

重み平均に基づくペアリングによる ルール並び替え法

渕野 敬¹ 原田 崇司² 田中 賢² 三河 賢治³

¹ 神奈川大学理学部情報科学科

² 神奈川大学大学院理学研究科

³ 新潟大学学術情報基盤機構情報基盤センター

2018 年 10 月 12 日

目次

- ① パケット分類
 - パケット分類とルールリスト
 - 遅延
 - ルール順序最適化問題
- ② ペアリング法
 - 基本的な考え方
 - ペアリング法の問題点
 - 従属関係を考慮した r_{min} の選択方法
 - 提案手法: 従属関係を考慮した r_{max} の選択方法
- ③ 計算機実験
- ④ まとめと今後の課題

ネットワーク機器のパケット分類

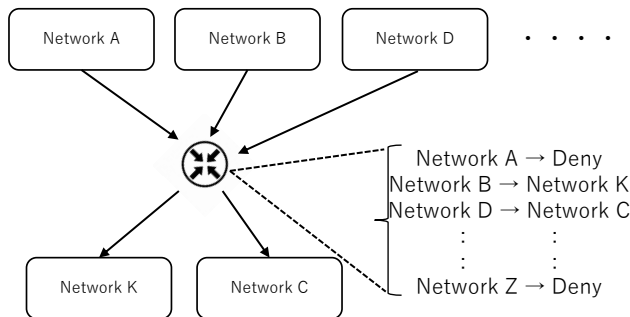


図 1: ネットワーク機器のパケット分類

ポリシーにしたがってパケットを分類

パケット分類

ポリシー パケットの分類についての明示的な取り決め
ルールリスト ポリシーにしたがってパケットを分類するルールの一覧

1. ポリシーを満たすルールリストを作成
2. ルールリストを線形探索して分類
 $(r_1, r_2, r_3, \dots, r_n)$ の順にパケットと照合)

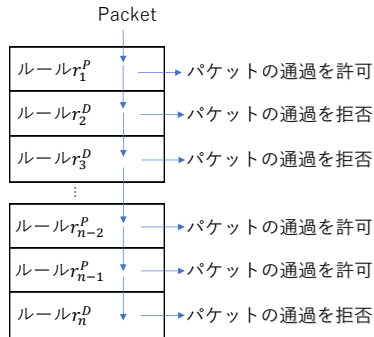


図 2: パケット分類 (フィルタリングの場合)

ルールリストの形式

表 1: ルールリスト

フィルタ $\mathcal{R}$
$r_1^P = 0*101$
$r_2^D = 01***$
$r_3^D = 10**0$
$r_4^D = 00**1$
$r_5^D = 111*1$
$r_6^P = 11*0*$
$r_7^P = 0*1**$
$r_8^P = 101**$
$r_9^D = 101**$

ルール

- ルール番号 $i \in \mathbb{N}$
- パケットは長さ w のビット列
- 条件 $c \in \{0, 1, *\}^w$ の組. ただし, ‘*’ は don't care
- 形式 $r_i^e = c, e \in \{P, D\}$

e.g. パケット 11101 は, $r_1 \cdots r_4$ には合致せず r_5 に合致
 $\Rightarrow r_5$ のアクションを適用

▷ σ は並び替え操作. $\mathcal{R}_\sigma$ は σ によって並び替えたルールリスト.

e.g. $\sigma = (3, 4, 1, 6, 5, 7, 8, 2, 9)$

表 2: 並び替え前

フィルタ $\mathcal{R}$
$r_1^P = 0*101$
$r_2^D = 01***$
$r_3^D = 10**0$
$r_4^D = 00**1$
$r_5^D = 111*1$
$r_6^P = 11*0*$
$r_7^P = 0*1**$
$r_8^P = 101**$
$r_9^D = 101**$

表 3: σ で並び替え

フィルタ $\mathcal{R}_\sigma$
$r_3^D = 10**0$
$r_8^P = 101**$
$r_1^P = 0*101$
$r_2^D = 01***$
$r_5^D = 111*1$
$r_4^D = 00**1$
$r_6^P = 11*0*$
$r_7^P = 0*1**$
$r_9^D = 101**$

評価パケット数 $|r_i|_F$

評価パケット数 $|r_i|_F$

分布 F のもとで、ルールリスト $\mathcal{R}$ の $1 \sim (i-1)$ 番目までのルールに合致せず、 r_i^e に合致するパケットの数

表 4: ルールリスト

フィルタ $\mathcal{R}$
$r_1^P = 0*101$
$r_2^D = 01***$
$r_3^D = 10**0$
$r_4^D = 00**1$
$r_5^D = 111*1$
$r_6^P = 11*0*$
$r_7^P = 0*1**$
$r_8^P = 101**$
$r_9^D = *****$

表 5: パケットの分布 F

00000 : 20	00001 : 7	00010 : 17	00011 : 5
00100 : 45	00101 : 2	00110 : 55	00111 : 4
01000 : 1	01001 : 1	01010 : 1	01011 : 1
01100 : 2	01101 : 3	01110 : 1	01111 : 1
10000 : 3	10001 : 20	10010 : 3	10011 : 23
10100 : 20	10101 : 60	10110 : 15	10111 : 25
11000 : 10	11001 : 10	11010 : 24	11011 : 18
11100 : 20	11101 : 15	11110 : 18	11111 : 35

表 6: F のもとで計算された各
ルールの評価パケット数

フィルタ $\mathcal{R}$	$ r_i _F$
$r_1^P = 0*101$	5
$r_2^D = 01***$	10
$r_3^D = 10**0$	6
$r_4^D = 00**1$	20
$r_5^D = 111*1$	50
$r_6^P = 11*0*$	40
$r_7^P = 0*1**$	100
$r_8^P = 101**$	120
$r_9^D = *****$	130

表 7: パケットの分布 F

00000 : 20	00001 : 7	00010 : 17	00011 : 5
00100 : 45	00101 : 2	00110 : 55	00111 : 4
01000 : 1	01001 : 1	01010 : 1	01011 : 1
01100 : 2	01101 : 3	01110 : 1	01111 : 1
10000 : 3	10001 : 20	10010 : 1	10011 : 23
10100 : 1	10101 : 80	10110 : 1	10111 : 40
11000 : 10	11001 : 10	11010 : 24	11011 : 18
11100 : 20	11101 : 15	11110 : 18	11111 : 35

R_8 が評価するパケットは $\{10101, 10111\}$
 $|r_8|_F = 80 + 40 = 120$

遅延



i 番目のルールによって評価型が決まるパケットは i 回の照合を受ける. 1 回の照合を遅延 1 と考え, ネットワーク機器の遅延 $L(\mathcal{R}, F)$ を定義

定義 : 遅延 $L(\mathcal{R}, F)$

$$L(\mathcal{R}, F) = \sum_{i=1}^{n-1} i|r_i|_F + (n-1)|r_n|_F$$

表 8: ルールリスト

フィルタ $\mathcal{R}$	$ r_i _F$
$r_1^P = 0*101$	5
$r_2^D = 01***$	10
$r_3^D = 10**0$	6
$r_4^D = 00**1$	20
$r_5^D = 111*1$	50
$r_6^P = 11*0*$	40
$r_7^P = 0*1**$	100
$r_8^P = 101**$	120
$r_9^D = *****$	130
$L(\mathcal{R}, F) = 3293$	

表 9: パケットの分布 F

00000 : 20	00001 : 7	00010 : 17	00011 : 5
00100 : 45	00101 : 2	00110 : 55	00111 : 4
01000 : 1	01001 : 1	01010 : 1	01011 : 1
01100 : 2	01101 : 3	01110 : 1	01111 : 1
10000 : 3	10001 : 20	10010 : 1	10011 : 23
10100 : 1	10101 : 80	10110 : 1	10111 : 40
11000 : 10	11001 : 10	11010 : 24	11011 : 18
11100 : 20	11101 : 15	11110 : 18	11111 : 35

$$\begin{aligned}
 L(\mathcal{R}, F) &= \sum_{i=1}^{9-1} i|r_i|_F + (9-1)|r_9|_F \\
 &= |r_1|_F + 2 \times |r_2|_F + 3 \times |r_3|_F + \cdots + 8 \times |r_8|_F + 8 \times |r_9|_F \\
 &= 3293
 \end{aligned}$$

表 10: ポリシー

00000 : D	00001 : D	00010 : D	00011 : D
00100 : P	00101 : P	00110 : P	00111 : D
01000 : D	01001 : D	01010 : D	01011 : D
01100 : D	01101 : P	01110 : D	01111 : D
10000 : D	10001 : D	10010 : D	10011 : D
10100 : D	10101 : P	10110 : D	10111 : P
11000 : P	11001 : P	11010 : D	11011 : D
11100 : P	11101 : D	11110 : D	11111 : D

表 11: $\mathcal{R}_1$

フィルタ $\mathcal{R}_1$
$r_1^P = 0*101$
$r_2^D = 01***$
$r_3^D = 10**0$
$r_4^D = 00**1$
$r_5^D = 111*1$
$r_6^P = 11*0*$
$r_7^P = 0*1**$
$r_8^P = 101**$
$r_9^D = *****$

表 12: $\mathcal{R}_2$

フィルタ $\mathcal{R}_2$
$r_3^D = 10**0$
$r_1^P = 0*101$
$r_4^D = 00**1$
$r_2^D = 01***$
$r_5^D = 111*1$
$r_6^P = 11*0*$
$r_8^P = 101**$
$r_7^P = 0*1**$
$r_9^D = *****$

ポリシーを満たすルールの並びは複数存在.

⇒ ポリシーを維持する範囲でルールを自由に並び替えることができる.

表 13: ルールリスト

フィルタ $\mathcal{R}$	$ r_i _F$
$r_1^P = 0*101$	5
$r_2^D = 01***$	10
$r_3^D = 10**0$	6
$r_4^D = 00**1$	20
$r_5^D = 111*1$	50
$r_6^P = 11*0*$	40
$r_7^P = 0*1**$	100
$r_8^P = 101**$	120
$r_9^D = *****$	130
$L(\mathcal{R}, F) = 3293$	

表 14: 並び替えたルールリスト

フィルタ $\mathcal{R}$	$ r_i _F$
$r_3^D = 10**0$	6
$r_8^P = 101**$	120
$r_1^P = 0*101$	5
$r_2^D = 01***$	10
$r_4^D = 00**1$	20
$r_5^D = 111*1$	50
$r_6^P = 11*0*$	40
$r_7^P = 0*1**$	100
$r_9^D = *****$	130
$L(\mathcal{R}, F) = 2821$	

同じポリシーを表現し、 $\mathcal{R}_1$ よりも $\mathcal{R}_2$ の方が遅延 $L(\mathcal{R}, F)$ が小さい

⇒ **ルールを並び替えると遅延が減少する可能性**

遅延が減少するルール順序を求める方法を考察

ルール順序最適化問題

ルール順序最適化問題

入力 ルールリスト $\mathcal{R}$, 頻度分布 F

出力 $\sum_{i=1}^{n-1} \sigma(i) |r_i|_F + (n-1) |r_n|_F$ を最小化し, $\mathcal{R}$ のポリシーを維持するルールの順序

表 15: 入力のルールリスト

フィルタ $\mathcal{R}$	$ r_i _F$
$r_1^P = 0*101$	5
$r_2^D = 01***$	10
$r_3^D = 10**0$	6
$r_4^D = 00**1$	20
$r_5^D = 111*1$	50
$r_6^P = 11*0*$	40
$r_7^P = 0*1**$	100
$r_8^P = 101**$	120
$r_9^D = *****$	130
$L(\mathcal{R}, F) = 3293$	

表 16: σ で並び替え

フィルタ $\mathcal{R}_\sigma$	$ r_i _F$
$r_5^D = 11101$	50
$r_6^P = 11*0*$	40
$r_3^D = 10**0$	6
$r_8^P = 10***$	120
$r_1^P = 0*101$	5
$r_4^D = 00**1$	20
$r_2^D = 01***$	10
$r_7^P = 0*1**$	100
$r_9^D = *****$	130
$L(\mathcal{R}, F) = 2665$	

重複関係 $\mathcal{O} \subseteq \mathcal{R} \times \mathcal{R}$ と従属関係 $\mathcal{D} \subseteq \mathcal{R} \times \mathcal{R}$

赤 重複
黒 従属

表 17: 入力されたルールリスト

フィルタ $\mathcal{R}$	$ r_i _F$
$r_1^P = 0*101$	5
$r_2^D = 01***$	10
$r_3^D = 10**0$	6
$r_4^D = 00**1$	20
$r_5^D = 111*1$	50
$r_6^P = 11*0*$	40
$r_7^P = 0*1**$	100
$r_8^P = 101**$	120
$r_9^D = *****$	130
$L(\mathcal{R}, F) = 3293$	

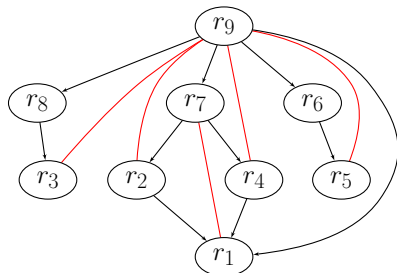


図 3: 従属関係と重複関係のグラフ

e.g. r_7^P は r_1^P に重複, r_8^P は r_3^D に従属

従属関係 $\mathcal{D} \subseteq \mathcal{R} \times \mathcal{R}$ とポリシー違反

従属関係を無視して r_8^P をルールリストの一番上に配置すると

表 18: ルールリスト

フィルタ $\mathcal{R}$	$ r_i _F$
$r_1^P = 0*101$	5
$r_2^D = 01***$	10
$r_3^D = 10**0$	6
$r_4^D = 00**1$	20
$r_5^D = 111*1$	50
$r_6^P = 11*0*$	40
$r_7^P = 0*1**$	100
$r_8^P = 101**$	120
$r_9^D = *****$	130
$L(\mathcal{R}, F) = 3293$	

表 19: ポリシー違反を起こすルールの並び

フィルタ $\mathcal{R}$	$ r_i _F$
$r_8^P = 101**$	120
$r_1^P = 0*101$	5
$r_2^D = 01***$	10
$r_3^D = 10**0$	6
$r_4^D = 00**1$	20
$r_5^D = 111*1$	50
$r_6^P = 11*0*$	40
$r_7^P = 0*1**$	100
$r_9^D = *****$	130
$L(\mathcal{R}, F) = 1894$	

パケット

{10100, 10110} の
評価型が Deny から
Permit に変わる
⇒ **ポリシー違反**

ルール順序最適化問題を考えるとき、
従属関係が重要。

① パケット分類

- パケット分類とルールリスト
- 遅延
- ルール順序最適化問題

② ペアリング法

- 基本的な考え方
- ペアリング法の問題点
- 従属関係を考慮した r_{min} の選択方法
- 提案手法: 従属関係を考慮した r_{max} の選択方法

③ 計算機実験

④ まとめと今後の課題

遅延をより少なくする基本的な考え方

ルールリストの上位にあるほど、
パケットが受ける照合回数が少ない



評価パケット数の大きいルールが
上位にあると、多くのパケットが
受ける照合回数が減少

e.g.

100 個のパケットが r_{100} に合致する
とき、遅延は $100 \times 100 = 10000$ に
なるが、 r_1 に合致するならば遅延は
 $100 \times 1 = 100$.

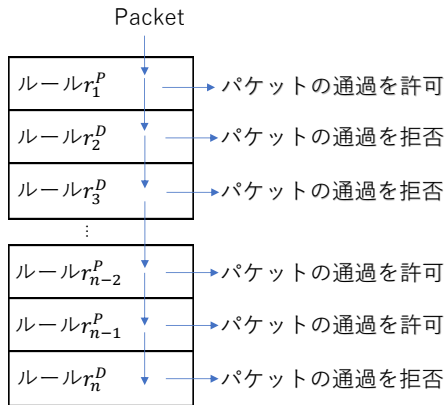


図 4: ルールリストのモデル

ルールを評価パケット数降順に並び替えると遅延がより減少



従属関係を見捨てて並び替えるとポリシー違反が発生



ペアリングし、単一化することで降順に並びかえる

ペアリング法の基本的な考え方

表 20: $|r_i|_F$ の降順に並び
替えられない例

フィルタ $\mathcal{R}$	$ r_i _F$
$r_2^D = 01***$	10
$r_3^D = 10**0$	6
$r_6^P = 11*0*$	40
$r_8^P = 101**$	120



表 21: r_1 と r_8 を
ペアリングした例

フィルタ $\mathcal{R}$	$ r_i _F$
$r_3^D = 10**0$	63
$r_8^P = 101**$	
$r_6^P = 11*0*$	40
$r_2^D = 01***$	10

r_8 と r_3 をペアリングし、単一化させることで r_8 をより上位に配置

ペアリング法

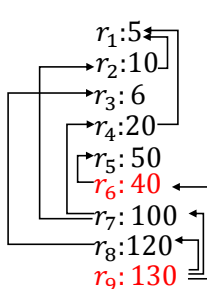


図 5: ルールリストの例

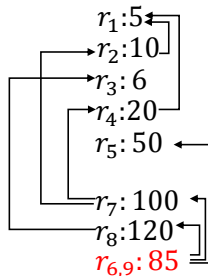


図 6: r_6 と r_9 を
ペアリング

r_9 と r_6 でペアリングを行う. ペアリング後のルールは二つの
従属関係を引き継ぎ, 重みは $\frac{|r_9|+|r_6|}{2} = 85$ となる.

問題点



ペアリング法では、ペアリング対象 (r_{min}, r_{max}) を決定するとき、以下のような要素が考慮されておらず、十分に遅延が減少しない

1. r_{min} を決めるとき、ペアリング候補がどのようなルールに従属しているのか
2. r_{max} を決めるとき、各々のルールがどのようなルールに従属しているのか

単一のルールのみで r_{min} を決定してはいけない

表 22: ルールリストの例

フィルタ $\mathcal{R}$	$ r_i _F$
$r_1^P = 0*101$	5
$r_2^D = 01***$	10
$r_3^D = 10**0$	6
$r_4^D = 00**1$	20
$r_5^D = 111*1$	50
$r_6^P = 11*0*$	40
$r_7^P = 0*1**$	100
$r_8^P = 101**$	120
$r_9^D = *****$	130
$L(\mathcal{R}, F) = 3293$	

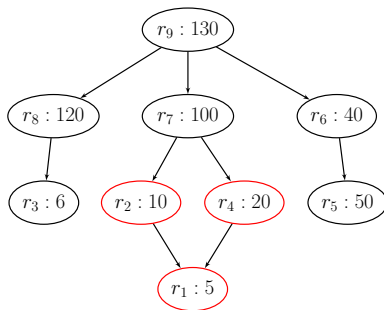
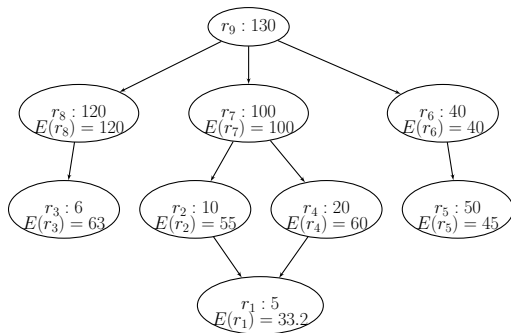


図 7: 従属関係 (r_i は重み)

r_7 の重みは大きいですが, r_1, r_2, r_4 の重みが小さい
⇒ より遅延を減らすことができる

従属関係を考慮した r_{min} の選択方法



$$E(r_1) = \frac{|r_7| + |r_2| + |r_4| + |r_1|}{4} = 33.2$$

$$E(r_2) = \frac{|r_7| + |r_2|}{2} = 55$$

$$E(r_3) = \frac{|r_8| + |r_3|}{2} = 63$$

.

.

.

$$E(r_8) = \frac{|r_8|}{1} = 120$$

図 8: r_{max} から到達可能な
ルールに評価値を設定

評価値最小のルール r_1 から従属関係を遡り, r_9 とペアリング可能な
ルール r_7 を r_{min} とする

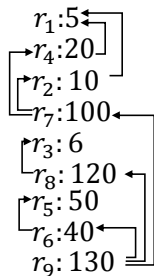


図 9: ペアリング法

$$L(\mathcal{R}, F) = 2825$$

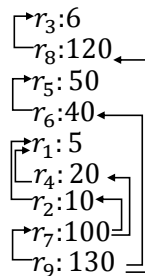


図 10: 提案手法

$$L(\mathcal{R}, F) = 2471$$

提案手法



$$E(r_i) = \frac{r_i \text{ から到達可能なルールの重みの和}}{r_i \text{ から到達可能なルールの数}}$$

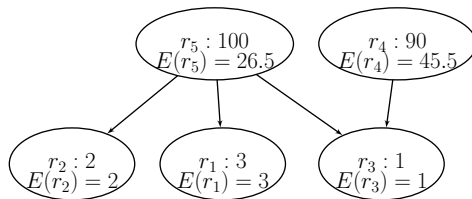


図 11: 各ルールの評価値を記したグラフ

提案手法では r_4 の方が r_5 よりも評価値が大きいため, r_4 と r_3 のペアリングが先に行われ, r_4 を r_5 の上位に配置することができる.

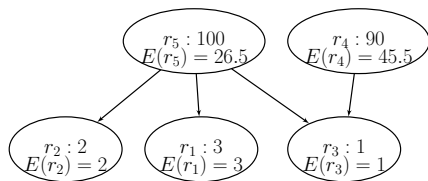


図 12: 評価値を追記

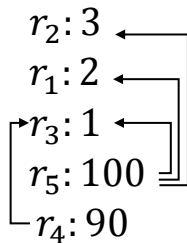


図 13: ペアリング法

$$L(\mathcal{R}, F) = 770$$

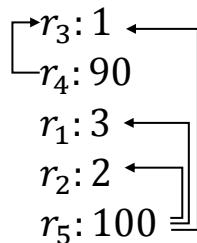


図 14: 提案手法

$$L(\mathcal{R}, F) = 588$$

1 パケット分類

- パケット分類とルールリスト
- 遅延
- ルール順序最適化問題

2 ペアリング法

- 基本的な考え方
- ペアリング法の問題点
- 従属関係を考慮した r_{min} の選択方法
- 提案手法: 従属関係を考慮した r_{max} の選択方法

3 計算機実験

4 まとめと今後の課題

実験環境



OS :CentOS Linux 7

CPU :Intel Core i5-3470 3.20GHz × 4

主記憶容量 :2GB

実装言語 :Java

- ルール数長 104, ルール数が 1000~5000 のルールリストを ClassBench[9] で生成
- Mohan[8], SGM[1], 田中 [5], ペアリング法 [10], 日景 [6], 提案手法を実装
- それぞれの遅延 $L(\mathcal{R}, F)$ と並び替え時間を計測

実験結果 (遅延)

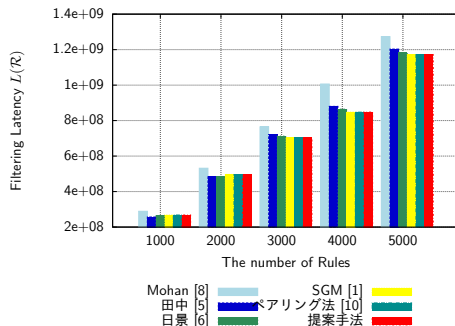


表 23: ペアリング法と提案手法が並び替えたの遅延の表

ルール数	ペアリング法	提案手法
1000	$2.67993e + 08$	$2.67994e + 08$
2000	$4.96356e + 08$	$4.96395e + 08$
3000	$7.04155e + 08$	$7.04167e + 08$
4000	$8.47685e + 08$	$8.47983e + 08$
5000	$1.17175e + 09$	$1.17173e + 09$

図 15: 遅延 (回)

左図で示すように本手法は **SGM** と同程度に遅延を減らしている。

実験結果 (並び替え時間)

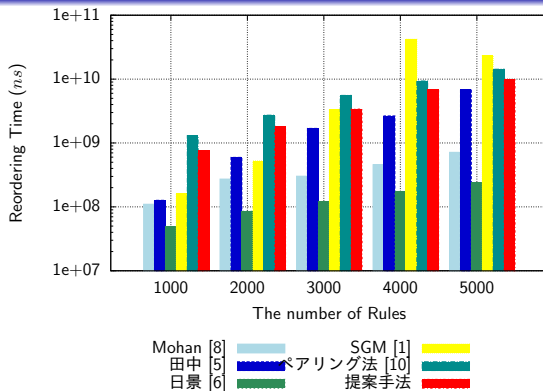


図 16: 並び替え時間 (ns)

ルール数が 4000 と 5000 のとき，提案手法は
SGM よりも少ない計算時間で並び替えを行っている。

1 パケット分類

- パケット分類とルールリスト
- 遅延
- ルール順序最適化問題

2 ペアリング法

- 基本的な考え方
- ペアリング法の問題点
- 従属関係を考慮した r_{min} の選択方法
- 提案手法: 従属関係を考慮した r_{max} の選択方法

3 計算機実験

4 まとめと今後の課題

まとめと今後の課題



まとめ

- ルール順序最適化問題に対する重み平均を用いたペアリング法を提案

今後の課題

- 先行関係がより複雑になったとき，十分に遅延を減らせるような手法の考案
- 重み変動を考慮した並び替え法の考案

参考文献 I



-  R. Mohan, A. Yazidi, B. Feng, and B.J. Oommen, “Dynamic ordering of firewall rules using a novel swapping window-based paradigm,” Proceedings of the 6th International Conference on Communication and Network Security, pp.11–20, New York, NY, USA, 2016, ACM.
-  K. Tanaka, K. Mikawa, and K. Takeyama, “Optimization of packet filter with maintenance of rule dependencies,” IEICE Communications Express, vol.2, no.2, pp.80-85, Feb 2013.
-  日景喬一, 山田敏規, “D-1-6 ルール間の依存関係を保持したファイアウォールの負荷最小化のためのアルゴリズム (d-1. コンピューテーション, 一般セッション),” 電子情報通信学会総合大会講演論文集, vol.2016, no.1, p.6, mar 2016.

参考文献 II



-  A. Tapdiya, and E. Fulp, “Towards optimal firewall rule ordering utilizing directed acyclical graphs,” Computer Communications and Networks, 2009. ICCCN 2009. Proceedings of 18th International Conference on, pp.1-6, Aug 2009.
-  T. Harada, K. Tanaka, R. Ogasawara, and K. Mikawa, “Fast rule reordering method via pairing dependent rules,” submitted.

重みの変動について

Filter $\mathcal{R}_{id}$	$ E(\mathcal{R}_{id}, i) _F$	Filter $\mathcal{R}_\sigma$	$ E(\mathcal{R}_\sigma, i) _F$
$r_1^P = * 0 * 1$	4	$r_3^P = 0 * 0 0$	2
$r_2^P = 0 0 0 0$	1	$r_1^P = * 0 * 1$	4
$r_3^P = 0 * 0 0$	1	$r_2^P = 0 0 0 0$	0
$r_4^D = 0 * 1 *$	3	$r_4^D = 0 * 1 *$	3
$r_5^P = * 1 * 1$	3	$r_5^P = * 1 * 1$	3
$r_6^P = * * * 1$	0	$r_6^P = * * * 1$	0
$r_7^D = * * * *$	4	$r_7^D = * * * *$	4

表のように並び替えたとき, r_2^P で評価されていた 0000 が, r_3^P に評価されるようになるので重み $E(\mathcal{R}, i)$ が変動

⇒ ルールリストが与えられたときの評価パケット数ではルール順序を最適化させるのに不十分

複数のルールに従属しているルールの従属関係が複雑な場合

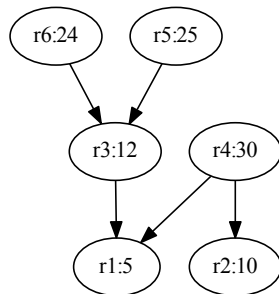
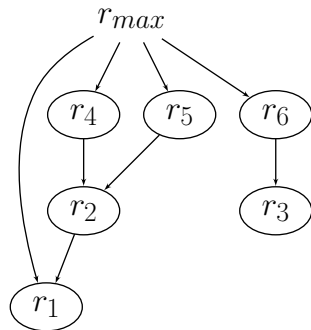


図 17: 提案手法が遅延を十分に減少できないルールリストの従属関係

ペアリングできないルール



r_{max} と r_1 は従属関係があるが
 r_2 からも到達可能である

r_{max} と r_1 がペアリングを行い、
新たに r_{new} が生成されたとする。
 $r_2 \sim r_4$ は r_{new} に対してどの
ような配置をしてもポリシー違反となる

⇒ 他のルールを経由して到達可能なルールはペアリング可能なルールではない

$$|r_1| < |r_2| < \dots < |r_6| < |r_{max}|$$