in utilization between NMU Types

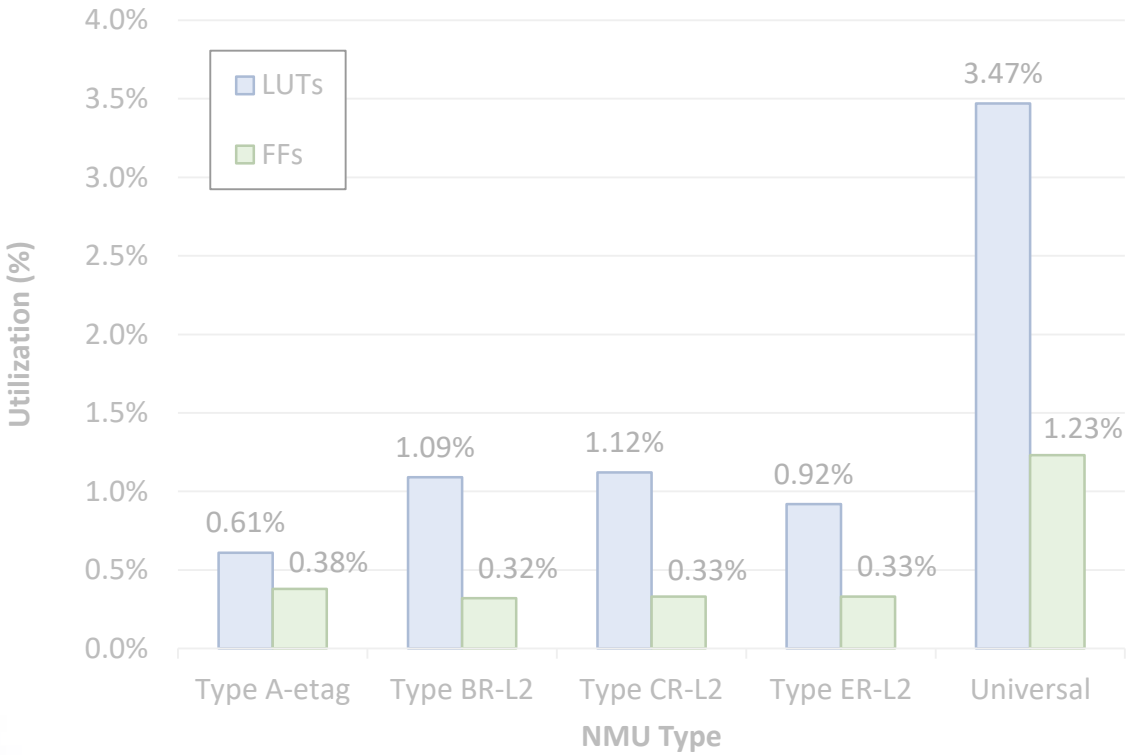
NMU Evaluation – Access Control Type



Overhead of Universal NMU is about 3-4x (but still small)

NMU Evaluation – Access Control Type

Area Comparison



Latency (cycles)

	Egress	Ingress
Type A-etag	1	4-6
Type BR-L2	5-10	6-8
Type CR-L2	5-10	6-8
Type ER-L2	6-7	8-10
Universal	13-18	19-25

Impact on latency of Universal NMU is more pronounced

NMU Evaluation – Routability

Area Comparison:

	Without Routing		With Routing		Overhead	
	LUTs	FFs	LUTs	FFs	LUTs	FFs
Type B-L2	3516	2883	7199	4311	2.04x	1.50x
Type C-L2	3687	2867	7424	4378	2.01x	1.53x
Type E-L2	3392	3113	6133	4316	1.81x	1.39x

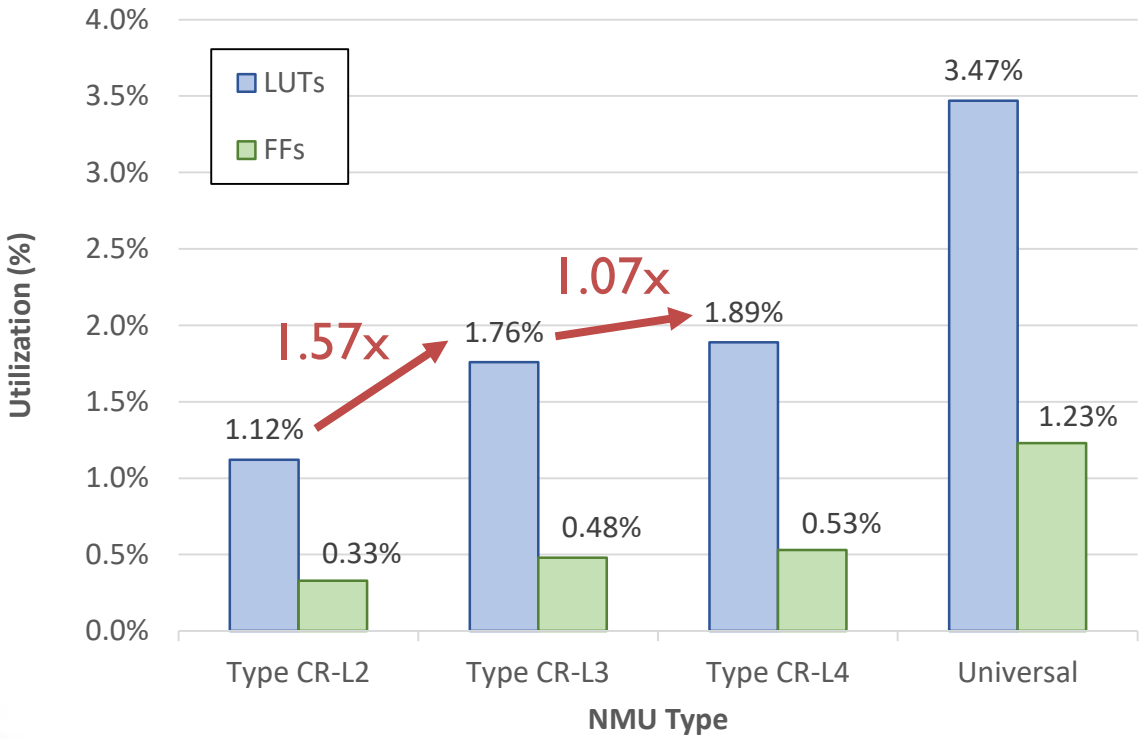
NMU Evaluation – Routability

Latency Comparison (in cycles):

	Without Routing		With Routing		Overhead	
	Egress	**Ingress**	**Egress**	**Ingress**	**Egress**	**Ingress**
Type B-L2	1-6	2-4	5-10	6-8	4 cycles	4 cycles
Type C-L2	1-6	2-4	5-10	6-8	4 cycles	4 cycles
Type E-L2	1	4-6	6-7	8-10	5-6 cyc.	4 cycles

NMU Evaluation – Network Layer

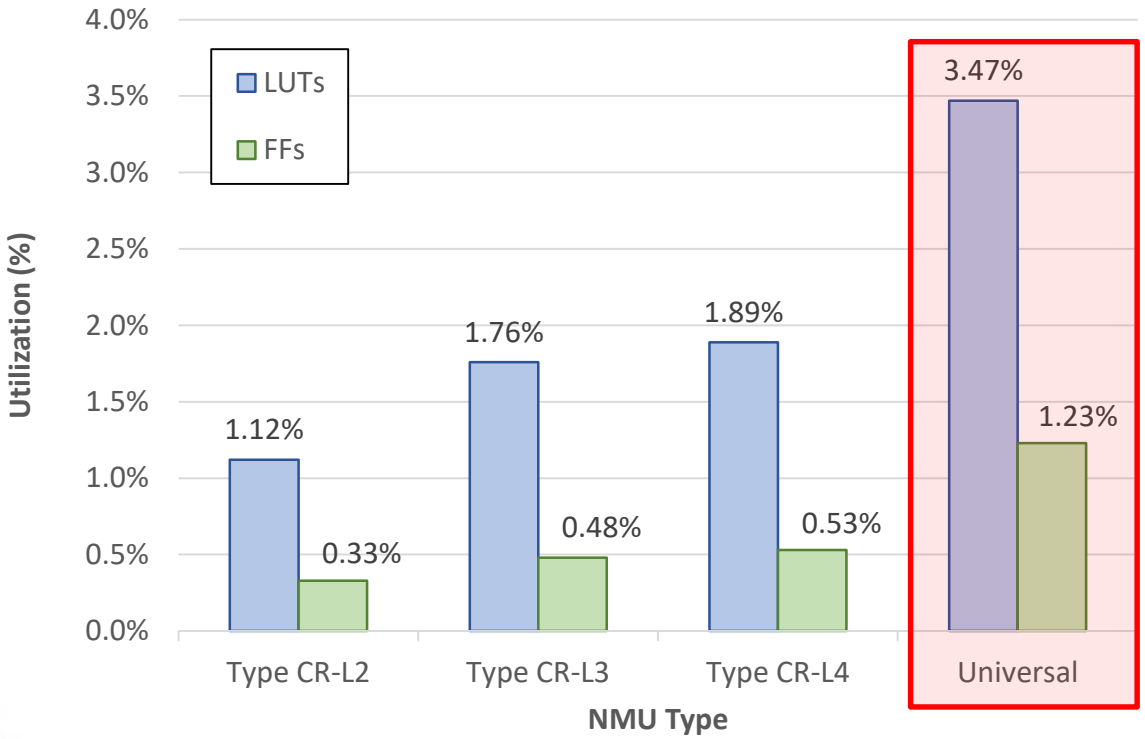
Area Comparison



Overhead of IP Layer inspection significant, but not significant for Transport Layer

NMU Evaluation – Network Layer

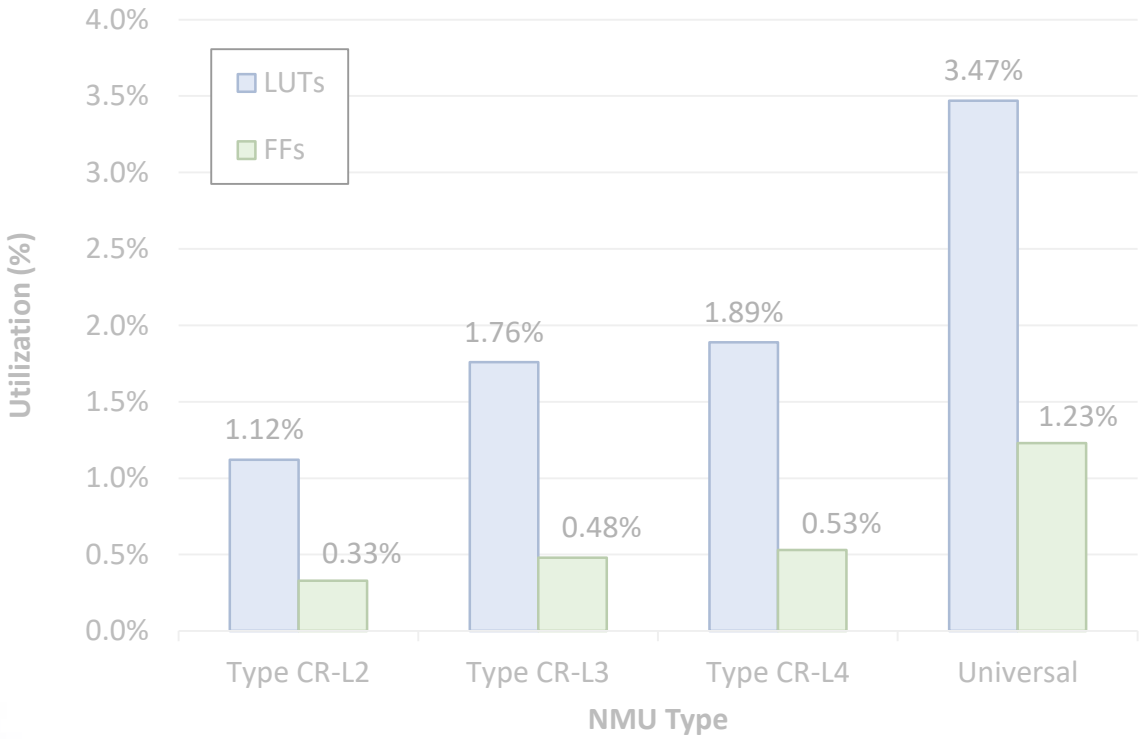
Area Comparison



Universal NMU overhead is still high, 1.8-2x

NMU Evaluation – Network Layer

Area Comparison



Latency (cycles)

	Egress	Ingress
Type CR-L2	5-10	6-8
Type CR-L3	6-11	7-12
Type CR-L4	6-11	7-12

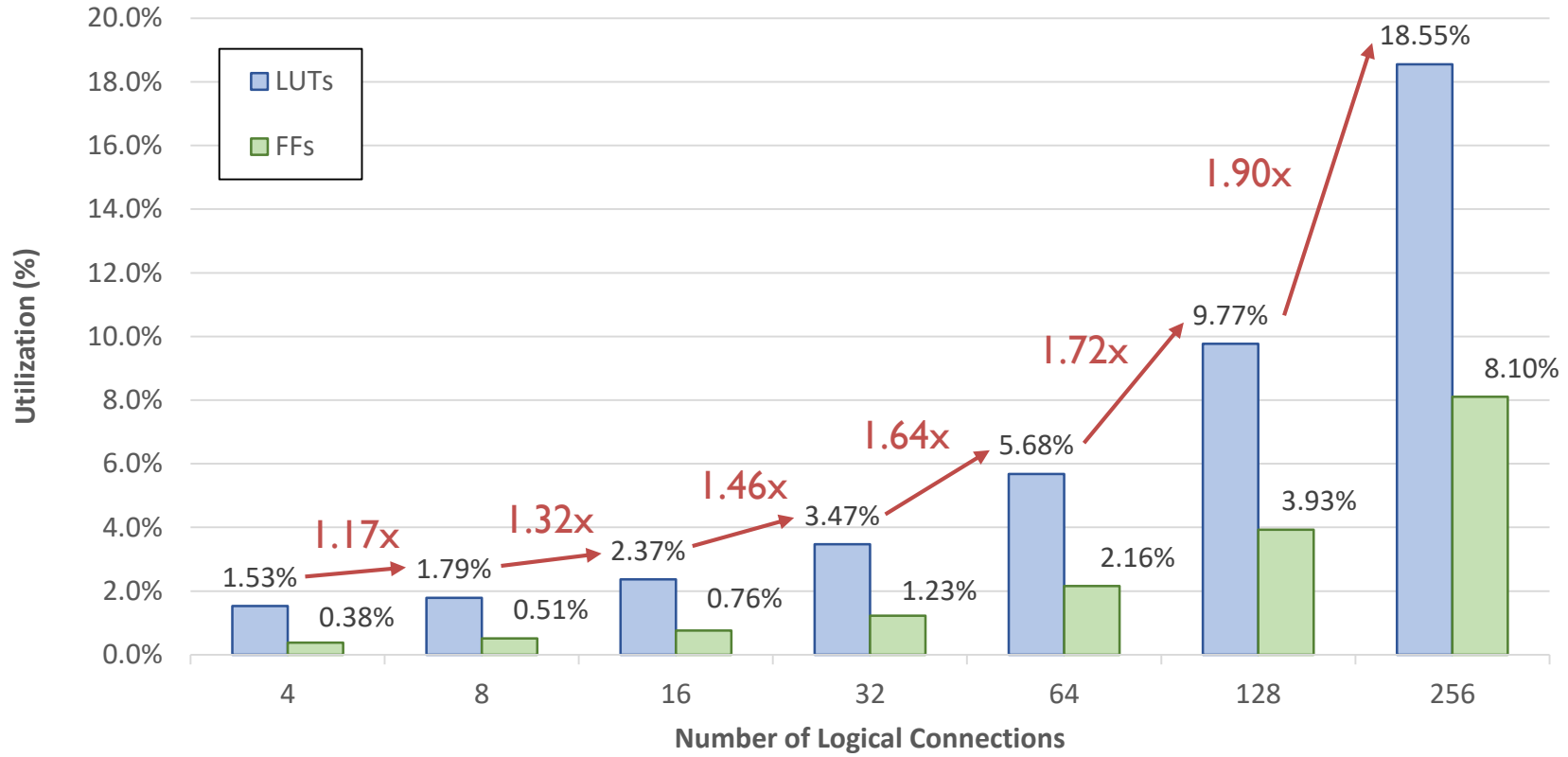
Not much difference in latency

NMU Variety Evaluation Summary

- Area
  - Routability has biggest impact on area utilization
  - Jumping from MAC to IP processing also has a big impact, though the jump from IP to Transport protocol is less severe
  - All implementations have low area overhead
- Latency
  - Routability has single biggest impact on latency as well
  - Universal NMU has a big latency hit

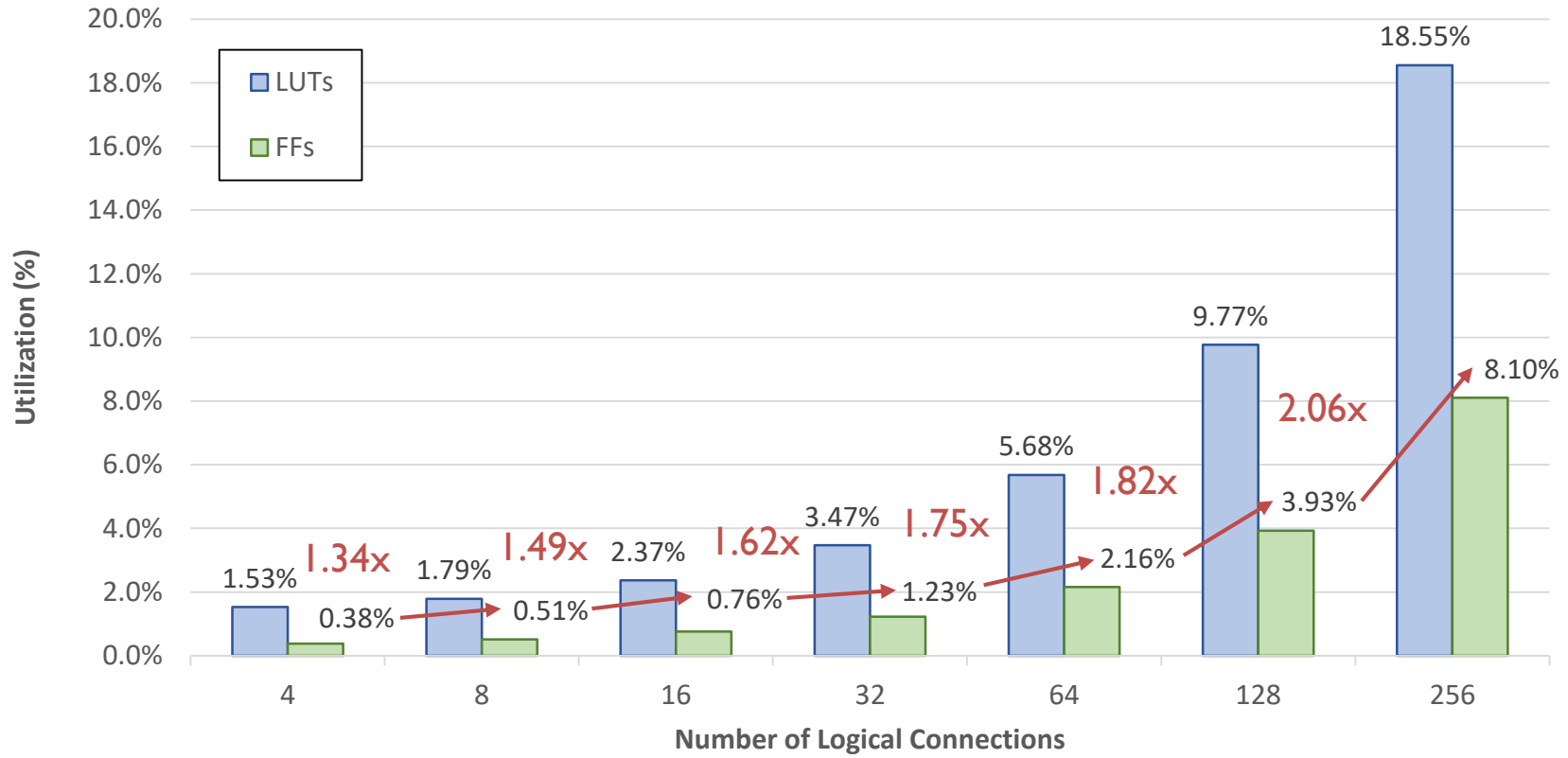
Universal NMU Scalability

- Scaling Number of Logical Connections



Universal NMU Scalability

- Scaling Number of Logical Connections



Outline

- Motivation for NMU
- NMU Architecture Types
- Our Hardware Implementation
- Evaluation of NMU Types
- Conclusions

Conclusions

- The NMU is a low overhead network security solution for direct-connected FPGAs, across many configurations
- Differences between NMU configurations are quite small, though Universal NMU does add significantly more latency
- Universal NMU can scale to 256 connections, with area hit
- Universal NMU effectively implements all NMU functionalities identified, may be candidate for hardening

Acknowledgments

We'd like to thank and acknowledge Xilinx, Hauwei, and NSERC for the funding, material, and support provided for this project

Questions?

rozhkoda@eecg.toronto.edu