Exploiting Order Independence for Scalable and Expressive Packet Classification

Author: Kirill Kogan, Sergey I. Nikolenko, Ori Rottenstreich, William Culhane, and Patrick Eugster

Presenter: Qing Lyu
2017/04/12

IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 24, NO. 2, APRIL 2016

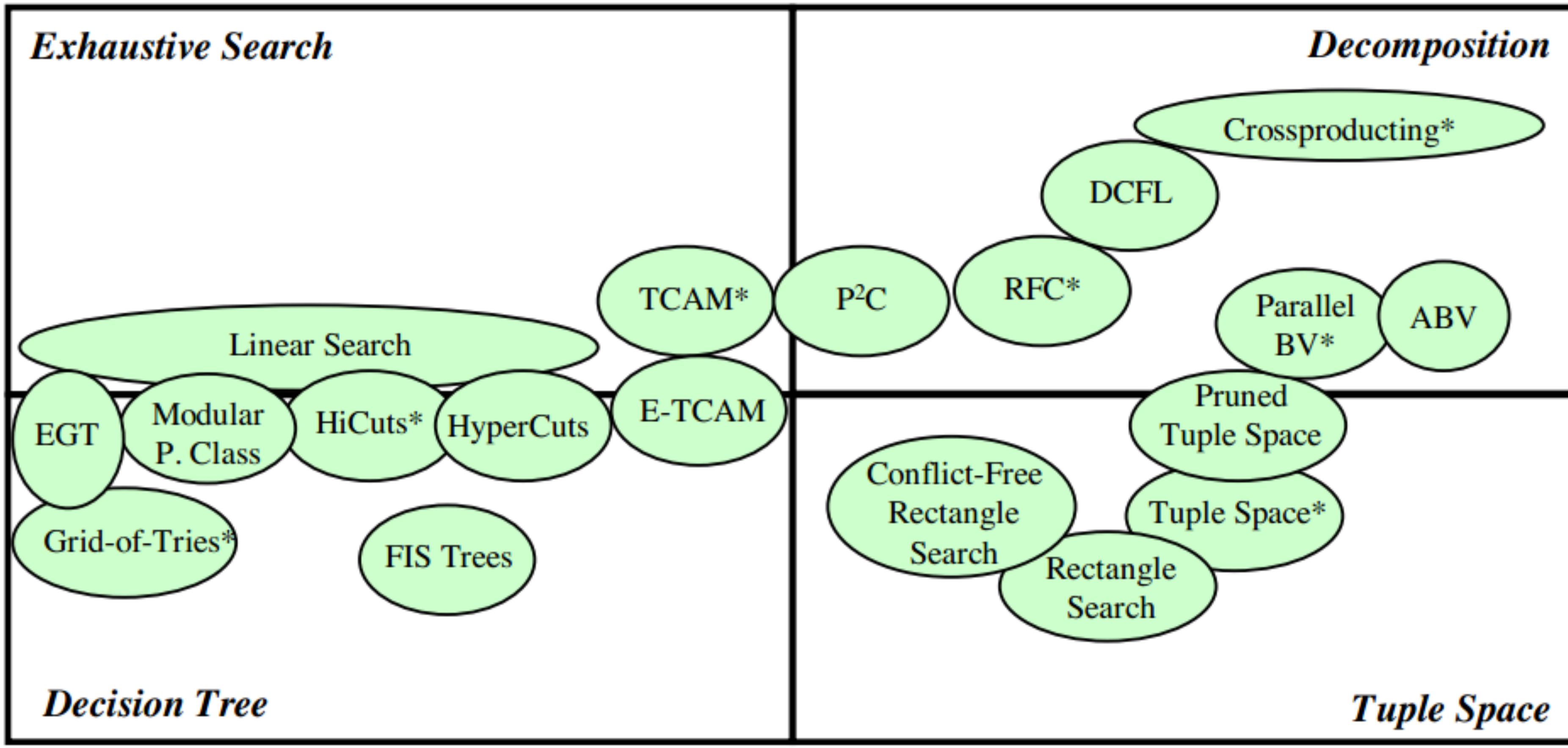
Outline

- Introduction
- Order-independence
- Problem formulation and theorem
- Solutions
- Simulations
- Conclusions

Outline

- Introduction
- Order-independence
- Problem formulation and theorem
- Solutions
- Simulations
- Conclusions

Packet Classification



Packet Classification

TCAM

- NFV, power consumption and heat
- cannot efficiently represent range rules, especially multiple fields
- exponential space growth

Software (SW)

- tradeoffs between (memory) space and (lookup) time.

TCAM

Advantages: efficiently represent multi-field classification with prefixes

Disadvantages: suffers an exponential blowup from range expansion

each range-based field in a rule introduces an additional multiplicative factor

SW-Complexity bounds

Software-based packet classifier with rules and fields $\mathrm{K} \geq 3$

$\mathrm{O}(N^k)$ space, $O(\log N)$ time

$O(N)\ space,\ O(log^{k-1}N)$ time

E.g. 100 rules and 4 fields,

$\mathrm{O}(N^k)$ is about 100MB, $O(log^{k-1}N)$ time is about 350 memory accesses

Ranges

Standard five-tuples with two fields which include ranges

Desired implementations for classifiers on ranges:

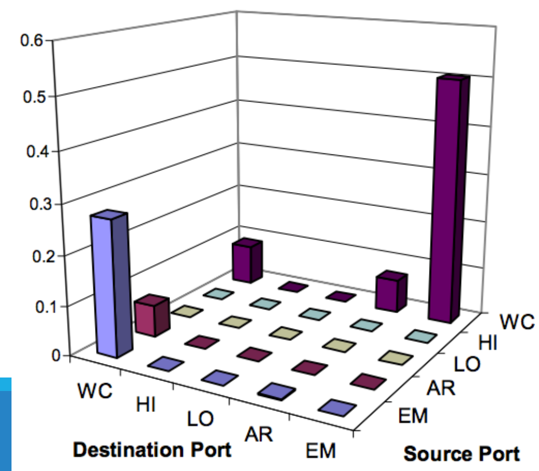
IP or MAC addresses

dates

packet lengths

etc

- WC, wildcard
- HI, ephemeral user port range [1024 : 65535]]
- LO, well-known system port range [0 : 1023]]
- AR, arbitrary range
- EM, exact match



Outline

- Introduction
- Order-independence
- Problem formulation and theorem
- Solutions
- Simulations
- Conclusions

Contributions

Exploit the order-independence of classifiers

Reduce the number of classification fields represented by prefixes or ranges

Implemented in linear space and with worst-case guaranteed logarithmic time

Allows the addition of more fields including range constraints without impacting space and time complexities

Model Description

A packet header contains k fields

A field $i, 1 \leq \mathrm{i} \leq k$ is a string of W_i bits

A classifier $\mathcal{K}$ is an ordered set of rules, $R_1, \dots, R_N$

A rule R_j is a is an ordered set of k fields and an associated action A_j

A field F_i is represented by a range of values on W_i bits

each rule contains k ranges $R = (I_1, \dots, I_k), I_i = [l_i, u_i]$

Denotation

$\mathcal{K}^{-F}$

the classifier obtained from by $\mathcal{K}$ by removing a set of fields F from each rule

$\mathcal{K}^{+F}$

the classifier obtained from $\mathcal{K}$ by extending its rules with F (with values defined separately for every rule)

Denotation

$intersect$

at least one header that matches both rules R_1, R_3

$disjoint$

R_1, R_2

$order - independent$

R_1, R_2 are order-independent if the corresponding sets of matching headers are disjoint, every header matches at most one of them.

$$R_1 = (100*)$$
$$R_2 = (01**)$$
$$R_3 = (1***)$$

Denotation

A classifier $\mathcal{K}$ is called order-independent if any two of its rules are filter-order-independent

$\mathcal{K}(S)$ denotes a classifier that uses only a subset S of fields in classification

Order-independence

$\mathcal{K}'$: with the same rules as $\mathcal{K}$ sorted in a different order

Then, any packet header p is matched by the same rule in $\mathcal{K}'$ and $\mathcal{K}$

The condition is satisfied when rules do not intersect

i.e., for each pair of rules there is at least one field in which the corresponding ranges (or prefixes) are disjoint.

Transitively order- dependent $R_1, R_2 \quad R_2, R_3$

Cont.

$\mathcal{K}$

R_1=([1,3],[4,5])

R_2=([5,6],[4,5]) order-independent

$\mathcal{K}'$

R_3=([1,3],[4,5])

R_4=([2,4],[4,5]) order-dependent as (3,4) matches both rules

Two Encoding Schemes

Binary Encoding

Worst case $2(W-1)$ TCAM entries

SRGE Encoding

Worst case $2(W-2)$ TCAM entries

Multi-field (k-field) range, upper bound entries #

Binary: $(W-2)^k$, SRGE: $(W-4)^k$

Binary Encoding

Binary Encoding

$W - bit$ range is encoded as a union of disjoint subtrees in the binary tree of 2^W leaves

Each subtree is represented by a single prefix TCAM entry

the maximal number of entries to encode a range (worst-case expansion) is $2(W - 1)$

Binary Encoding

$W = 5$

Range $[16,23]$

A single entry $(01 ***)$

Range $[1,30]$

$2W - 2$ entries

(00001),(0001*),(001**),(01***),(10***),(110**),(1110*),(11110)

SRGE

Improve the worst-case bound to $2(W-2)$ entries by representing values in Gray code.

Entries are not necessarily prefixes and do not necessarily represent disjoint subsets of the range

Range $[1,30]$

$2W-4$ entries

Example1

$\mathcal{K} = (R_1, R_2, R_3)$ with two fields of 5 bits each

Binary encoding

$R_1 = ([1,3],[4,31])$

6 entries

1st field:(00001,0001*)

2nd field:(00100,01***,1****)

$R_2 = ([4,4],[2,30])$

7 entries

1st field:(00100)

2nd field:(00010,001**,01***,10***,110**,1110*,11110)

$R_3 = ([7,9],[5,21])$

10 entries

1st field:(00111,0100*)

2nd field:(00101,0011*,01***,100**,1010*)

Example1

$\mathcal{K}^{+1} = (R_1^{+1}, R_2^{+1}, R_3^{+1})$ with one additional fields of 5 bits

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

42+28+50=120 entries

Outline

- Introduction
- Order-independence
- Problem formulation and theorem
- Solutions
- Simulations
- Conclusions

Tested Classifiers

12 classifiers from Classbench generated with real parameters

Each with 50K rule on 6 fields

5 real-life classifiers from Cisco Systems

Tested Classifiers

TABLE I

LEFT TO RIGHT: # OF RULES (TOTAL AND IN A MAXIMAL ORDER-INDEPENDENT SET); TCAM SPACE UNDER TWO ENCODINGS (BINARY [53] AND SRGE [6]) FOR CLASSBENCH CLASSIFIERS: STANDARD REPRESENTATION AND REDUCED BY THEOREM 2; TCAM SPACE FOR THE SAME CLASSIFIERS EXTENDED WITH 2 NEW RANDOM SYNTHETIC 16-BIT RANGE FIELDS: STANDARD AND REDUCED BY THEOREM 1

	Rules		Original classifier						Classifier expanded by two 16-bit ranges					
			Original $\mathcal{K}$			By Theorem 2			Original $\mathcal{K}^{+2}$			By Theorem 1		
	Total	Order-ind., all fields	Width, bits	Space, Kb Binary	Space, Kb SRGE	OI wid, bits	Space, Kb Binary	Space, Kb SRGE	Width, bits	Space, Kb Binary	Space, Kb SRGE	OI wid, bits	Space, Kb Binary	Space, Kb SRGE
acl1	49870	49779	120	7922	7655	31	1517	1517	152	1752769	1462501	31	3857	3509
acl2	47276	44178	120	11289	11289	82	4189	4189	152	2499525	2159699	82	145925	126505
acl3	49859	47674	120	10771	10571	91	5018	5008	152	2391264	2027239	91	177771	152514
acl4	49556	46670	120	10079	9904	97	5379	5370	152	2234839	1895078	97	216994	186701
acl5	40362	38962	120	6121	6121	63	2950	2950	152	1359256	1172403	63	127818	110663
fw1	47778	43675	120	19454	19438	72	3926	3911	152	4303234	3720573	72	194313	164893
fw2	48885	48826	120	10866	10866	52	2498	2498	152	2399603	2071400	52	7382	6695
fw3	46038	41615	120	15090	15073	84	4161	4145	152	3337763	2873120	84	170688	144596
fw4	45340	42857	120	33500	33368	76	4025	4008	152	7438741	6390798	76	195130	165332
fw5	45723	39962	120	12478	12445	76	3759	3745	152	2745105	2366939	76	180420	152652
ipc1	49840	48294	120	8041	7924	50	2580	2579	152	1789100	1521153	50	52378	45391
ipc2	50000	50000	120	5859	5859	36	1757	1757	152	1301839	1123612	36	1757	1757
cisco1	584	538	120	78	78	52	34	34	152	17523	15088	52	1650	1441
cisco2	269	249	120	68	68	21	7	7	152	15662	13510	21	477	415
cisco3	95	92	120	11	11	30	3	3	152	2505	2137	30	76	71
cisco4	364	329	120	79	79	38	18	18	152	17827	15385	38	1484	1237
cisco5	148	120	120	19	19	17	5	5	152	4303	3717	17	695	590

Fields Expansion Theorem ---Theorem 1

let $\mathcal{K}^{+m}$ be a classifier that results from an order-independent classifier $\mathcal{K}$ by adding new fields m of any width

Then, $\mathcal{K}$ with a false-positive check of a single matched rule is a semantically equivalent representation of $\mathcal{K}^{+m}$

Theorem 1

Introducing additional fields based on prefixes or ranges to an order–dependent classifier

1. affects only the encoding size of its order–dependent part

2. new fields in the order-independent part can be ignored without affecting the classification outcome

3. The space and lookup time complexity of classification in the order-independent do not increase

Theorem 1

Previous new fields based on prefixes or ranges significant increase the software solution

$O(log^{k-1}N)$ look up time in linear memory

Likewise, TCAM-based solution, range converted to prefixes, new field based on ranges adds an additional multiplicative factor for the required TCAM space

Theorem 1

Support for additional fields amenable to range rules would greatly improve classification expressiveness

e.g., ranges on dates, packet length, etc

Theorem 1

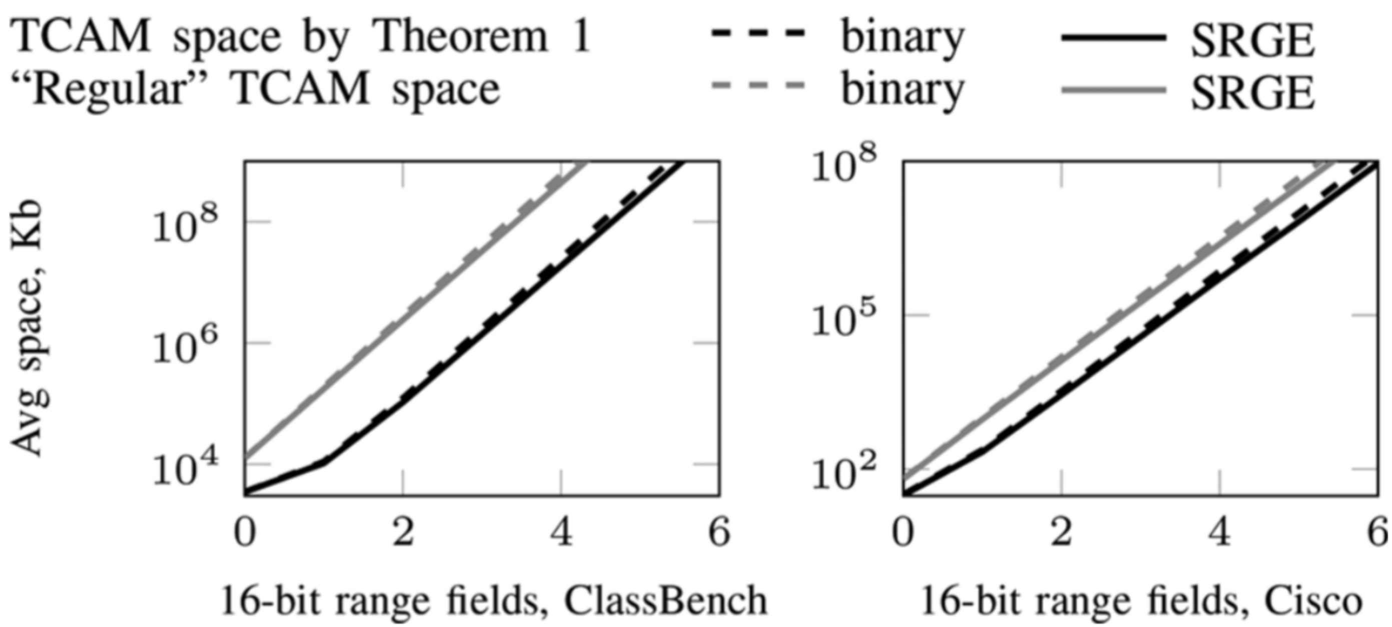


Fig. 1. TCAM space for ClassBench and CISCO classifiers as a function of the number of additional 16-bit range fields.

Theorem 1

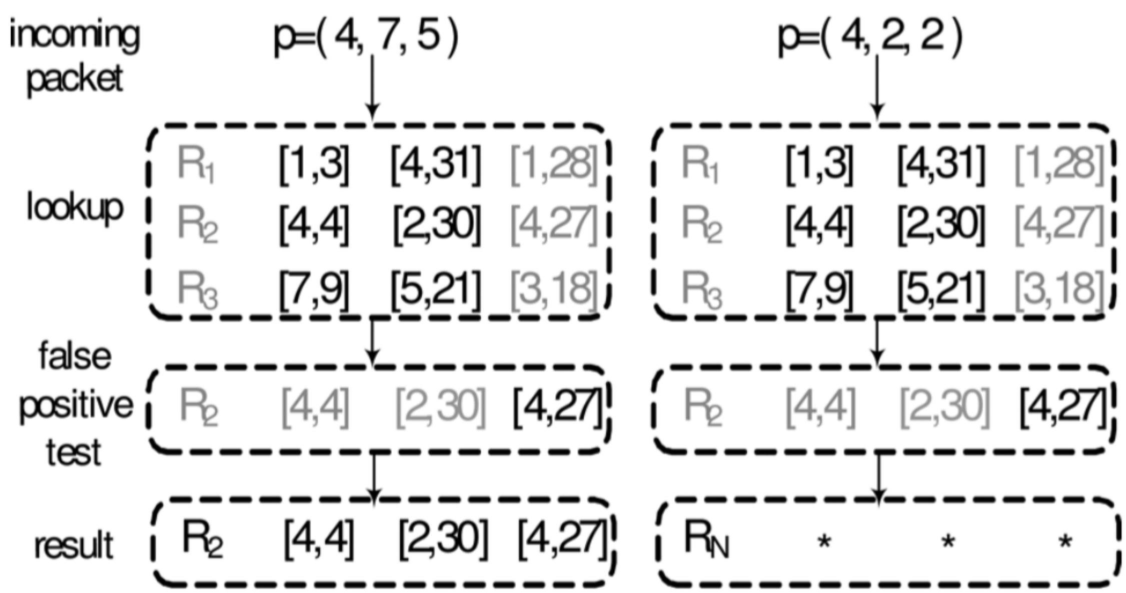


Fig. 2. Visualizing the lookup procedure on rules with a subset of fields in Example 1. Fields checked in a certain step are in black; those irrelevant for the step, in grey. Packet $(4, 2, 2)$ matches R_2 but fails the false positive check on the added field of R_2^{+1}, so the classifier returns the catch-all rule.

Example2

$\mathcal{K} = (R_1, R_2, R_3)$ with three fields of 5 bits

$R_1 = ([1,3], [4,31], [1,28])$

$R_2 = ([4,4], [2,30], [4,27])$

$R_3 = ([7,9], [5,21], [3,18])$

42+28+50=120 entries

$\mathcal{K}^{-\{2,3\}} = (R_1^{-\{2,3\}}, R_2^{-\{2,3\}}, R_3^{-\{2,3\}})$

$R_1^{-\{2,3\}} = ([1,3])$

$R_2^{-\{2,3\}} = ([4,4])$

$R_3^{-\{2,3\}} = ([7,9])$

2+1+2=5 entries

Fields Reduction Theorem ---Theorem 2

let $\mathcal{K}^{-m}$ be a classifier that results from an order-independent classifier $\mathcal{K}$ by removing m fields

If $\mathcal{K}^{-m}$ is order-independent, then, $\mathcal{K}^{-m}$ with a false-positive check of a single matched rule is a semantically equivalent representation of $\mathcal{K}$

Theorem 2

If the reduced classifier $\mathcal{K}^{-m}$ contains at most two fields

Then, we can efficiently implement lookup in time logarithmic N in with (near-) linear memory

For TCAM based solutions, reduces the required TCAM space proportionally

Theorem 2

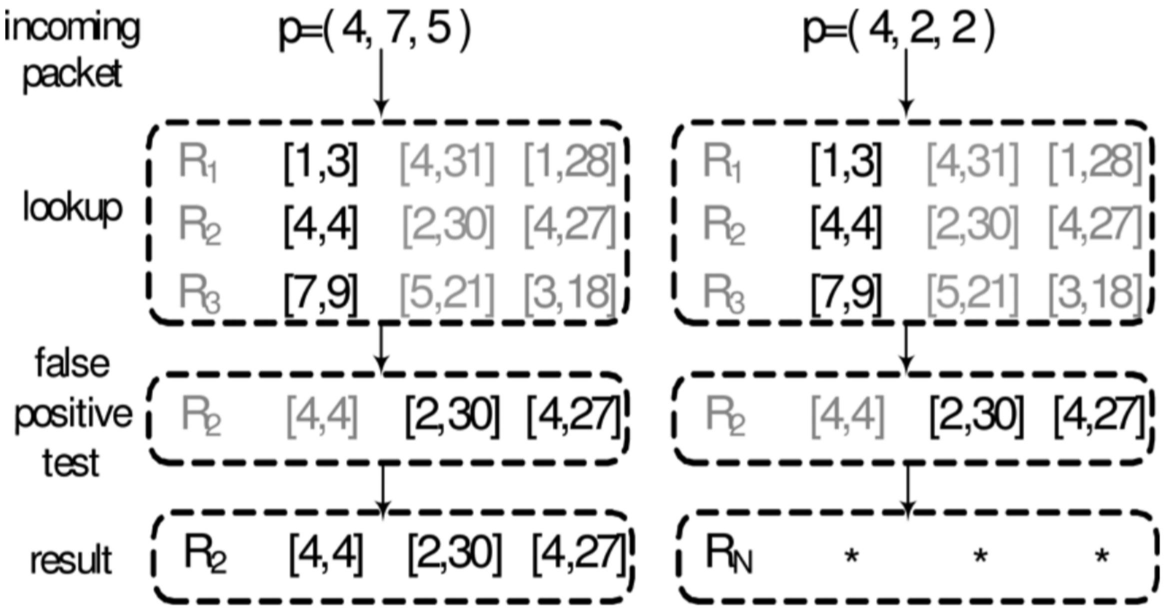


Fig. 3. Visualizing the lookup procedure on rules with a subset of fields in Example 2. Fields to be checked in a step are shown in black; irrelevant for the step, in grey. Note that packet $(4, 2, 2)$ matches $R_2^{-\{2,3\}}$ but fails the false positive check, so the classifier returns the catch-all rule.

Fields Subset Minimization (FSM)

Find a maximal subset of fields M of an order-independent classifier $\mathcal{K}$

such that

$\mathcal{K}^{-m}$ is order-independent， if there are several such subsets,

choose M with maximal total width (to minimize lookup word width)

multi-group representation β groups

FSM+multi-group representation

β groups:

1. each rule belongs to a single group

2. the rules of each group are order-independent on a subset of k fields of $\mathcal{K}$

3. different groups can reuse the same fields to keep order-independence

$l - MGR$: find the minimal number of disjoint groups that each group is order-independent on l fields

FSM+multi-group representation

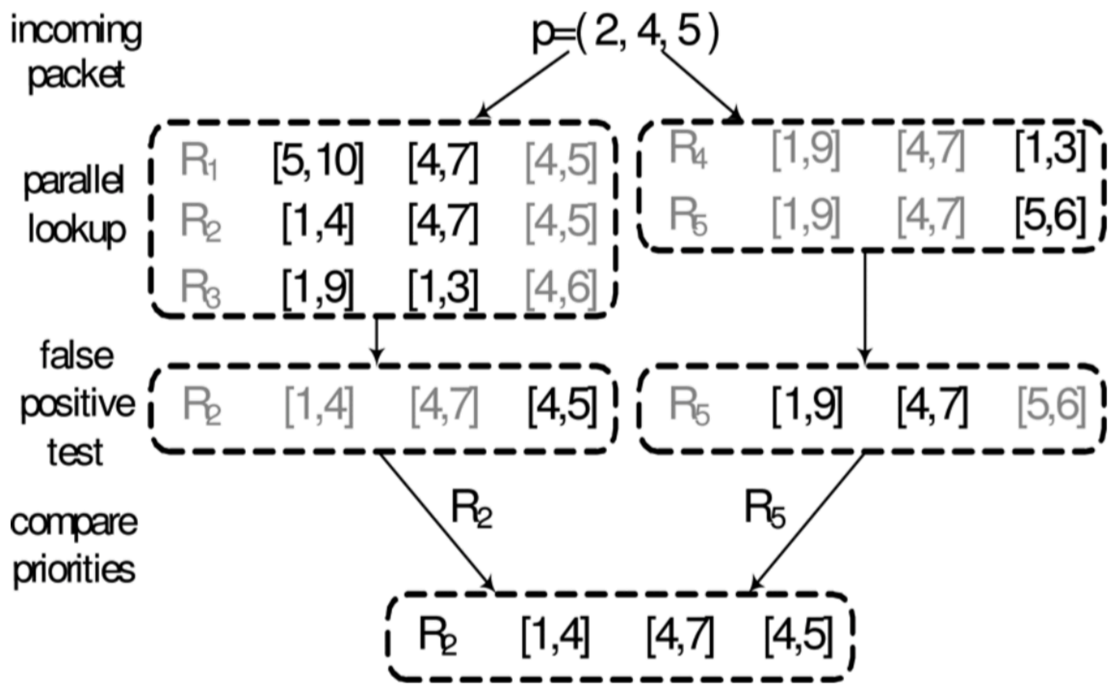


Fig. 4. A visualization of the lookup procedure with a multi-group representation of the rules in Example 3. Fields to be checked on a certain step are in black; those that are irrelevant for this step, in grey.

Maximum Rules Coverage

$\mathcal{K}$: order-independent part $\mathcal{I}$ and order-dependent part $\mathcal{D}$

1.find a maximal order-independent subset of rules on all k fields

2.find an order-independent part that already has the desired properties

Maximum Rules Coverage

$l - MRC$: find a maximal subset $\mathcal{I} \subset \mathcal{K}$ which is order-independent on at most l fields

$(\beta, l) - MRC$: find a maximal subset $\mathcal{I} \subset \mathcal{K}$ that can be assigned to at most β groups, where each group is order-independent on at most l fields.

$(\beta, l) - MRCC$: no two rules $R_1 \subset \mathcal{I}$, $R_2 \subset \mathcal{D}$, such that R_1 “intersects” with R_2, and $R_2 \prec R_1$

Bit-level

$\mathrm{R}_1 = (100*, 001*), \mathrm{R}_2 = (1010{,}0001)$

$\mathrm{R}_3 = (000*, ****), \mathrm{R}_4 = (001*, ****)$

First field

$R_1^{-1} = (100*), R_2^{-1} = (1010)$

$R_3^{-1} = (000*), R_4^{-1} = (001*)$

First and third bit

Treat $\mathcal{K}$ as 8 one-bit fields, $R_1^{-6} = (10), R_2^{-6} = (11), R_3^{-6} = (00), R_4^{-6} = (01)$

Representing Classifier as Boolean Expression

each field is represented by a prefix, so fields can be represented by a string of individual bits

Rule set can be written as an unordered disjunction (OR, $\vee$) of individual rules, which expressed as a conjunction (AND, $\wedge$)

A rule $s = s_1 \ldots s_k, s_j \in (0,1,*)$, can be expressed as a conjunction $s = f_s(x_1, \ldots, x_k)$=$\wedge_{s_i=1} x_i \wedge_{s_i=0} \bar{x}_i$

Min-DNF (disjunctive normal form): find a minimal size DNF representation for Boolean function

Compare FSM with Boolean Expression

$\mathrm{R}_1 = (100*, 001*), \mathrm{R}_2 = (1010{,}0001)$

$\mathrm{R}_3 = (000*, ****), \mathrm{R}_4 = (001*, ****)$

The MinDNF heuristic applicable here is the resolution rule that can be applied to R_3 and R_4

$R_3' = (00**, ****)$

This classifier has width 8; if we discard bits with identical values (second bit in the first field), we still get width 7

Cont.

TABLE II
EXPERIMENTAL RESULTS OF MINDNF REDUCTION IN THE ORDER-INDEPENDENT SUBSETS OF CLASSIFIERS

	Original classifier					MinDNF reduced						OI
	Rules, orig.	Rules, OI	Rules, binary	Rules, SRGE	Wid, bits	Rules, binary	Wid, bits	Red., bits	Rules, SRGE	Wid, bits	Red., bits	wid, bits
acl1	49870	49779	67511	65240	120	67505	90	90	65234	90	90	31
acl2	47276	44178	90772	90772	120	90233	104	104	90230	104	104	82
acl3	49859	47674	85252	83630	120	85226	106	106	83605	106	106	91
acl4	49556	46670	77837	76424	120	77755	106	106	76338	106	106	97
acl5	40362	38962	47514	47514	120	46261	96	94	46249	96	94	63
fw1	47778	43675	159525	159519	120	159461	112	112	159458	112	112	72
fw2	48885	48826	92646	92646	120	92316	88	88	92316	88	88	52
fw3	46038	41615	122495	122477	120	122259	112	112	122244	112	112	84
fw4	45340	42857	278887	277921	120	277799	104	104	276807	104	104	76
fw5	45723	39962	99574	99412	120	99421	112	112	99273	112	112	76
ipc1	49840	48294	66718	65734	120	66715	112	112	65731	112	112	50
ipc2	50000	50000	50000	50000	120	50000	112	112	50000	112	112	36
cisco1	584	538	603	603	120	475	104	86	474	104	86	52
cisco2	269	249	565	565	120	564	104	84	564	104	84	21
cisco3	95	92	92	92	120	92	88	68	92	88	68	30
cisco4	364	329	629	629	120	629	104	84	629	104	84	38
cisco5	148	120	139	139	120	139	104	84	139	104	84	17

Outline

- Introduction
- Order-independence
- Problem formulation and theorem
- Solutions
- Simulations
- Conclusions

Exact Algorithm

Use a binary search to solve FSM

If the classifier is order-independent for some removed subset with $\frac{k}{2}$ fields, try to find a subset of size $\frac{3}{4}k,$ otherwise check order-independence for subsets of size $\frac{k}{4}$, and so on

Worst case $\binom{k}{\frac{k}{2}}+\binom{k}{\frac{k}{4}}+\ldots+\binom{k}{1} < 2^{k-1}$

FSMBINSEARCH

Exact Algorithm

Check order-independent

N rules, m removed fields (originally k)

Algorithm 1 IsOrderIndependent(M)

for $i = 1 \rightarrow N - 1$ **do**
    for $j = i + 1$ to N **do**
        if $R_i^{-M} \cap R_j^{-M} \neq \emptyset$ **then return** False
return True

$O(N^2(k-m))$ FSMBINSEARCH $O(k2^{k-1}N^2)$

Exact Algorithm

When fields increase, FSM requires approximation heuristics

$2 - MRC$: NP-complete

$(\beta, l) - MRC$: NP-complete as well

exponential on N

Approximate Solutions

FSM is reducible to SetCover in $O(k \cdot N^2)$ and has an approximation factor of $2\ln(N) + 1$

Construct set cover problem

Define $U = \{(i,j) | i < j, i,j \in [1,N]\}$, define k sets $S_1, \ldots, S_k$ to cover U

Polynomial time $O(k \cdot N^2)$

For $l \in [1,k], S_l = \left\{(i,j) \middle| i < j, i,j \in [1,N], R_i(l) \cap R_j(l) = \emptyset\right\}$

S_l contains all pairs of rules that do not intersect in field l

Cont.

Algorithm 3 Algorithm GreedySetCover

$\mathcal{X} \leftarrow \emptyset$; $\mathcal{T} \leftarrow \mathcal{S}$
while $\mathcal{T}$ not empty **do**
    Choose S_i in $\mathcal{T}$ that contains maximal number of uncovered elements in $\mathcal{U}$
    $\mathcal{X} \leftarrow \mathcal{X} \cup \{S_i\}$
    $\mathcal{T} \leftarrow \mathcal{T} \setminus S_i$
return $\mathcal{X}$

each step selects an additional set with maximal number of uncovered elements

If sets of fields required for order-independence in different groups are pairwise disjoint

$2 - MGR$ is reducible to SetCover in $O(\frac{k(k+1)}{2} \cdot N^2)$ and time and has an approximation factor of $2\ln(N) + 1$

$l - MSC$ (Maximum Set Coverage), a set S of k sets whose union equals $\mathcal{U}$, select at most l sets of S such that as many elements of $\mathcal{U}$ as possible are covered (the union of selected sets has maximal size)

A variant of Algorithm 3 that stops after l subsets are added solves the $l - MSC$ problem and has an approximation factor of $1 - \frac{1}{e} + o(1)$

Use an algorithm for the $l - MSC$ problem as a heuristic to solve the $l - MRC$ problem

worst-case guaranteed lookup time $\log(N)$ and a linear space requirement when l=2

$l - MRC$ to solve $(\beta, l){-}MRC$, once any unassigned rule “intersects” with any rule assigned to the current group, open a new group if the total number of groups does not exceed β

The algorithm for $l - MGR$ is the same as for $(\beta, l){-}MRC$ but stop to create new groups all rules of the original classifier is covered

Classifier Configuration

implementing service–level–agreement (SLA) policies–––relatively static

offline computation for more efficient representations

1. maximal size of the order-independent part

2. minimal subset of fields that preserve order-independence

3. minimal number of order-independent groups based on at most two fields

4. assignment of rules to a predefined number of groups based on at most two fields

Classifier Configuration

those required for switching---dynamic

Remove: $\mathcal{I}$, $\mathcal{D}$

Insert: more complicated, if order-dependent with $\mathcal{I}$, insert it to $\mathcal{D}$, if $\mathcal{D}$ is full recompute $\mathcal{I}$, $\mathcal{D}$; if order-independent with $\mathcal{I}$, insert it to $\mathcal{I}$.

Modify existing rules： modified in fields (not) required for order-independence, make sure whether the modified rule intersect with other rules in $\mathcal{I}$.

Outline

- Introduction
- Order-independence
- Problem formulation and theorem
- Solutions
- Simulations
- Conclusions

Simulations

12 classifiers from Classbench, each with 50 K rules on 6 fields

and on 5 real life classifiers provided by Cisco Systems

Cont.

TABLE III

SIMULATION RESULTS. COLUMNS, LEFT TO RIGHT: TOTAL NO. OF RULES $|\mathcal{K}|$; $|\mathcal{I}|$, WHERE $\mathcal{I} = \text{MRC}(\mathcal{K}, \{0, \ldots, 5\})$; MINIMAL SIZE SUBSET OF FIELDS W.R.T. WHICH $\mathcal{I}$ IS ORDER-INDEPENDENT; $|\mathcal{I}'|$, WHERE $\mathcal{I}' = \text{MRC}(\mathcal{K}, \{0, 1\})$ (I.E., ORDER-INDEPENDENT W.R.T SOURCE IP AND DESTINATION IP); FOR THE ONE-FIELD RESULT $\{\mathcal{G}\} = \text{MGR}(\mathcal{K})$: TOTAL NUMBER OF ORDER-INDEPENDENT GROUPS $|\{\mathcal{G}\}|$, NO. OF GROUPS COVERING 95% AND 99% OF THE RULES, NO. OF GROUPS OF SIZE ≤ 2 AND OF SIZE ≤ 5; SIMILAR STATISTICS FOR THE TWO-FIELD MGR; SIMILAR STATISTICS FOR ONE-FIELD AND TWO-FIELD MGR RUN ON $\mathcal{I}$

	Total	k-MRC			MGR on the entire set										MGR on the k-MRC result									
					1-MGR					2-MGR					1-MGR					2-MGR				
	rules	size	FSM	$\{0,1\}$	all	95%	99%	≤ 2	≤ 5	all	95%	99%	≤ 2	≤ 5	all	95%	99%	≤ 2	≤ 5	all	95%	99%	≤ 2	≤ 5
acl1	49870	49779	0, 1, 4	49768	16	1	1	6	9	12	1	1	5	7	5	1	1	1	2	2	1	1	0	0
acl2	47276	44178	0, 1, 3, 4	43819	67	5	13	19	32	39	2	5	10	20	17	4	7	1	2	6	1	2	1	1
acl3	49859	47674	0, 1, 2, 3, 4	46114	31	3	5	11	20	18	2	3	7	10	11	3	4	1	2	6	1	3	1	1
acl4	49556	46670	0, 1, 2, 3, 4	40518	42	8	14	6	15	16	3	5	3	5	21	7	10	0	3	6	2	3	0	0
acl5	40362	38962	0, 1, 3, 4	22725	43	18	27	2	6	11	4	6	1	1	37	17	25	0	2	10	4	6	1	1
fw1	47778	43675	0, 1, 2, 3, 4	44713	71	2	5	18	42	41	2	3	10	19	7	2	2	2	2	4	1	2	0	1
fw2	48885	48826	0, 1, 2, 4	48755	20	3	3	15	15	12	1	1	5	9	7	3	3	2	3	2	1	1	0	0
fw3	46038	41615	0, 1, 2, 3, 4	40581	101	2	17	42	57	56	2	7	11	28	17	2	2	2	4	7	1	2	0	0
fw4	45340	42857	0, 1, 2, 3, 4	43912	109	2	40	9	28	57	1	17	10	16	20	2	2	2	3	8	1	1	0	1
fw5	45723	39794	0, 1, 2, 3, 4	39007	94	2	13	38	58	49	2	8	6	21	8	2	2	2	3	5	1	2	1	2
ipc1	49840	48294	0, 1, 3, 4	48385	22	2	2	9	16	16	1	3	6	11	6	1	2	1	2	4	1	1	0	0
ipc2	50000	50000	0, 1	50000	2	2	2	0	0	1	1	1	0	0	2	2	2	0	0	1	1	1	0	0
cisco1	584	538	0, 1, 3, 4	406	15	8	13	2	8	10	4	7	3	4	9	5	7	2	3	5	2	4	1	2
cisco2	269	249	0, 1, 4	246	4	2	3	1	1	2	2	2	0	0	4	2	3	1	2	2	1	2	0	1
cisco3	95	92	0, 1, 3, 4	89	5	3	5	2	3	3	2	3	1	2	4	2	4	2	2	2	1	2	0	1
cisco4	364	329	0, 1, 3, 4	324	7	3	5	2	4	4	2	3	1	1	5	2	3	2	3	3	1	2	1	2
cisco5	148	120	0, 1	120	3	2	3	0	1	2	2	2	0	0	2	2	2	0	0	1	1	1	0	0

Cont.

TABLE IV

SIMULATION RESULTS: SIZES IN KBITS CORRESPONDING TO RESULTS SHOWN IN TABLE III. TCAM SPACE AND POWER CONSUMPTION FIGURES ARE BASED ON 80-BIT TCAM CELLS WITH BLOCKS OF SIZE 2 K CELLS [35] AND POWER CONSUMPTION OF 1.85 WT PER MBIT OF THE FINAL TCAM (IN FULL BLOCKS) [2]

	Original						k-MRC						MRC, $\{0,1\}$						MGR, 95%						MGR, 99%					
	Binary			SRGE			Binary			SRGE			Binary			SRGE			Binary			SRGE			Binary			SRGE		
	Kbits	Blocks	Wt	Kbits	Blocks	Wt	Kbits	Blocks	Wt	Kbits	Blocks	Wt	Kbits	Blocks	Wt	Kbits	Blocks	Wt	Kbits	Blocks	Wt	Kbits	Blocks	Wt	Kbits	Blocks	Wt	Kbits	Blocks	Wt
acl1	7922.1	68	20.1	7656.0	66	19.5	2647.8	34	10.1	2559.1	33	9.8	2121.3	34	10.1	2050.4	33	9.8	2122.5	35	10.4	2051.6	34	10.1	2122.5	35	10.4	2051.6	34	10.1
acl2	11289.1	97	28.7	11289.1	97	28.7	5615.9	51	15.1	5615.9	51	15.1	3587.5	52	15.4	3587.5	52	15.4	3124.2	49	14.5	3124.2	49	14.5	3034.3	49	14.5	3034.3	49	14.5
acl3	10771.6	92	27.2	10571.2	91	26.9	6775.5	50	14.8	6651.1	49	14.5	3537.6	50	14.8	3464.1	49	14.5	2931.9	47	13.9	2880.7	46	13.6	2880.2	47	13.9	2827.1	46	13.6
acl4	10079.3	87	25.8	9904.9	85	25.2	6430.7	48	14.2	6322.5	47	13.9	4215.7	52	15.4	4130.9	51	15.1	2794.1	44	13.0	2751.3	43	12.7	2712.8	44	13.0	2666.9	43	12.7
acl5	6121.9	53	15.7	6121.9	53	15.7	3152.2	29	8.6	3152.2	29	8.6	3797.4	39	11.5	3797.4	39	11.5	1747.1	27	8.0	1747.1	27	8.0	1651.5	27	8.0	1651.5	27	8.0
fw1	19454.1	167	49.4	19438.6	166	49.1	11976.3	87	25.8	11961.1	87	25.8	5727.9	87	25.8	5712.4	87	25.8	5213.5	84	24.9	5213.5	84	24.9	5203.0	84	24.9	5202.7	84	24.9
fw2	10866.8	93	27.5	10866.8	93	27.5	5076.4	47	13.9	5076.4	47	13.9	2922.7	47	13.9	2922.7	47	13.9	2906.9	48	14.2	2906.9	48	14.2	2906.9	48	14.2	2906.9	48	14.2
fw3	15090.0	129	38.2	15073.1	129	38.2	9348.0	68	20.1	9332.0	68	20.1	4661.8	69	20.4	4645.0	68	20.1	4110.7	66	19.5	4110.7	66	19.5	4034.9	65	19.2	4034.2	65	19.2
fw4	33500.3	286	84.7	33368.8	285	84.4	20427.4	147	43.5	20341.2	146	43.2	9491.1	147	43.5	9441.1	146	43.2	8899.5	142	42.0	8869.9	142	42.0	8853.8	143	42.3	8821.4	143	42.3
fw5	12478.5	107	31.7	12445.4	107	31.7	7810.9	57	16.9	7785.5	57	16.9	4007.0	58	17.2	3985.8	57	16.9	3369.6	54	16.0	3365.3	54	16.0	3334.9	54	16.0	3329.7	54	16.0
ipc1	8041.1	69	20.4	7924.2	68	20.1	3871.2	36	10.7	3815.8	35	10.4	2283.1	36	10.7	2250.2	35	10.4	2157.3	36	10.7	2126.1	35	10.4	2157.3	36	10.7	2126.1	35	10.4
ipc2	5859.4	51	15.1	5859.4	51	15.1	1562.5	26	7.7	1562.5	26	7.7	1562.5	26	7.7	1562.5	26	7.7	1562.5	27	8.0	1562.5	27	8.0	1562.5	27	8.0	1562.5	27	8.0
cisco1	78.3	1	0.3	78.3	1	0.3	40.6	1	0.3	40.6	1	0.3	40.0	1	0.3	40.0	1	0.3	22.3	1	0.3	22.3	1	0.3	20.4	1	0.3	20.4	1	0.3
cisco2	68.6	1	0.3	68.6	1	0.3	24.4	1	0.3	24.4	1	0.3	20.7	1	0.3	20.7	1	0.3	19.0	1	0.3	19.0	1	0.3	18.4	1	0.3	18.4	1	0.3
cisco3	11.1	1	0.3	11.1	1	0.3	5.4	1	0.3	5.4	1	0.3	3.5	1	0.3	3.5	1	0.3	3.1	1	0.3	3.1	1	0.3	2.9	1	0.3	2.9	1	0.3
cisco4	79.7	1	0.3	79.7	1	0.3	40.4	1	0.3	40.4	1	0.3	26.1	1	0.3	26.1	1	0.3	21.5	1	0.3	21.5	1	0.3	21.0	1	0.3	21.0	1	0.3
cisco5	19.6	1	0.3	19.6	1	0.3	7.6	1	0.3	7.6	1	0.3	7.6	1	0.3	7.6	1	0.3	5.5	1	0.3	5.5	1	0.3	5.2	1	0.3	5.2	1	0.3

Outline

- Introduction
- Order-independence
- Problem formulation and theorem
- Solutions
- Simulations
- Conclusions

Conclusions

New properties of classifiers that ignores superfluous information in classification lookup

Simplifies classifier matching by firstly solve the rule selection problem which is based only on parts of rules then check false positive on the remainder of the fields not considered

The proposed techniques may be reused in neighboring areas (matching in databases or data mining, Boolean minimization)

Thank you！

Reference

- [1] K. Kogan, S. Nikolenko, O. Rottenstreich, W. Culhane, and P. Eugster, Exploiting Order Independence for Scalable and Expressive Packet Classification. *IEEE Trans on Networks*, Vol. 24, No. 2, pp.1251-1266, 2016.
- [2] K. Kogan, S. Nikolenko, O. Rottenstreich, W. Culhane, and P. Eugster, SAX-PAC (Scalable And eXpressive PAcket Classification), in *Proc. Of ACM SIGCOMM 2014,* 2014.
- [3] M. Kerber, Heuristics for Resolution in Propositional Logic*, in Proc. of the 32nd annual German conference on Advances in artificial intelligence*, 2009.
- [4] J. A. Robinson, “A machine-oriented logic based on the resolution principle,” *J. ACM*, vol. 12, no. 1, pp. 23–41, 1965.
- [5] A. Bremler-Barr and D. Hendler, “Space-efficient TCAM-based classification using gray coding,” *IEEE Trans. Computers*, vol. 61, no. 1, pp. 18–30, 2012.

Reference

- [6] D. Taylor and J. Turner, A Packet Classification Benchmark. *IEEE/ACM Trans on Networks,* Vol. 15, No. 3, pp. 499-511, 2007.