L-028

パケットフィルタリング最適化問題における多項式時間アルゴリズム

Polynomial time algorithm for packet-filtering optimization problem

小出淳一* 田中賢† 三河賢治‡
Junichi Koide Ken Tanaka Kenji Mikawa

1 はじめに

不正アクセスやスパムメール等の対策として，パケットフィルタリングは必要であるが，各パケットを許可/拒否するルールの並び順は管理者の経験に頼っており，これまで理論的に検討されることは少なかった．

田中ら [1] は，パケットフィルタリングを構成する各ルールを論理式の集合としてモデル化し，フィルタリング負荷を軽減するルールの再構成法を示したが，それらの計算量については触れていない．

本稿では，ルール間の関係がルールを再構成するアルゴリズムの計算量にどのような影響を与えるかを明らかにする．ルールの集合を作り，集合を並び替える多項式時間の負荷軽減方法を提案する．

2 フィルタリングモデル

パケットを許可/拒否するルールを長さ m のビット列

$$R_i = b_1 b_2 ... b_m, b_i \in \{0, 1, -\} \quad (1)$$

のように表す．式中‘ $-$ ’をドントケアと呼び，0, 1 両方を表す記号である．パケットの許可/拒否を明示する場合は，特に R_i^A（許可），R_i^D（拒否）で表す．また，ルールを順序づけて並べたものをフィルタリングテーブル $\boldsymbol{R}$ と呼び

$$\boldsymbol{R} = \langle R_1, R_2, ..., R_n \rangle \quad (2)$$

のように表す．例えば，$R_3^A = 1-0-$ は，$\boldsymbol{R}$の 3 番目のルールで，$1-0-$ に一致するパケットを許可する（図 1 参照）．$\boldsymbol{R}$を通過するパケット流について，R_i で評価するパケット数を評価パケット数と呼び，$\|R_i\|$ と表す．パケットの到着頻度分布が一様の場合，フィルタリング負荷は

$$L(\boldsymbol{R}) = \sum_{i=1}^{n} i \|R_i\| \quad (3)$$

*新潟大学大学院自然科学研究科
†神奈川大学理学部
‡新潟大学総合情報処理センター

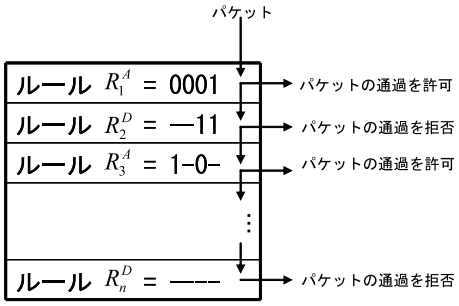

図 1: パケットフィルタリングのモデル

で求める．

3 多項式時間の再構成法

3.1 ルールの関係

$\boldsymbol{R}$を通過するパケット流について，評価パケット数の多いルールを上位に移動すると $L(\boldsymbol{R})$ が減少する．ところがルール間の関係によって，ルールを移動できないものが存在する．ルールを入れ替えて評価パケット数が変わるような入れ替えはできない．例えば，すべてのパケットを許可するルールを最上位に移動してしまうと，以後のルールは意味をなさない．

ルール間の関係は以下の 4 つに分類できる．二つのルール R_i，R_j について，次の二項関係を定義する．

[**独立**] 任意のパケットについて，互いに同じパケットを評価しないならば，R_i, R_j は独立である．

[**重複**] 任意のパケットについて一つでも同じパケットを評価するならば，R_i, R_j は重複である．すなわち，独立ではない R_i, R_j は重複である．

[**包含**] 重複のうち，R_i で評価可能なすべてのパケットが R_j で評価可能ならば，R_j は R_i を包含する．

[**一部重複**] 重複のうち，R_i でのみ評価可能なパケットと，R_j でのみ評価可能なパケットが存在するならば，R_i, R_j は一部重複である．

一部重複を含まないフィルタリングテーブルにおいて，各ルールの評価パケット数を多項式時間で求める

方法は，著者らによって確立された．しかしながら，一部重複を含む場合の評価パケット数を求める方法は知られていない．

3.2 独立ルール集合

$x \in S_i, y \in S_j$ について，x と y が独立となるように $\boldsymbol{R}$の各ルールを $S_1, ..., S_r$ に分割する．この集合を独立ルール集合と呼ぶ．

独立ルール集合を用いれば，各集合に属するルールの評価パケット数の平均値が大きいものを上位に移動することによって $L(\boldsymbol{R})$ が減少する．ところが，独立ルール集合に属する要素同士は独立である必要はないので，当然一部重複の関係も存在しうる．一部重複を含む場合の評価パケット数を多項式時間で求める方法は知られていないので，予想平均評価パケット数を導入する．集合に属するルールの中で最もドントケアが多いルールのドントケア数が d のとき，$\frac{2^d}{|S_i|}$ を予想値として扱い，予想平均評価パケット数と呼ぶ．

3.3 独立ルール集合の作成

独立ルール集合の構造を表す木 T を定義する．T は $\boldsymbol{R}$ の各ルールを葉として持ち，左にあるルールほど上位のルールである．T の構成方法は次の通りである．

1. T の根 $\leftarrow R_1$;
2. while R_i (i=2,...,n) do
3. while (T を帰りがけ順にたどる) do
   { 現在の節 a について }
4. if a は葉である then
5. if R_i と a が重複である then
6. $c(a) \leftarrow 1; c(p(a)) \leftarrow c(p(a)) + 1;$
7. else
8. if $c(a) > 0$ then
9. $c(p(a)) \leftarrow c(p(a)) + 1;$
10. end;
11. Insert(R_i)
12. end;

Insert(R){ 現在の節 a について }

1. if $c(a) = 0$ then
2. if a が独立節 then
3. 末節として R を挿入
4. else
5. a と $p(a)$ の間に新節 b を挿入
6. b の末節に R を挿入
7. if $c(a) = 1$
8. if a が葉である then
9. a と $p(a)$ の間に新節 b を挿入
10. b の末節に R を挿入
11. else
12. 重複がある節で Insert(R) を行う

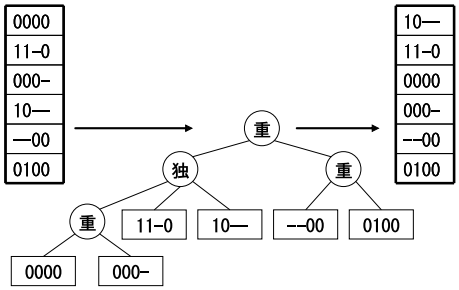

図 2: 独立ルール集合の作成

13. if $c(a) > 1$ then
14. a と $p(a)$ の間に新節 b を挿入
15. b の末節に R_i を挿入

すべての節を帰りがけ順にたどる計算量は $O(n)$ 時間となる．重複の判定は $O(m)$ 時間であるので，T を構成するための計算量は $O(n^2m)$ 時間となる．

3.4 独立ルール集合を用いた負荷軽減

独立節毎に集合を予想平均評価パケット数の降順に並べ替える．左の葉から右の葉へとルールを並べる．予想平均評価パケット数は T を構成する時点で，計算量を変えずに計算できる．以下の手順で，再構成を行う．

1. while (T を行きがけ順にすべての節をたどる) do
2. { 現在の節 a について }
3. if a が独立節 then
4. 枝を予想平均評価パケット数の降順に並べ替え
5. if $a =$ 葉 then
6. $\boldsymbol{R}$ の末尾に a を加える
7. end;

図 2 の場合，適用すると負荷が 28 が 20 へと減る．すべての節を行きがけ順にたどる計算量は $O(n)$ 時間なので，上記のアルゴリズムの計算量は $O(n)$ 時間となる．ただし，いくつかの例で負荷が軽減することを確認したが，負荷が減るための十分条件は今のところ見つかっていない．

4 おわりに

これまでパケットフィルタリングの負荷を軽減するアルゴリズムの計算量は理論的に検討されることはなく，単純なアルゴリズムでは指数時間を要していた．本稿では，予想平均評価パケット数という指標を用いることで多項式時間のアルゴリズムを提案した．

参考文献

[1] 田中賢， 伊藤聖，“ ネットワーク機器の負荷を軽減するフィルタリングルール再構成法 ”， 信学論，Vol. J88-B， No. 9，pp. 905-912 (2005).