包含関係に限定したルールリスト再構築法

原田 崇司 [1]　田中 賢 [1]　三河 賢治 [2]

[1] 神奈川大学大学院 理学研究科 理学専攻 情報科学領域

[2] 新潟大学学術情報機構情報基盤センター

2018 年 6 月 15 日

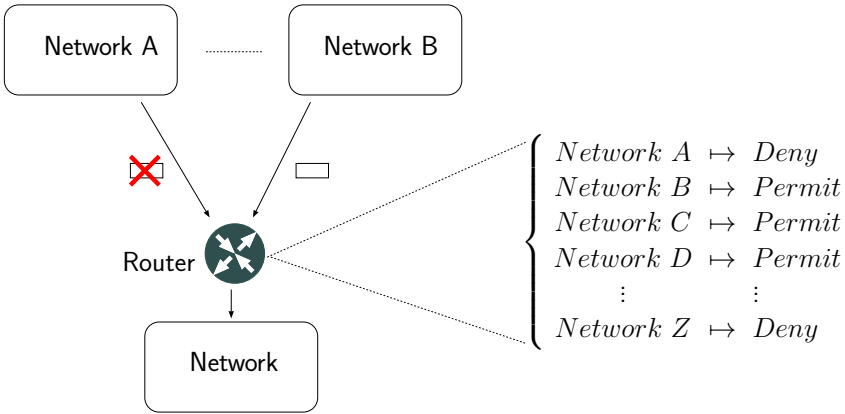
CAS・SIP・MSS・VLD, 北海道大学

研究テーマ

フィルタリングルールリストの再構築による

パケットフィルタリングの高速化

パケットフィルタリング



ポリシーに従ってパケットをフィルタリング

パケットフィルタリングの高速化

“フィルタリングの速度” については後で定義

- TCAM などの特殊なハードの使用による高速化
- トライや決定木などのデータ構造の使用による高速化
- フィルタリングルールリストの並び替えによる高速化
- フィルタリングルールリストの再構築による高速化 ⇐

ルールの形式

一般にパケットフィルタリングには，パケットヘッダの

送信元アドレス	(e.g. 131.10.42.40)
宛先アドレス	(e.g. 95.184.130.35)
送信元ポート番号	(e.g. 2020)
宛先ポート番号	(e.g. 22)
プロトコル	(e.g. TCP)

を使用するので，フィルタリングルールは，これら 5 つの項目を指定

(e.g. r_1 : 131.10.42.40/32, 95.184.130.35/32, 0 : 65535, 1724 : 1724, UDP)

抽象化してパケット（ヘッダ）を $0, 1$ の系列，ルールを $0, 1, *$ の系列とみる

ルールの形式

- ルールは，
  - ルール番号 $i \in \mathbb{N}$
  - 条件 $c \in \{0, 1, *\}^w$
  - アクション（評価型）$e \in \{P, D\}$

  の組．ただし，'$*$' は don't care
- パケットは長さ w のビット列

Filter $\mathcal{R}$	
r_1^P	$0\ *\ 1\ *$
r_2^D	$0\ 0\ 0\ 0$
r_3^D	$*\ 0\ 0\ *$
r_4^P	$*\ 1\ *\ 0$
r_5^P	$1\ *\ 1\ *$
r_6^D	$*\ *\ 1\ *$

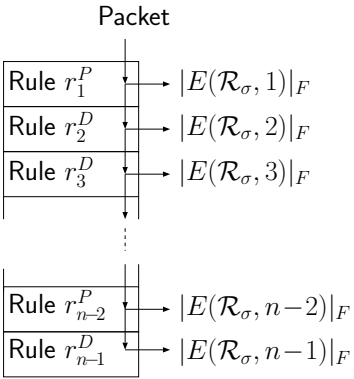
パケットフィルタリング

$$
\begin{aligned}
&0000 \mapsto P, \quad 0001 \mapsto P,\\
&0010 \mapsto D, \quad 0011 \mapsto P,\\
&0100 \mapsto P, \quad 0101 \mapsto P,\\
&0110 \mapsto D, \quad 0111 \mapsto D,\\
&1000 \mapsto D, \quad 1001 \mapsto P,\\
&1010 \mapsto D, \quad 1011 \mapsto P,\\
&1100 \mapsto D, \quad 1101 \mapsto P,\\
&1110 \mapsto D, \quad 1111 \mapsto P
\end{aligned}
\tag{1}
$$

Table 1: 式 1 を満たすルールリスト

Filter $\mathcal{R}$	
r_1^P	$0\ 0\ 0\ *$
r_2^P	$0\ 0\ 1\ 1$
r_3^D	$0\ *\ 1\ *$
r_4^P	$0\ 1\ 0\ *$
r_5^P	$1\ *\ *\ 1$
r_6^D	$*\ *\ *\ *$

パケットフィルタリングの遅延



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を定義

$$L(\mathcal{R}_\sigma, F) = \sum_i^{n-1} i|E(\mathcal{R}_\sigma, \sigma^{-1}(i))|_F + (n-1)|E(R_\sigma, \sigma^{-1}(n))|_F$$

σ はルールの順序，F はパケットの頻度分布

評価パケット数（ルール重み）

評価パケット数 $|E(\mathcal{R}_\sigma, i)|_F$

分布 F において，順序 σ のルールリスト $\mathcal{R}_\sigma$ の $1 \sim (\sigma(i)-1)$ 番目のルールに合致せず，r_i^e に合致するパケットの数

Filter $\mathcal{R}$	$\lvert E(\mathcal{R}_{id}, i)\rvert_F$
$r_1^P = * \; 0 \; * \; 1$	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
$L(\mathcal{R}_{id}, F) = 60$	

e.g. F：一様分布，σ：id

r_4^D に合致するパケットは

$$\{0010, 0011, 0110, 0111\}.$$

しかし，0011 は r_1^P に評価されるので，r_4^D によって評価型が決まるパケットは

$$\{0010, 0110, 0111\}$$

の 3 つ．以上より $|E(\mathcal{R}_{id}, 4)| = 3$

ルールリストとポリシー

Filter $\mathcal{R}$	$\lvert E(\mathcal{R}, i)\rvert_F$
$r_1^D = 1 * * *$	8
$r_2^D = 0\ 1 * *$	4
$r_3^P = 0\ 0\ 0\ 0$	1
$r_4^D = * * * *$	3
$L(\mathcal{R}, F) = 28$	

Filter $\mathcal{R}'$	$\lvert E(\mathcal{R}', i)\rvert_F$
$r_1^P = 0\ 0\ 0\ 0$	1
$r_2^D = * * * *$	15
$L(\mathcal{R}', F) = 16$	

$$0000 \mapsto P, 0001 \mapsto D, 0010 \mapsto D, 0011 \mapsto D, 0100 \mapsto D, 0101 \mapsto D, 0110 \mapsto D, 0111 \mapsto D,$$
$$1000 \mapsto D, 1001 \mapsto D, 1010 \mapsto D, 1011 \mapsto D, 1100 \mapsto D, 1101 \mapsto D, 1110 \mapsto D, 1111 \mapsto D \quad (2)$$

フィルタ $\mathcal{R}$ と $\mathcal{R}'$ は同じポリシー (2) を表現

⟹ ルール再構築によりフィルタリング遅延を減らせる可能性有

ルールリスト最適化問題

ルールリスト最適化問題

入力： ルールリスト $\mathcal{R}$，頻度分布 F
出力： $L(\mathcal{R}', F)$ を最小化する，$\mathcal{R}$ とポリシーが等しい $\mathcal{R}'$

Filter $\mathcal{R}$	$\lvert E(\mathcal{R}, i)\rvert_F$
$r_1^D = 1 * * *$	8
$r_2^D = 0\ 1 * *$	4
$r_3^P = 0\ 0\ 0\ 0$	1
$r_4^D = * * * *$	3
$L(\mathcal{R}, F) = 28$	

Filter $\mathcal{R}'$	$\lvert E(\mathcal{R}', i)\rvert_F$
$r_1^P = 0\ 0\ 0\ 0$	1
$r_2^D = * * * *$	15
$L(\mathcal{R}', F) = 16$	

右のルールリストがおそらく最適なルールリスト..

本発表のテーマは，この最適化問題に対する発見的解法

ルールリストとポリシー

Filter $\mathcal{R}_{id}$	$\lvert E(\mathcal{R}_{id}, i)\rvert_F$
$r_1^P = * \; 0 \; * \; 1$	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
$L(\mathcal{R}_{id}, F) = 60$	

Filter $\mathcal{R}_{\sigma}$	$\lvert E(\mathcal{R}_{\sigma}, i)\rvert_F$
$r_1^P = * \; 0 \; * \; 1$	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
$L(\mathcal{R}_{\sigma}, F) = 51$	

$$0000 \mapsto P, 0001 \mapsto P, 0010 \mapsto D, 0011 \mapsto P, 0100 \mapsto P, 0101 \mapsto P, 0110 \mapsto D, 0111 \mapsto D,$$
$$1000 \mapsto D, 1001 \mapsto P, 1010 \mapsto D, 1011 \mapsto P, 1100 \mapsto D, 1101 \mapsto P, 1110 \mapsto D, 1111 \mapsto P \quad (3)$$

フィルタ $\mathcal{R}_{id}$ と $\mathcal{R}_{\sigma=(1\ 5\ 4\ 2\ 3\ 6\ 7)}$ は同じポリシー (3) を表現

⇒ ルールを並び替えるとフィルタリング遅延を減らせる可能性有り

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

Filter $\mathcal{R}_{id}$	$\lvert E(\mathcal{R}_{id}, i)\rvert_F$
$r_1^P = * 0 * 1$	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

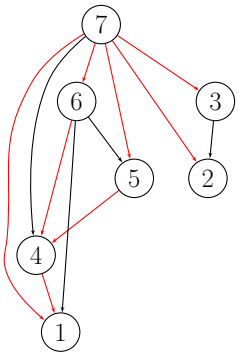
Filter $\mathcal{R}_\sigma$	$\lvert E(\mathcal{R}_\sigma, i)\rvert_F$
$r_1^P = * 0 * 1$	4
$r_4^D = 0 * 1 *$	3
$r_5^P = * 1 * 1$	3
$r_3^P = 0 * 0 0$	2
$r_2^P = 0 0 0 0$	0
$r_6^P = * * * 1$	0
$r_7^D = * * * *$	4

この最適化問題は $\mathcal{NP}$ 困難だが...

ルール間の先行制約：重複関係 $\mathcal{O}$ と従属関係 $\mathcal{D}$

先行制約のグラフが根付木だと多項式時間で解ける

Filter $\mathcal{R}_{id}$	$\lvert E(\mathcal{R}_{id}, i)\rvert_F$
$r_1^P = * \; 0 \; * \; 1$	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



黒　：　重複

赤　：　従属

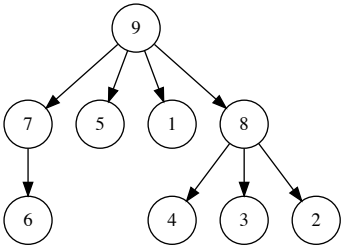
e.g. r_6^P は r_5^P に重複，
r_6^P は r_4^D に従属

ルール順序最適化問題における先行制約

ルール順序最適化問題は $\mathcal{NP}$ 困難

ルール間の先行制約のグラフが根付木

多項式時間で解ける

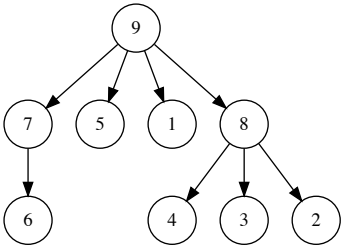


ルール順序最適化問題における先行制約

ルール順序最適化問題は $\mathcal{NP}$ 困難

ルール間の先行制約のグラフが根付木

多項式時間で解ける



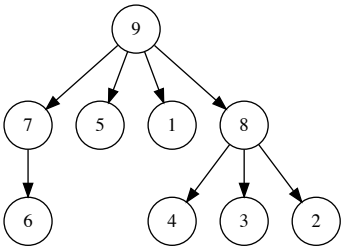
⇨ 先行制約のグラフが根付木となるようにルールリストを書き換え

ルール順序最適化問題における先行制約

ルール順序最適化問題は $\mathcal{NP}$ 困難

ルール間の先行制約のグラフが根付木

多項式時間で解ける



⇨ 先行制約のグラフが根付木となるようにルールリストを書き換え

⇨ 書き換えたルールリストの順序を最適化

包含関係に限定したルールリスト

Table 2: 包含関係に限るルールリスト

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

Filter $\mathcal{R}$
$r_7^P = 0\ *\ *\ 1$
$r_8^P = 1\ 1\ 1\ 1$
$r_9^D = 1\ 1\ 1\ *$
$r_{10}^P = 0\ *\ 0\ 0$
$r_{11}^D = 1\ *\ *\ *$
$r_{12}^D = *\ *\ *\ *$

Table 3: 包含関係に限らぬルールリスト

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

r_i と r_j が重複するなら，$M(r_i) \subset M(r_j)$ に限るとき，
そのルールリストを「包含に限るルールリスト」とよぶ

表 3 の r_1 と r_4 は重複するが，$M(r_1) = \{1000, 1100\} \not\subset M(r_4) = \{1100, 1110\}$

書換アルゴリズム

包含関係に限るルールリストの先行制約のグラフは根付木

以下のルールリスト書換アルゴリズムを考案

1. ルールリストを包含関係に限るルールリストへと書き換え
2. 評価型（従属関係）を考慮しても根付木となるよう修正

ルールリスト書換

Table 4: 包含関係に限らぬルールリスト

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

Table 5: 包含関係に限るルールリスト

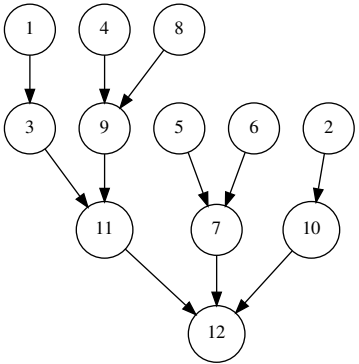
Filter $\mathcal{R}$
$r_{1_1}^D = 1\ 0\ 0\ 0$
$r_{1_2}^D = 1\ 1\ 0\ 0$
$r_2^P = 0\ 1\ 0\ 0$
$r_3^P = 0\ *\ 0\ *$
$r_4^P = 1\ 1\ *\ 0$
$r_5^P = 0\ *\ 1\ 1$
$r_6^D = 0\ 1\ 0\ 1$

r_1 から走査し,「包含関係に限る」性質を破るルールの $*$ を $0, 1$ へと展開

評価型を考慮

Table 6: ルールリスト

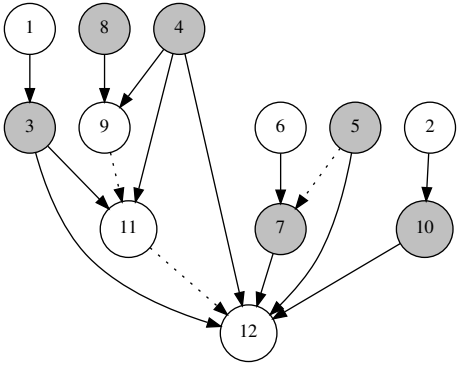
Filter $\mathcal{R}$	Filter $\mathcal{R}$
$r_1^D = 1\ 1\ 0\ 0$	$r_7^P = 0\ *\ *\ 1$
$r_2^P = 0\ 1\ 0\ 0$	$r_8^P = 1\ 1\ 1\ 1$
$r_3^P = 1\ *\ 0\ *$	$r_9^D = 1\ 1\ 1\ *$
$r_4^P = 1\ 1\ 1\ 0$	$r_{10}^P = 0\ *\ 0\ 0$
$r_5^P = 0\ *\ 1\ 1$	$r_{11}^D = 1\ *\ *\ *$
$r_6^D = 0\ 1\ 0\ 1$	$r_{12}^D = *\ *\ *\ *$



評価型を考慮

Table 6: ルールリスト

Filter $\mathcal{R}$	Filter $\mathcal{R}$
$r_1^D = 1\ 1\ 0\ 0$	$r_7^P = 0 * * 1$
$r_2^P = 0\ 1\ 0\ 0$	$r_8^P = 1\ 1\ 1\ 1$
$r_3^P = 1 * 0 *$	$r_9^D = 1\ 1\ 1 *$
$r_4^P = 1\ 1\ 1\ 0$	$r_{10}^P = 0 * 0\ 0$
$r_5^P = 0 * 1\ 1$	$r_{11}^D = 1 * * *$
$r_6^D = 0\ 1\ 0\ 1$	$r_{12}^D = * * * *$



評価型を考慮すると（従属関係のグラフだと）根付木とならないことが

節点（ルール）の挿入と縮約

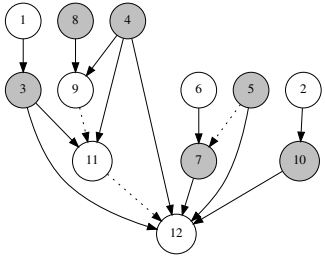


Figure 1: オリジナル

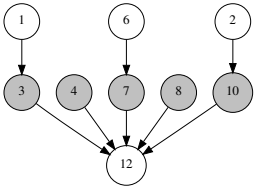


Figure 2: 縮約

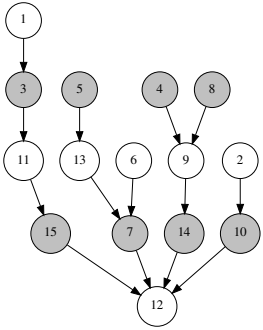
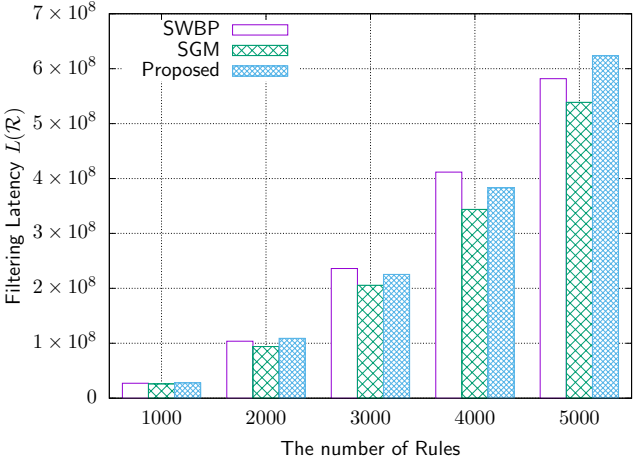


Figure 3: 挿入

節点（ルール）の挿入と縮約により，先行制約のグラフを根付木へ

実験結果



既存手法で最良の SGM より悪いが，最新の SWBP より 4000 では良い

まとめと今後の課題

まとめ

包含関係に着目した，ルールリスト最適化問題に対する発見的解法を提案

今後の課題

- 包含関係に拘らずに，先行関係のグラフが根付木となるようなルールリスト書換法の提案
- 根付木にも拘らずに，ルール順序を最適化できるように，ルールリストを書換える手法の提案