L-003

ポリシに影響しない従属関係削除に基づくルール並び替え法
A Rule Reordering Method via Deleting Dependencies Unaffecting the Policy

渕野 敬[1)]　原田 崇司[2)]　田中 賢[1)]　三河 賢治[3)]
Takashi Fuchino　Takashi Harada　Ken Tanaka　Kenji Mikawa

1　はじめに

パケット分類とは，ネットワーク機器に到着するパケットの振る舞いをポリシーに対応するルールリストによって決定する機能である．線形探索などによるパケット分類では，パケットとルールとの比較回数が増加するとパケット分類による通信の遅延が生じる．ルールリストのポリシを保持しながら，この遅延を最小化するルールの並びを求める問題，ルール順序最適化問題が研究されている．この問題に対する多くの手法 [1, 2, 3, 4, 5, 6] がポリシを保持するために従属関係に基づく先行制約に従ってルールを並び替えているが，その先行制約を満たしていなくてもポリシを保持するルールの並びが存在する．本稿では，上位に配置されたルールによってポリシに影響しなくなる先行制約を探索し，それらの制約を削除する手法を提案する．また，パケット分類アルゴリズムのベンチマーク ClassBench[7] を用いた計算機実験により提案手法の有効性を示す．

2　パケット分類

パケット分類は図 1 のようにモデル化される．ネットワーク機器に到着したパケットはルールリストの上位のルールから順に照合され，最初に合致したルールのアクションが適用される．

パケットを $\{0,1\}^w$ 上の長さ w のビット列とする．ルールリストの各ルールは，ルール番号 $i \in [n]$, 合致するパケットの条件 $b_1 b_2 \ldots b_w \in \{0,1,*\}^w$ とパケットに適用されるアクション $a^i \in \{A_1, A_2, A_3, \ldots, A_m\}$ の三つからなる．ただし，n はルールの数とし，$[n] = \{1, 2, \ldots, n\}$, $A_1, A_2, A_3, \ldots, A_m$ を m 個の異なるアクションとする．$*$ はパケットのビット 0, 1 のいずれにも合致することを意味する．以下では，表記の単純化のためにアクションやアクションに付随する添え字を省略することがある．

ルールとルールリストの例を表 1 に示す．表 1 の例では，ルールリストが到着するパケットに与えるアクションは Permit と Deny の二種類でそれぞれパケットの通過の許可と拒否を意味する．パケット分類では，到着するすべてのパケットに対して，適用するアクションを決めなければならないので，ルールリストの最後にすべてのパケットに合致するデフォルトルール r_n^D を配置する．表 1 では，r_9^D がデフォルトルールである．

ルールリスト $\mathcal{R}$ は，パケットの集合 $\mathcal{P}$ からアクションの集合 $\{A_1, A_2, \ldots, A_m\}$ への函数と見なすこともでき，この函数をルールリストのポリシーという．ルールリスト $\mathcal{R}$ が，パケット p に与えるアクションを $\mathcal{R}(p)$ と表す．ルールを並び替える前後で $R(p)$ が異なるようなパケット p が存在するとき，そのような順序をポリシを保持しない，またはポリシに違反する順序という．

ルールリスト $\mathcal{R}$ と順序が与えられると，ルール $r_i \in \mathcal{R}$ によってアクションが決まるパケットの集合が定まる．この集合を $E(\mathcal{R}, r_i^{a^i})$ と記す．表記 $E(\mathcal{R}_\sigma, r_i^{a^i})$ において，$r_i^{a^i}$ のルール番号 i 以外の部分は冗長なので，以降 $E(\mathcal{R}, r_i^{a^i})$ を $E(\mathcal{R}, i)$ と表す．

表 2　到着パケットの頻度分布 $\mathcal{F} : \mathcal{P} \to \mathbb{N}$

$0000 \mapsto 30$	$0001 \mapsto 12$	$0010 \mapsto 36$	$0011 \mapsto 43$
$0100 \mapsto 400$	$0101 \mapsto 15$	$0110 \mapsto 25$	$0111 \mapsto 30$
$1000 \mapsto 10$	$1001 \mapsto 15$	$1010 \mapsto 50$	$1011 \mapsto 20$
$1100 \mapsto 10$	$1101 \mapsto 10$	$1110 \mapsto 70$	$1111 \mapsto 17$

到着パケットの頻度分布 $\mathcal{F}$，ルールリスト $\mathcal{R}$ と順序が与えられたとき，ルール r_i^a によってアクションが定まるパケットの数を r_i^a の評価パケット数または重みと呼び，$|E(\mathcal{R}, i)|_\mathcal{F}$ と記す．

パケットとルールとの照合を遅延 1 と考え，到着パケットの頻度分布 $\mathcal{F}$ でのルールリスト $\mathcal{R}$ の遅延 $L(\mathcal{R}, \mathcal{F})$ を (1) のように定義する．

定義 2.1　(分類遅延)

$$L(\mathcal{R}, \mathcal{F}) = \sum_{i=1}^{n-1} i|E(\mathcal{R}, i)|_\mathcal{F} + (n-1)|E(\mathcal{R}, n)|_\mathcal{F} \qquad (1)$$

この遅延が最小となるルールの並びを求めるために，次の最適化問題を定義する．

定義 2.2　(ルール順序最適化問題)
入力：　ルールリスト $\mathcal{R}$, 頻度分布 $\mathcal{F}$
出力：　ポリシを保ち $L(\mathcal{R}, \mathcal{F})$ を最小化する順序

ルール並び替えにおいて，ルールリスト内のどのルールを入れ替えるとポリシー違反が起こるかが重要なので，ルール上の重複関係と従属関係を定義する．

定義 2.3　(ルールの重複) $r_i^{a^i}$ と $r_j^{a^j}$ の両方に合致するパケットが存在するとき，$r_i^{a^i}$ と $r_j^{a^j}$ は重複するという．

定義 2.4　(ルールの従属) $r_i^{a^i}$ と $r_j^{a^j}$ が重複しており，$i < j$ 且つ $a^i \neq a^j$ のとき，$r_j^{a^j}$ は $r_i^{a^i}$ に従属するという．

重複または従属関係を維持する順序はポリシー違反を起こさないので，ルール順序最適化問題に対する発見的解法の多くは，重複または従属関係を保ちながらルールを並び替えている．ただし，重複関係・従属関係を破っていてもポリシーを保つ順序が存在する可能性があることに注意されたい．

3　ポリシに影響しない従属関係の探索

既存のルール並び替え法は，ポリシを保持するため，従属関係による先行制約に従ってルールを並び替えてい

1)　神奈川大学大学院 理学研究科
2)　高知工科大学 情報学群
3)　新潟大学 情報基盤センター

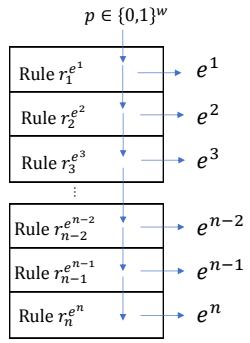

図 1　パケット分類モデル

表 1　ルールリスト

フィルタ $\mathcal{R}$	$\lvert r_i \rvert_F$
$r_1^P = 111*$	87
$r_2^P = 10*0$	60
$r_3^D = 1**0$	5
$r_4^D = 110*$	55
$r_5^D = 011*$	55
$r_6^P = *1*0$	400
$r_7^P = *0*0$	76
$r_8^P = 00*1$	65
$r_9^D = ****$	50

る．しかし，従属関係にあるルールの共通部分に含まれるパケットが上位に配置されているルールにすべて合致するとき，従属関係にあるルールを入れ替えてもポリシが保持される．このように従属関係を満たさなくてもポリシを保持するようなルールの並びは存在し，また，その中でより遅延の小さいルールの並びが存在する場合がある．しかし，既存のルール並び替え法ではそのようなルールの並びを求めることができない．そこで，ポリシに影響しない従属関係を探索し，削除することでより遅延の小さいルールの並びを求める SAT ソルバを用いた手法と発見的解法を提案する．

4　SAT ソルバを用いた探索法

着目した従属関係がポリシに影響しているかどうかの判定はその従属関係にあるルールの共通部分に含まれているパケットのうち，上位に配置されているルールに合致しないパケットが存在するかどうかの充足可能性判定に変換することができる．この手法は複数のルールが部分的に共通部分に合致するような複雑な従属関係でも判定が可能である．しかし，SAT ソルバは入力される論理式のリテラルに対し，計算量が指数となっているため，ルール数が増加すると現実的な時間で動作が終了しなくなる．

5　共通部分を包含するルールの探索法

この手法は，それぞれの従属関係に対し，共通部分に含まれるパケットにすべて合致するルールを探索する．そして，探索したルールが従属関係にあるルールよりも上位に配置されていれば，その従属関係を削除する．この手法はルール数を n，ルールの bit 数を w とすると，計算量は $O(w^2n^3)$ となり，多項式時間で動作が終了する．しかし，SAT ソルバを用いた手法に比べ，一般に精度が低下する．

6　計算機実験

提案手法の有効性を確かめるために，Java 言語を用いて計算機実験を行った．計算機実験のために ClassBench [7] を用いて生成したルール数 100 ~ 500 のルールリストをそれぞれ 10 個ずつ合計 50 個生成した．また，それぞれのルールリストに対して ClassBench を用いてヘッダ数 10 万のヘッダリストを生成した．これらのルールリストに対して，SGM を用いてルールを並び替える．このとき，ポリシに影響しない従属関係を削除することで遅延が減少するか計測した．各々のルール数における平均値を表 3 に示す．また，それぞれの手法の並び化時間を計測し，その平均値を表 4 に示す．表 3 より，ポリシに影響しない従属関係を削除してルールを並び替えた方が遅延が減少している．また，SAT ソルバを用いた並び替え法は共通部分を包含するルールの探索法よりも遅延を減少させている．しかし，表 4 より，共通部分を包含するルールの探索法は SAT ソルバを用いた並び替え法よりも高速にルールを並び替えている．

7　まとめ

本稿では，ポリシに影響しない従属関係削除に基づくルール並び替え法を提案した．提案した二つの手法は共により遅延の小さいルールの並びを求めることができた．本稿で述べた発見的解法は複数のルールが部分的に共通部分に従属しているような従属関係に対して，ポリシに影響するかどうかを正しく判定できていない．このような従属関係において，ポリシに影響するかどうかの判定を行えるような手法を考案することが今後の課題である．また，共通部分に含まれるパケットが他のルールにどのように合致するかは，生成されたルールリストの傾向に依存しているため，どのような従属関係を含むルールが生成されるか調査することも課題である．

参考文献

[1] A. Tapdiya, and E. Fulp, “Towards optimal firewall rule ordering utilizing directed acyclical graphs,” Computer Communications and Networks, 2009. ICCCN 2009. Proceedings of 18th Internatonal Conference on, pp.1-6, Aug 2009.

[2] 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.

[3] 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.

[4] 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.

[5] R. Mohan, A. Yazidi, B. Feng, and J. Oommen, “On optimizing firewall performance in dynamic networks by invoking a novel swapping window-based paradigm,” International Journal of Communication Systems, vol.31, no.15, p.e3773, 2018, e3773 dac.3773.

[6] 渕野敬，原田崇司，田中賢，三河賢治，“従属部分グラフ列挙によるルール並べ替え法,” 電子情報通信学会論文誌 D，vol.103，no.4，pp.228–237，2020.

[7] D.E. Taylor, and J.S. Turner, “Classbench: A packet classification benchmark,” IEEE/ACM Trans. Netw., vol.15, no.3, pp.499–511, June 2007.

表 3　それぞれの手法で並び替えたルールリストの遅延

rule	SGM	useSAT	heuristic
100	$3.72427e+06$	$3.71522e+06$	$3.71547e+06$
200	$6.26120e+06$	$6.19822e+06$	$6.22564e+06$
300	$8.99778e+06$	$8.88456e+06$	$8.91101e+06$
400	$1.07783e+07$	$1.06241e+07$	$1.07349e+07$
500	$1.24831e+07$	$1.23364e+07$	$1.24339e+07$

表 4　それぞれの手法のルール並び替え時間

rule	SGM	useSAT	heuristic
100	$3.49486e+06$	$2.0934e+10$	$1.86858e+07$
200	$6.85717e+06$	$1.31292e+11$	$4.77025e+07$
300	$1.73752e+07$	$4.30588e+11$	$8.37047e+07$
400	$2.56541e+07$	$8.30809e+11$	$1.21384e+08$
500	$3.56020e+07$	$1.30691e+12$	$1.55902e+08$