A Heuristic Algorithm for Relaxed Optimal Rule Ordering Problem

Takashi Harada[†], Ken Tanaka[†] and Kenji Mikawa[‡]

[†]Kanagawa University, Japan
[‡]Niigata University, Japan

Research Theme

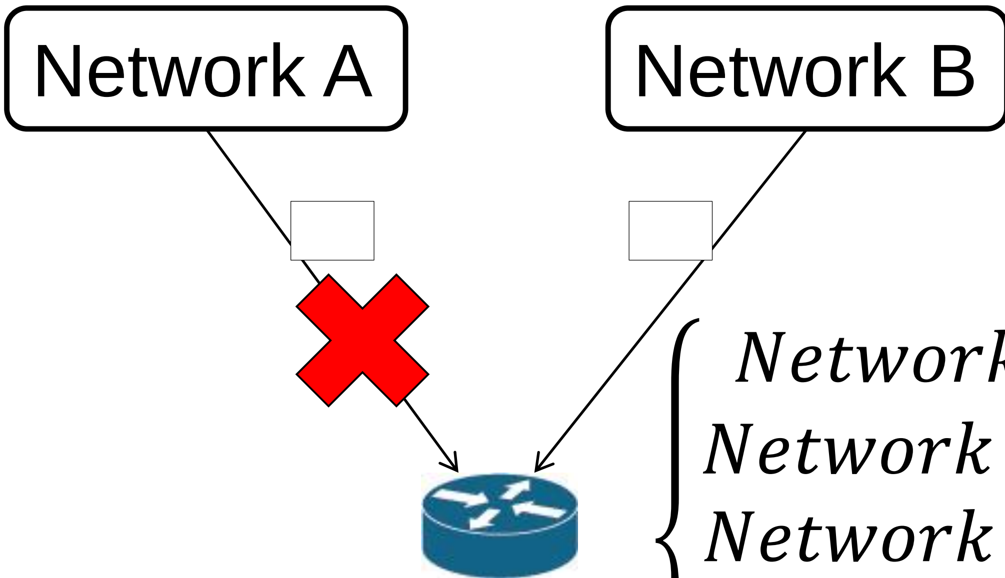
Acceleration of packet filtering

by reordering rules in a rule list

Table of Contents

- Packet Filtering Model
- Optimal Rule Ordering (ORO)
- Relaxed Optimal Rule Ordering (RORO)
- Heuristic algorithm for RORO
- Experiments
- Conclusion and Future Work

Packet Filtering



$$\begin{cases} Network\ A : Deny \\ Network\ B : Permit \\ Network\ C : Permit \\ \vdots \\ Network\ Z : Deny \end{cases}$$

Filtering packets according to the policy

Acceleration of Packet Filtering

- Using a special hardware, like TCAM
- Using software-based algorithm
- Reordering rules in a rule list
- Reconstructing a rule list

Form of packet and Rule

We regard packet as a bit string of length w.

e.g. $w = 4, \qquad p = 0100$

We regard condition of rule as a string on $\{0,1,*\}^w$.

$$r_i^e = b_1 b_2 \cdots b_w \quad (b_i \in \{0,1,*\}, e \in \{\, P, D \,\})$$

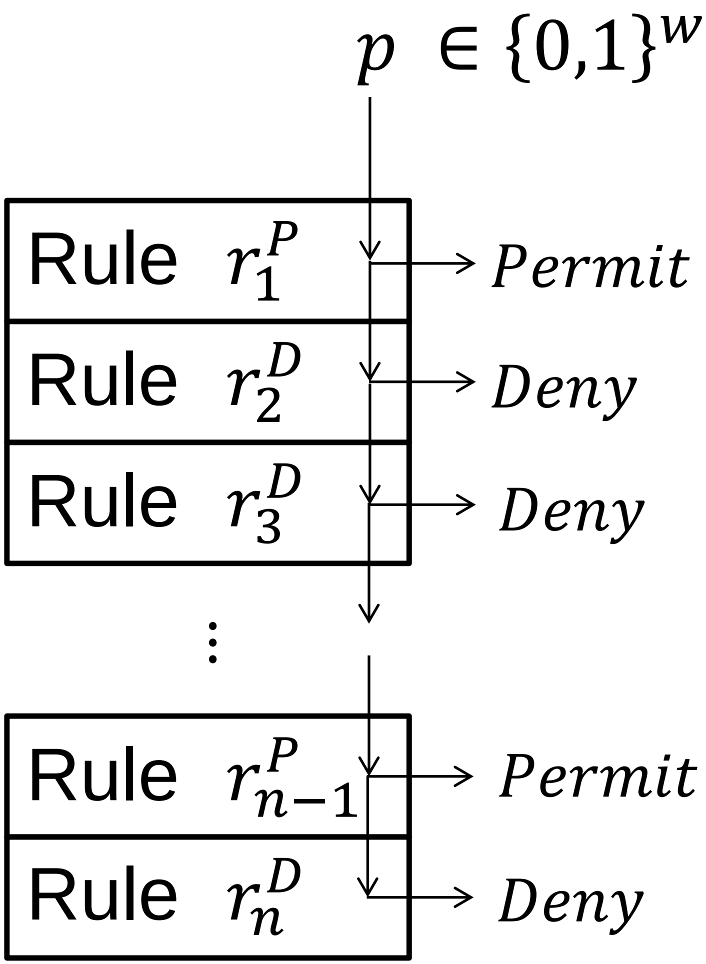
e.g. $w = 4, \qquad r_2^P = *\, 1\, *\, 0$

Packet Filtering

Take a bit sequence $(01010001 \cdots 101000001)$,

and return $Permit$ or $Deny$ according to a policy.

Model of Packet Filtering



A packet is compared with each rule in order,

assigned the evaluation type of the first matched rule.

Packet Filtering

e.g.

$p = 0111,$
D is assigned to p.

$R(0111) = D$

Filter R	$\lvert E(R,i)\rvert_U$
$r_1^P = * \, 0 * 1$	4
$r_2^P = 0\,0\,0\,0$	1
$r_3^P = 0 * 0\,0$	1
$r_4^D = 0 * 1 *$	3
$r_5^P = * \, 1 * 1$	3
$r_6^P = * * * \, 1$	0
$r_7^D = * * * *$	4

Filtering Policy

$$
\begin{array}{ll}
0000 \mapsto P & 1000 \mapsto D \\
0001 \mapsto P & 1001 \mapsto P \\
0010 \mapsto D & 1010 \mapsto D \\
0011 \mapsto P & 1011 \mapsto P \\
0100 \mapsto P & 1100 \mapsto D \\
0101 \mapsto P & 1101 \mapsto P \\
0110 \mapsto D & 1110 \mapsto D \\
0111 \mapsto D & 1111 \mapsto P
\end{array}
$$

Filter R	$\lvert E(R,i) \rvert_U$
$r_1^P = * \ 0 * 1$	4
$r_2^P = \ 0\ 0\ 0\ 0$	1
$r_3^P = 0 * 0\ 0$	1
$r_4^D = 0 * 1 *$	3
$r_5^P = * \ 1 * 1$	3
$r_6^P = * * * \ 1$	0
$r_7^D = * * * *$	4

The table on the right shows the policy on the left.

Policy Violation

$0000 \mapsto P$	$0001 \mapsto P$	$0010 \mapsto D$	$0011 \longmapsto P$
$0100 \mapsto P$	$0101 \mapsto P$	$0110 \mapsto D$	$0111 \mapsto D$
$1000 \mapsto D$	$1001 \mapsto P$	$1010 \mapsto D$	$1011 \mapsto P$
$1100 \mapsto D$	$1101 \mapsto P$	$1110 \mapsto D$	$1111 \mapsto P$

$0000 \mapsto P$	$0001 \mapsto P$	$0010 \mapsto D$	$0011 \longmapsto P$
$0100 \mapsto P$	$0101 \mapsto P$	$0110 \mapsto D$	$0111 \mapsto P$
$1000 \mapsto D$	$1001 \mapsto P$	$1010 \mapsto D$	$1011 \mapsto P$
$1100 \mapsto D$	$1101 \mapsto P$	$1110 \mapsto D$	$1111 \mapsto P$

Filter R	Filter R'
$r_1^P = * \ 0 * 1$	$r_1^P = * \ 0 * 1$
$r_2^P = 0\,0\,0\,0$	$r_2^P = 0\,0\,0\,0$
$r_3^P = 0 * 0\,0$	$r_3^P = 0 * 0\,0$
$r_4^D = 0 * 1 *$	$r_5^P = * \ 1 * 1$
$r_5^P = * \ 1 * 1$	$r_4^D = 0 * 1 *$
$r_6^P = * * * 1$	$r_6^P = * * * 1$
$r_7^D = * * * *$	$r_7^D = * * * *$

Interchanging r_4^D and r_5^P, 0111's action is changed from D to P

Policy violation occurs !

$M(r_i)$

$M(r_i)$ is a set of packets that can match r_i^e regardless of upper rules.

e.g.

$$M(r_5) = \{\, 0101, 0111, 1101, 1111 \,\}.$$

Filter R	$\lvert E(R,i)\rvert_U$
$r_1^P = * \; 0 * 1$	4
$r_2^P = \; 0\;0\;0\;0$	1
$r_3^P = 0 * 0\;0$	1
$r_4^D = 0 * 1 *$	3
$r_5^P = * \; 1 * 1$	3
$r_6^P = * * * \; 1$	0
$r_7^D = * * * *$	4

$E(R, i)$

$E(R, i)$ is a set of packets that are evaluated by r_i^e.

e.g.
$E(R, 5) = \{\, 0101, 1101, 1111 \,\}$.

Because 0111 is evaluated r_4^D,
$E(R, 5)$ is different from $M(r_5)$.

Filter R	$\lvert E(R, i) \rvert_U$
$r_1^P = * 0 * 1$	4
$r_2^P = 0\,0\,0\,0$	1
$r_3^P = 0 * 0\,0$	1
$r_4^D = 0 * 1 *$	3
$r_5^P = * 1 * 1$	3
$r_6^P = * * * 1$	0
$r_7^D = * * * *$	4

Packet Arrival Distribution F

A packet arrival distribution F is mapping from $\{0,1\}^w$ to $\mathbb{N}$.

e.g.

$$F(0010) = 0,$$
$$F(0011) = 3,$$
$$F(0100) = 10$$

$$\begin{array}{rrrr} 0000 \mapsto & 20 & 1000 \mapsto & 0 \\ 0001 \mapsto & 0 & 1001 \mapsto & 1 \\ 0010 \mapsto & 0 & 1010 \mapsto & 13 \\ 0011 \mapsto & 3 & 1011 \mapsto & 0 \\ 0100 \mapsto & 10 & 1100 \mapsto & 0 \\ 0101 \mapsto & 2 & 1101 \mapsto & 0 \\ 0110 \mapsto & 0 & 1110 \mapsto & 0 \\ 0111 \mapsto & 0 & 1111 \mapsto & 7 \end{array}$$

$|P|_F$

Let P be a set of packets and F be a packet arrival distribution.

$$\boxed{|P|_F \equiv \sum_{p \in P} F(p)}$$

e.g.

$P = \{1110, 1111\}$

$|P|_F = 0 + 7 = 7$

$$\begin{array}{rcrcrcr} 0000 & \mapsto & 20 & \quad & 1000 & \mapsto & 0 \\ 0001 & \mapsto & 0 & & 1001 & \mapsto & 1 \\ 0010 & \mapsto & 0 & & 1010 & \mapsto & 13 \\ 0011 & \mapsto & 3 & & 1011 & \mapsto & 0 \\ 0100 & \mapsto & 10 & & 1100 & \mapsto & 0 \\ 0101 & \mapsto & 2 & & 1101 & \mapsto & 0 \\ 0110 & \mapsto & 0 & & 1110 & \mapsto & 0 \\ 0111 & \mapsto & 0 & & 1111 & \mapsto & 7 \end{array}$$

Computing $|E(R,i)|_U$

Filter R
$r_1^P = * 0 * 1$
$r_2^P = 0\ 0\ 0\ 0$
$r_3^P = 0 * 0\ 0$
$r_4^D = 0 * 1 *$
$\mathrm{r}_5^{\mathrm{P}} = * 1 * 1$
$r_6^P = * * * 1$
$r_7^D = * * * *$

Computing $|E(R,i)|_U$ →

$\|E(R,i)\|_U$
4
1
1
3
3
0
4

Computing $|E(R,n)|_U$ is $\#P$-Complete

Filtering Latency $L(R_\sigma, F)$

Regard comparison of a packet and some rule as the latency 1,

$$L(R_\sigma, F) \equiv \sum_{i=1}^{n-1} \sigma(i)|E(R_\sigma, i)|_F + (n-1)|E(R_\sigma, n)|$$

where, R is a rule list, F is a packet arrival distribution and σ is an order of rules.

Filtering Latency $L(R_\sigma, F)$

Uniform Distribution U

$0000 \mapsto 1$ $\quad$ $1000 \mapsto 1$
$0001 \mapsto 1$ $\quad$ $1001 \mapsto 1$
$0010 \mapsto 1$ $\quad$ $1010 \mapsto 1$
$0011 \mapsto 1$ $\quad$ $1011 \mapsto 1$
$0100 \mapsto 1$ $\quad$ $1100 \mapsto 1$
$0101 \mapsto 1$ $\quad$ $1101 \mapsto 1$
$0110 \mapsto 1$ $\quad$ $1110 \mapsto 1$
$0111 \mapsto 1$ $\quad$ $1111 \mapsto 1$

Filter R	$\lvert E(R,i)\rvert_U$
$r_1^P = * \, 0 * 1$	4
$r_2^P = \ 0\,0\,0\,0$	1
$r_3^P = 0 * 0\,0$	1
$r_4^D = 0 * 1 \, *$	3
$r_5^P = * \, 1 * 1$	3
$r_6^P = * * * \, 1$	0
$r_7^D = * * * *$	4

$$L(R,U) = 1 \cdot 4 + 2 \cdot 1 + 3 \cdot 1 + 4 \cdot 3 + 5 \cdot 3 + 6 \cdot 0 + 6 \cdot 4$$
$$= 60$$

Policy and Reordering rules

Filter R	$\lvert E(R,i)\rvert_U$
$r_1^P = * \ 0 * 1$	4
$r_2^P = \ 0 \ 0 \ 0 \ 0$	1
$r_3^P = 0 * 0 \ 0$	1
$r_4^D = 0 * 1 *$	3
$r_5^P = * \ 1 * 1$	3
$r_6^P = * * * \ 1$	0
$r_7^D = * * * *$	4
$L(R,U) = 60$	

Filter R_π	$\lvert E(R_\pi,i)\rvert_U$
$r_1^P = * \ 0 * 1$	4
$r_4^D = 0 * 1 *$	3
$r_5^P = * \ 1 \ * 1$	3
$r_3^P = 0 * 0 \ 0$	2
$r_2^P = \ 0 \ 0 \ 0 \ 0$	0
$r_6^P = * * * \ 1$	0
$r_7^D = * * * *$	4
$L(R_\pi,U) = 51$	

R and $R_{\pi=(1\ 5\ 4\ 2\ 3\ 6\ 7)}$ denote the same policy

Optimal Rule Ordering

Reordering rules can reduce the latency caused by filtering.

⇓

Find the order of rules that minimize the filtering latency!

Overlap Relation

If there is a packet p that matches both r_i and r_j, r_i and r_j are said to be overlapped.

e.g.

Because, there is packet 0000 that matches r_2^P and r_3^P, r_2^P and r_3^P are overlapped.

Filter R
$r_1^P = * \, 0 * 1$
$r_2^P = \; 0\,0\,0\,0$
$r_3^P = 0 * 0\,0$
$r_4^D = 0 * 1 *$
$r_5^P = * \, 1 * 1$
$r_6^P = * * * \, 1$
$r_7^D = * * * *$

Optimal Rule Ordering (conventional)

Optimal Rule Ordering	
Input	Rule list R and packet arrival distribution F
Output	Order of rules σ that minimizes $L(R_\sigma, F)$ s.t. $\forall i, j,\ i < j\ \land O(r_i, r_j) \Rightarrow\ \sigma(i) <\ \sigma(j)$

where $O(r_i, r_j)$ denotes that r_i and r_j are overlapped

$\boxed{\forall i, j,\ i < j\ \land O(r_i, r_j) \Rightarrow\ \sigma(i) <\ \sigma(j)}$ means that if r_i and r_j are overlapped, r_j can't be placed ahead of r_j.

Interchangeable condition (1)

Even though $O(r_i, r_j)$, we can place r_j above of r_i.

e.g.

Although r_2^P and r_3^P are overlapped, interchanging r_2^P and r_3^P holds policy.

Filter R
$r_1^P = * 0 * 1$
$r_2^P = 0 0 0 0$
$r_3^P = 0 * 0 0$
$r_4^D = 0 * 1 *$
$r_5^P = * 1 * 1$
$r_6^P = * * * 1$
$r_7^D = * * * *$

Filter R'
$r_1^P = * 0 * 1$
$r_3^P = 0 * 0 0$
$r_2^P = 0 0 0 0$
$r_4^D = 0 * 1 *$
$r_5^P = * 1 * 1$
$r_6^P = * * * 1$
$r_7^D = * * * *$

Dependency Relation

If r_i^e and r_j^f are overlapped and e is different from f, r_i^e and r_j^f are said to be dependent.

Filter R
$r_1^P = * \ 0 * 1$
$r_2^P = \ 0 \ 0 \ 0 \ 0$
$r_3^P = 0 * 0 \ 0$
$r_4^D = 0 * 1 *$
$r_5^P = * \ 1 * 1$
$r_6^P = * * * \ 1$
$r_7^D = * * * *$

e.g.

Because, r_4^D and r_5^P are overlapped and those actions are different, r_4^D and r_5^P are dependent

Interchanging r_4^D and r_5^P cause policy violation!!

Dependency Relation

Even if r_i^e and r_j^f are overlapped, if r_i^e and r_j^f are not dependent, we can place r_j^f ahead of r_i^e

?

r_i^e and r_j^f are dependent $\Leftrightarrow$ r_j^f can't be placed ahead of r_i^e

Left direction $\Leftarrow$ is true, but …

Interchangeable condition (2)

Filter R	$\lvert E(R,i)\rvert_U$
$r_1^P = * \, 0 * 1$	4
$r_2^P = \ 0\,0\,0\,0$	1
$r_3^P = 0 * 0\,0$	1
$r_4^D = 0 * 1 *$	3
$r_5^P = * \, 1 * 1$	3
$r_6^P = * * * \, 1$	0
$r_7^D = * * * *$	4

Filter R'	$\lvert E(R',i)\rvert_U$
$r_1^P = * \, 0 * 1$	4
$r_2^P = \ 0\,0\,0\,0$	1
$r_3^P = 0 * 0\,0$	1
$r_4^D = 0 * 1 *$	3
$r_5^P = * \, 1 * 1$	3
$r_7^D = * * * *$	4
$r_6^P = * * * \, 1$	0

Even though r_6^P and r_7^D are dependent, because r_6^P evaluate no packet ($E(R,6) = \phi$), we can exchange r_6^P and r_7^D.

Note that we can't still place r_6^P above of r_4^P

Relaxed Optimal Rule Ordering

Relaxed Optimal Rule Ordering	
Input	Rule list R and packet arrival distribution F
Output	Order of rules σ that minimizes $L(R_\sigma, F)$ s.t. holding the filtering policy.

Holding the filtering policy is the most important point

Varying weights of rules

Filter R	$\lvert E(R\ i)\rvert_U$
$r_1^P = * 0 * 1$	4
$r_2^P = 0 0 0 0$	1
$r_3^P = 0 * 0 0$	1
$r_4^D = 0 * 1 *$	3
$r_5^P = * 1 * 1$	3
$r_6^P = * * * 1$	0
$r_7^D = * * * *$	4

Filter R'	$\lvert E(R', i)\rvert_U$
$r_1^P = * 0 * 1$	4
$r_3^P = 0 * 0 0$	2
$r_2^P = 0 0 0 0$	0
$r_4^D = 0 * 1 *$	3
$r_5^P = * 1 * 1$	3
$r_6^P = * * * 1$	0
$r_7^D = * * * *$	4

Interchanging overlap rules may cause varying weights of rules as noted above.

Improved Reordering Methods Based on [8]

The algorithm [8] ignores the variation of weights (In [8], the weight of rule r_i^e is denoted w_i as a constant.)

Propose an algorithm considering the variation of weights

[8] K. Tanaka, K. Mikawa, and M. Hikin, “A heuristic algorithm for reconstructing a packet filter with dependent rules,” IEICE Trans. Commun., vol 96, no. 1, pp. 155-162, Jan 2013

Interchange Adjacent Rules

In the [8], if r_i^e and r_{i+1}^f are interchangeable $\mathrm{w_i} < w_{i+1}$ holds, interchange r_i^e and r_{i+1}^f.

Consider the variation of weights

Let r_i^e and r_{i+1}^e are ith and $i+1$th rule respectively and overlapped. If

$$i|E(R,i)|_F + (i+1)|E(R,i+1)|_F$$
$$> i(|E(R,i)|_F + |E(R,i)\ \cap M(r_{i+1})|_F) + (i+1)(|E(R,i) \setminus M(r_k)|_F)$$

holds, interchange r_i^e and r_{i+1}^e.

Interchange Adjacent Rules

Filter R	$\|E(R\ i)\|_U$
$r_1^P = * 0 * 1$	4
$r_2^P = 0\ 0\ 0\ 0$	1
$r_3^P = 0 * 0\ 0$	1

Filter R'	$\|E(R', i)\|_U$
$r_1^P = * 0 * 1$	4
$r_3^P = 0 * 0\ 0$	2
$r_2^P = 0\ 0\ 0\ 0$	0

Because $w_2 = 1 < 1 = w_3$ doesn't holds, the [8] doesn't interchange r_2^P and r_3^P.

In contrast to this, proposed method interchange them by considering the variation of weights.

Interchange of Single Rule and Consecutive Rules

Filter R	$\lvert E(R\ i)\rvert_F$
$r_3^P = 1\ 0\ 0\ *$	4
$r_2^P = 1 * 1\ 1$	3
$r_1^D = 1\ 0\ 0 *$	2
$r_4^P = * 0 * *$	28

Filter R	$\lvert E(R\ i)\rvert_F$
$r_3^P = 1\ 0\ 0\ *$	4
$r_1^D = 1\ 0\ 0 *$	2
$r_4^P = * 0 * *$	30
$r_2^P = 1 * 1\ 1$	1

Let $F(1011) = 2$.

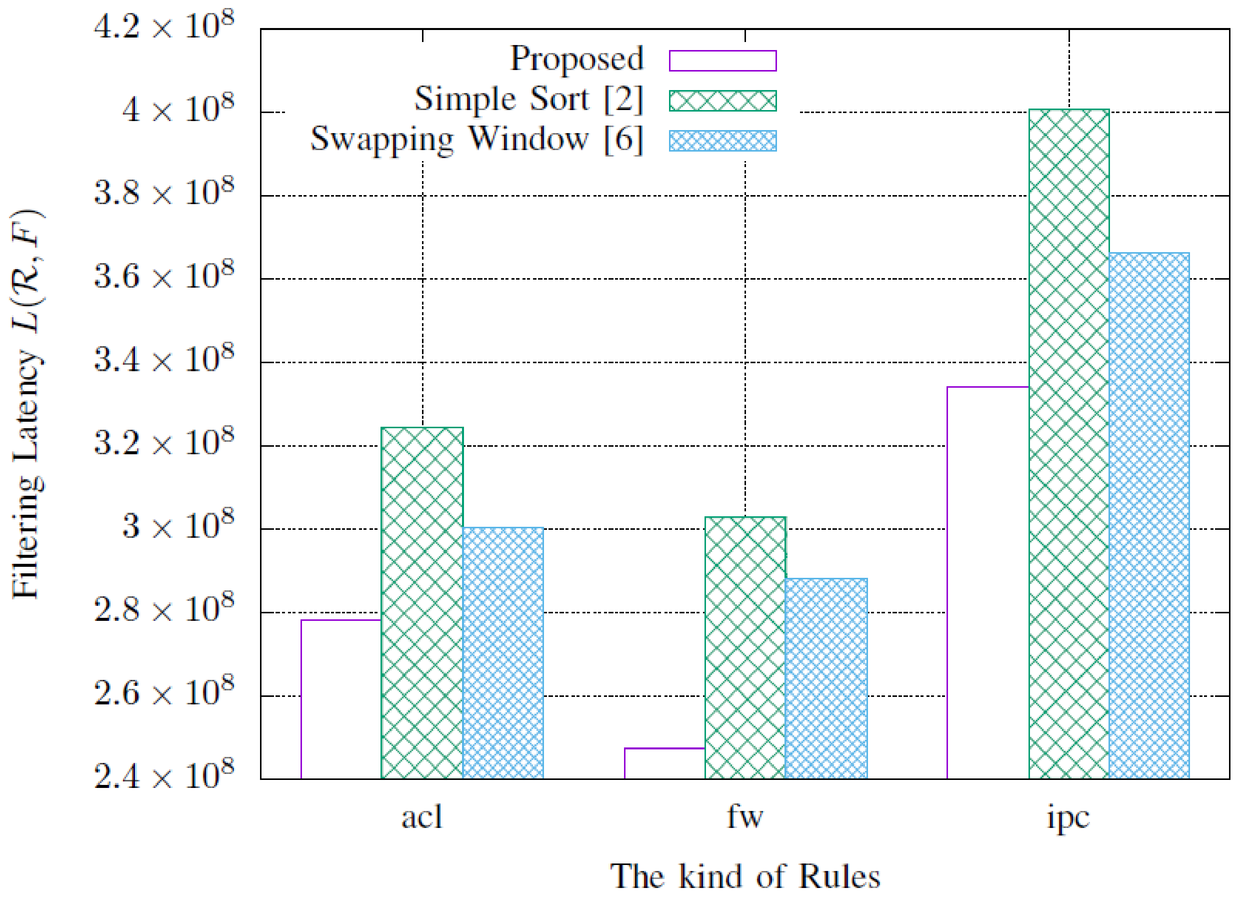
In this case, we also consider the variation of weights.

Experiments

Latency		Acl	fw	ipc
Fixed weight	Method [8]	2.99843×10^8	2.60785×10^8	3.46560×10^8
varying weight	method [8]	2.94295×10^8	2.60504×10^8	3.44709×10^8
	proposed	2.78112×10^8	2.47181×10^8	3.33953×10^8

To solve RORO, the variation of weights should be considered

Experiments



Proposed method decreases the latency compared with [2] and [6]

[2] E. W. Fulp, "Optimization of network firewall policies using directed acyclic graphs," in In proc, IEEE Internet Management Conf, extended abstract, 2005
[6] R. Mohan A. Yazidi, B. Feng, and B. J. Oommen, “Dynamic ordering of firewall rules using a novel swapping window-based paradigm,” in Proceedings of ICCNS ‘16. NY, ACM, 2016, pp.11-20

Conclusion and Future work

Conclusion

- Introduced Relaxed Optimal Rule Ordering Problem (RORO)
- Proposed a heuristic for RORO

Future Work

- Develop a heuristic for a large rule list

Thank you for listening