Algorithms for Routing Lookups and Packet Classification

October 3, 2000

Pankaj Gupta
Department of Computer Science
Stanford University
pankaj@stanford.edu
http://www.stanford.edu/~pankaj

High Level Outline

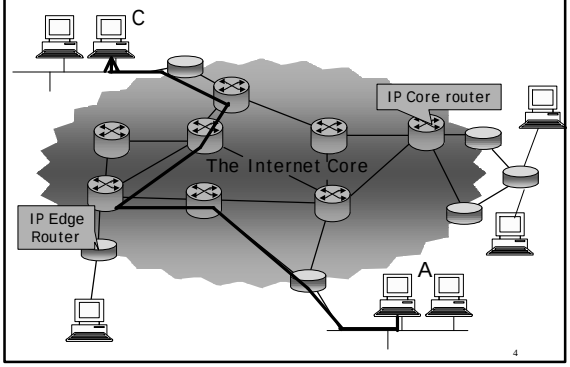
Part I. Routing Lookups
- Two lookup algorithms

Part II. Packet Classification
- One classification algorithm

Routing Lookups: Outline

- Introduction
  - Background
  - Motivation
  - Definition of the problem
- Algorithm #1
- Algorithm #2
- Conclusions of Part I

Internet: Mesh of Routers



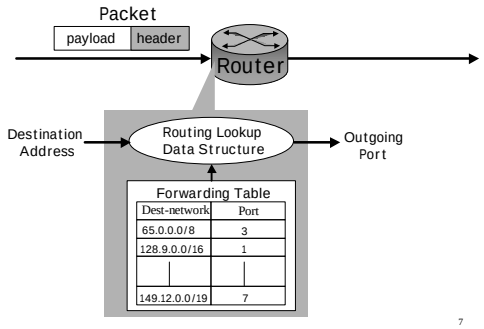
Inside an IP Router

1. Accept packet arriving on an incoming link.
2. Lookup packet destination address in the forwarding table, to identify outgoing port(s).
3. Manipulate packet header: e.g., decrement TTL, update header checksum.
4. Send packet to the outgoing port(s).
5. Buffer packet in the queue.
6. Transmit packet onto outgoing link.

Part I of the Talk

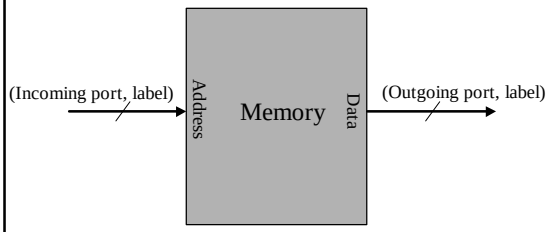
Fast and efficient algorithms that an IP router uses to lookup the destination address in order to decide where to forward the packets next.

Forwarding Engine



7

The Search Operation is not a Direct Lookup



IP addresses: 32 bits long ⇒ 4G entries

8

The Search Operation is also not an Exact Match Search

Exact match search: search for a key in a collection of keys of the same length.

Relatively well studied data structures:

- Hashing
- Balanced binary search trees

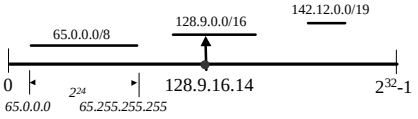
9

Example Forwarding Table

Destination IP Prefix	Outgoing Port
65.0.0.0/8	3
128.9.0.0/16	1
142.12.0.0/19	7

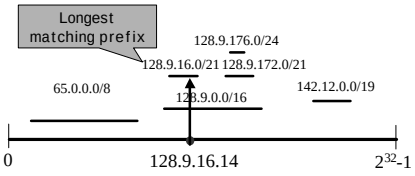
Prefix length

IP prefix: 0-32 bits



10

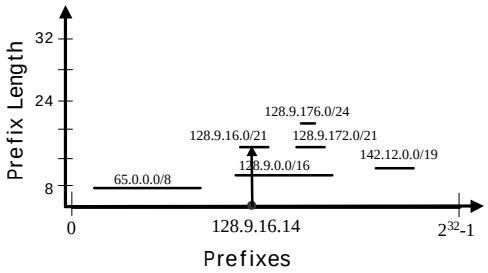
Prefixes can Overlap



Routing lookup: Find the longest matching prefix (aka the most specific route) among all prefixes that match the destination address.

11

Difficulty of Longest Prefix Match



12

Metrics for Lookup Algorithms

- Preprocessing time
- Storage requirements
- Lookup rate
- Update time

13

Lookup Rate Required

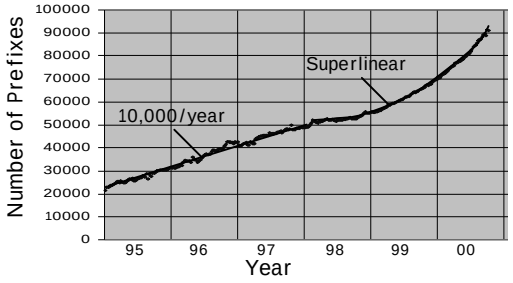
Year	Line	Line-rate (Gbps)	40B packets (Mpps)
1998-99	OC12c	0.622	1.94
1999-00	OC48c	2.5	7.81
2000-01	OC192c	10.0	31.25
2002-03	OC768c	40.0	125

31.25 Mpps ⇒ 33 ns

DRAM: 50-80 ns, SRAM: 5-10 ns

14

Size of the Forwarding Table



Source: http://www.telstra.net/ops/bgptable.html

15

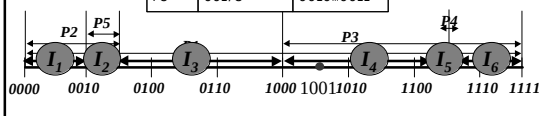
Routing Lookups: Outline

- Introduction
- Algorithm #1
  - Motivation
  - Details
  - Performance
- Algorithm #2
- Conclusions

16

Binary Search on Prefix Intervals[1]

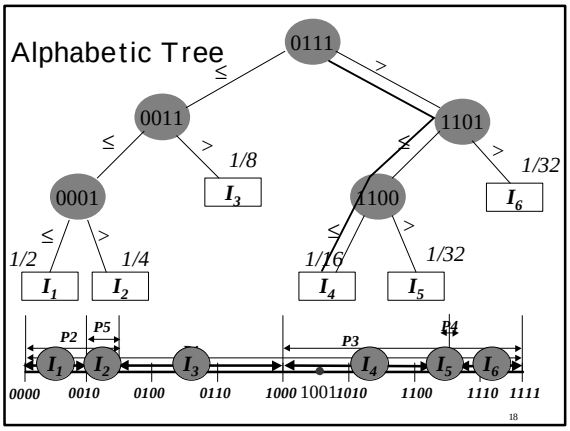
	Prefix	Interval
P1	/0	0000…1111
P2	00/2	0000…0011
P3	1/1	1000…1111
P4	1101/4	1101…1101
P5	001/3	0010…0011



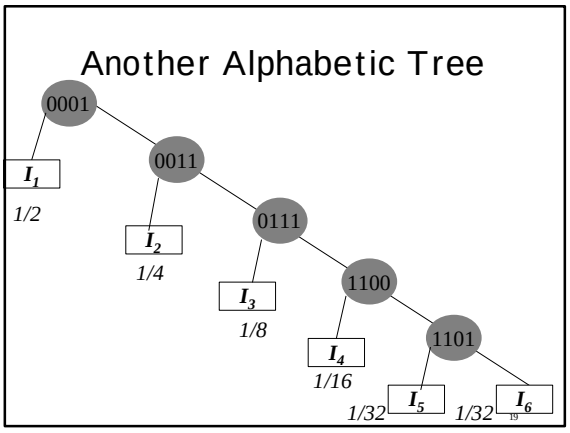
1. [Lampson et al., Proc. Infocom, 1998]

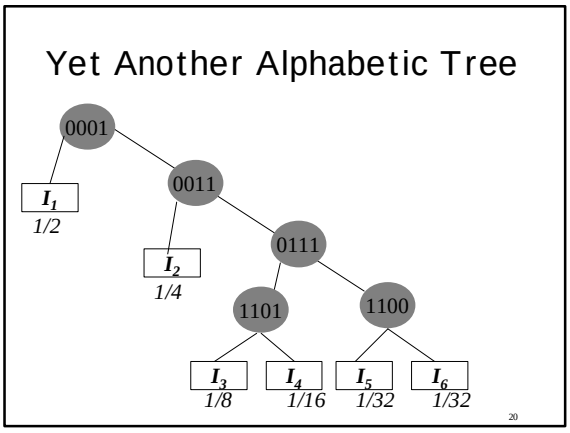
17

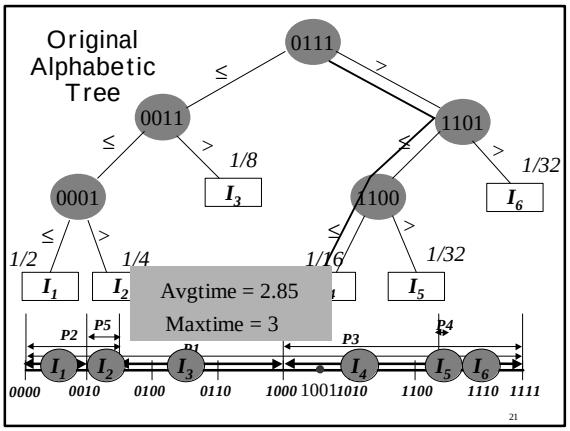
Alphabetic Tree

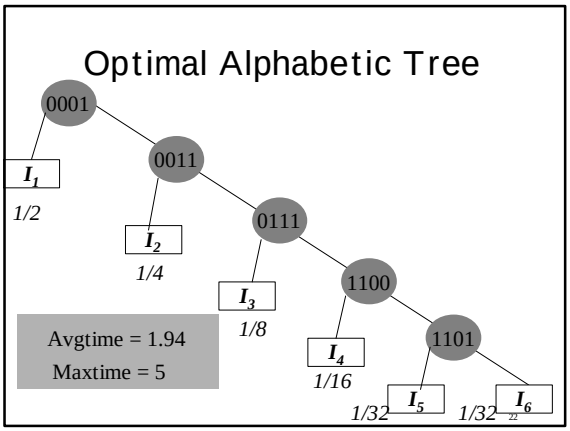


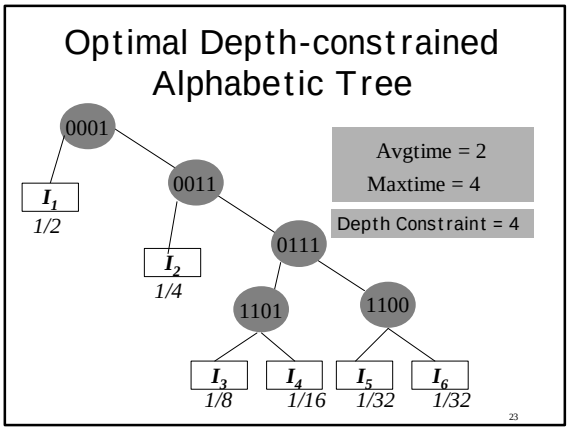
18

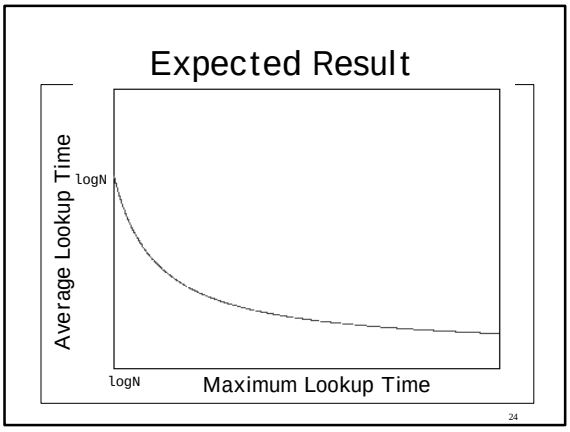

Another Alphabetic Tree
0001
0011
0111
1100
1101
I1
1/2
I2
1/4
I3
1/8
I4
1/16
I5
1/32
I6
1/32
19


Yet Another Alphabetic Tree
0001
0011
0111
1101
1100
I1
1/2
I2
1/4
I3
1/8
I4
1/16
I5
1/32
I6
1/32
20


Original Alphabetic Tree
0111
0011
1101
0001
1100
≤
>
1/8
1/32
1/2
1/4
1/16
1/32
Avgtime = 2.85
Maxtime = 3
P2
P5
P1
P3
P4
0000
0010
0100
0110
1000
1001
1010
1100
1110
1111
21


Optimal Alphabetic Tree
0001
0011
0111
1100
1101
I1
1/2
I2
1/4
I3
1/8
I4
1/16
I5
1/32
I6
1/32
Avgtime = 1.94
Maxtime = 5
22


Optimal Depth-constrained Alphabetic Tree
0001
0011
0111
1101
1100
I1
1/2
I2
1/4
I3
1/8
I4
1/16
I5
1/32
I6
1/32
Avgtime = 2
Maxtime = 4
Depth Constraint = 4
23


Expected Result
Average Lookup Time
logN
logN
Maximum Lookup Time
24

Problem Statement

access time to reach leaf i

Depth constraint

Minimize Average Lookup Time = $\sum_i l_i p_i$ s.t. $l_i \le D \ \forall i$

probability of accessing leaf i

Previous Work:

- Depth-constrained Huffman trees
- Optimal solutions
  [Larmore and Przytycka94] O(nDlogn) with large constant factors.

25

Goal: Near-optimal Depth-constrained Alphabetic Tree

Why near-optimal ?

- Simpler to find than an optimal solution.
- Probabilities are approximate.

26

Algorithm MinDPQ

Fact [Yeung91]: Given $\{p_k\}$, can choose $\{l_k\}$ such that: $H(p) \le C < H(p) + 2$

$$C = avgLookupTime$$
$$H(p) = -\sum_i p_i \log p_i$$

$$l_k = \begin{cases} \lceil -\log_2 p_k \rceil & k = 1, n \\ \lceil -\log_2 p_k \rceil + 1 & 1 < k < n \end{cases}$$

But:

$$p_k < 2^{-D} \Rightarrow l_k > D$$

Depth constraint (D) violated

27

Algorithm MinDPQ (contd.)

Original distribution $\{p_k\}$, possibly $p_{min} < 2^{-D}$

Transform Probabilities

Transformed distribution $\{q_k\}$, $q_{min} \ge 2^{-D}$

Explicit Solution

$$q_k^* = \max\left\{\frac{p_k}{m}, 2^{-D}\right\} \text{ where } m \text{ is s.t. } \sum_k q_k^* = 1$$

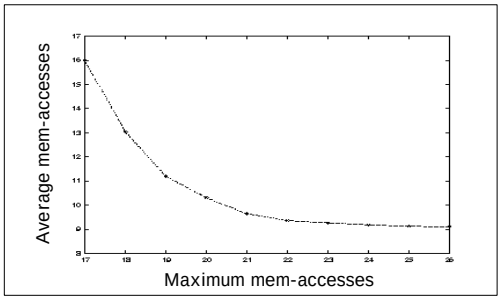
Within 2 memory accesses of optimal!

μ can be found in O(nl

$$C^* = \sum_k p_k l_k^* \le D(p\|q) + H(p) + 2 \le C^{opt} + 2$$

28

Algorithm MinDPQ: Experimental Results



29

Summary of Algorithm MinDPQ

- A practical algorithm to minimize average lookup time while simultaneously keeping maximum lookup time bounded.
- Provably within two memory accesses of the optimal algorithm.

30

Routing Lookups: Outline

- Introduction
- Algorithm #1
- Algorithm #2
  - Motivation
  - Previous Work
  - Details
  - Performance
- Conclusions

31

What if we are simply interested in the fastest worst-case lookup algorithm?

32

Previous Work

- K. Sklower. "A tree-based packet routing table for Berkeley unix," Proc. Usenix, pp 93-9, 1991.
- W. Doeringer, G. Karjoth and M. Nassehi. "Routing on longest-matching prefixes," IEEE/ACM Transactions on Networking, vol. 4, no. 1, pp 86-97, 1996.
- M. Degermark, A. Brodnik, S. Carlsson, S. Pink. "Small forwarding tables for fast routing lookups," Proc. Sigcomm, pp 3-14, 1997.
- M. Waldvogel, G. Varghese, J. Turner, B. Plattner. "Scalable high-speed IP routing lookups," Proc. Sigcomm, pp 25-36, 1997.

33

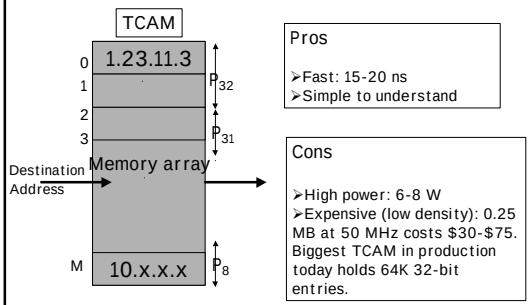
Previous Work (contd.)

- B. Lampson, V. Srinivasan, G. Varghese. "IP lookups using multiway and multicolumn search," Proc. Infocom, vol. 3, pp 1248-56, 1998.
- V. Srinivasan, G.Varghese. "Fast IP lookups using controlled prefix expansion", Sigmetrics, 1998.
- S. Nilsson, G. Karlsson. "IP-address lookup using LC-tries," IEEE JSAC, vol. 17, no. 6, pp 1083-92, 1999.

Fastest: 298 ns (3.3 Mpps) with 2 MB for forwarding table with 38K prefixes (300 MHz Pentium-II with 512 KB cache)

34

Lookups with Ternary CAM



Pros

- Fast: 15-20 ns
- Simple to understand

Cons

- High power: 6-8 W
- Expensive (low density): 0.25 MB at 50 MHz costs $30-$75. Biggest TCAM in production today holds 64K 32-bit entries.

35

Motivation: Speed and Simplicity

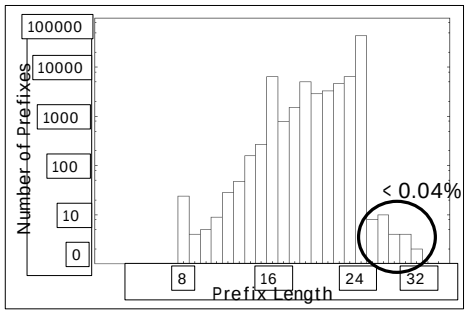
Optimized for implementation in dedicated hardware:

- Routing lookup function fairly well-defined
- Seems necessary for highest performance anyway

Goal: One routing lookup every memory access

36

Key Idea #1



MAE-EAST routing table (source: www.merit.edu)

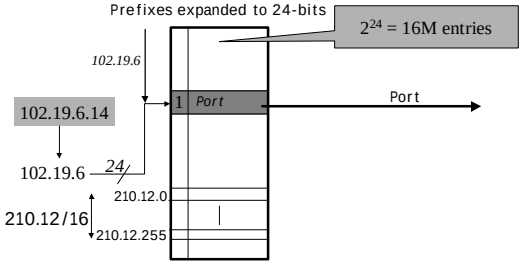
37

Key Idea #2

- Memory is cheap (approx $1/MByte), and getting cheaper
  - Makes sense to use memory inefficiently to gain speed

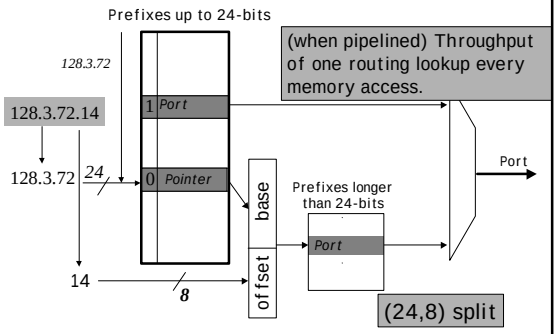
38

Routing Lookups in Hardware



39

Routing Lookups in Hardware



40

Routing Lookups in Hardware

Pros	Cons
➢Simple hardware implementation ➢20 Mpps with 50ns DRAM ➢Unlimited number of prefixes less than or equal to 24 bits long	➢Large memory required (7-33 MB) ➢Depends on prefix-length distribution ➢Slow worst-case updates

41

Routing Lookups: Outline

- Introduction
- Algorithm #1
- Previous work
- Algorithm #2
- Conclusions

42

Summary of Contributions

- Algorithm to minimize average lookup time while keeping worst case bounded: of independent interest in information theory.
- Hardware lookup algorithm: first proposed algorithm that performs a routing lookup in one memory access.

43

High Level Outline

Part I. Routing Lookups
- Two lookup algorithms

Part II. Packet Classification
- One classification algorithm

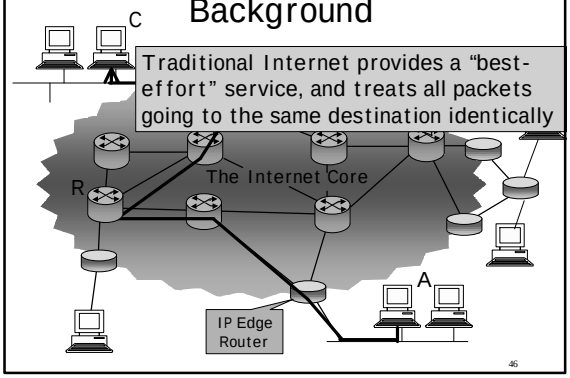
44

Packet Classification: Outline

- Introduction
  - Background
  - Motivation
  - Problem definition
- Previous work
- Proposed algorithm
- Conclusions

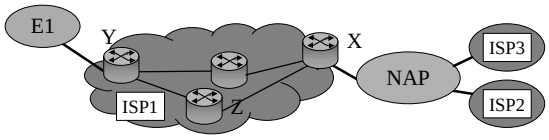
45

Background



46

Motivation: Desire for Additional Services



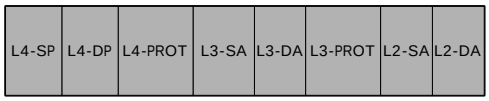
Service	Example
Differentiated Service	Ensure that traffic from ISP2 is given higher priority over traffic from ISP3.
Packet Filtering	Deny all web traffic from ISP3 at interface X.
Policy-based routing	Ensure that al… sent via interface Z.

Src IP address

Transport Layer Protocol

47

Packet Header Fields



Transport layer header | Network layer header | MAC header

DA = Destination address
SA = Source address
PROT = Protocol
SP = Source port
DP = Destination port

L2 = layer 2 (e.g., Ethernet)
L3 = layer 3 (e.g., IP)
L4 = layer 4 (e.g., TCP)

48

Multi-field Packet Classification

	Field 1	Field 2	…	Field k	Action
Rule 1	5.3.40.0/21	2.13.8.11/32	…	UDP	A_1
Rule 2	5.168.3.0/24	152.133.0.0/16	…	TCP	A_2
…	…	…	…	…	…
Rule N	5.168.0.0/16	152.0.0.0/8	…	ANY	A_N

Example: packet (5.168.3.32, 152.133.171.71, … , TCP)

Packet Classification: Find the action associated with the highest priority rule matching an incoming packet header.

49

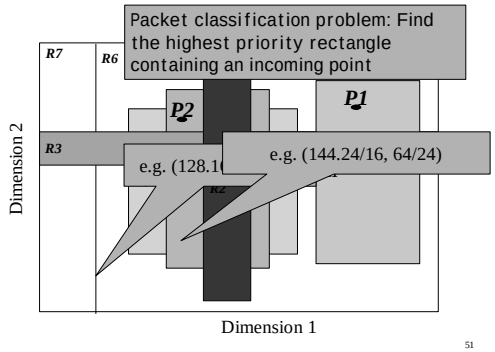
Routing Lookup: Instance of 1D Classification

- One-dimension (destination address)
- Forwarding table ≡ classifier
- Routing table entry ≡ rule
- Outgoing port ≡ action
- Prefix-length ≡ priority

Example of multi-dimensional classification: Firewall for packet-filtering

50

Geometric Interpretation



51

Goal: Packet Classification Algorithms

- Small preprocessing time
- Low storage requirements
- High speed
- Scale to multiple header fields

52

Packet Classification: Outline

- Introduction
- Previous work
- Proposed algorithm
- Conclusions

53

Previous Work

- T.V. Lakshman, D. Stiliadis. "High-speed policy-based packet forwarding using efficient multi-dimensional range matching," Proc. Sigcomm, pp 191-202, 1998.
- V. Srinivasan, S. Suri, G. Varghese and M. Waldvogel. "Fast and scalable layer four switching," Proc. Sigcomm, pp 203-214, 1998.
- V. Srinivasan, G. Varghese, S. Suri. "Packet classification using tuple space search", Proc. Sigcomm, pp 135-146, 1999.

54

Previous Work (contd.)

- M. M. Buddhikot, S. Suri, and M. Waldvogel. "Space decomposition techniques for fast layer-4 switching," PfHSN '99, pp 25-41, 1999.
- A. Feldmann and S. Muthukrishnan. "Tradeoffs for packet classification," Proc. Infocom, vol. 3, pp 1193-202, 2000.
- T. Woo, "A modular approach to packet classification: algorithms and resuts," Proc. Infocom, vol. 3, pp 1203-22, 2000.

55

Previous Algorithms: Summary

- Good for two fields, but do not scale to more than two fields, OR
- Good for very small classifiers (< 50 rules) only, OR
- Have non-deterministic classification time, OR
- Either too slow or consume too much storage

56

Classification Algorithms: Speed vs Storage Tradeoff

Point Location: Lower bounds for N regions in d dimensions.

O(log N) time with $O(N^d)$ storage, or
$O(\log^{d-1}N)$ time with O(N) storage

N = 100, d = 4, N^d = 100 MBytes and
$\log^{d-1}N$ = 350 memory accesses

57

Recursive Flow Classification: Motivation

- Lower bounds are achieved by pathological classifier datasets.
- Real-life datasets have structure and redundancy.
- Good heuristics may do better than worst-case bounds for real-life datasets.

Goal: A practical algorithm that exploits the structure of real-life datasets to achieve both high speed and low storage requirements.

58

Packet Classification: Outline

- Introduction
- Previous work
- Algorithm Recursive Flow Classification
  - Motivation
  - Real-life datasets: characteristics and structure
  - Algorithm details
  - Performance
  - Pros and cons
- Conclusions

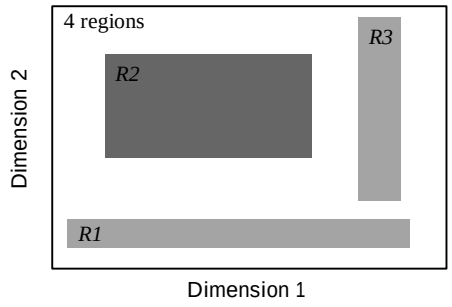
59

Classifier Dataset

- 793 classifiers from 101 ISP and enterprise networks with a total of 41,505 rules
- 40 classifiers: more than 100 rules. Biggest classifier had 1733 rules
- Maximum of 4 fields per rule: source IP address, destination IP address, protocol and destination transport port number

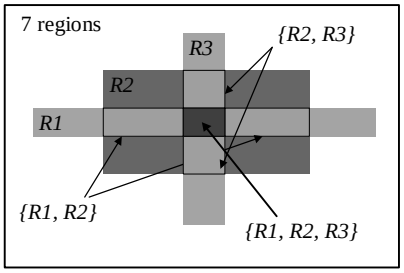
60

Structure of the Classifiers



61

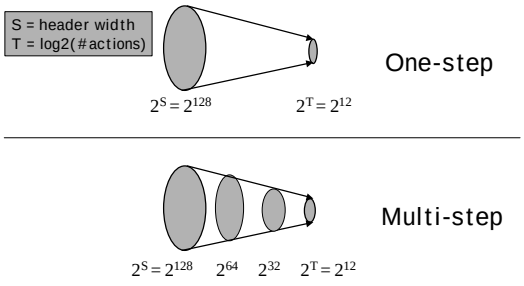
Structure of the Classifiers



dataset: 1733 rule classifier = 4316 distinct regions (worst case is 10^{11} !)

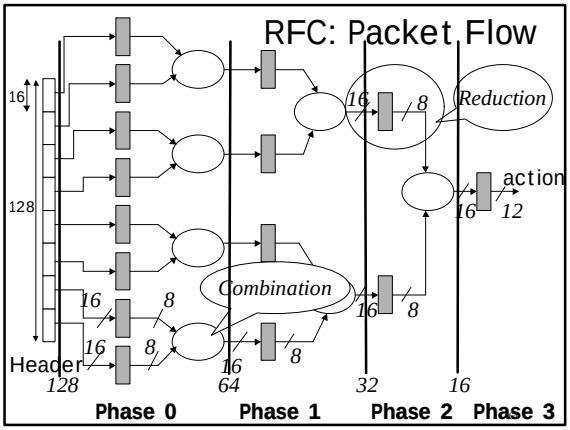
62

Recursive Flow Classification (RFC): Basic Idea



63

RFC: Packet Flow

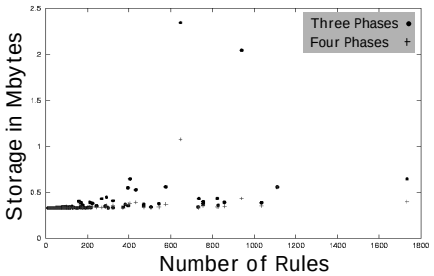


RFC: Classification Speed

Pipelined hardware: 31 Mpps (worst case OC192) using two 0.5Mbyte SRAMs and two 8Mbyte SDRAMs at 125MHz.

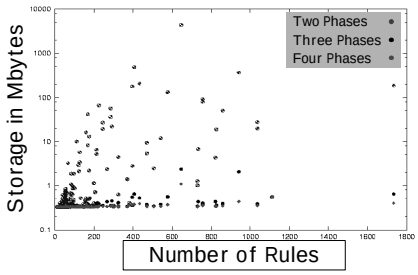
65

RFC: Storage Requirements



66

Two Phase RFC ≡ Crossproducting [Srini98]



67

RFC: Pros and Cons

Pros	Cons
➢Exploits structure of real-life datasets ➢Scales to multiple fields ➢Fast classification (designed for parallel and pipelined accesses)	➢Depends on structure of classifiers ➢Large pre-processing time ➢Slow incremental insertions

68

Packet Classification: Outline

- Introduction
- Previous work
- Proposed algorithm
  - Motivation
  - Real-life classifiers: characteristics and structure
  - Algorithm details
  - Performance
  - Pros and cons
- Conclusions

69

Summary of Contributions on Packet Classification

Recursive Flow Classification: First proposed algorithm that achieves fast multi-field classification and low storage requirements, by deliberately exploiting the structure of real-life datasets.

70

Other Contributions on Packet Classification

- P. Gupta and N. McKeown. "Packet classification using hierarchical intelligent cuttings," Proc. Hot Interconnects VII, August 99. Also in IEEE Micro, pp 34-41, vol. 20, no. 1, Jan/Feb 2000.
- P. Gupta and N. McKeown. "Dynamic algorithms with worst-case performance for packet classification," Proc. IFIP Networking, May 2000.

71

Publications for Algorithms Discussed Here

- P. Gupta, B. Prabhakar, and S. Boyd. "Near-optimal routing lookups with bounded worst case performance," Proc. Infocom, vol. 3, pp 1184-92, March 2000.
- P. Gupta, S. Lin, and N. McKeown. "Routing lookups in hardware at memory access speeds," Proc. Infocom, vol. 3, pp 1241-8, April 1998.
- P. Gupta and N. McKeown. "Packet Classification on Multiple Fields," Proc. Sigcomm, vol. 29, pp 147-60, September 1999.

72

Unrelated Contributions

- P. Gupta and N. McKeown. "Design and implementation of a fast crossbar scheduler," Proc. Hot Interconnects VI, August 98. Also in IEEE Micro, pp 20-28, vol. 19, no. 1, Jan/Feb 1999.
- D. Shah and P. Gupta, "Fast updates on ternary CAMs for packet lookups and classification," Proc. Hot Interconnects VIII, August 2000.

73