TurboNet Team 高性能网络研究小组
TNT High-performance Network Research Team

MegaTurbo: A Scalable FPGA-based Engine for MegaFlow Classifier in Open vSwitch

Sheng Lan, Ying Li, Zhongxian Liang, Wenjun Li*, Yao Xin, Ying Wan, Hui Li, Weizhe Zhang

* The first three authors are students, and they conducted this work under the supervision of their advisor: Wenjun Li (wenjunli@pku.org.cn)

ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (**FPGA'26**)
Seaside, California, USA, February 22-24, 2026

TurboNet Team

CONTENTS

Review on Open vSwitch (OVS)

- **Two execution paths in OVS packet processing: Slow path + Fast path**

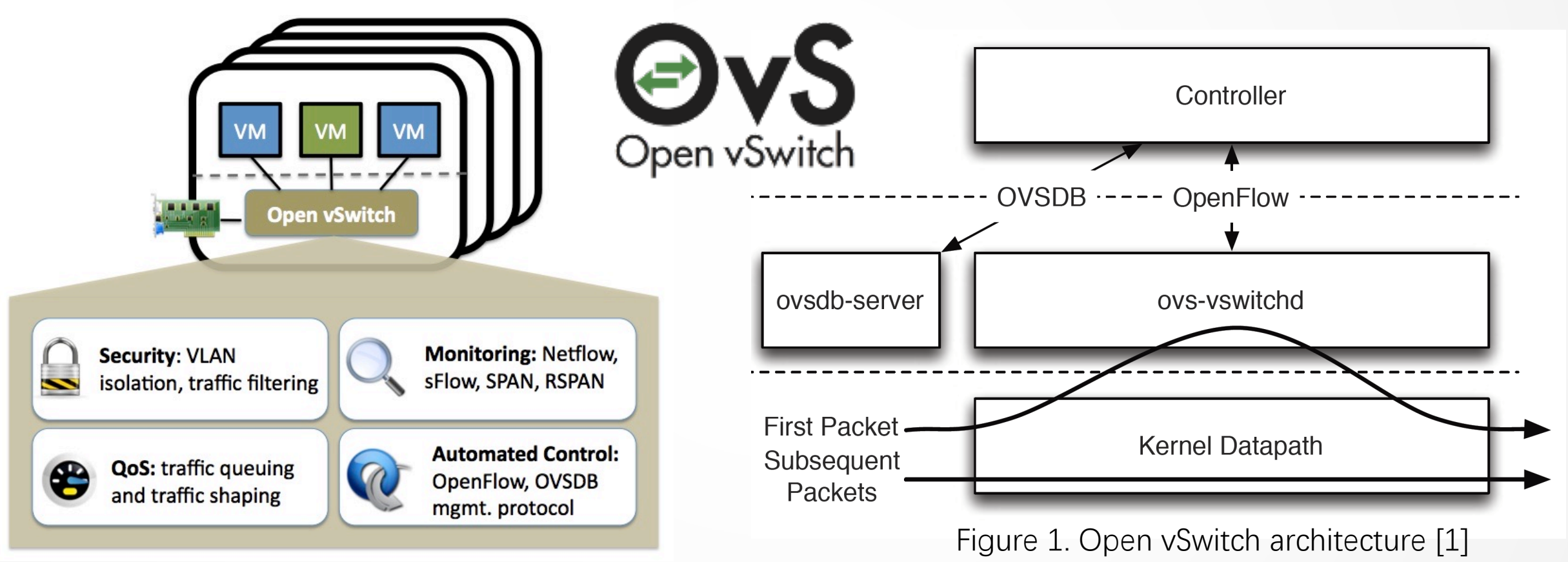


Figure 1. Open vSwitch architecture [1]

[1] Ben Pfaff, et al., "The Design and Implementation of Open vSwitch," In USENIX NSDI, 2015.

Packet Classification in OVS

- **Key for OpenFlow rule table lookup and MegaFlow cache table lookup**

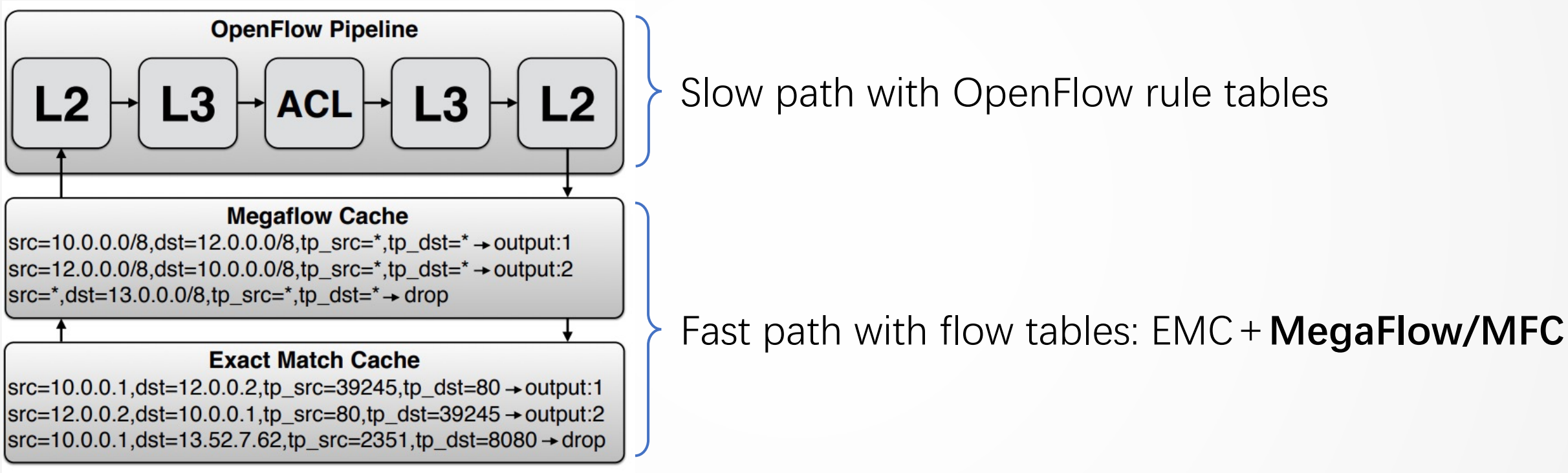


Figure 2. Cache hierarchy in OVS [2]

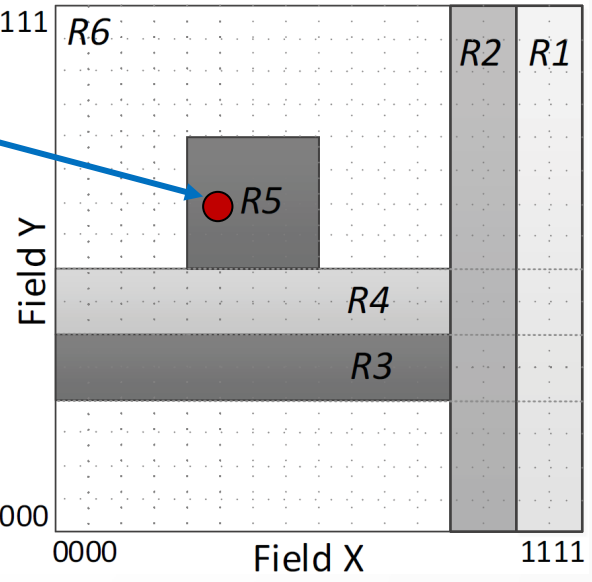
[2] Nick Shelly and et al., "Flow Caching for High Entropy Packet Fields," In ACM HotSDN, 2014.

Review on the Packet Classification Problem

❖ **Algorithmic table lookup ←→ Geometric point location**

Rules	Field X	Field Y	*Action*
R1	111*	*	*action1*
R2	110*	*	*action2*
R3	*	010*	*action3*
R4	*	011*	*action4*
R5	**01****	**10****	***action5***
R6	*	*	*action6*

e.g., Packet P_i
<0101,1010>

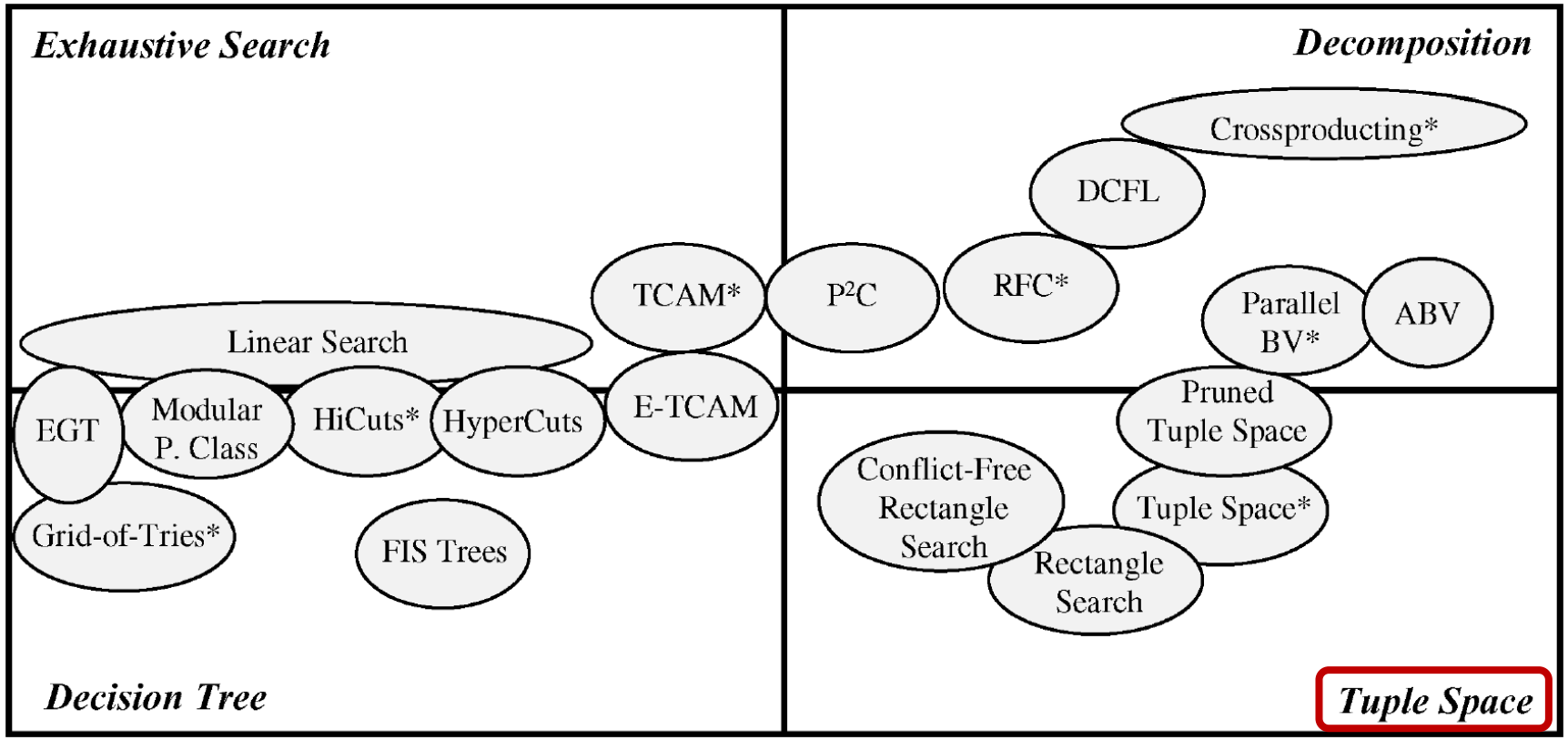


❖ **Metrics for multi-field packet classification**

- Time: Throughput, Memory access, Construction time
- Space: Memory consumption
- Others: Updatable, More fields, Larger classifier, Power consumption, etc.

Review on Existing Packet Classifications

❖ Well-known taxonomy from David E. Taylor [3]



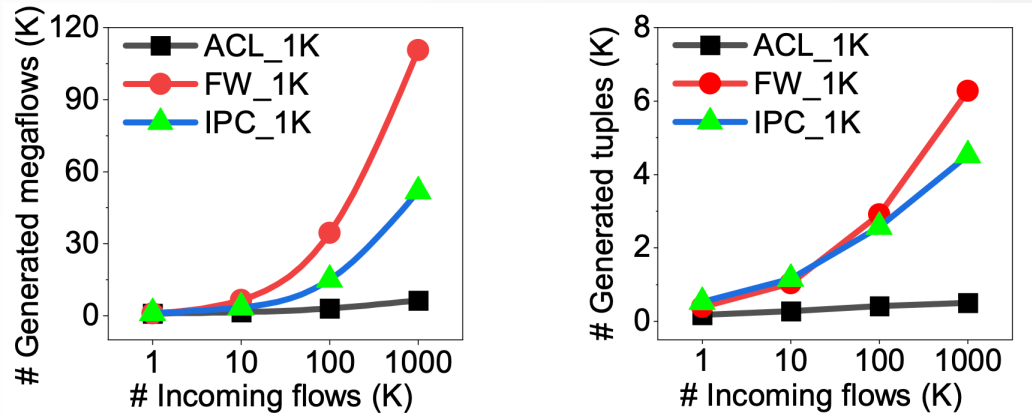
TSS can support fast rule updates

❖ Packet classification in OVS: A variant of Tuple Space Search(TSS)

[3] David E.Taylor, "Survey and Taxonomy of Packet Classification Techniques," ACM Computing Surveys, 37(3):238-275, 2005.

Existing HW Acceleration Techniques for MFC

- **Multicore/ASIC for TSS algorithm**
  - Suffer from a tuple explosion problem
- **TCAM**
  - Expensive, power-hungry, limited capacity
- **FPGA**
  - RTC: update-friendly, but low throughput & non-deterministic latency [4,5]
  - Pipeline: high throughput, but poor support for dynamic rule updates [6]
  - There is no solution specifically tailored for MFC (**a highly distinctive classifier**)



(a) Number of flow rules in MFC (b) Number of tuples in MFC

Figure 1: MFC evaluation results under varying traffic loads

[4] Yao Xin, et al., "Recursive Multi-Tree Construction with Efficient Rule Sifting for Packet Classification on FPGA," In IEEE/ACM TON, 2024.
[5] Yao Xin, et al., "A Parallel and Updatable Architecture for FPGA-Based Packet Classification With Large-Scale Rule Sets," In IEEE Micro, 2023.
[6] Yaxuan Qi, et al., "Multi-dimensional Packet Classification on FPGA: 100 Gbps and Beyond," In IEEE FPT, 2010.

Thus, a hardware accelerator specifically tailored for MFC is desired, forming the basis of MegaTurbo.

Our Design Principles

Domain-specific Design → Low Resource Consumption

Algorithm-hardware Co-design → Fast & Deterministic Rule Lookup

CPU-FPGA Collaborate Framework → Dynamic Rule Update

TurboNet Team

CONTENTS

Key Insights on MegaFlow Classifier

- **MegaFlow rules do not overlap with each other**
  - At most one rule matches a packet
  - No priority resolution in the classifier
  - Packet classification reduces to locating a packet in a unique region
  - **Inspiration**: MegaFlow classification naturally maps to a decision tree

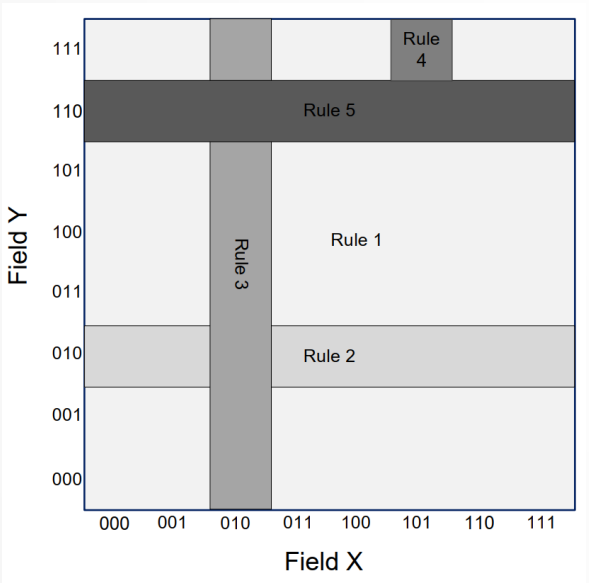


Figure 3. Original OpenFlow ruleset

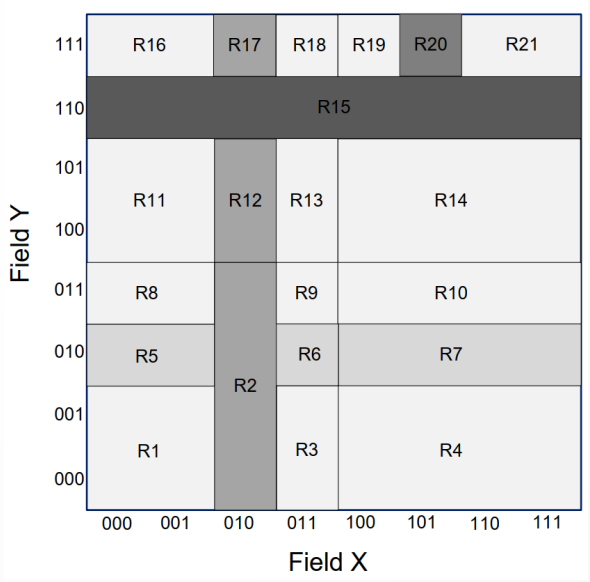
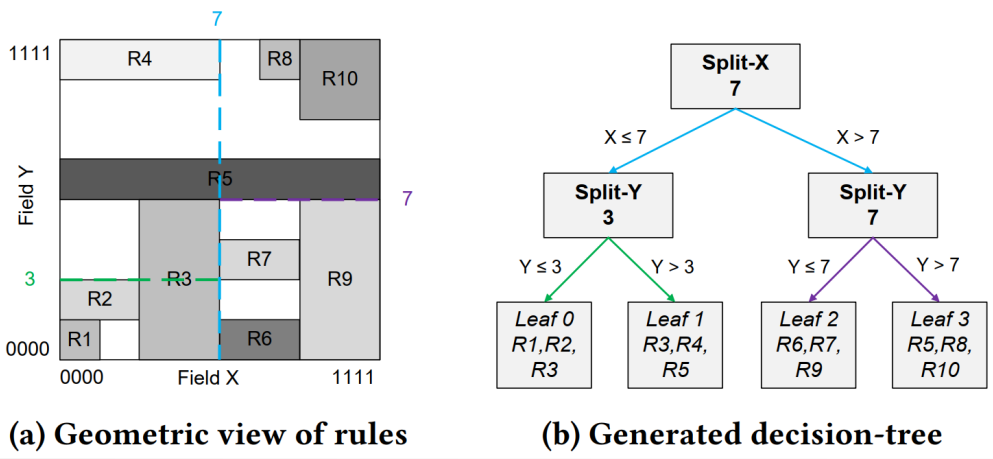


Figure 4. Partitioned MegaFlow ruleset

MegaFlow-dedicated Algorithm: MegaTree

- **MegaTree: Balanced trees with identical max depth, without rule replication**
  - HyperSplit-based tree construction → **Balanced trees** → Friendly for hardware
    - Recursively splits the rule space along one dimension
    - Chooses split points to balance subspaces
  - Kick mechanism → **No rule replication** → Friendly for dynamic rule update
    - when a split causes replication
    - when a leaf exceeds capacity



(a) Geometric view of rules (b) Generated decision-tree

Figure 5. Algorithm foundation: HyperSplit [7]

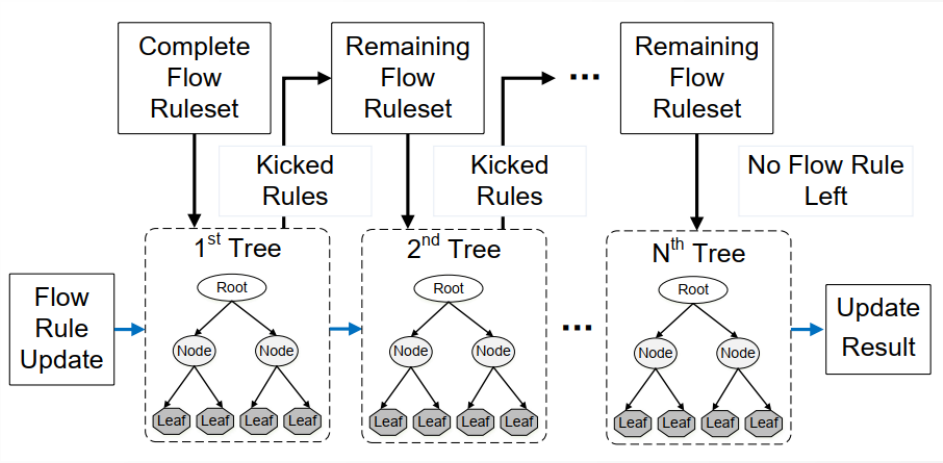


Figure 6. HyperSplit-based MegaTree algorithm

[7] Yaxuan Qi, "Packet Classification Algorithms: From Theory to Practice," In IEEE INFOCOM, 2009.

Example: HyperSplit *VS* MegaTree

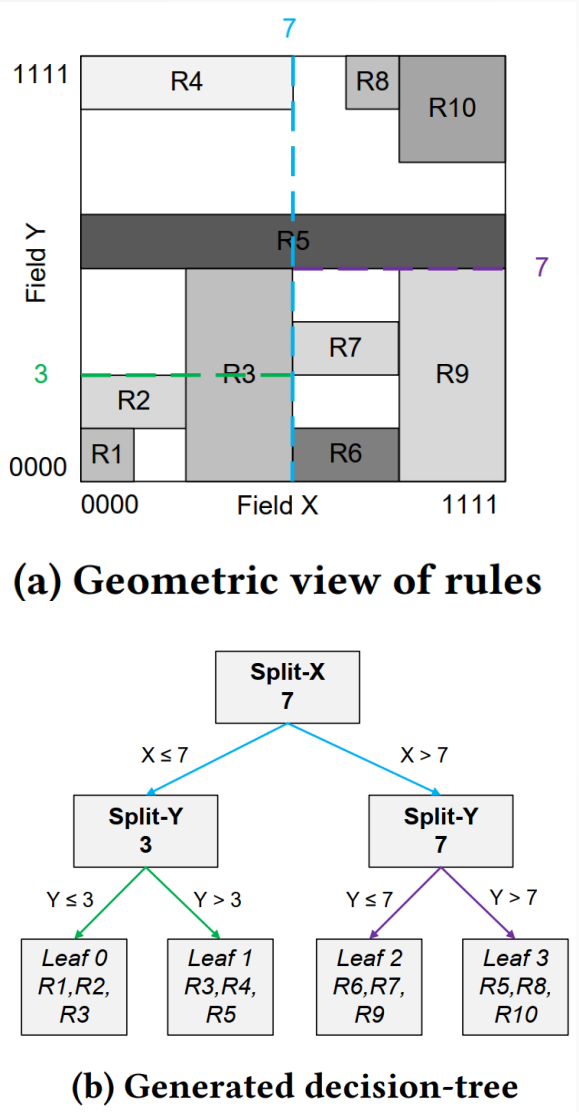


Figure 7. HyperSplit example

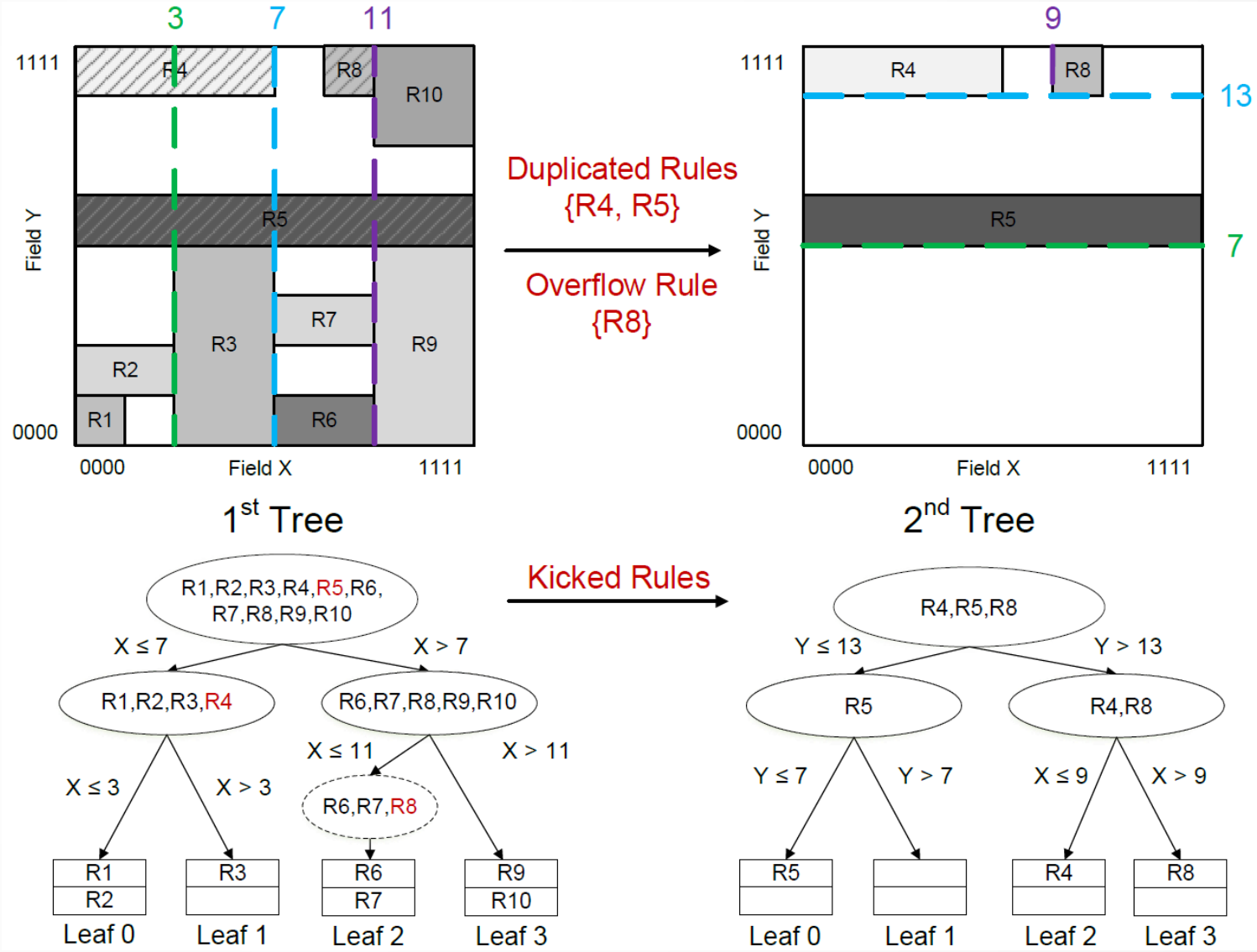


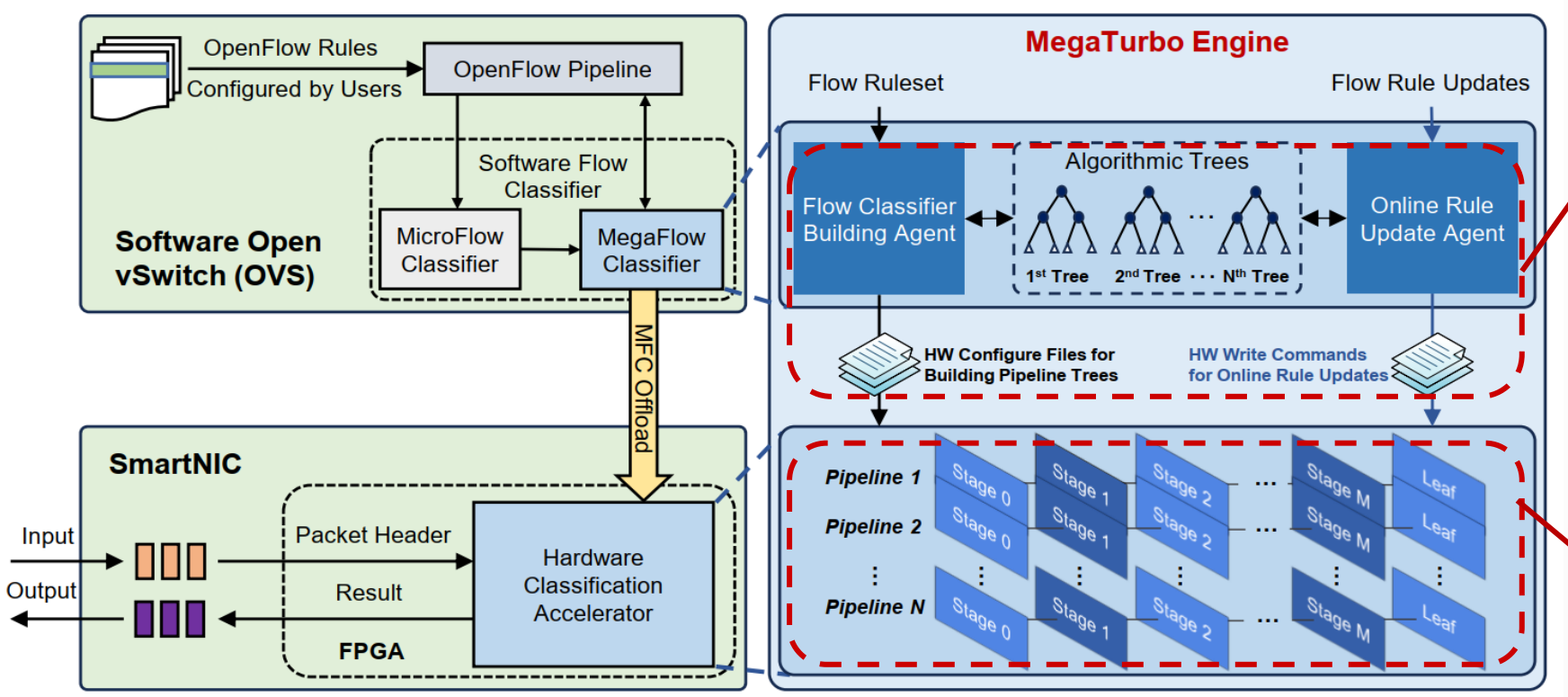
Figure 8. MegaTree example

TurboNet Team

CONTENTS

Overall Framework

❖ MegaTurbo engine: a CPU-FPGA collaborate framework



MegaFlow rules are compiled into hardware-friendly trees, and hardware configure files

Packets and rule updates are processed by a parallel, fully pipelined FPGA engine

Figure 9. System architecture of MegaTurbo

FPGA Design: Parallel and Hierarchical Pipelines

- **Decision trees in MegaTree are mapped onto hierarchical FPGA pipelines**
  - Each pipeline stage is isolated, enabling high clock frequency
  - Multiple tree pipelines run in parallel, and results are merged via a simple MUX

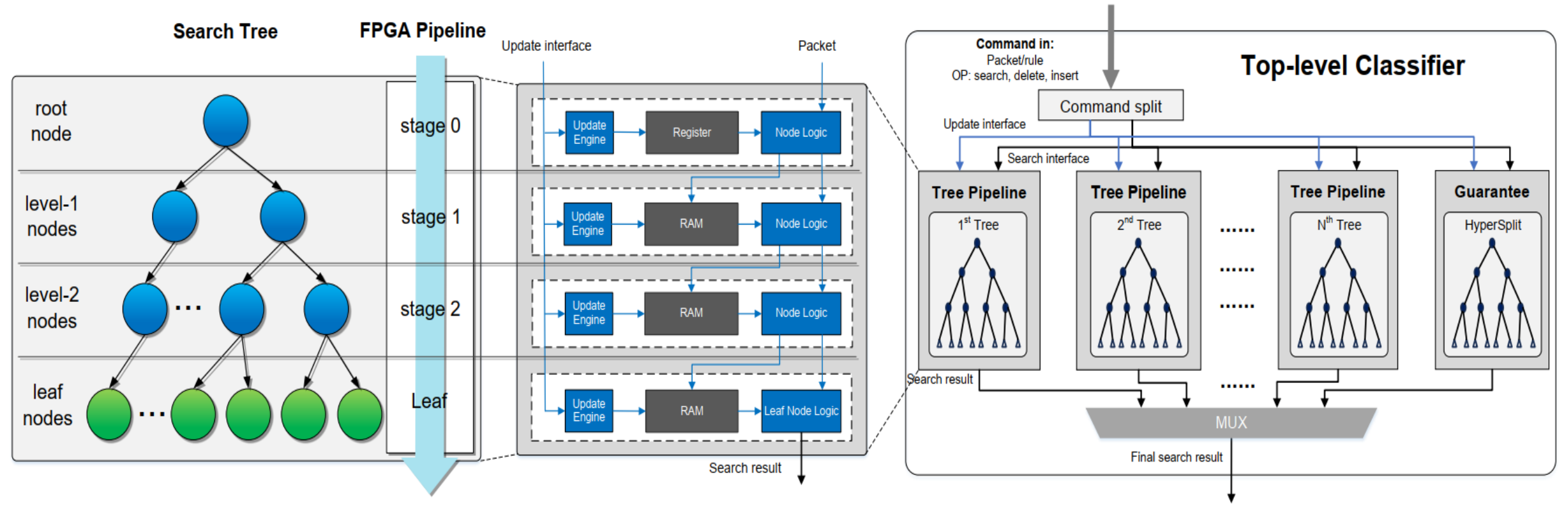


Figure 10. Hardware architecture

FPGA Design: Inter/Leaf RTL Module

❖ Operations

- Internal module
  - Previous-stage address → local RAM lookup
  - Simple comparison → next-level address
- Leaf module
  - Matched in parallel and results are aggregated
  - Simple MUX (without complex priority resolution)

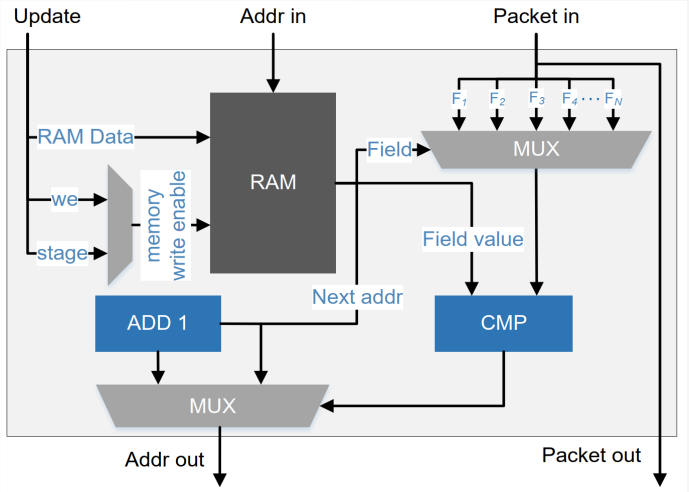


Figure 11. Internal pipeline stage

❖ Advantages

- Simple logic → low resource usage
- Isolated stages → high clock frequency
- Independent stages → localized updates

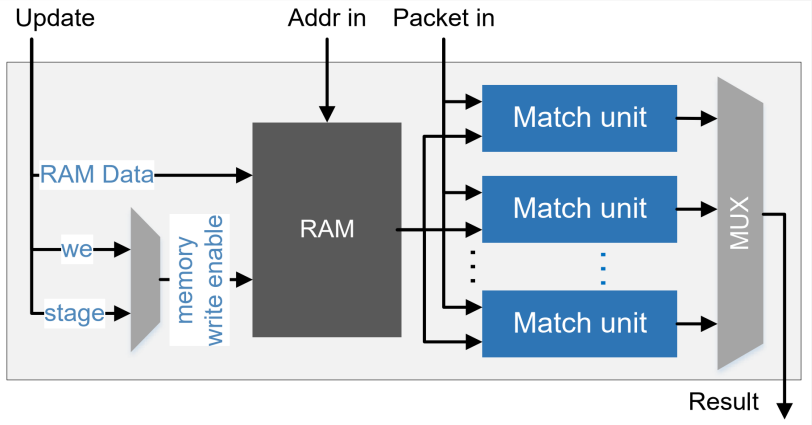


Figure 12. Leaf rule matching

A Running Example: From Algorithm to Hardware

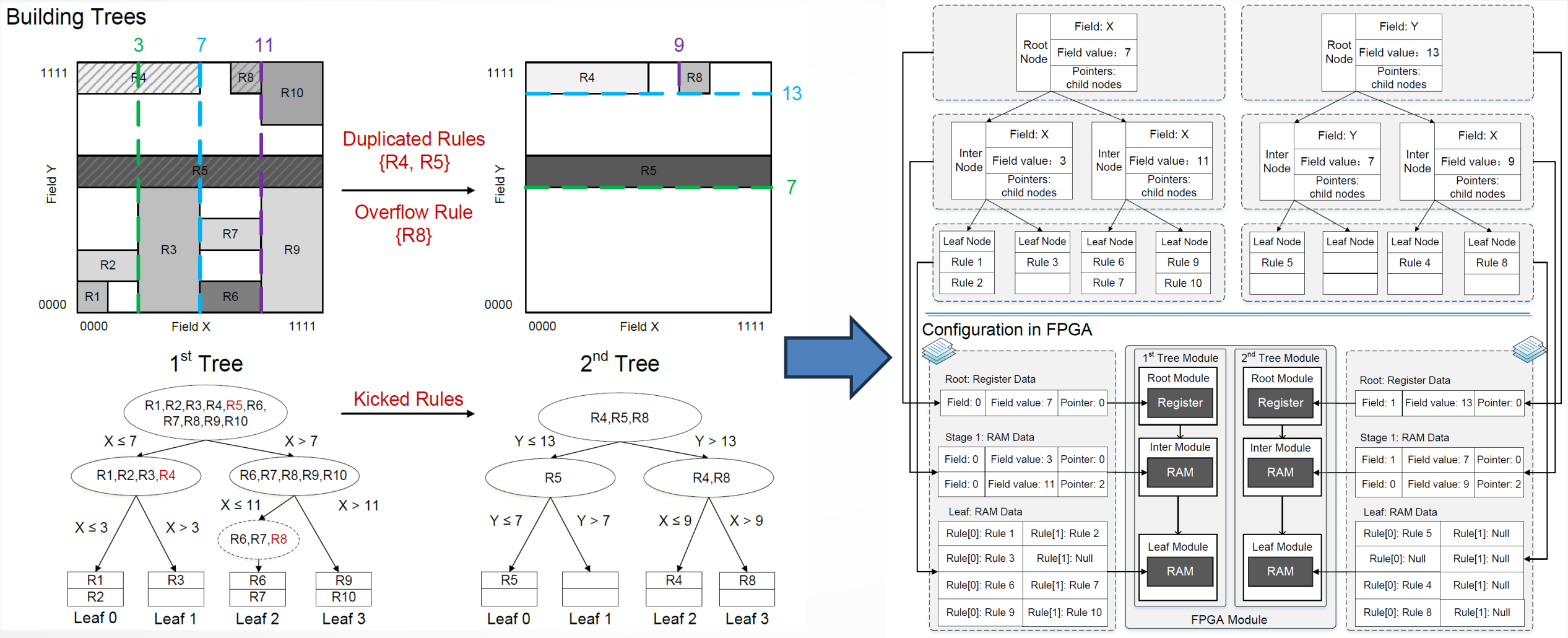


Figure 13. A running example of MegaTurbo: From MegaTree algorithm to RTL modules (binth=2, MAX depth=2)

Why MegaTurbo Supports Dynamic Updates?

- **Key 1: No rule replication (Software/Algorithm Level)**
  - Each rule stored exactly once
  - Update complexity independent of tree depth
- **Key 2: Stage-local memory organization (Hardware Level)**
  - Local RAM per pipeline stage
  - Updates affect only one stage
  - No global rebuild or rebalancing
- **Key 3: Decoupled update & lookup paths (Architecture Level)**
  - Dedicated update interface
  - Lookup pipeline continues running
  - No classification stall

TurboNet Team

CONTENTS

FPGA Implementation

- Implemented on Xilinx Virtex UltraScale+ VU9P at **250 MHz**
- Low and stable resource usage across all **100K-scale** rulesets

Ruleset	CLB LUTs (1182240)	CLB Registers (2364480)	BRAM (2160)	LUTRAM (591840)	Max Frequency
ACL1_100K	3.23%	2.05%	29.17%	0.39%	250.90 MHz
ACL2_100K	3.23%	2.05%	29.17%	0.39%	250.94 MHz
ACL3_100K	3.23%	2.05%	29.17%	0.39%	250.94 MHz
ACL4_100K	3.23%	2.05%	29.17%	0.39%	250.94 MHz
ACL5_100K	3.23%	2.05%	29.17%	0.39%	250.94 MHz
FW1_100K	3.23%	2.05%	29.17%	0.39%	250.94 MHz
FW2_100K	3.23%	2.05%	29.17%	0.39%	250.94 MHz
FW3_100K	3.23%	2.05%	29.17%	0.39%	250.94 MHz
FW4_100K	3.53%	2.31%	31.71%	0.21%	250.38 MHz
FW5_100K	3.53%	2.29%	31.71%	0.21%	254.39 MHz
IPC1_100K	3.23%	2.05%	29.17%	0.39%	250.94 MHz
IPC2_100K	3.54%	2.32%	31.71%	0.21%	251.26 MHz

Table 1. FPGA resource utilization across 100K MegaFlow rulesets

Results: High Throughput with Efficient Updates

- Sustains ~**500 MPPS** classification throughput across all workloads
- Supports **>300KUPS** dynamic rule updates without throughput degradation
- Maintains **stable** performance across diverse MegaFlow rulesets

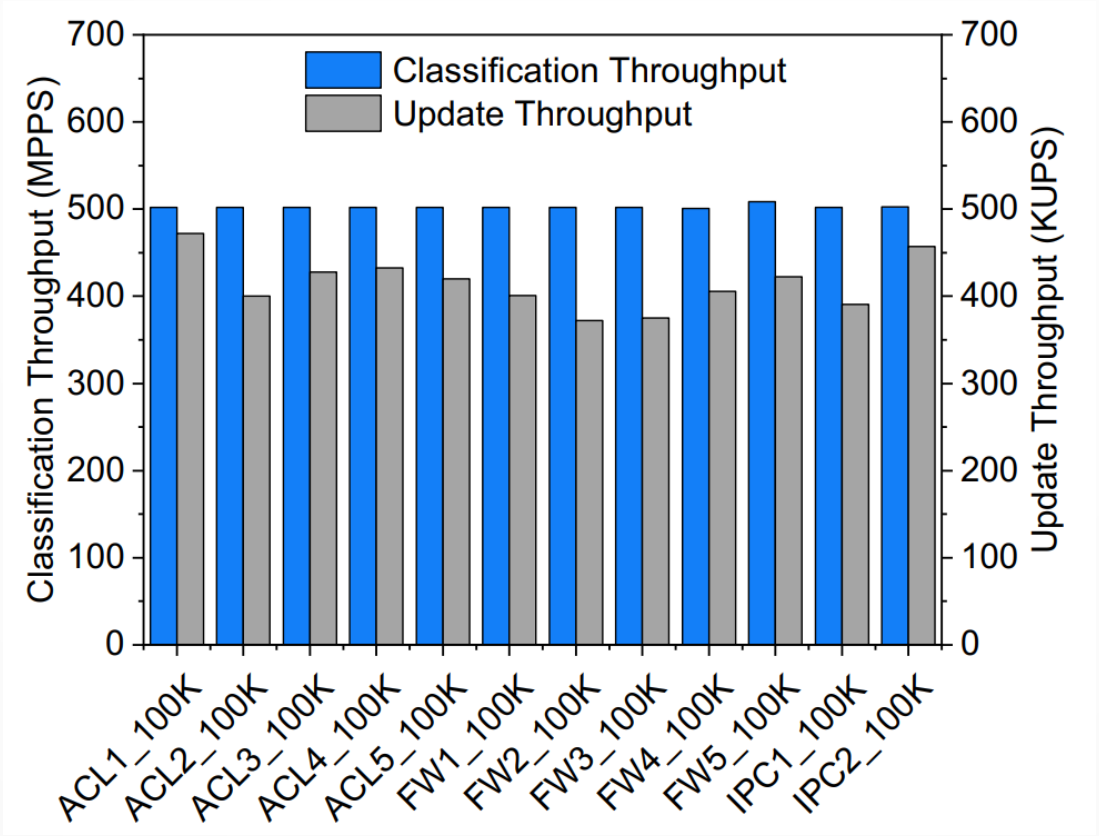


Figure 14. Throughput for 100K MegaFlow rulesets

Comparison With Related Work

❖ Domain-specific+Co-design → Excellent trade-off+Performance advantages

Approach	Ruleset Scale	Device	Resource Consumption				Classification Throughput (MPPS)	Microsecond-scale On-the-fly Update
			LUT	Registers	BRAM	URAM		
Proposed MegaTurbo	100K	Ultrascale+ VU9P	26828	49367	630	0	501.88	✓
HACL [57]		UltraScale+ XCVU35P	190811	335203	416	156	280.74	×
KickTree_Systolic [58]		Ultrascale+ VU9P	506670	550812	1836	936	121.40	✓
TcbTree [59]		Ultrascale+ VU9P	66834	111719	990	792	45.6	✓
KickTree_Parallel [61]		Ultrascale+ VU9P	607596	819039	1815	762	172.9	✓
MBitTree on FPGA [51]		Virtex-7 XC7V690T	37828	75656	818	0	175.9	×
REC [3]	10K	Virtex-5 XC5VFX200T	7044	\	173	0	323.5	×
		Virtex-6 XC6VLX760	7044	\	173	0	388.2	×
UTPC [18]		Stratix III EP3SE260H780	40070	\	852	0	433	×
D^2BS [64]		Virtex-5 XC5VSX240T	\	\	\	\	263.7	×
HyperCuts on FPGA [17]		Virtex-5 XC5VFX200T	10307	\	407	0	250.7	×
ParaSplit [9]		Virtex-5 XC5VSX240T	48380	\	399	0	200.4	×
CubeCuts [4]		Virtex-5 XC5VFX200T	45656	\	195	0	368.8	×
HyperSplit on FPGA [38]		Virtex-6 XC6VLX760	2988	5976	103	0	230.5	×

Table 2. Comparison of decision tree based approaches on FPGA

TurboNet Team

CONTENTS

Conclusion

- **MegaTurbo is a domain-specific design**
  - To the best of our knowledge, MegaTurbo is the first FPGA-based accelerator specifically designed for the OVS MegaFlow classifier
- **MegaTurbo is a CPU-FPGA collaborate framework**
  - To the best of our knowledge, MegaTurbo is the first solution that supports microsecond-scale online rule updates for a pipelined decision tree architecture
- **MegaTurbo is an algorithm-hardware co-design**
  - By mapping multiple balanced trees onto multiple parallel pipelines, MegaTurbo achieves high throughput, deterministic latency, and dynamic rule updates

One More Thing: Our Opinions/Ongoing Works on OVS

- **The fast path can be further improved, especially the MegaFlow Classifier**
  - MFC still has much room for domain-specific and co-design optimization
  - Scalable rule update guarantee mechanisms for hardware are desired
  - Offloading critical MegaFlow rules to valuable hardware assisted by network measurement (e.g., Sketch) is an attractive optimization direction
- **The slow path is emerging as a critical bottleneck in the OVS system**
  - Efficient cache rule consistency verification mechanism is highly desired
  - Offloading the slow path to hardware is a radical yet promising solution
- **OVS-oriented research TestBench and platform are highly desired**
  - Current TestBenches like ClassBench fail to capture modern cloud use cases
  - An FPGA platform specifically tailored for OVS acceleration is highly valuable

高性能网络研究小组
TNT High-performance Network Research Team

高性能网络研究小组微信公众号二维码

THANKS!

https://wenjunli.com/MegaTurbo