Towards Effective Packet Classification

J. Li, Y. Qi, and B. Xu

Network Security Lab

RIIT, Tsinghua University

Dec, 2005

Outline

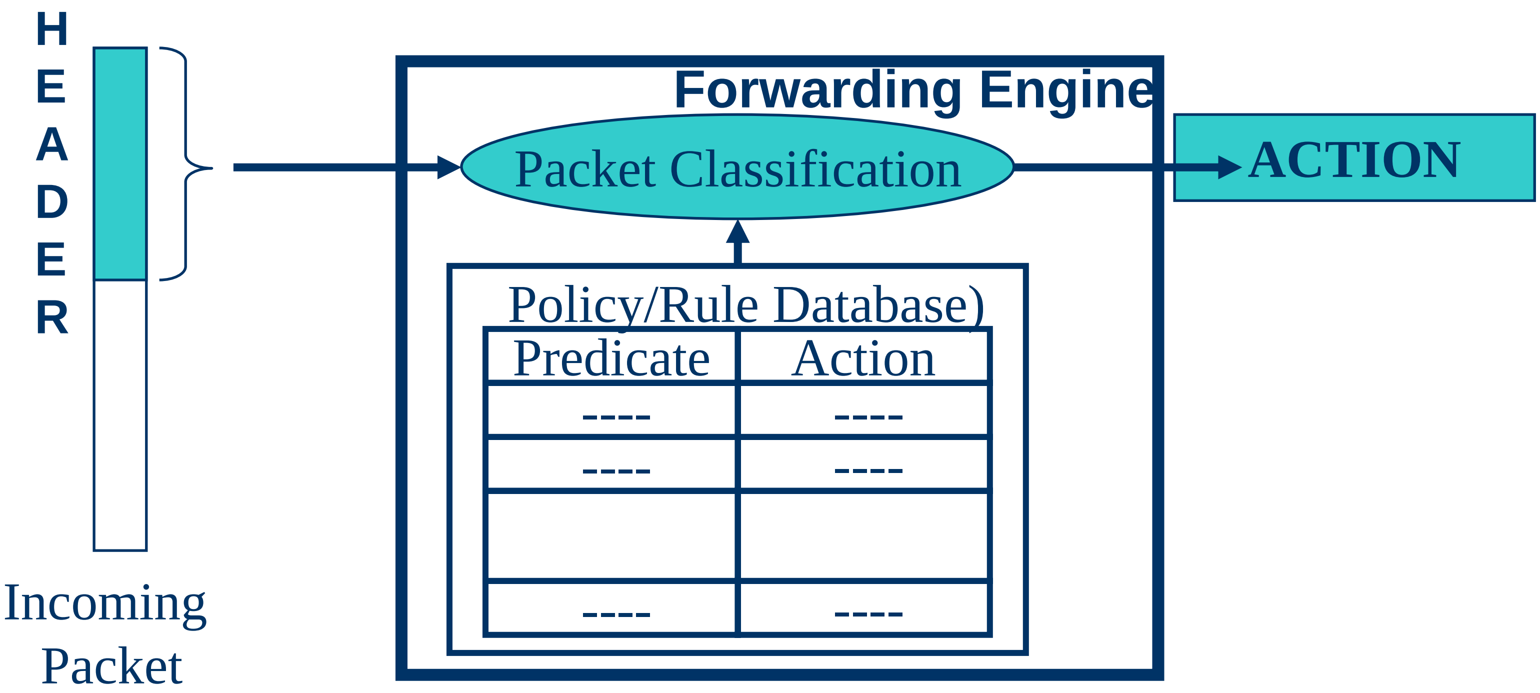
- Algorithm Study
  - Understanding Packet Classification
  - Worst-case Complexity Analysis
  - Existing Algorithmic Solutions
  - Our Novel Algorithms
- Network Processor Implementation
  - Current Hardware Limits
  - External & Internal Traffics
  - Intel IXP Implementation
- Summary

Outline

- Algorithm Study
  - Understanding Packet Classification
  - Worst-case Complexity Analysis
  - Existing Algorithmic Solutions
  - Our Novel Algorithms
- Network Processor Implementation
  - Current Hardware Limits
  - External & Internal Traffics
  - Intel IXP Implementation
- Summary

1.1 Understanding Packet Classification

- Packet Classification Overview



1.1 Understanding Packet Classification

- An Example Rule Set

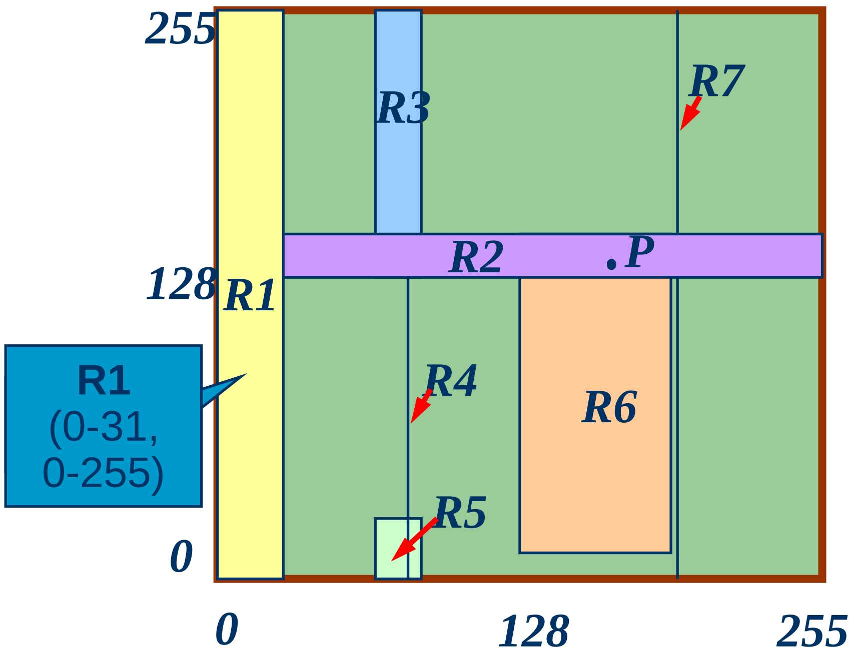
	Field 1	*Field 2*	*…*	*Field k*	*Action*
Rule 1	152.163.190.69/21	152.163.80.11/32	…	UDP	A1
Rule 2	152.168.3.0/24	152.163.0.0/16	…	TCP	A2
…	…	…	…	…	…
Rule N	152.168.0.0/16	152.0.0.0/8	…	ANY	An

E.g. A packet P(152.168.3.32, 152.163.171.71, …, TCP) would have action A2 (also matches An but A2 has higher priority) applied to it.

1.1 Understanding Packet Classification

- Rules In the Search Space

Rule	*Xrange*	*Yrange*
R1	0-31	0-255
R2	0-255	128-131
R3	64-71	128-255
R4	67-67	0-127
R5	64-71	0-15
R6	128-191	4-131
R7	192-192	0-255



Spaces: Single/multiple dimensions (fields); Span of each dimensions.
Rules: Prefix/Range matching; Structural characteristics.
Packets: Dynamic characteristics.

1.2 Worst-case Complexity Analysis

Point Location: among N non-overlapping regions in F dimensions takes

- Either O(log$N$) time with O(N^F) space
- Or O(N) space with O($\log^{F-1}N$) time
- E.g. N=1000，F=4：1000G space，1000 accesses

De-overlapping: N overlapping regions need up to $(2N\text{-}1)^F$ non-overlapping region to represent.

Range-to-Prefix: N rules in range [0, 2^W] need up to $(N(2W\text{-}1))^F$ prefixes to represent.

F: number of fields; W: bit length of each field

1.2 Worst-case Complexity Analysis

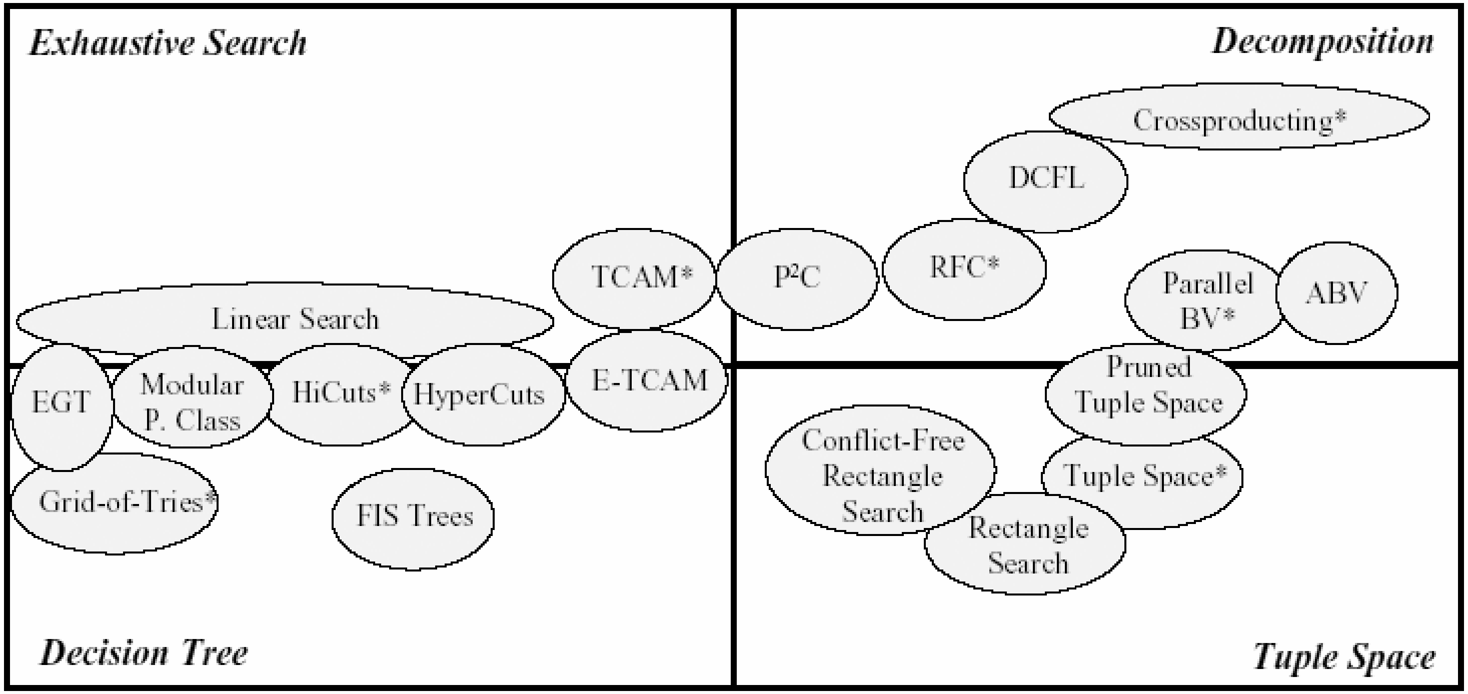
- Conclusion

The theoretical bounds tell us that it is not possible to arrive at a practical worst case solution. Fortunately, we don’t have to; No single algorithm will perform well for all cases. Hence a hybrid scheme might be able to combine the advantages of several different approaches.

—— P. Gupta

1.3 Existing Algorithmic Solutions

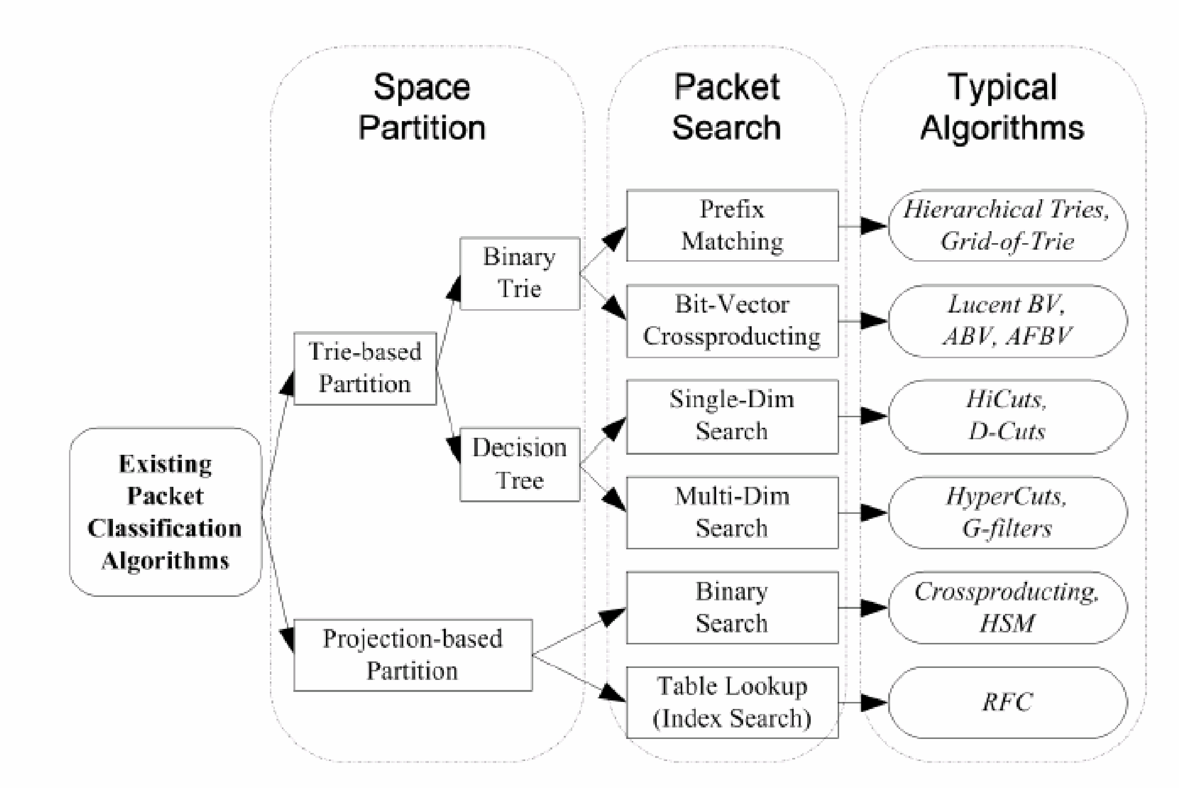
- Algorithm Categorization (1)



Categorization Based on Packet Search Data-structures [17]

1.3 Existing Algorithmic Solutions

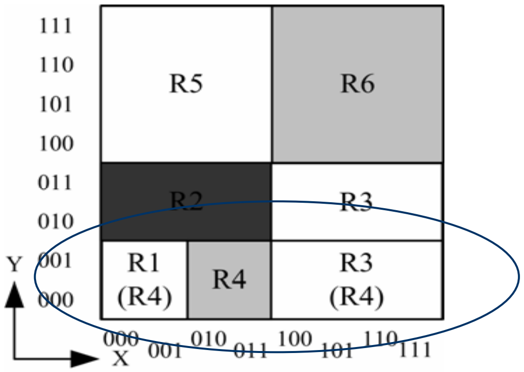
- Algorithm Categorization (2)



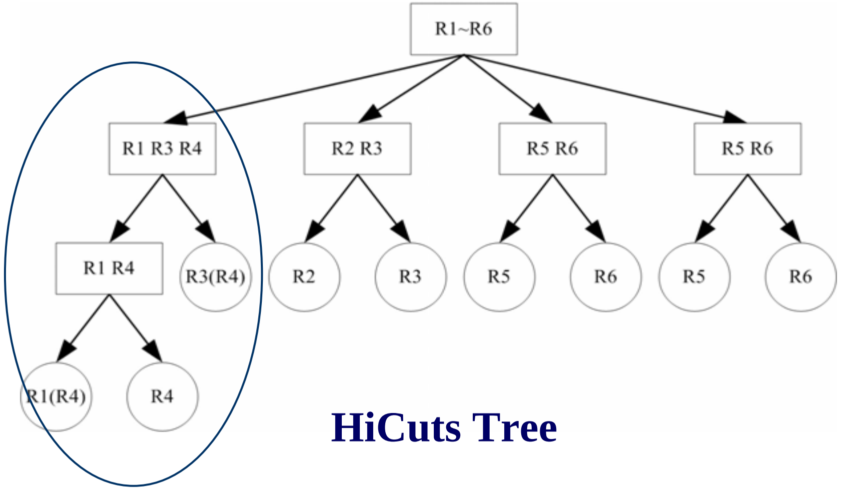
Categorization Based on Space Partition

1.4 Novel Algorithms: D-Cuts

- Dynamic Cuttings: Ideas

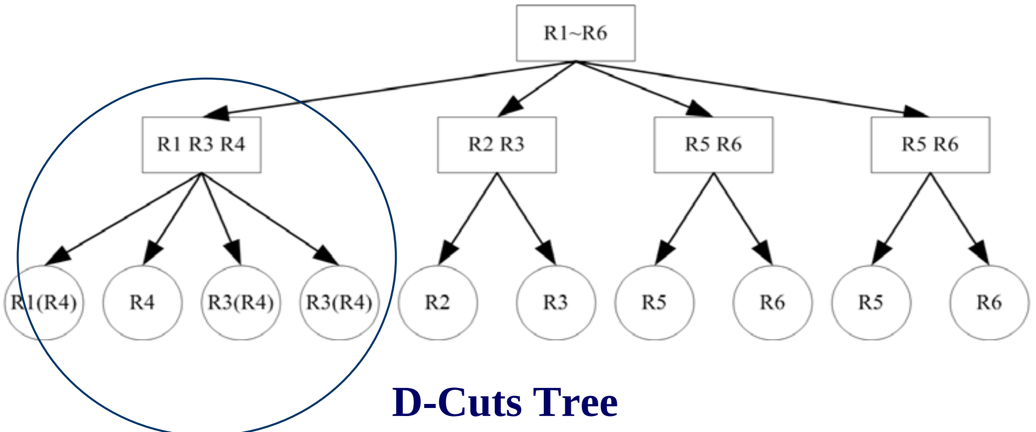


Rules



HiCuts Tree

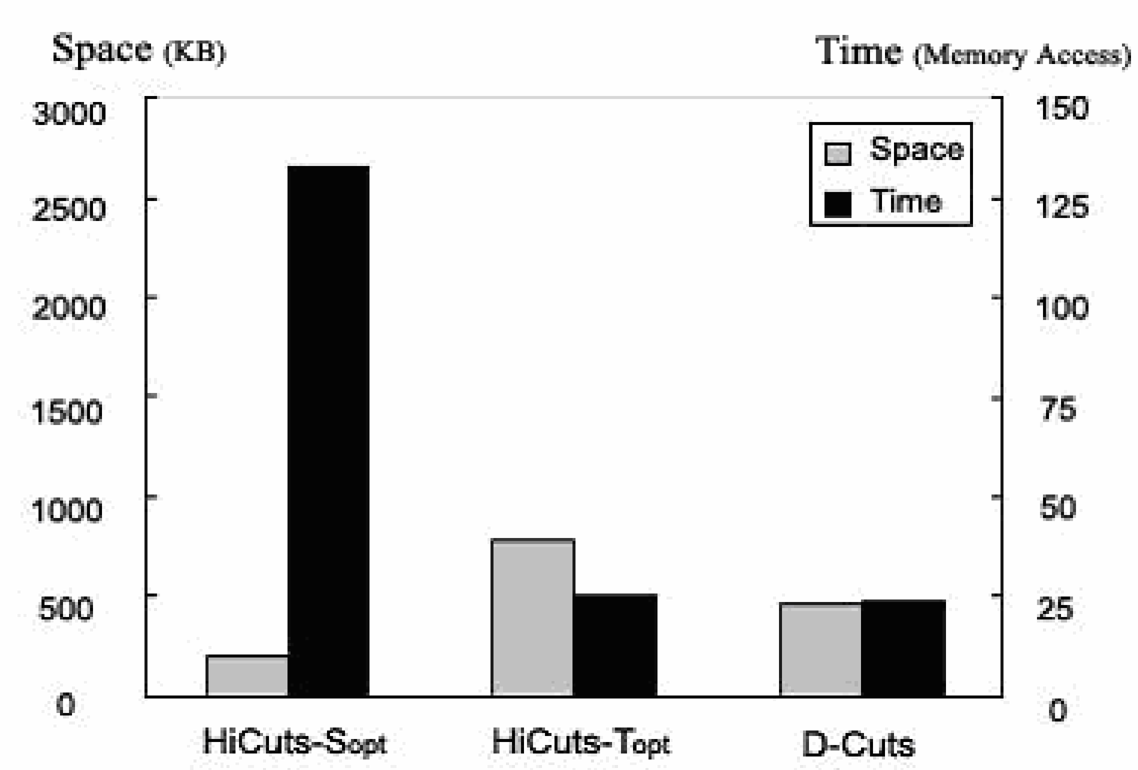
If most traffic matches {R1, R3, R4}, i.e. goes through subspace {X(000:111), Y(000:001)} then we can rebuild the HiCuts tree to cut down the worst case depth.



D-Cuts Tree

1.4 Novel Algorithms: D-Cuts

- Dynamic Cuttings: Performance

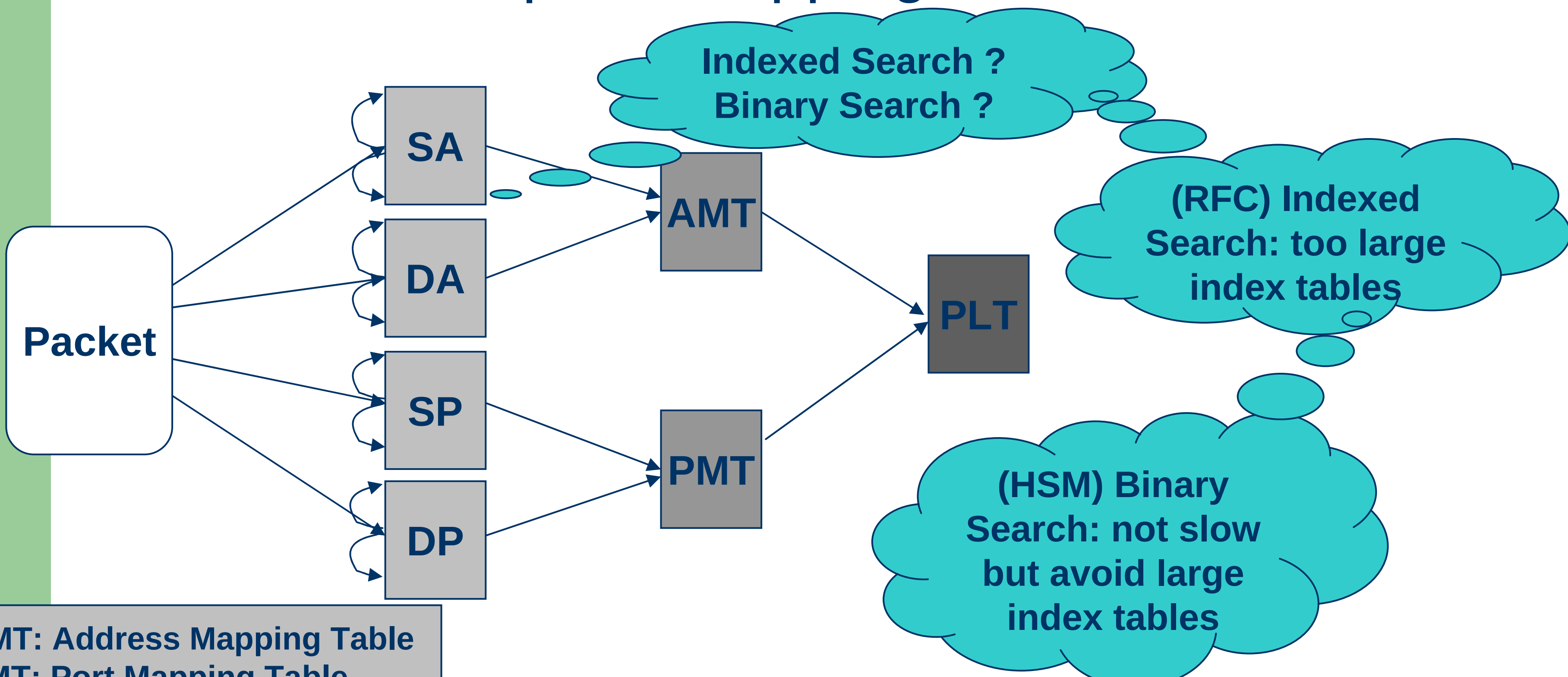


Sopt: Optimized for space

Topt: Optimized for time

1.4 Novel Algorithms: HSM

- Hierarchical Space Mapping: Ideas



AMT: Address Mapping Table
PMT: Port Mapping Table
PLT: Policy Lookup Table

1.4 Novel Algorithms: HSM

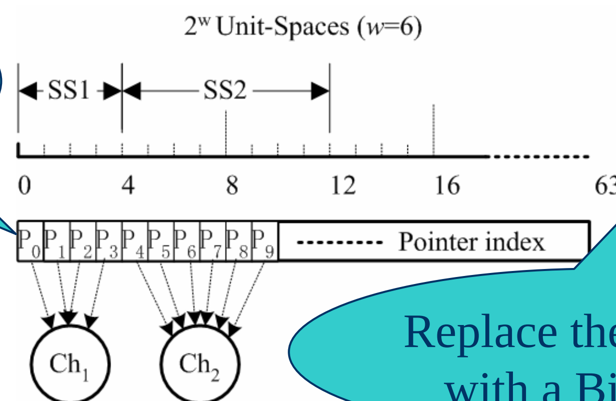
- Hierarchical Space Mapping: Performance

Classifiers	Number of rules	RFC(kB)	HSM(kB)	Percentage Improved
FW1	68	802	41	95%
FW2	136	838	111	87%
FW3	340	1,186	262	78%
CR1	500	1,060	119	89%
CR2	1,000	2,122	923	61%
CR3	1,535	3,454	1,947	44%
CR4	1,945	6,320	3,957	37%

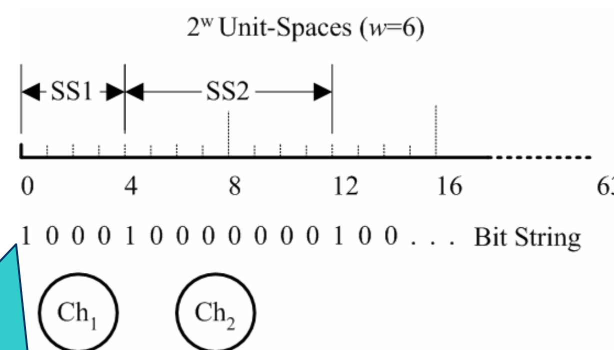
1.4 Novel Algorithms: sBits

- Shifting Bits: Ideas

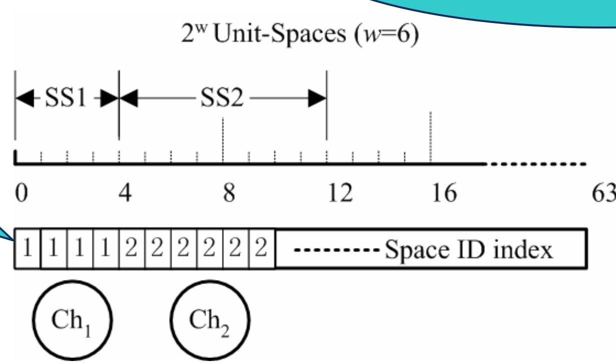
Pointer array: Too Large



Replace the pointers with a Bit string



Replace the pointers with Indexes



Note:

- 32:1 compression rate
- No additional memory access
- Hardware supported

1.4 Novel Algorithms: sBits

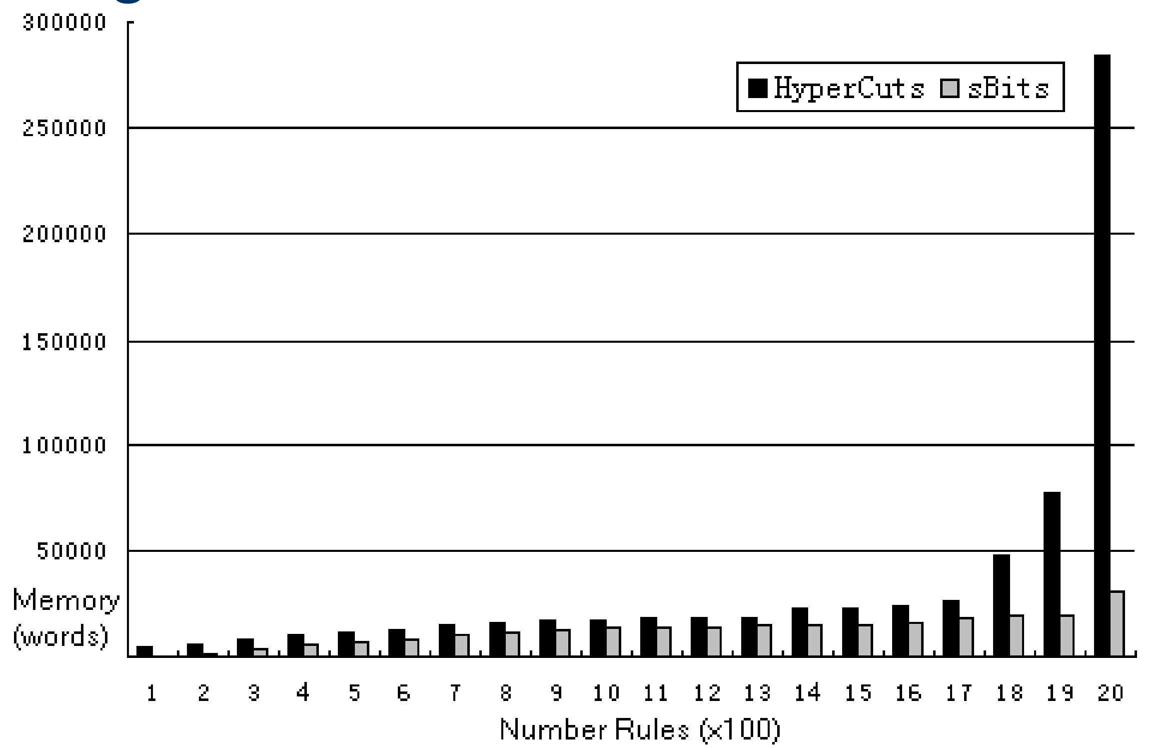
- Shifting Bits: Performance

	No. Rules	HiCuts	HyperCuts	sBits
FW1	68	5,443	35,401	420
FW2	136	10,779	69,782	924
FW3	340	24,645	172,932	2,331
CR1	500	29,409	89,005	3,612
CR2	1000	979,736	871,541	28,287
CR3	1530	13,606,858*	480,225	29,204
CR4	1945	5,928,724*	672,442	43,183

sBits vs. HiCuts/HyperCuts: memory usage (Unit: 32-bit long-word)

1.4 Novel Algorithms: sBits

- Shifting Bits: Performance



sBits vs. HyperCuts: memory usage against rulesets of different size

1.4 Novel Algorithms: Summary

- Tree-based Algorithms (HiCuts, D-Cuts)
  - Memory efficient
  - No explicit worst-case bound, not fast enough
- Table-based Algorithms (RFC, HSM)
  - Fast search speed
  - Not memory efficient
- Hybrid Algorithms (sBits)
  - Combine the advantages of several different approaches.
  - Maybe hard to implement (too complicated)

Outline

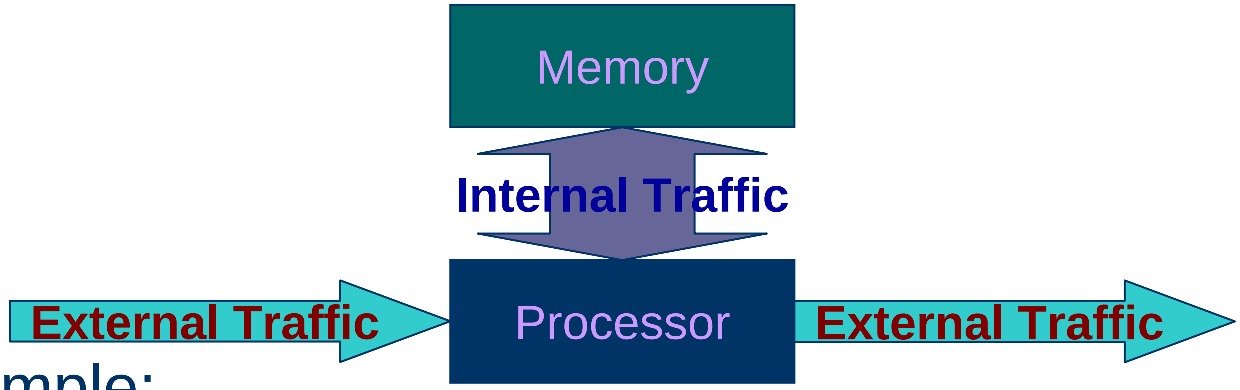
- Algorithm Study
  - Understanding Packet Classification
  - Worst-case Complexity Analysis
  - Existing Algorithmic Solutions
  - Our Novel Algorithms
- Network Processor Implementation
  - Current Hardware Limits
  - External & Internal Traffics
  - Intel IXP Implementation
- Summary

2.1 Current Hardware Limits

- TCAM
  - Board area
  - Power
  - Range matching
- ASIC/FPGA
  - R&D cost
  - Update
- General Purpose CPU
  - Continuity of both time and space
- Network Processor
  - Highly integrated processing units
  - Date plane & Control plane
  - Handle rarely associative network traffics

2.2 External & Internal Traffics

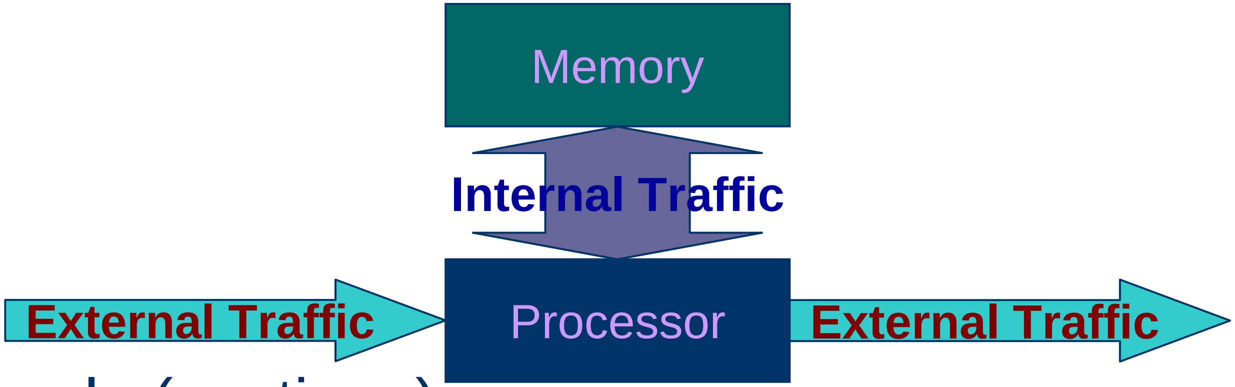
- Traffic In a Router



- Example:
  - Assume: 1 rule = 64 Byte in Memory
  - Assume: 1 packet= 64 Byte going through Processor
  - By Linear Search: process 1 packet needs to read 1K rules in worst-case.
  - (Internal Traffic) : (External Traffic) = 1000:1

2.2 External & Internal Traffics

- Traffic In a Router



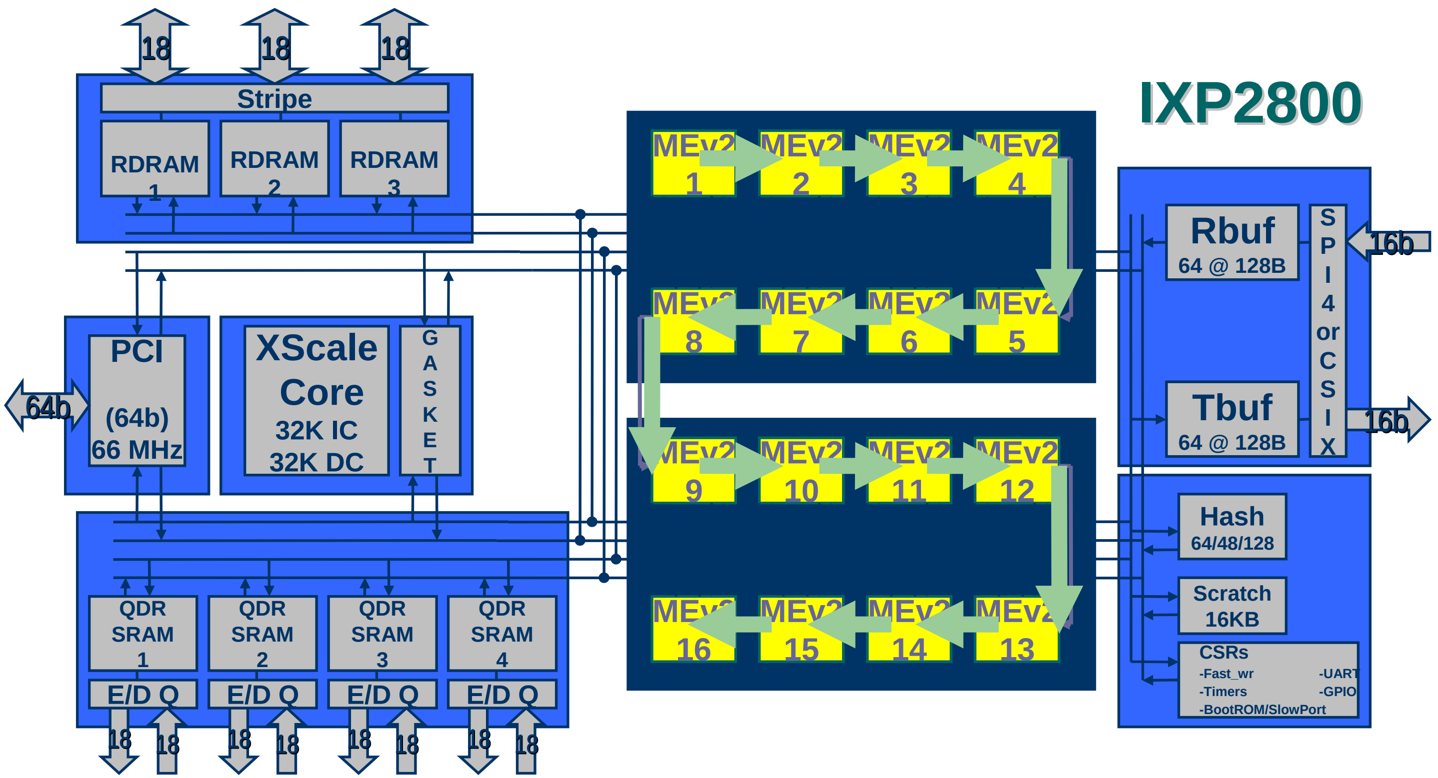
- Example (continue):
  - Assume SRAM Bandwidth in NP = 20GByte/s
  - If (Internal Traffic) : (External Traffic) = 1000:1
    - External Traffic < (20G/1000) = 20MByte/s (160Mbps)
  - Else if (Internal Traffic) : (External Traffic) = 40:1
    - External Traffic < (20G/40) = 0.5GByte/s (4Gbps)

2.2 External & Internal Traffics

- Existing Algorithms (dealing with 2,000 rules)
  - Table-based Algorithms:
    - (Internal Traffic) : (External Traffic) = 1:1~5:1
    - Best temporal performance
    - Require up to 30MB SRAM
  - Tree-based Algorithms:
    - (Internal Traffic) : (External Traffic) = 20:1~30:1
    - Require less than 10MB SRAM
    - Unstable performance, no worst-case bound

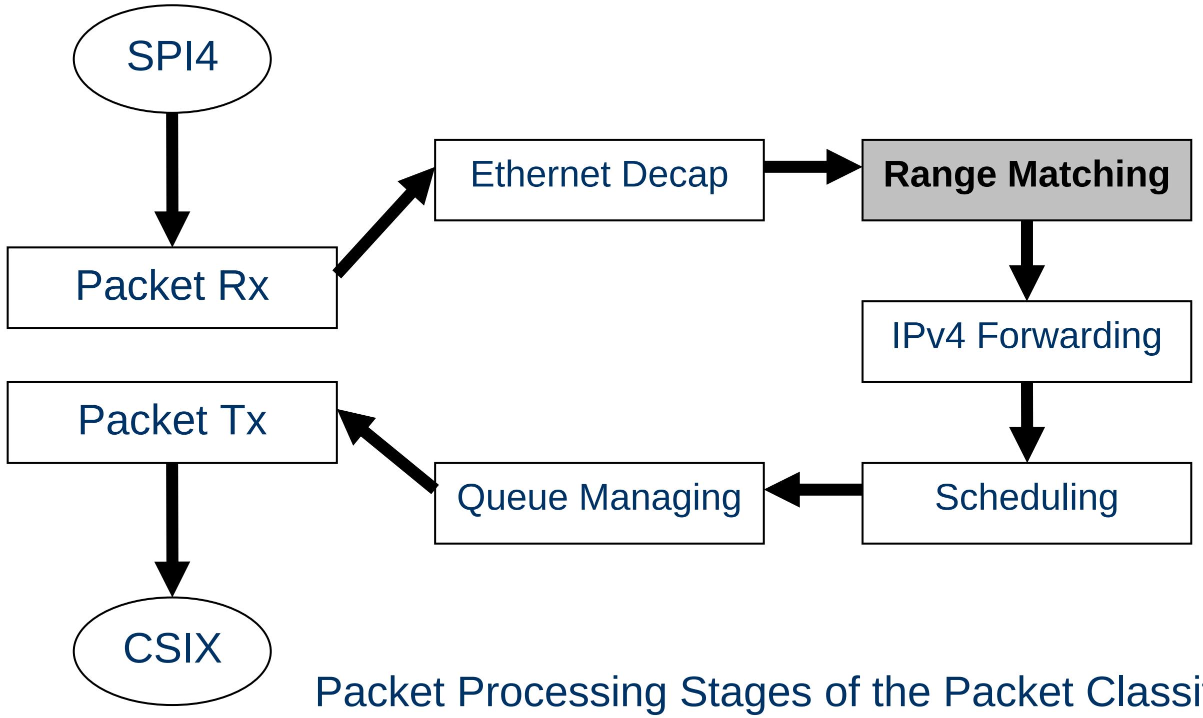
2.3 Intel IXP Implementation

- Intel Network Processor Architecture



2.3 Intel IXP Implementation

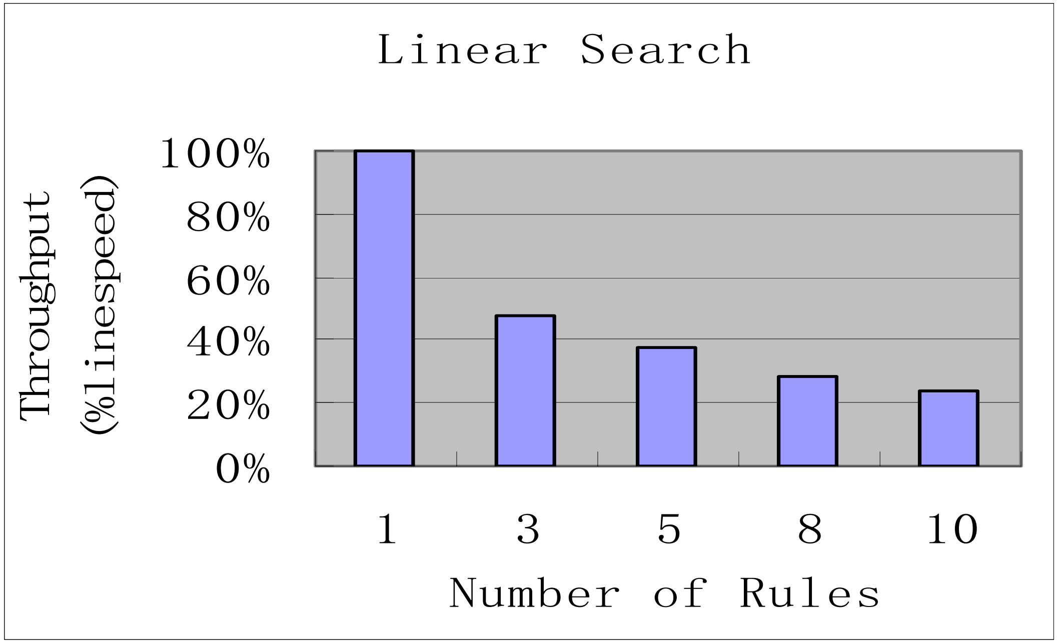
- IXP2xxx Packet Processing Stages



Packet Processing Stages of the Packet Classification Application.
Packet classification algorithms are running in Rage Matching PPS.

2.3 Intel IXP Implementation

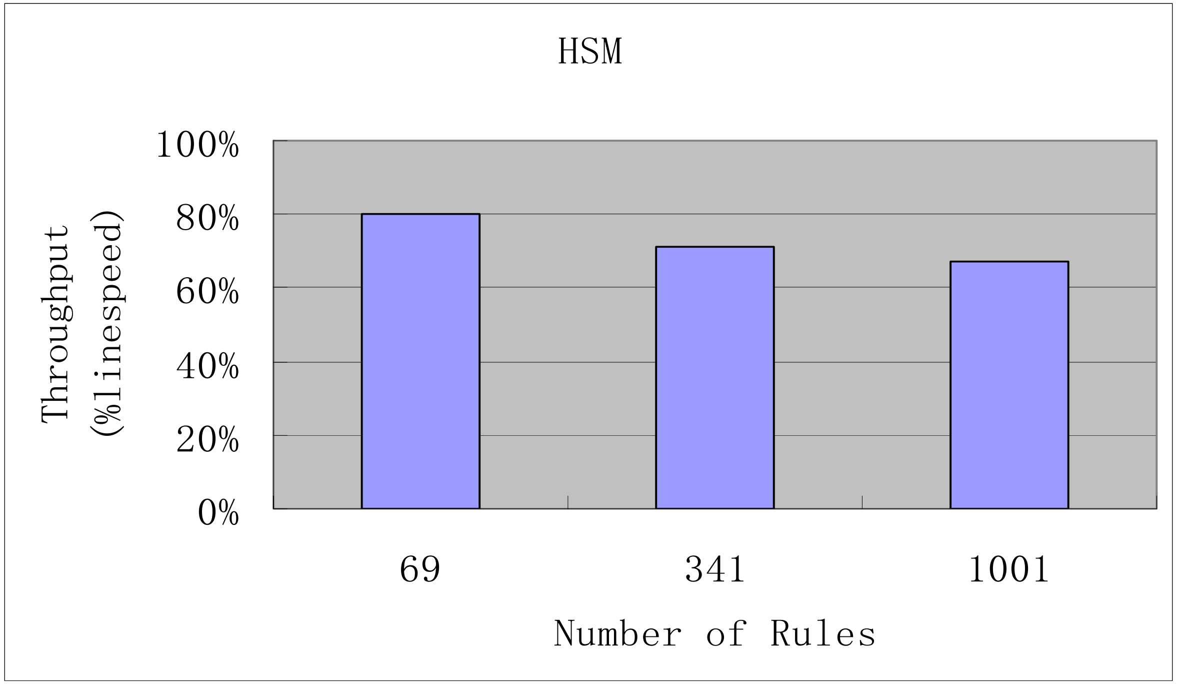
- Simulation Result: Linear Search



Performance Evaluation of Linear Search Algorithm. Each incoming packet just matches the default rule, so that the worst-case performance is obtained. Deterministic worst-case bound: O(*N*).

2.3 Intel IXP Implementation

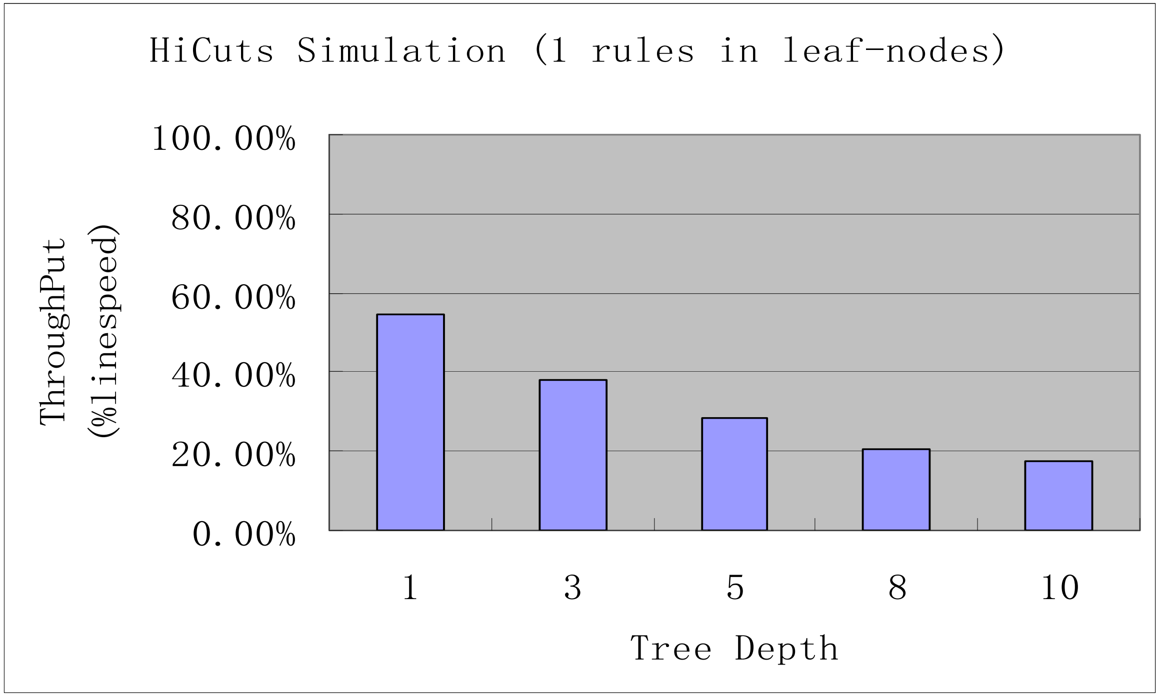
- Simulation Result: HSM



Performance Evaluation of HSM Algorithm. Deterministic worst-case bound: O(log*N*).

2.3 Intel IXP Implementation

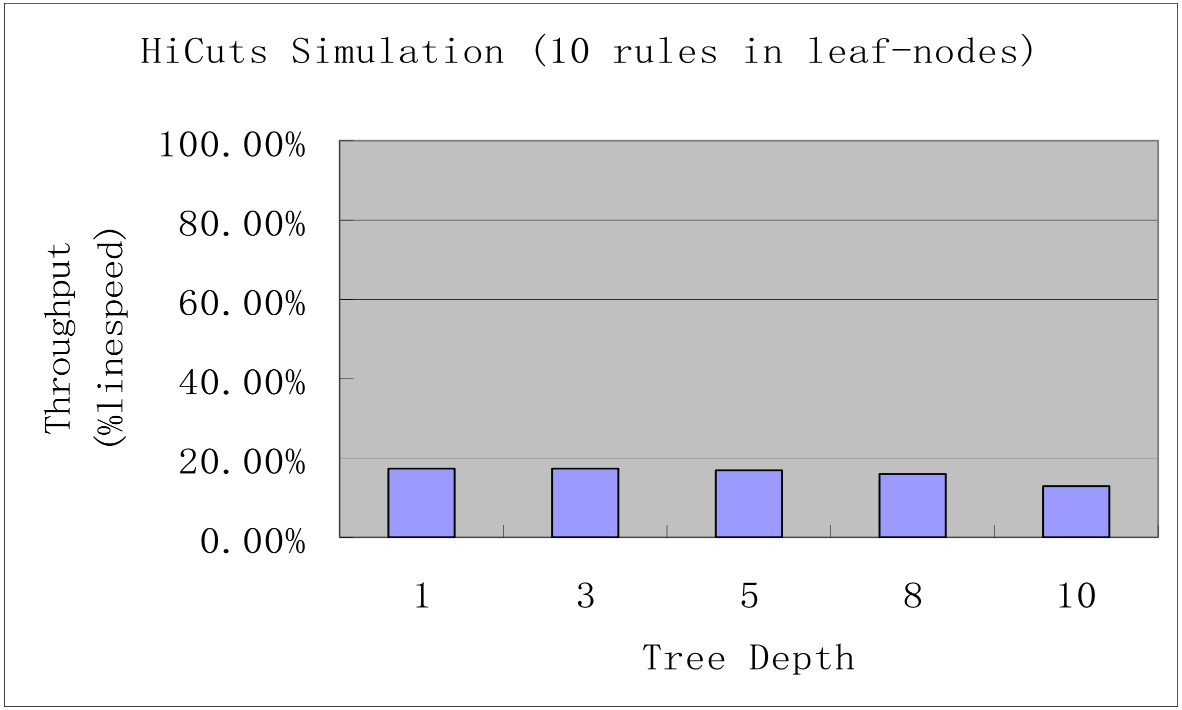
- Simulation Result: HiCuts (worst-case path)



Performance Evaluation of HiCuts Algorithm. Non-deterministic worst-case bound. 1k rules often need a 10-level decision tree.

2.3 Intel IXP Implementation

- Simulation Result: HiCuts (worst-case path)



And what's more, in the worst-case, it often needs up to 10 times of linear searches after tracing down the decision tree.

Outline

- Algorithm Study
  - Understanding Packet Classification
  - Worst-case Complexity Analysis
  - Existing Algorithmic Solutions
  - Our Novel Algorithms
- Network Processor Implementation
  - Current Hardware Limits
  - External & Internal Traffics
  - Intel IXP Implementation
- Summary

3 Summary

- No single algorithm will perform well for all cases:
  - We search for algorithms that are “fast enough” and use “not too much” memory.
  - Search Speed should be guaranteed in the worst-case.
- Hardware limits require flexible algorithms:
  - Designing an effective algorithm should consider the features and limits of the hardware: e.g. SRAM size…
  - Implementation of an algorithm should make full use of all hardware units: e.g. Local Memory for Port Indexing…

References

[1] P. Gupta and N. McKeown, “Packet Classification Using Hierarchical Intelligent Cuttings”, Proc. Hot Interconnects, 1999

[2] M.H. Overmars and A.F. van der Stappen, “Range Searching and Point Location among Fat Objects”, J. of Algorithms, 21(3), 1996.

[3] P. Gupta and N. McKeown, “Packet Classification on Multiple Fields”, Proc. ACM SIGCOMM 99, 1999.

[4] B. Xu, D. Jiang, J. Li, “HSM: A Fast Packet Classification Algorithm”, Proc. IEEE 19th International Conference on Advanced Information Networking and Applications (AINA), 2005.

[5] V. Srinivasan, et al., "Fast and Scalable Layer Four Switching“, Proc. ACM SIGCOMM, 1998.

[6] F. Baboescu, and G. Varghese, "Scalable Packet Classification“, Proc. ACM SIGCOMM, 2001.

[7] S. Singh, F. Baboescu, and G. Varghese, "Packet Classification Using Multidimensional Cutting“, Proc. ACM SIGCOMM, 2003.

[8] F. Baboescu, S. Singh, and G. Varghese, “Packet Classification for Core Routers: Is There An Alternative To CAMs?” Proc. INFOCOM, 2003.

[9] V.Srinivasan, S.Suri, and G.Varghese, “Packet Classification using Tuple Space Search”, Proc. SIGCOMM, 1999.

[10] S. Singh and F. Baboescu, “Packet Classification Repository”, http://ial.ucsd.edu/classification

[11] A. Feldman and S. Muthukrishnan, “Tradeoffs for Packet Classification”, Proc. INFOCOM, 2000.

[12] T. Lakshman and D. Stiliadis, “High Speed Policy-based Packet Forwarding using Efficient Multi-dimensional Range Matching”, Proc. SIGCOMM,1998.

[13] T.Y.C Woo, “A Modular Approach to Packet Classification: Algorithms and Results”, Proc. IEEE INFOCOM, 2000.

[14] F. Geraci, M. Pellegrini, and P. Pisati, “Packet Classification via Improved Space Decomposition Techniques”, Proc. IEEE INFOCOM, 2005.

[15] Y. Qi and J. Li, “Dynamic Cuttings: Packet Classification with Network Traffic Statistics”, Proc. 3rd International Trusted Internet Workshop (TIW), 2004.

[16] P. Gupta and N. McKewon, “Algorithms for Packet Classification”, IEEE Network, March/April 2001, 2001.

[17] D. E. Taylor , “Survey & Taxonomy of Packet Classification Techniques”, Washington University in Saint-Louis, US, 2004.

[18] M.E. Kounavis, A. Kumar, H. Vin, R. Yavatkar and A.T. Campbell, “Directions in Packet Classification for Network Processors”, Proc. Second Workshop on Network Processors (NP2), 2003.

Thanks

junl@tsinghua.edu.cn