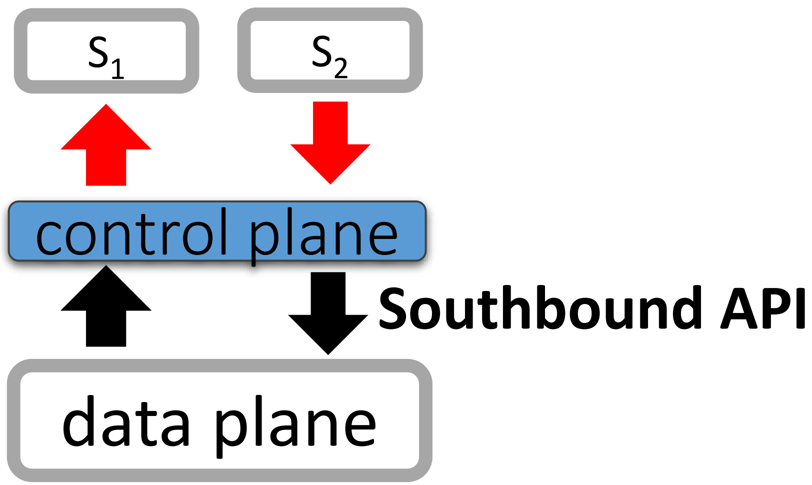
SAX-PAC (Scalable And eXpressive PAcket Classification)

Kirill Kogan	Purdue University and IMDEA Networks
Sergey Nikolenko	Steklov Institute of Mathematics at St. Petersburg and National Research University Higher School of Economics
Ori Rottenstreich	Technion and Mellanox
William Culhane	Purdue University
Patrick Eugster	Purdue University and TU Darmstadt

Outline

- Current state of the art in packet classification
- Impact of structural properties on representation efficiency
- Classifiers as Boolean expressions
- Proposed solutions
- Evaluation
- Summary and future work

Representation of expressiveness on data plane



$$
\begin{aligned}
R_1 &= ([1,3],[4,31],[1,28]) \\
R_2 &= ([4,4],[2,30],[4,27]) \\
R_3 &= ([7,9],[5,21],[3,18])
\end{aligned}
$$

priority

SW-based vs. TCAM-based solutions

SW-based: $N = 4$ rules $K = 2$ fields ~~prefixes~~ ~~ranges~~

$$
\begin{aligned}
R_1 &= (\; 100*, & 001* \;) \\
R_2 &= (\; 1010, & 0001 \;) \\
R_3 &= (\; 000*, & **** \;) \\
R_4 &= (\; 001*, & **** \;)
\end{aligned}
$$

Memory	Lookup time
$O(N)$	$O(log^{k-1}N)$
$O(N^k)$	$O(logN)$

TCAM-based: $N = 3$ rules $K = 3$ prefixes ~~ranges~~

$$
\begin{aligned}
R_1 &= ([1,3],[4,31],[1,28]) \\
R_2 &= ([4,4],[2,30],[4,27]) \\
R_3 &= ([7,9],[5,21],[3,18])
\end{aligned}
$$

Encoding	#TCAM entries
Binary	42+28+50=120
Gray	24+8+32=64

Order-independence

If the rules of a classifier do not ``intersect'', their order is not important.

$$\begin{aligned} R_1 &= ([1,3],[4,31],[1,28]) \\ R_2 &= ([4,4],[2,30],[4,27]) \\ R_3 &= ([7,9],[5,21],[3,18]) \end{aligned}$$

- Example: prefixes of the same length
- Implicit creation of order-dependence for service policies

	cisco1	cisco2	cisco3	fw	ipc	acl
Order-independent rules	120	249	329	39962	48294	49779
Total	148	269	364	45723	49840	49870
Order-independent %	81	93	90	87	97	99

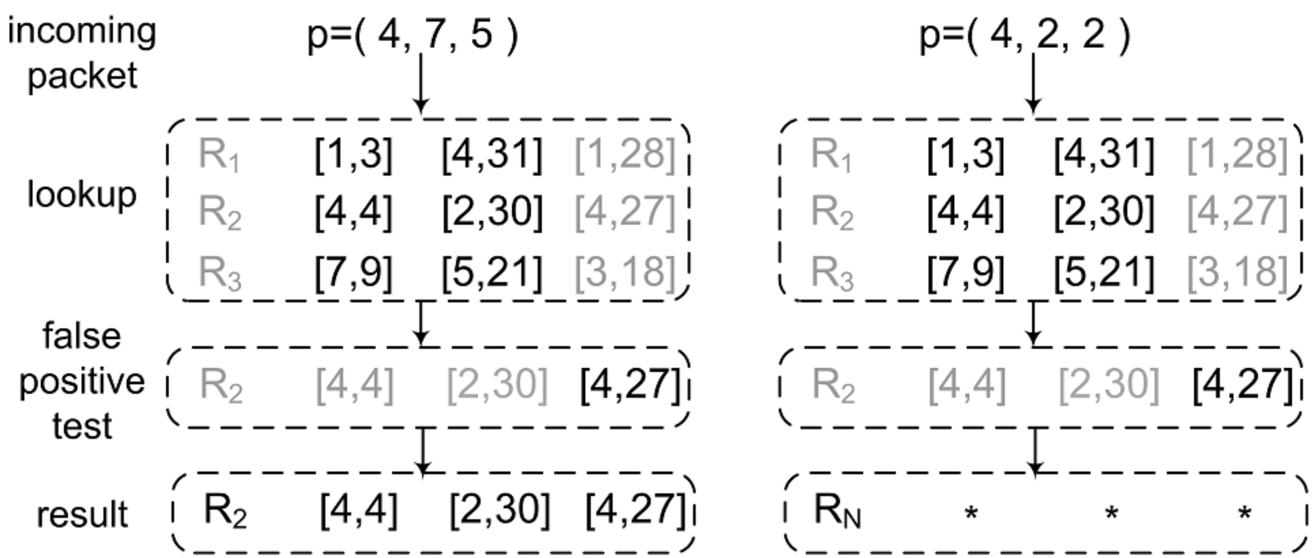
Exploiting order-independence

- Adding new fields keep order-independence
- At most one rule is matched and it can be false-positive
- We can reduce space by skipping new fields

$$
\begin{aligned}
R_1 &= ([1,3],[4,31]) \\
R_2 &= ([4,4],[2,30]) \\
R_3 &= ([7,9],[5,21])
\end{aligned}
$$

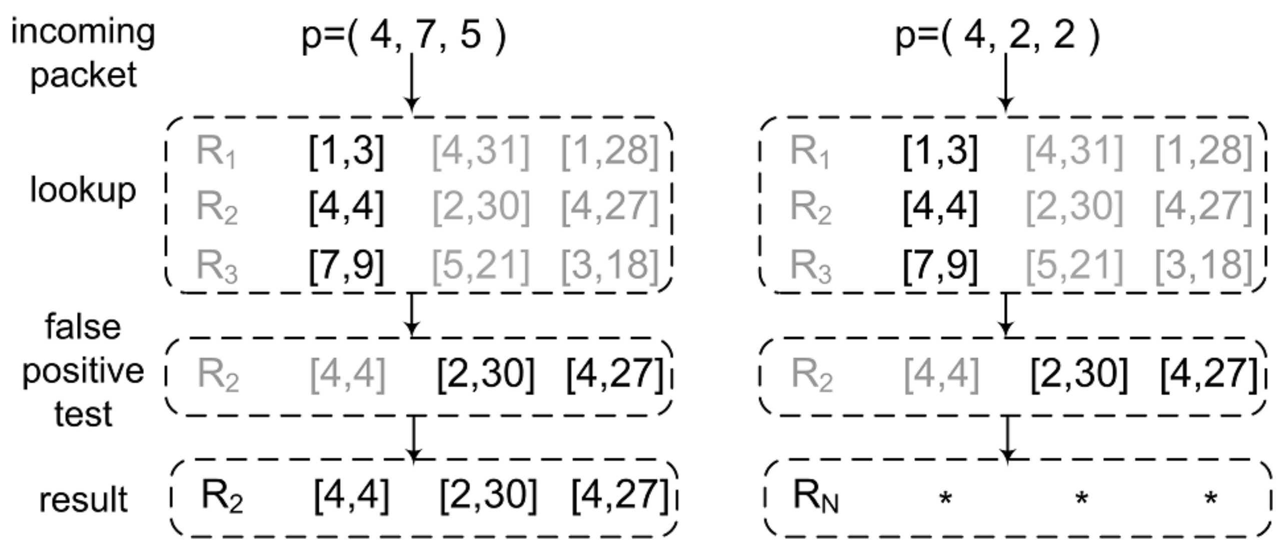
$$
\begin{aligned}
R_1^{+1} &= ([1,3],[4,31],[1,28]) \\
R_2^{+1} &= ([4,4],[2,30],[4,27]) \\
R_3^{+1} &= ([7,9],[5,21],[3,18])
\end{aligned}
$$

#Fields	Bin Encoding	Gray Encoding
2	6+7+10=23	6+4+8=18
3	42+28+50=120	24+8+32=64



Fields subset minimization (FSM)

Problem 1: Find a maximal subset M of fields of an order-independent classifier K s.t. K^{-M} is order-independent



Fields	Binary Encoding	Gray Encoding
{1,2,3}	42+28+50=120	24+8+32=64
{1,2}	6+7+10=23	6+4+8=18
{1}	2+1+2=5	2+1+2=5

Multi-group Representation

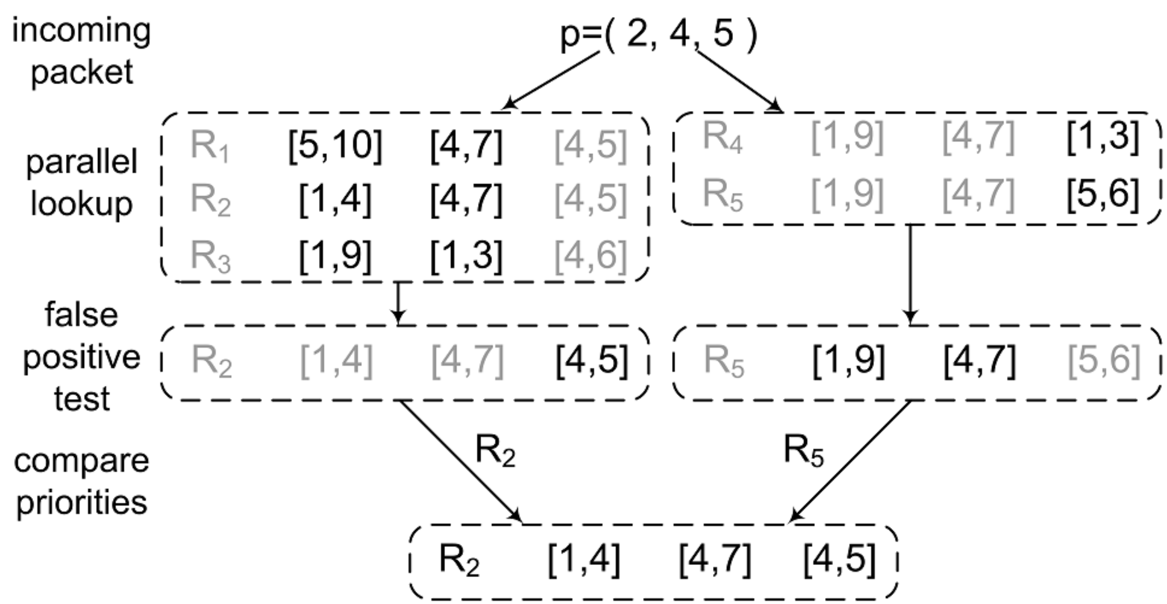
A classifier K is irreducible or order-dependent?

Problem 2: Given a classifier K on k fields and $0 < l < k$. Find an assignment of rules to a minimal number of disjoint groups s.t. every group is order-independent on l fields.

$$k = 3, l = 2$$

$$\begin{aligned} R_1 &= ([5,10],[4,7],[4,5]) \\ R_2 &= ([1,4],[4,7],[4,5]) \\ R_3 &= ([1,9],[1,3],[4,6]) \\ R_4 &= ([1,9],[4,7],[1,3]) \\ R_5 &= ([1,9],[4,7],[5,6]) \end{aligned}$$

$$R_1 \cap R_5 \neq \emptyset$$



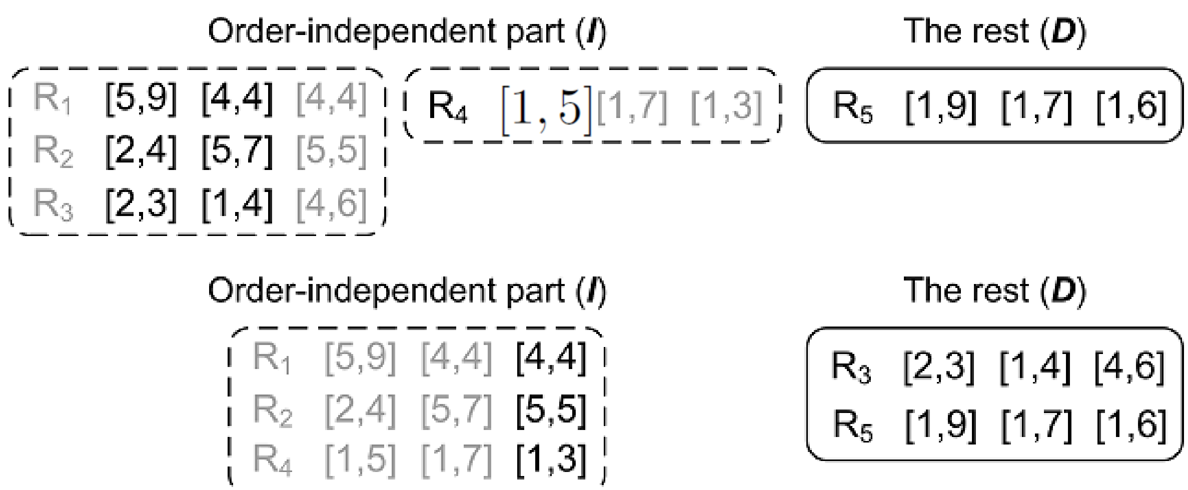
Multi-group representation of rules subset

Number of groups > supported level of parallelism β?

Problem 3: Given a classifier K on k fields, $0 < l < k$, and $\beta > 0$.
Find an assignment of a maximal subset of rules to β disjoint groups s.t. every group is order-independent on l fields.

$$k = 3, l = 2$$

$$\begin{aligned}
R_1 &= ([5,9],[4,4],[4,4]) \\
R_2 &= ([2,4],[5,7],[5,5]) \\
R_3 &= ([2,3],[1,4],[4,6]) \\
R_4 &= ([1,5],[1,7],[1,3]) \\
R_5 &= ([1,9],[1,7],[1,6])
\end{aligned}$$



$$\{R_1, R_2, R_3, R_4\}$$

Max order-independent set on 3 fields

Classifiers as Boolean expressions

$$
\begin{array}{rcllll}
R_1 & = & (& 100*, & 001* &) \\
R_2 & = & (& 1010, & 0001 &) \\
R_3 & = & (& 000*, & **** &) \\
R_4 & = & (& 001*, & **** &)
\end{array}
$$

$$
\begin{array}{l}
(x_1 \wedge \bar{x}_2 \wedge \bar{x}_3 \wedge \bar{x}_5 \wedge \bar{x}_6 \wedge x_7) \vee \\
(x_1 \wedge \bar{x}_2 \wedge x_3 \wedge \bar{x}_4 \wedge \bar{x}_5 \wedge \bar{x}_6 \wedge \bar{x}_7 \wedge x_8) \vee \\
(\bar{x}_1 \wedge \bar{x}_2 \wedge \bar{x}_3) \vee \\
(\bar{x}_1 \wedge \bar{x}_2 \wedge x_3)
\end{array}
$$

MinDNF: For a given Boolean function f, find a minimal size of DNF for f

MinDNF:

$$
\begin{array}{ll}
R_1 & = (100*, 001*) \\
R_2 & = (1010, 0001) \\
R_{34} & = (00**, ****)
\end{array}
$$

FSM with per field resolution:

$$
\begin{array}{lcl}
R_1^{-1} & = & (100*) \\
R_2^{-1} & = & (1010) \\
R_3^{-1} & = & (000*) \\
R_4^{-1} & = & (001*)
\end{array}
$$

FSM with per bit resolution:

$$
\begin{array}{lcl}
R_1^{-6} & = & (10) \\
R_2^{-6} & = & (11) \\
R_3^{-6} & = & (00) \\
R_4^{-6} & = & (01)
\end{array}
$$

Exact algorithm

FSM is NPC (reduction from SetCover)

Algorithm FSMBinSearch(k, min, max)

$m = \lfloor \frac{min+max}{2} \rfloor$
if $min = max$ **then return** m
for $M \subseteq \{1, \ldots, k\}, |M| = m$ **do**
    if IsOrderIndependent(M) **then**
        $min = m$
        return FSMBinSearch(k, min, max)
$max = m - 1$
return FSMBinSearch(k, min, max)

Theorem: $FSMBinSearch(k, 0, k-1)$ runs in time $O(k2^{k-1}N^2)$

Approximate algorithms

$$\begin{array}{llll} & S_1 & S_2 & S_3 \\ R_1 & = ([1,3], & [4,31], & [1,28]) \\ R_2 & = ([4,4], & [2,30], & [4,27]) \\ R_3 & = ([7,9], & [5,21], & [28,31]) \end{array}$$

Heuristics: algorithms for $SetCover$ and $MaxSetCoverage$

$$U = \{(i,j) | i < j, i,j \in [1,N]\} = \{(1,2),(1,3),(2,3)\}$$

Define $k\ sets\ S_1, .., S_k$ (one per field) to cover U
$S_i, 1 \leq i \leq k$, contains all pairs of rules that do not intersect in this field

$$S_1 = \{(1,2),(1,3),(2,3)\}\ S_2 = \emptyset\ S_3 = \{(2,3)\}$$

Theorem: FSM is reducible to $SetCover$ in $O(kN^2)$ time with approximation factor $2\ln(N) + 1$

Evaluation

6 classifiers with real parameters (see paper for more examples)

#		Max OI part 5 fields			Multi-group representation									
	Total				1-field groups					2-field groups				
	rules	OI size	FSM	{0,1}	all	95%	99%	≤ 2	≤ 5	all	95%	99%	≤ 2	≤ 5
1	584	538	0,1,3,4	406	15	8	13	2	8	10	4	7	3	4
2	269	249	0,1,4	246	4	2	3	1	1	2	2	2	0	0
3	364	329	0,1,3,4	324	7	3	5	2	4	4	2	3	1	14
4	49870	49779	0,1,4	49768	16	1	1	6	9	12	1	1	5	7
5	47276	44178	0,1,3,4	43819	67	5	13	19	32	39	2	5	10	20
6	48885	48826	0,1,2,4	48755	20	3	3	15	15	12	1	1	5	9

Summary

- Semantically equivalent: more fields imply less efficient representation
- Representation with false-positive: more fields – more efficient representation
- Structural properties can significantly improve time-space trade-off
- No restrictions on representation of every group (we define only additional abstraction layer)

Ongoing and future work

- Consider special cases as a FIB representation

G. Retvari et al., **Compressing IP Forwarding Tables: Towards Entropy Bounds and Beyond** SIGCOMM 13

- Identify additional structural properties
- Composition of structural properties
- Application to neighboring fields: data bases, program optimization, etc.

Thank you