Athena: Add More Intelligence to RMT-based Network Data Plane with Low-bit Quantization

Yunkun Liao, Hanyue Lin, Jingya Wu, Wenyan Lu, Huawei Li, Xiaowei Li, Guihai Yan

SKLP, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China

2024/12/3

EURO-PAR CONFERENCE 2024

ADAPT Architecture for Data Analytics and Processing Technology

SKLP 处理器芯片全国重点实验室 STATE KEY LAB OF PROCESSORS, ICT, CAS

ICT 中国科学院计算技术研究所 INSTITUTE OF COMPUTING TECHNOLOGY, CHINESE ACADEMY OF SCIENCES

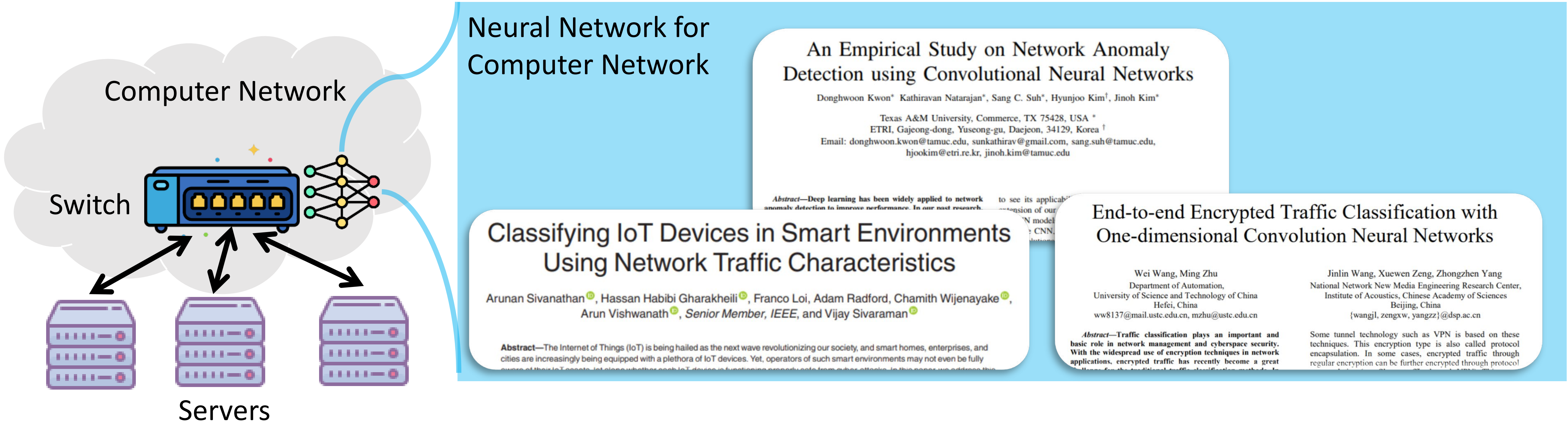
中国科学院大学 University of Chinese Academy of Sciences

中科驭数 YUSUR

Contens

Background and Motivation

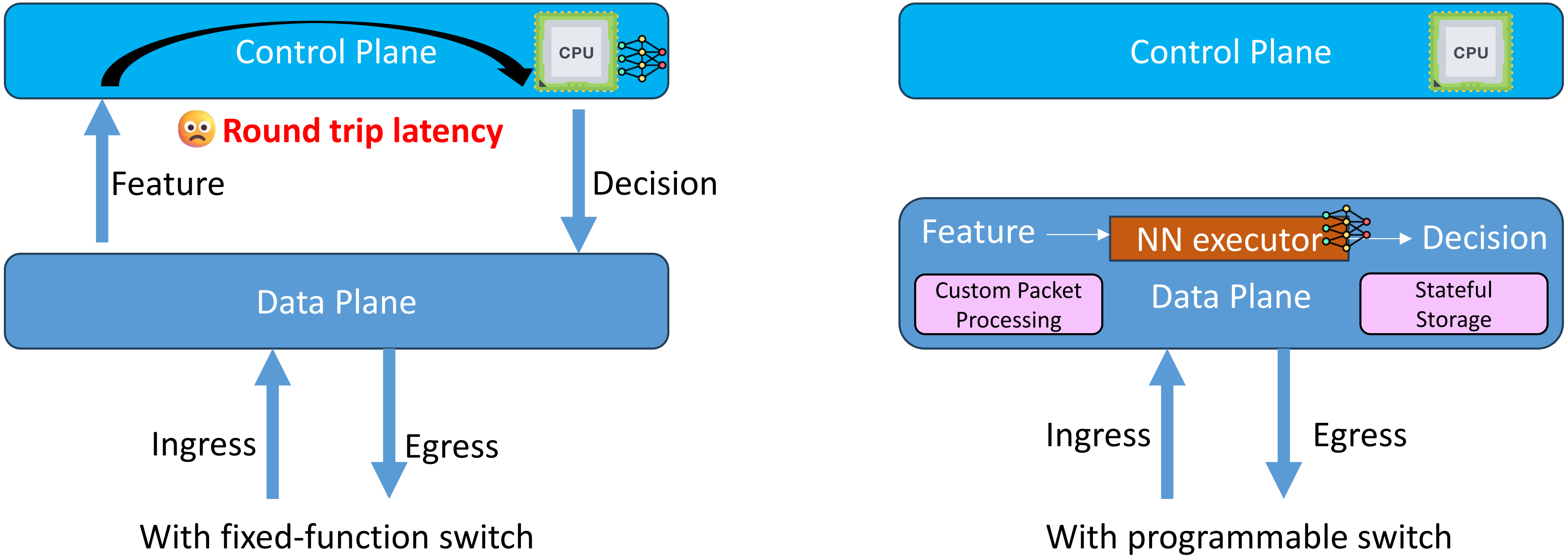
Neural Network for Computer Network



Neural Network (NN) has been used for several **network traffic analysis** tasks: anomaly detection, traffic classification, …

- Enable “end-to-end” learning.
- identify complex patterns from network packets.
- Make better decision than handcrafted heuristics.

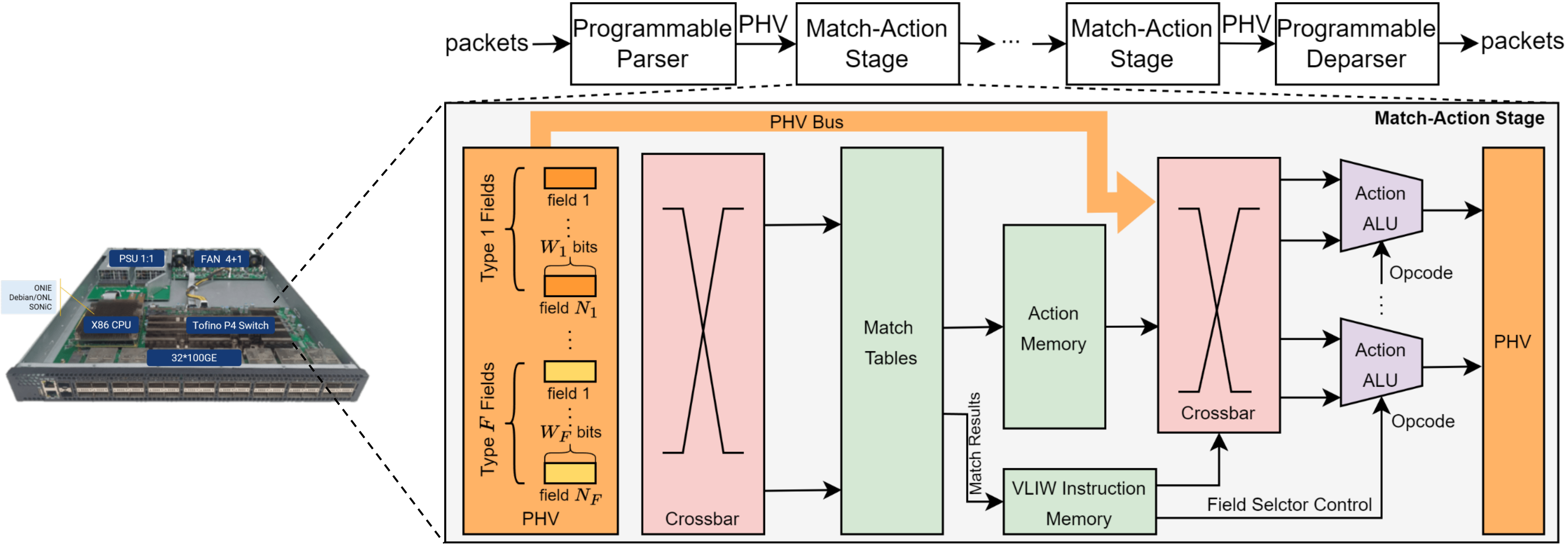
NN Inference on the Programmable Switch



Programmable switch can run NN inference within the network data plane

High throughput, Low latency

RMT Pipeline Architecture



1. Reconfigurable Match-Action Table (RMT) architecture forms the data plane of programmable switch.
2. RMT architecture is composed of multiple pipelined match-action stages.
3. Each stage has multiple Action ALU that accepts Packet Header Vector (PHV) fields as operands and modifies PHV field.
4. The bit width of PHV Field is limited and static.

Challenge: The Computing Power of RMT

Category	Description
logical	and, or, xor, not, ...
shadd/sub	signed or unsigned shift
arith	inc, dec, min, max
deposit-byte	any length, source & dest offset
rot-mask-merge	IPv4 $\leftrightarrow$ IPv6 translation uses
bitmasked-set	$S_1 \& S_2 \mid \overline{S_1} \& S_3$; metadata uses
move	if V_{S_1} $S_1 \rightarrow D$
cond-move	if $\overline{V_{S_2}} \& V_{S_1}$ $S_1 \rightarrow D$
cond-mux	if V_{S_2} $S_2 \rightarrow D$ else if V_{S_1} $S_1 \rightarrow D$

Table 1: Partial action instruction set.
(S_i means source i; V_x means x is valid.)

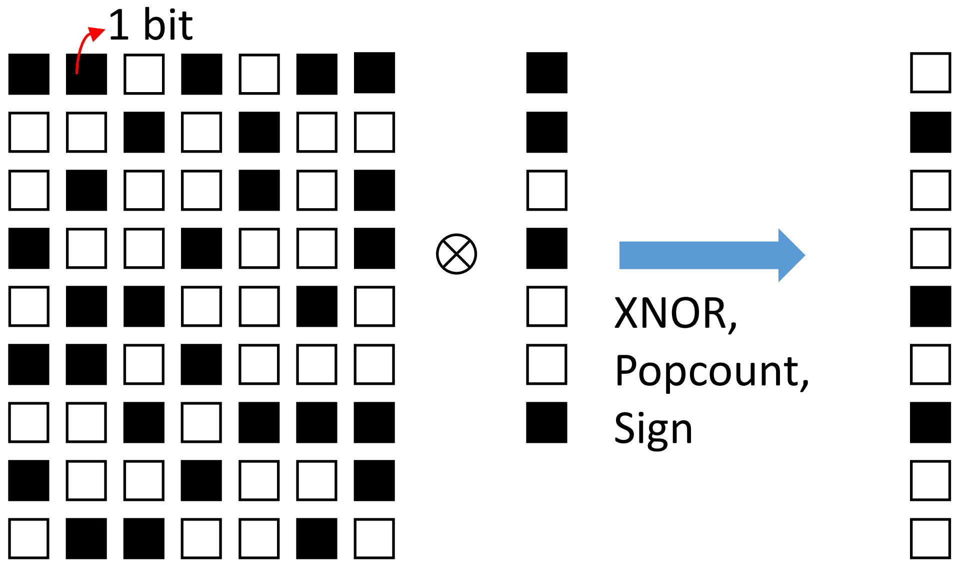
Action ALU

- Prefers bit operations
- No floating point
- No INT8, INT4, INT2 fixed point

RMT provides limited computing power for fixed-point Neural Network inference.

Prior Solutions for the Challenge

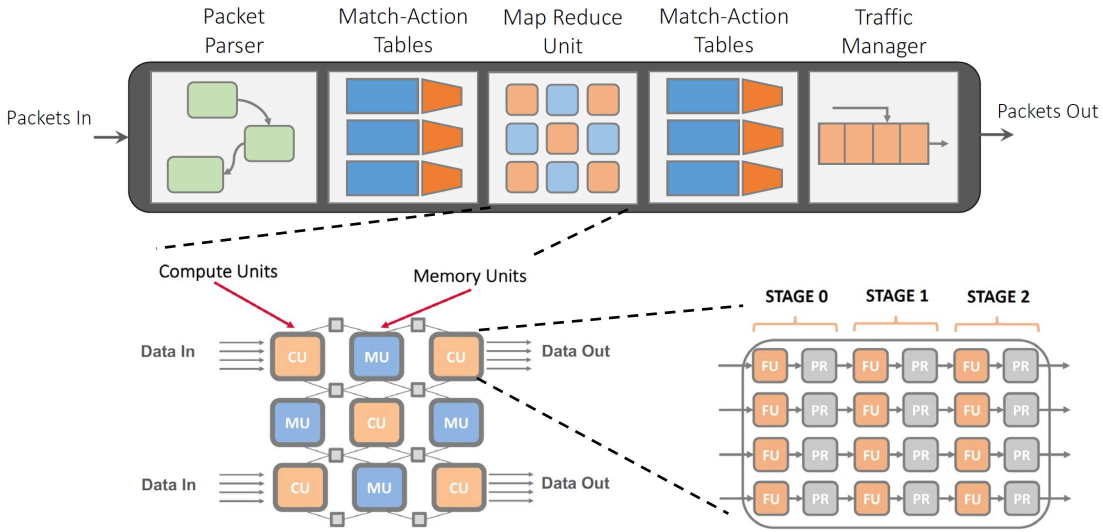
1. Reduce Model Complexity by Binary Neural Network
   - ➢ Represented work: N3IC[1]



- XNOR, popcount and sign for BNN inference
- Operations supported by RMT.

Low model accuracy

2. Augment RMT with off-pipeline accelerator
   - ➢ Represented work: Taurus[2]

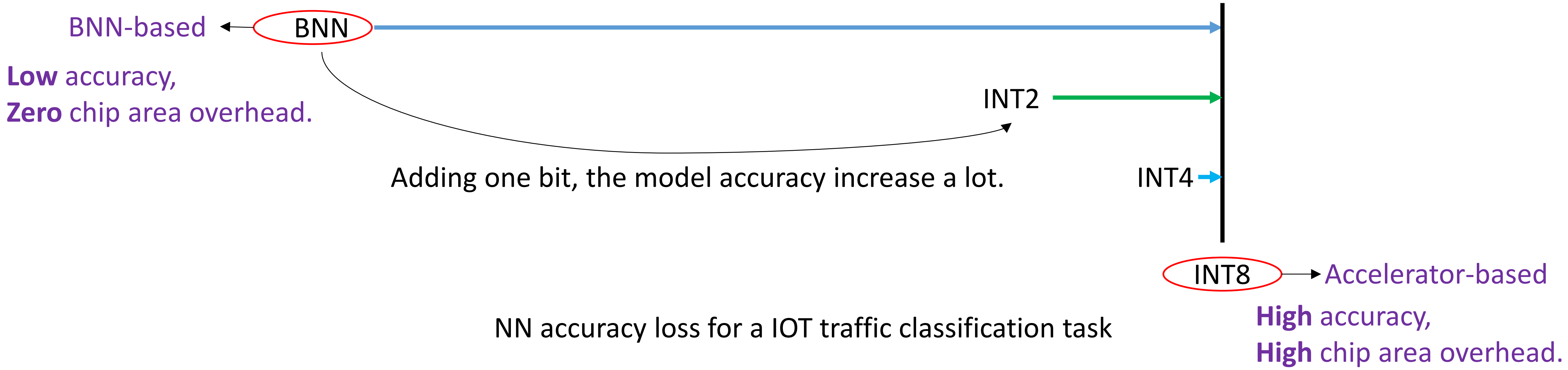


- Introduce Map Reduce Unit to RMT.
- Map Reduce Unit runs INT8 NN inference at line rate.

High area/programming overhead

[1] Siracusano, Giuseppe, et al. "Re-architecting traffic analysis with neural network interface cards." 19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22). 2022.
[2] Swamy, Tushar, et al. "Taurus: a data plane architecture for per-packet ML." Proceedings of the 27th ACM International Conference on Architectural Support for Programming Languages and Operating Systems. 2022.

Our Motivation： Low-bit Neural Network



NN accuracy loss for a IOT traffic classification task

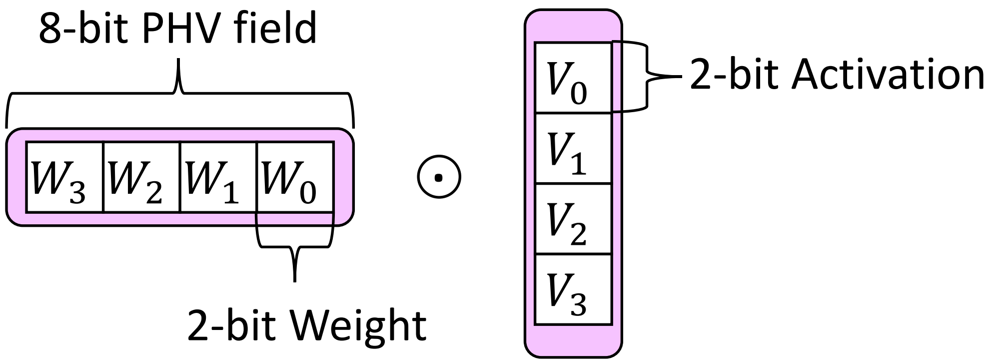
Question: How to efficiently execute low-bit NN inference in native RMT architecture?

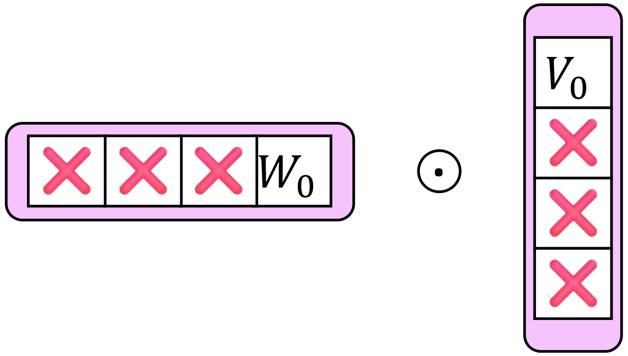
Relative high accuracy and **tiny** chip area overhead.

Our Design: Athena

Challenge #1: Low-bit Vector Multiplication

🙁 RMT does not support vectorized low-bit operations.





With vectorized low-bit operations, PHV fields are fully utilized.

Without vectorized low-bit operations, PHV fields are under utilized.

Running Low-bit NN Inference by Decomposition

RMT supports binary vector-vector operations.

Key idea: Spilt low-bit vector multiplication to multiple binary vector multiplications.

$$\mathbf{x} \cdot \mathbf{y} = \mathrm{bitcount}(\mathrm{and}(\mathbf{x}, \mathbf{y})), x_i, y_i \in \{0, 1\} \, \forall i.$$

BNN Vector Multiplication

Binary vector multiplication

$$\mathbf{x} \cdot \mathbf{y} = \sum_{m=0}^{M-1} \sum_{k=0}^{K-1} 2^{m+k} \boxed{\mathrm{bitcount}[\mathrm{and}(c_m(\mathbf{x}), c_k(\mathbf{y}))]},$$

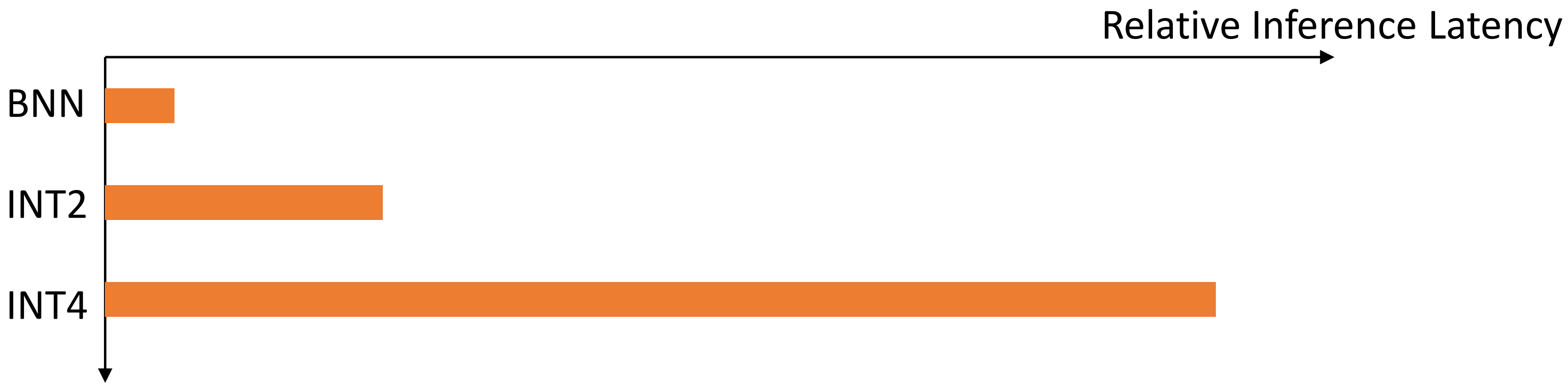
$$c_m(\mathbf{x})_i, c_k(\mathbf{y})_i \in \{0, 1\} \, \forall i, \, m, \, k.$$

M-bit x Vector and K-bit y Vector Multiplication

Athena achieves vectorized low-bit vector multiplication without hardware modifications.

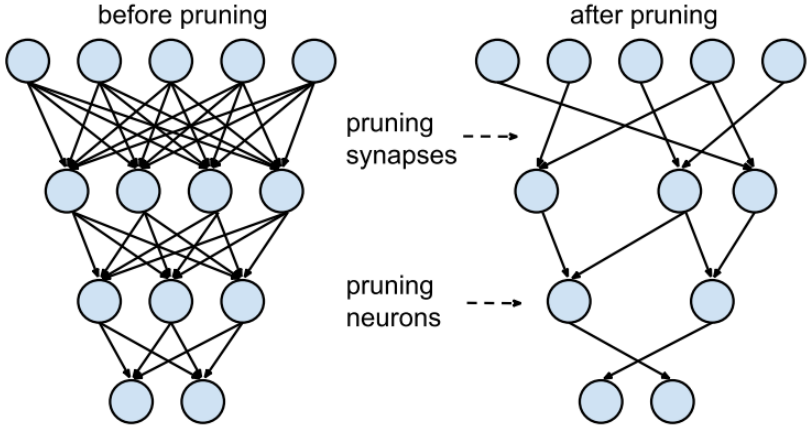
Challenge #2: Minimize the Inference Latency

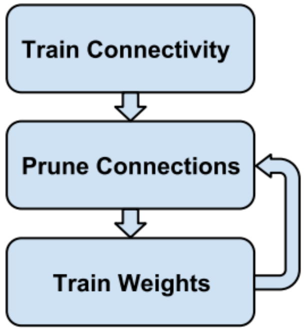
😟 Low-bit NN inference requires more computation than BNN inference.

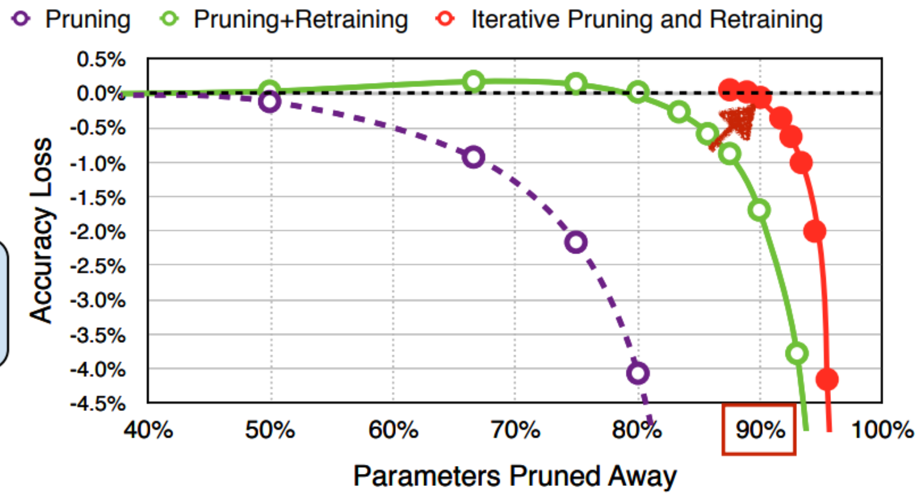


We should minimize the inference latency overhead.

Leverage NN Sparsity for Computation Reduction







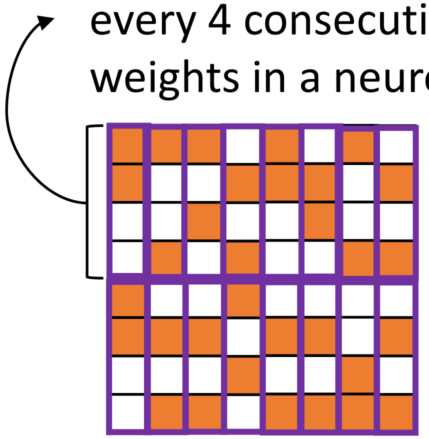
😀 NN models are sparse. Pruned NN models can maintain the accuracy.

Unexplored question: which sparsity granularity is suit for RMT computation?

RMT-friendly Sparsity Granularity

Athena identifies column-wise 2:4 Sparsity as the RMT-friendly sparsity granularity.

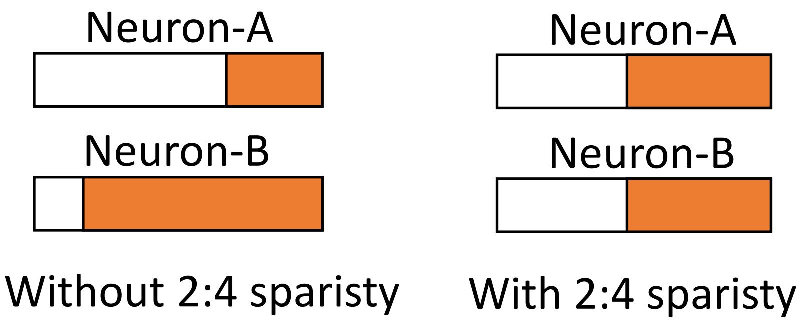
2 weights are left in every 4 consecutive weights in a neuron,



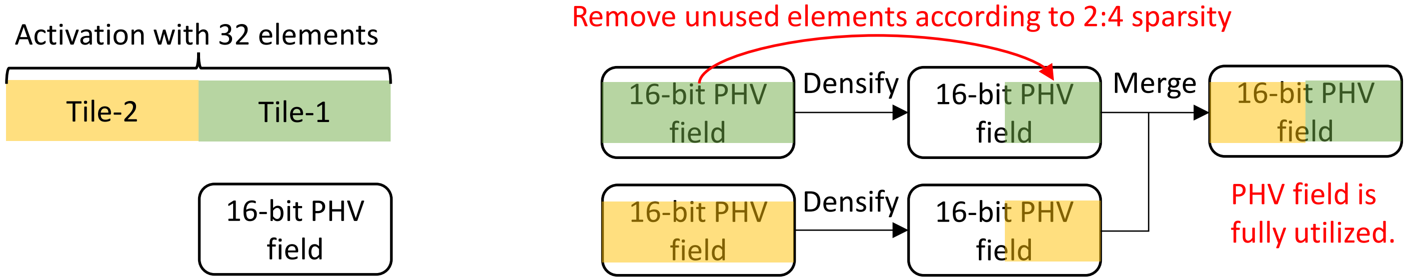
Column-wise 2:4 Sparsity

1. 2:4 Sparsity can achieve less popcount stages.

Popcount stages $\propto$ **Maximum** Number of unpruned elements in parallel neurons

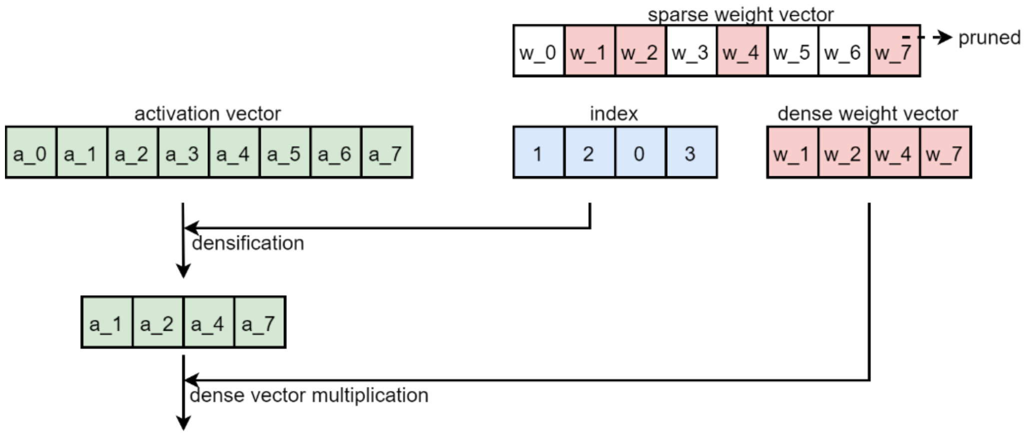


2. 2:4 Sparsity can support the **activation splitting** optimization, reducing the neuron-level tiling factor.

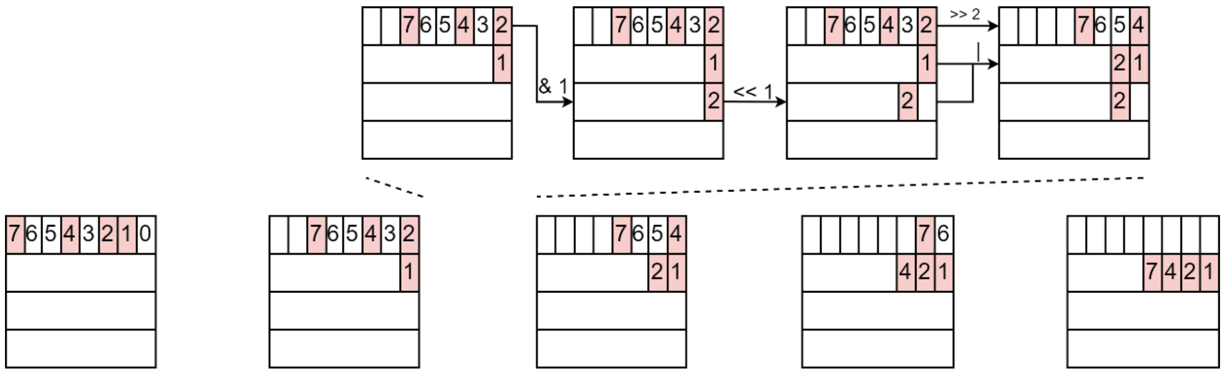


Inefficient Densification

Densification removes unused activation elements based on the weight sparsity distribution.



Densification process

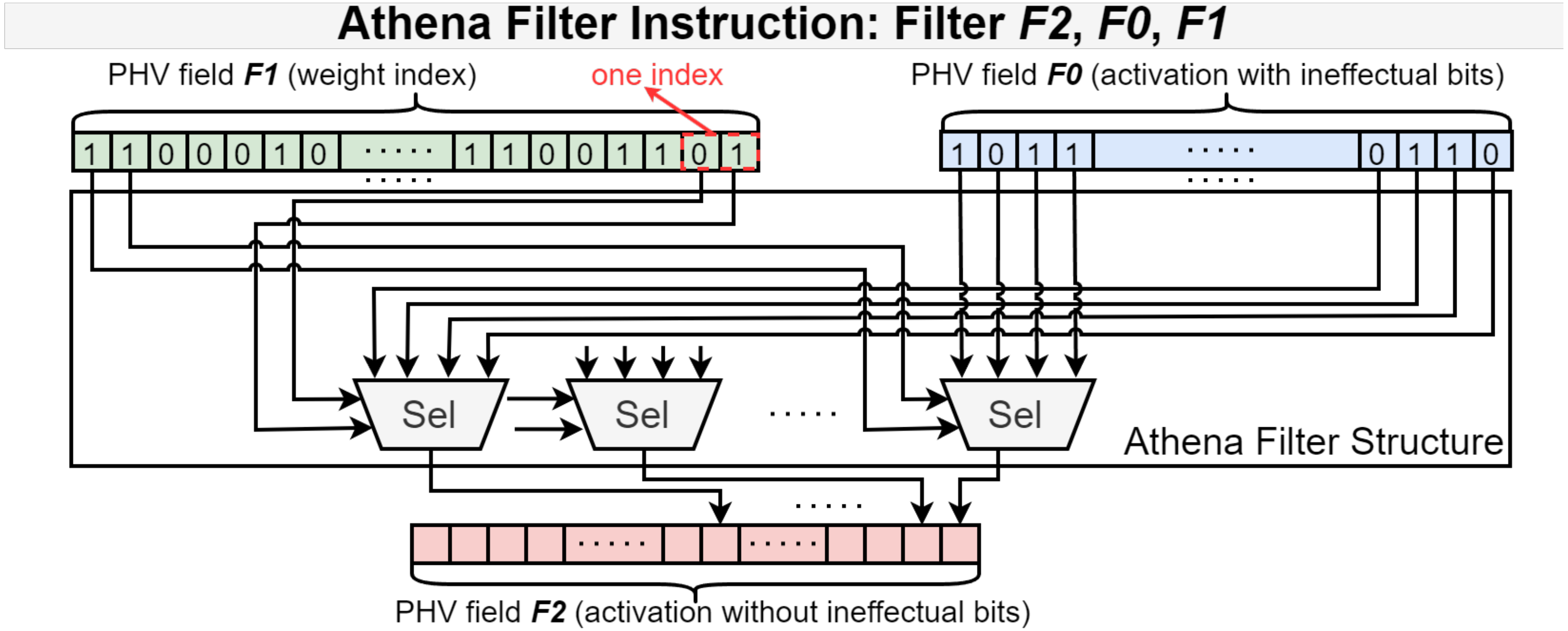


😟 Densification based on existing RMT ALU, several stages, **inefficient**

How to densify a PHV field in one stage?

Sparsity Filter Instruction for Densification

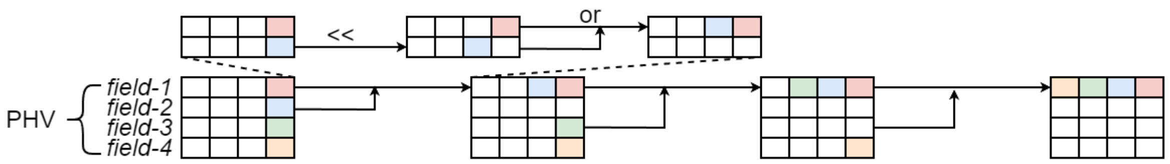
- Add **Athena filter instruction** to RMT's action ALU: *Filter F2, F0, F1*.
- **Tiny** chip area overhead, **zero** programming model overhead.

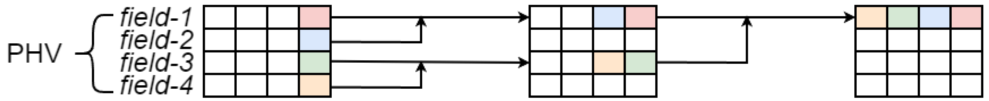


😀 We can densify a PHV field in one stage with Athena filter instruction.

Leverage Field-level Parallelism

Folding is the process of stacking bits from different neurons in the same position together.





N3IC[1] iteratively stacks bits on one PHV field.

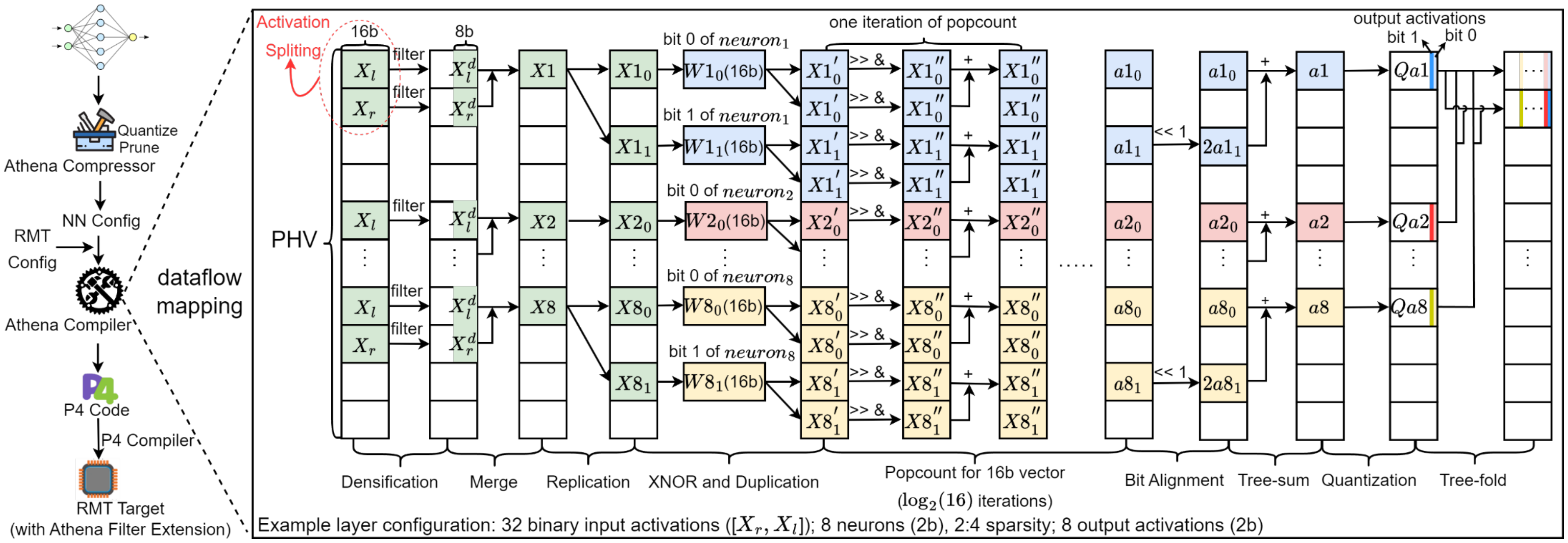
😦 Folding N bits requires $2(N-1)$ stages

Athena organizes the fold as a tree to fully use ALUs of idle fields.

😀 Folding N bits requires $2\log_2 N$ Stages

[1] Siracusano, Giuseppe, et al. "Re-architecting traffic analysis with neural network interface cards." 19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22). 2022.

Athena Overview



Example layer configuration: 32 binary input activations ($[X_r, X_l]$); 8 neurons (2b), 2:4 sparsity; 8 output activations (2b)

Athena deploys sparse low-bit multi-layer perceptron (MLP) neural network inference on RMT pipeline architecture.

- Athena compressor use DoReFa-Net and 2:4 sparsity pattern to quantize and prune the NN.
- Athena compiler represents NN dataflow in P4 language based on provided NN and RMT configurations.
- We envision the next-generation RMT target to support the Athena filter extension.

Evaluation

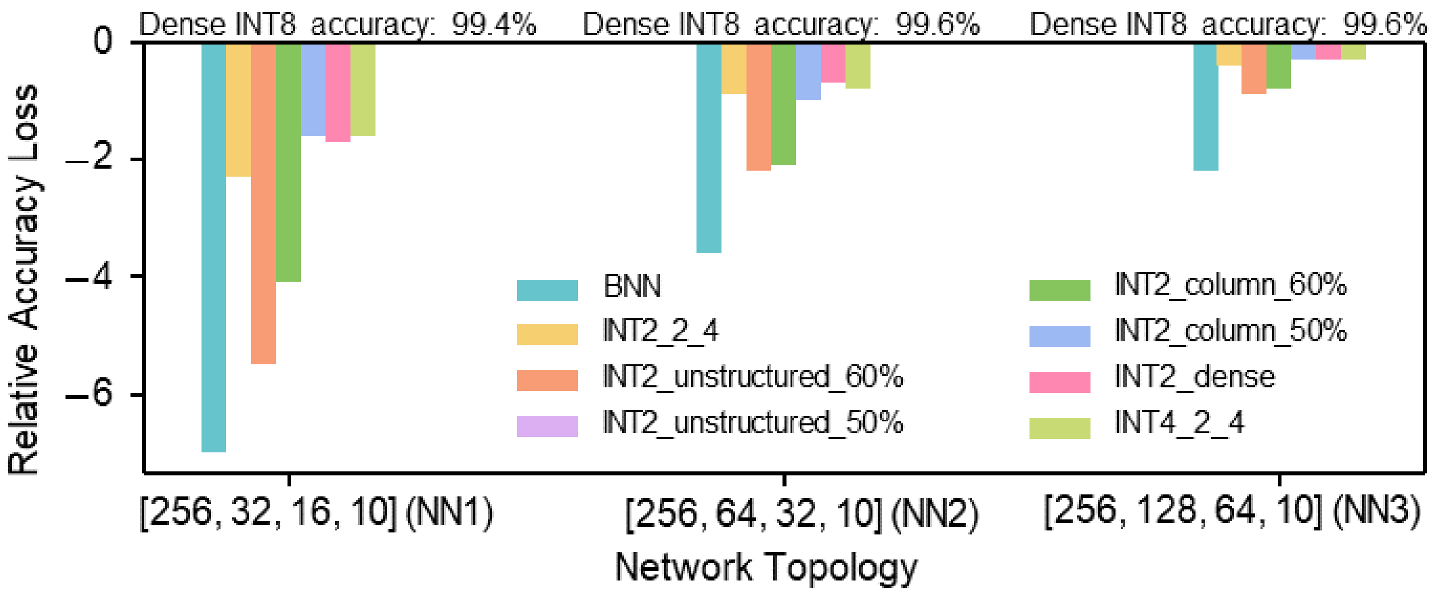
Three Questions Answered by Experiments

1. **Accuracy Improvement**: Compared with the BNN, how much accuracy improvement does the sparse low-bit NNs generated by Athena compressor achieve in computer network tasks?
2. **Inference Latency Overhead**: Compared with the BNN, how much inference latency overhead does the spare low-bit NN dataflow generated by the Athena compiler introduce?
3. **Athena Filter Extension Overhead**: How much overhead is the Athena filter extension?

Experiments Setup

- **Accuracy of Sparse Low-bit Model:**
  - Task**:** IoT traffic classification task on the UNSW IOT dataset.
  - Model: three-layer MLP, one input layer, one hidden layer, and one output layer.
  - Quantization: 2-bit
  - Sparsity: unstructured, 2:4 column-wise sparsity
- **Inference Latency Overhead:**
  - We build an analytical performance model to evaluate the required logical stages of NN computation on RMT.
  - PHV field width is configured as 32 bits according to commodity RMT-based switch.
  - The number of PHV fields is configured to fully unroll all the layers of BNN based on the dataflow of N3IC
- **Filter Extension Overhead:**
  - We implement the Athena filter in Verilog and synthesize the Verilog on Synopsys Design Compiler 2022.12-SP1 using an open 15nm FreePDK.

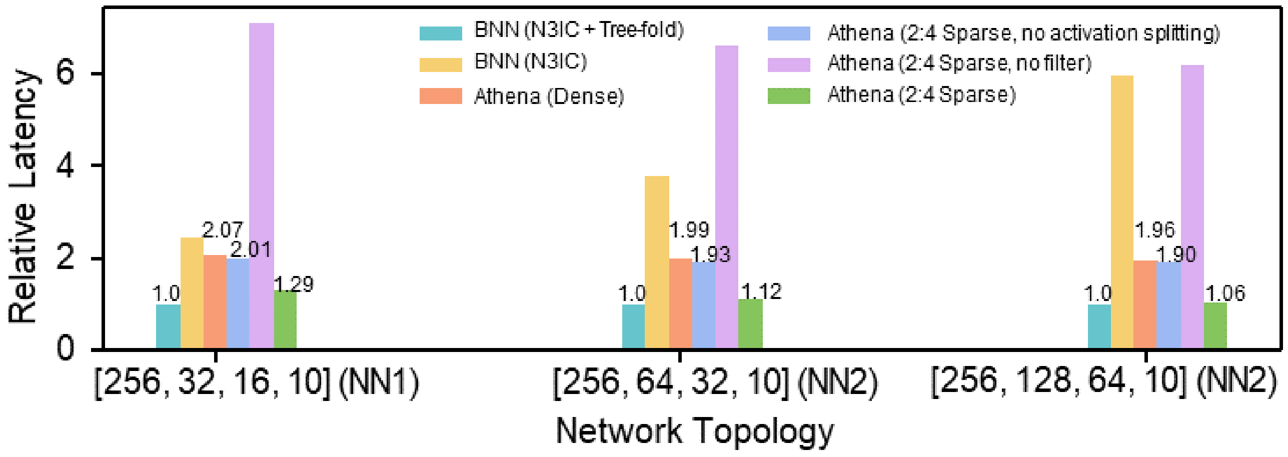
Higher IOT Traffic Classification Model Accuracy

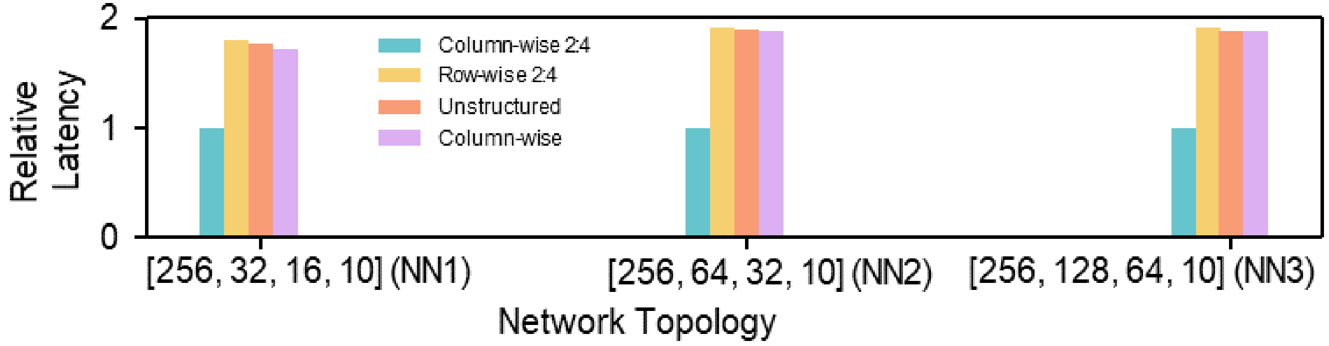


Results:

- The INT2 model with 2:4 sparsity has a 3.04X, 4.00X and 5.50X accuracy loss reduction compared with BNN.
- Second, 2:4 sparsity has comparable accuracy than the column-wise and unstructured pruning when the sparsity ratios are the same.

Optimized Inference Latency



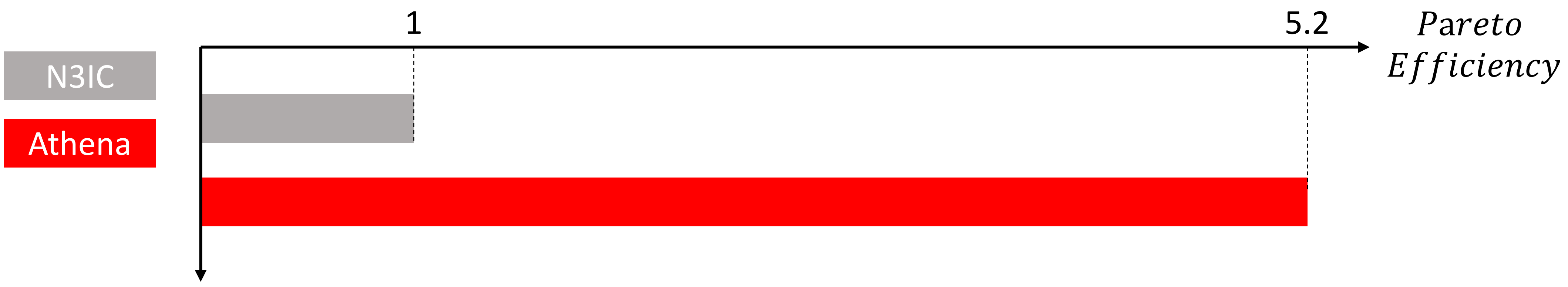


- Athena BNN can achieve 2.34X, 3.80X, and 5.95X inference latency reductions compared with N3IC BNN.
- The Athena filter is essential to ensure Athena (2:4 Sparse) has lower inference latency than Athena (Dense).
- The activation splitting can effectively reduce the inference latency by reducing the tiling factor.
- The column-wise 2:4 sparsity achieves lower latency than row-wise 2:4, unstructured and column-wise sparsity

Better Model Accuracy and Inference Latency Trade Off

$$Pareto\ Efficiency = \frac{Accuracy\ Loss\ Reduction}{Relative\ Inference\ Latency}$$

$Pareto\ Efficiency > 1$: better design trade off.



Tiny Athena Filter Overhead

	Chip Area Overhead	Clock Frequency >= 1GHz
Athena filter	$0.09mm^2$	Yes
Taurus	$4.8mm^2$	Yes

- We set the entire chip area of a RMT switch as $57.4\mathrm{mm}^2$(15 nm).
- Athena adds 0.2% chip area overhead.
- Athena reduces the chip area by 98%, compared with Taurus[3].

[2] Swamy, Tushar, et al. "Taurus: a data plane architecture for per-packet ML." Proceedings of the 27th ACM International Conference on Architectural Support for Programming Languages and Operating Systems. 2022.

Conclusion

Conclusion and Outlook

1. We explored the possibility of deploying low-bit NN on programmable switch to better support NN for computer network.
2. We proposed Athena, a full-stack solution for deploying low-bit NNs on the RMT pipeline.
3. We showed that Athena provides better design trade off than the STOA.
4. We can adapt Athena to other network data planes, e.g., the many-core architecture.

Takeaway: Intelligent Network Dataplane requires NN model.

Thank you

Yunkun Liao

liaoyunkun20s@ict.ac.cn