論　文

# 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

田中　賢† 伊藤　聖†

# Reconstruction Method of Filtering Rules in order to Mitigate Load of Network Gears

Ken TANAKA† and Suguru ITOH†

あらまし　ルータやマルチレイヤスイッチなどのネットワーク機器において，ポリシに従った効率の良いパケットフィルタリングを実現することは，安定したネットワークを運用するために不可欠である．ここでは，パケットフィルタリングの効率を議論するための理論的枠組みを提案する．ネットワーク機器のフィルタリングルールに用いられる，プロトコル，ポート，アドレス，各種のオプションなどを単純化し，論理式の集合としてフィルタリングルールを形式化する．パケットの転送可否を判断するまでに適用されるルールの数を，そのパケットがネットワーク機器に与える負荷と考え，これを軽減するようなフィルタリングルールの再構成法を示す．提案した枠組みは，機種依存性が高いネットワーク機器の設定や運用法に関する問題を，一つの枠組みの中で体系的に取り扱うことに役立つ．

キーワード　パケットフィルタリング，フィルタリングルール，アクセスリスト，セキュリティポリシ

## 1. まえがき

広帯域のアクセス網が普及するにつれ，ルータやマルチレイヤスイッチなどのネットワーク機器におけるパケットフィルタリングが重要度を増している．増大する不正アクセスからの防御とユーザの利便性の確保を両立するためには，ポリシに従った過不足のないフィルタリングルールを構成する必要がある．

パケットフィルタリングは，ネットワーク機器に一定の負荷を与えるが，これはパケット転送の遅延を引き起こし，サービス品質の低下につながる．大量のパケットを受ける場合には，この負荷がバッファあふれやコネクションタイムアウトの原因となる．

パケット転送時の遅延は，フィルタリングルールのサイズやフィルタリングルールを構成するルールの順序に依存する．フィルタリングルールを設計する際には，主としてポリシを忠実に反映することに重点が置かれており，ネットワーク機器のスループットの点でフィルタリングルールの最適性は必ずしも保証されない．フィルタリングの効率向上のためには，ルールの統合や順序の入換えが行われることが一般的だが，多くの場合それは設計者の経験に基づくものであり，手順の妥当性やフィルタリングの効率について理論的に検討されることはまれである．

堀川ら [1] は，ネットワークトポロジーとポリシの記述からアクセスリストを生成するシステムを提案しているが，ネットワーク機器の負荷軽減についてはアクセスリスト構成時に部分的に用いる知識として取り込んでおり，アクセスリスト全体の最適性については議論していない．

本論文ではまず，ネットワーク機器におけるフィルタリングルールの最適性を議論するための理論的枠組みを提案する．フィルタリングルールに用いられる，プロトコル，ポート，アドレス，各種のオプションなどを単純化し，ルールの集合としてフィルタリングルールを形式化する．パケットの転送可否を判断することは，ルールの集合によりパケットの評価型を求めることである．このとき，パケットの評価型を決定するまでに適用されるルールの数が，そのパケットがネットワーク機器に与える負荷と考えられる．そこで，ネットワーク機器に到達するすべてのパケットがネットワーク機器に与える負荷の総和を最小にするフィルタリングルールを，最適なフィルタリングルールと定

† 新潟大学工学部，新潟市
Faculty of Engineering, Niigata University, 8050 2-no-cho Ikarashi, Niigata-shi, 950–2181 Japan

義する．

一般に，ネットワーク機器に到達するパケットに付随する属性値をあらかじめ予測することは困難なので，ここでは，想定されるすべてのパケットがネットワーク機器に与える負荷の平均を考え，これを最小にするようなフィルタリングルールとして，準最適フィルタリングルールを定義する．

次に，形式化に基づき，フィルタリングルールを最適化する方法の適用範囲や妥当性を明らかにする．既存のフィルタリングルールを最適化する方法として，ルールの併合と入換えを取り上げ，それが適用できるための必要十分条件を示す．

最後に，二つの方法に基づくフィルタリングルールの再構成法を提案し，その有効性を例示する．負荷平均を目的関数と考えたとき，フィルタリングルールの最適化は目的関数を最小化するようなフィルタリングルールを求める問題となる．再構成法が，目的関数値をより小さくするようなフィルタリングルールを与えることを確認する．

フィルタリングルールを形式的に扱うことは，フィルタリングルールの設計にかかわる種々の手法を理論的に扱うことを可能にする．ポリシに沿った目的関数を定義すれば，負荷平均以外の観点からのフィルタリングルールの評価や，それに従った設計手法の評価も可能である．提案した枠組みは，機種依存性が高くアドホックになりがちなネットワーク機器の設定や運用法を，一つの枠組みの中で体系的に取り扱うことに役立つ．

## 2. パケットフィルタリングのモデル

本章では，パケットフィルタリングをモデル化し，フィルタリングルールからルータの負荷を見積もる関数を定義する．

### 2.1 最適フィルタリングルール

ネットワーク機器におけるパケットフィルタリングは図 1 のようにモデル化することができる．図中，$R_i$ は $i$ 番目のフィルタリングルールのルール，$n$ はフィルタリングルールを構成するルールの数である．$i$ をルール $R_i$ のルール番号と呼ぶ．$A$ と $D$ は各ルールがパケットに付与する評価型で，$A$ はパケットの転送許可を，$D$ はパケットの転送拒否をそれぞれ表す．ルールが与える評価型を明示する際は，$R_i$ に評価型を添える．すなわち，$R_i^A$ はパケットの転送を許可するルールを，$R_i^D$ は転送を拒否するルールを表す．ルール番号が $i$，$i+1$，$i+2$，$\cdots$ のように連続するとき，それらのルールは連続するという．

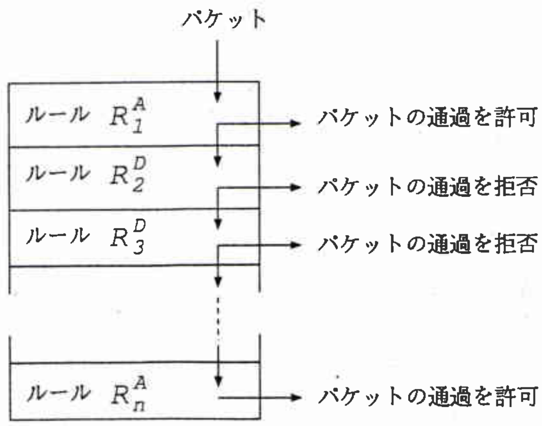


図 1 パケットフィルタリングのモデル
Fig. 1 Model of packet filtering.

ルータにパケットが到着すると，$R_1$ から順にルールを適用し，パケットの転送を許可するか拒否するかを判断する．ルール $R_n$ では，それ以前のルールで評価型が決まらなかったすべてのパケットについて，デフォルトの評価型を与える．

各ルールの条件式は，以下のような論理式とする．

$$R_i = b_0 b_1 \cdots b_m \quad (b_i \in \{0, 1, \text{-}\}) \tag{1}$$

アドレスは，マスク記号 - を伴った論理式とする．送信元あるいは送り先，ポート番号，プロトコルといった条件を，ビット列としてアドレスの上位に付加し一つの論理式として扱う．例えば，ポート番号のビット長を 2，アドレスのビット長を 8 とすると，

$$R_3^A = 0111\text{------} \tag{2}$$

は，フィルタリングルールの 3 番目のルールで，アドレスの上位 2 ビットが 11，ポート番号が 1 であるようなパケットの転送を許可することを表す．

すべてのパケットは，$n$ 個のうちのいずれかのルールによってその評価型が決まる．今，ネットワーク機器に到着するパケットのアドレスが頻度分布 $F$ であると，各ルールで評価型が決まるパケットの数を評価パケット数と呼び，これを以下のように定義する．

[定義 1]（評価パケット数） 頻度分布 $F$ に従って到着するすべてのパケットに対し，$R_i$ で評価型が決まるパケットの数を $R_i(F)$ 評価パケット数と呼び，$|R_i(F)|$ と表す．

ルールは $R_1$ から順に適用されるので，$|R_i(F)|$ は $R_i$ 以前のすべてのルールに依存する．一つのルール

を適用することによるネットワーク機器の負荷を 1 とすると，1 組のルール $R$ によって決まるフィルタリングの負荷 $L(R,F)$ は，$|R_i(F)|$ の重み付きの総和として以下のように定義できる．

［定義 2］（フィルタリング負荷）

$$L(R,F) = \sum_{i=1}^{n} i|R_i(F)| \tag{3}$$

$L(R,F)$ を最小にするようなルール $R$ は，ネットワーク機器の負荷を最小にする意味で最適なルールである．このようなルールによって構成されるフィルタリングルールを最適フィルタリングルールと呼ぶ．

### 2.2 準最適フィルタリングルール

最適フィルタリングルールを構成するためには，ネットワーク機器に到着するパケットのアドレスの頻度分布が必要である．一般に，この頻度分布は未知であることから，最適フィルタリングルールの構成は困難である．

ここでは，すべてのアドレスから同じ頻度でパケットが到着するとしたときの最適フィルタリングルールを考える．このとき，1 組の $R$ に対し各評価パケット数 $|R_i(F)|$ は $F$ に依存せず一意に定まる．以後，このような評価パケット数を $|R_i|$ と表す．このとき，ネットワーク機器の負荷 $L(R)$ は以下のように表される．

$$L(R) = \sum_{i=1}^{n} i|R_i| \tag{4}$$

$L(R)$ を最小にするようなルールは，フィルタリングの際にネットワーク機器にかかる負荷を全パケットについて平均化する．このようなルールによって構成されるフィルタリングルールを準最適フィルタリングルールと呼ぶ．

## 3. フィルタリングルールの最適化法

フィルタリングルールを構成する問題を最適化問題の一つと考えると，$L(R)$ は $R$ を変数とする目的関数となる．各 $|R_i|$ の間には依存関係が存在するため，$L(R)$ を最小にする $R_i$ を求めることは容易ではない．ここでは，以下の二つの方針に沿って，既存のフィルタリングルールの $L(R)$ を小さくするような再構成手順を考える．

（1） ルールの併合によりルール数を削減する．

（2） 上位のルールで評価型が決まるパケットが多くなるよう，ルールを入れ換える．

それぞれの方針は，ネットワーク機器の設定者が経験的に用いているものだが，再構成手順については設定者のスキルによるところが大きく，不適切な設定による不具合や負荷の増大の一因となる．ここでは，二つの方針に応じた再構成手順について，それぞれの手順が適用できるための既存フィルタリングルールに関する必要十分条件を与える．

### 3.1 Quine-McCluskey 法

本節では，Quine-McCluskey 法 [3] によるルールの削減と併合法を考える．

本論文では，フィルタリングルールの条件式はアドレスのみを用いて表現することとしている．アドレスは 2 進数で表されるので，パケットの評価は，アドレスの各ビットをリテラル，ルールを項とした論理関数による評価とみなすことができる．よって，フィルタリングルールの併合は論理関数の最小化の問題となる．

Quine-McCluskey 法は，任意の論理関数が与えられたとき，その論理関数の最小積和型表現を求める方法である．これを用いて，複数のフィルタリングルールを一つにまとめることができる．4 ビットの条件部からなる三つのルールを Quine-McCluskey 法により併合して二つとする例を図 2 に示す．

### 3.2 ルール併合の条件

フィルタリングルールが与えられたとき，評価型の等しいルール同士を無条件で併合することはできない．以下では，ルールの併合が可能となるための，評価型とルールの順序に関する必要十分条件を与える．

［定理 1］ Quine-McCluskey 法でルールを統合できるのは，ルールが与える評価型が同じで連続しているときであり，その逆も成り立つ．

（十分性） 評価型が異なるルールに対して Quine-McCluskey 法を適用すると，適用後の最小積和型表現のルールの評価型が決まらない．よって，評価型の異なるルールに対して Quine-McCluskey 法を適用することはできない．

連続でないルールに対して Quine-McCluskey 法を適用すると，適用後の最小積和型表現のルールの位置が決まらない．よって，連続でないルールに対して

$$\begin{array}{lcl} R_1^A = \texttt{00--} & & R_1^{A'} = \texttt{00--} \\ R_2^A = \texttt{010-} & \Rightarrow & R_2^{A'} = \texttt{0-0-} \\ R_3^A = \texttt{00-1} & & \end{array}$$

図 2 Quine-McCluskey 法によるルールの併合
Fig. 2 Rule merge via Quine-McCluskey method.

Quine-McCluskey 法を適用することはできない．
（必要性） $R_i$ から $R_j$ $(j > i)$ までの評価型が同じルールを，その最小積和型表現で表してもそのルールで評価型が決定するパケットは変わらない．

**3.3 ルール入換えの条件**

ルール併合の際，Quine-McCluskey 法を効果的に使うためには，評価型の等しいルールが連続するように既存のフィルタリングルールの順番を入れ換える必要がある．フィルタリングの機能を変えることなく，任意のルールを任意の場所に移動することはできない．ここでは，フィルタリングの機能を変えることなく，ルールを移動するためのフィルタリングルールに関する必要十分条件を与える．

連続する二つのルール $R_i$ と $R_{i+1}$ について，$R_{i+1}$ を $R_i$ の手前に移動するとき，$R_{i+1}$ を移動するルール，$R_i$ を $R_{i+1}$ が超えるルールと呼ぶ．ここで，連続する二つのルール $R_i$ と $R_{i+1}$ の重複を以下のように定義する．

［定義 3］（重複） $R_i$ で評価型が決まり，$R_{i+1}$ を $R_i$ の手前に移動したとき $R_{i+1}$ で評価型が決まるようなパケットが存在するとき，$R_i$，$R_{i+1}$ は重複しているといい，重複しているパケットの数を $|R_i \wedge R_{i+1}|$ で表す．

連続する二つのルールの移動について，以下の定理が成り立つ．

［定理 2］ 移動によりすべてのパケットの評価型が不変であるための必要十分条件は，

（1） 移動するルールと超えるルールの評価型が同じ．あるいは，

（2） 移動するルールと超えるルールの評価型が異なり，かつ移動するルールと超えるルールが重複していない．

である．

（証明） 移動するルールを $R_{i+1}$，超えるルールを $R_i$ とする．$R_i$，$R_{i+1}$ に対し，重複しているパケットの集合を $P_{i,i+1}$，重複しておらず $R_{i+1}$ で評価が決まるパケットの集合を $P_{i+1}$，重複しておらず $R_i$ で評価が決まるパケットの集合を $P_i$ と表す．

（十分性） $R_i$，$R_{i+1}$ の評価型が同じとき，$R_i$，$R_{i+1}$ のどちらのルールが上にきても $P_{i,i+1}$，$P_i$，$P_{i+1}$ のすべてのパケットの評価は変わらない．

$R_i$，$R_{i+1}$ の評価型が異なり，かつ $R_i$ と $R_{i+1}$ が重複していない，すなわち $P_{i,i+1} = \phi$ のとき，$R_i$，$R_{i+1}$ のどちらのルールが上にきても $P_{i+1}$ は $R_{i+1}$ で，$P_i$ は $R_i$ で評価が決まる．よってすべてのパケットの評価は変わらない．

（必要性） （1），（2）以外の条件は，$R_i$，$R_{i+1}$ の評価型が異なり，かつ $R_i$，$R_{i+1}$ が重複しているときである．このとき，$R_{i+1}$ が $R_i$ の上にあるならば，$P_{i+1}$，$P_{i,i+1}$ は $R_{i+1}$ で評価され，$P_i$ は $R_i$ で評価される．しかし，$R_i$ が $R_{i+1}$ の上にあるならば，$P_i$，$P_{i,i+1}$ は $R_i$ で評価され，$P_{i+1}$ は $R_{i+1}$ で評価される．条件より，$R_i$，$R_{i+1}$ の評価型は異なるため $P_{i,i+1}$ の評価が変わってしまう．

ゆえに，$R_i$，$R_{i+1}$ の評価型が異なりかつ $R_i$，$R_{i+1}$ が重複しているとき，$R_{i+1}$ は $R_i$ を超えて移動することはできない．

**3.4 $L(R)$ を小さくする入換え**

$L(R)$ を小さくする方法の一つは，上位のルールで評価型が決まるパケットが多くなるように，ルールを入れ換えることである．ここでは，連続した二つのルール，$R_i$，$R_{i+1}$ $(i = 1, 2, \cdots, n-2)$ について，入換えにより $L(R)$ が減少するための条件を与える．評価型が等しい場合，異なる場合それぞれについて，各ルールの評価パケット数に関し以下の必要十分条件が成り立つ．

［定理 3］ 評価型が等しい連続する二つのルール $R_i$，$R_{i+1}$ を並べ換えて，$L(R)$ が減少するための必要十分条件は

$$|R_i| < |R_{i+1}| + |R_i \wedge R_{i+1}| \tag{5}$$

である．

（十分性） $R_i$，$R_{i+1}$ の順番を入れ換えると $i$ 番目のルールの評価パケット数は $|R_{i+1}|+|R_i \wedge R_{i+1}|$，$i+1$ 番目のルールの評価パケット数は $|R_i| - |R_i \wedge R_{i+1}|$ となる．$i+1$ 番目のルールによって評価されるパケットが，$i$ 番目のルールにも評価されることを考慮すると，入換え後と入換え前の $L(R)$ の差は以下のようになる．

$$\begin{aligned} & i|R_i| + (i+1)|R_{i+1}| - i(|R_{i+1}| \\ & \quad + |R_i \wedge R_{i+1}|) + (i+1)(|R_i| - |R_i \wedge R_{i+1}|) \\ & = |R_{i+1}| - |R_i| + |R_i \wedge R_{i+1}| \end{aligned} \tag{6}$$

式 (6) > 0 であれば，$L(R)$ は減少する．すなわち，$|R_i| < |R_{i+1}|+|R_i \wedge R_{i+1}|$ ならば入換えにより $L(R)$ は減少する．

（必要性） $|R_i| \geq |R_{i+1}|+|R_i \wedge R_{i+1}|$ の場合の $L(R)$ の差を考えれば明らか．

［定理 4］ 評価型が異なる連続する二つのルール $R_i$, $R_{i+1}$ を並べ換えて，$L(R)$ が減少するための必要十分条件は

$$|R_i \wedge R_{i+1}| = 0 \text{ かつ } |R_i| < |R_{i+1}| \tag{7}$$

である．

（十分性） 重複のない連続する二つのルール $R_i$，$R_{i+1}$ の順番を入れ換えても評価型が決まるルールは変化しない．つまり，下位のルールにより評価されるパケット数が少ない方が $L(R)$ が小さくなる．よって，$|R_i| < |R_{i+1}|$ ならば，入換えにより $L(R)$ は減少する．

（必要性） 重複のある連続する二つのルールを入れ換えることはできない．

## 4. フィルタリングルール再構成手順

前章で明らかにした条件をもとにした，フィルタリングルールの再構成手順を提案する．

### 4.1 フィルタリングルール再構成手順の概要

定理 1 より，Quine-McCluskey 法によりフィルタリングルールが併合できるのは，評価型の等しい連続したルールが与えられるときのみである．よって，できるだけたくさんのルールを併合するためには，評価型が等しいルールが連続するようにあらかじめルールを入れ換えることが効果的である．そこで，以下の手順でフィルタリングルールを再構成する．

（再構成手順）

（1） 評価型の等しいフィルタリングルールが連続するようルールを入れ換える．

（2） Quine-McCluskey 法によりルールを併合する．

（3） （1），（2）を繰り返し行い，フィルタリングルールの数を最小にする．

（4） 上位ルールの評価パケット数が多くなるようフィルタリングルールを入れ換える．

### 4.2 ルール併合手順

一般に，ルールの併合は上位のルールから優先的に行うのがよい．上位のルールを併合できれば，それより下位のすべてのルールの順位が繰り上がるため，$L(R)$ の値が小さくなることが期待できるからである．このことを考慮した併合手順を以下に提案する．

（併合手順）

（1） まず，$R_1$ に着目する．$R_1$ と評価型が同じで $R_1$ の直前に移動可能なルールをすべて移動する．移動の可否は，定理 2 の条件に従う．

（2） 移動したルールの中で，最上位のルールから $R_1$ までのルールに対し Quine-McCluskey 法を適用してルールを併合する．適用後のルールを $R_1$, $R_2$, ..., $R_m$ とする．ここで $m$ はルールの数である．Quine-McCluskey 法を適用したルールの次に位置するルールを $R_i$ とする．

（3） $R_i$ に着目する．$R_i$ と評価型が同じで，かつ $R_{i-1}$ と $R_i$ の間に移動可能なすべてのルールを移動する．移動の可否は，定理 2 の条件に従う．

（4） 移動したルールの中で，最上位にあるルールから $R_i$ までのルールに対して Quine-McCluskey 法を適用してルールを併合する．適用後のルールを新たに $R_1$, $R_2$, ..., $R_m$ とし，Quine-McCluskey 法を適用したルールの直後のルールを $R_i$ とする．

（5） $i = m$ となるまで（3），（4）を繰り返す．

（6） $R_m$ から逆順に見て，$R_m$ と評価型が等しくかつ連続するルールをすべて削除する．

$R_m$ は常にデフォルトルールであり（5）までの併合には関与せず，（6）で $R_m$ と評価型が等しいルールとのみ併合される点に注意されたい．

### 4.3 ルール入換え手順

ルールの併合を行った後，各ルールの評価パケット数を考慮して，より多くのパケットの評価型が上位のルールで決まるよう，ルールの入換えを行えば，$L(R)$ を小さくすることができる．連続しないルールの入換えについては，$|R_i|$ $(2 \le i \le n)$ が $R_1$, $R_2$, ..., $R_{i-1}$ に依存するため，$L(R)$ が減少するか否かを判断することが難しい．ここでは，ルールの入換えを上位の連続するルールから繰り返し順に行うことで，$L(R)$ を小さくするような入換え手順を提案する．

（入換え手順） $|R_i|$ $(1 \le i \le n)$ について，$|R_i| = 0$ であるようなルールを削除し，削除後のルールを $R_1$, $R_2$, ..., $R_m$ とする．$R_j$ $(2 \le j \le m)$ について，$L(R)$ が小さくなるよう $R_k$ $(1 \le k \le j-1)$ と評価パケット数を比較しながら上位に向かって入換えを行う．入換えを行うか否かは，定理 3, 4 の条件に従う．$R_j$ の位置が決まった後，$R_j$ の次のルールから，再び入換えを行う．$j = m$ となった時点で入換えを終了する．

デフォルトルール $R_m$ は，評価型が異なるすべてのルールと重複するので，入換えにはかかわらない点に注意されたい．

提案した手順により，ルールの併合と入換えを行う例を付録に示す．

## 5. むすび

本論文では，パケットフィルタリングのモデルをもとに最適フィルタリングルール構成の問題を形式化した．既存のフィルタリングルールを準最適フィルタリングルールに近づけるために，ルールの併合と入換えが可能となるための必要十分条件を与えた上で，これに基づく再構成手順を提案した．再構成により $L(R)$ が減少することを示し，提案手法の有効性を例示した．

モデルの中で，各ルールの条件部として任意の論理式を許しているが，実際のフィルタリングルールでは最上位ビットからの連続するネットマスクを伴ったアドレスによる条件のみ記述することができる[2]．今後，このような条件に限ったルールに基づくモデルを構築する必要がある．

本論文では，準最適フィルタリングルールを求める手順については議論していない．フィルタリングルールの仕様から準最適フィルタリングルールを求めるアルゴリズムとその複雑さについて検討する必要がある．

最適フィルタリングルールを求める代わりに，ネットワーク機器から得られる到着パケットの頻度分布をもとに，パケットの近似的な確率分布を考慮したフィルタリングルールを構成する方法を検討する必要がある．このとき，準最適フィルタリングルールを求める問題は，一様分布のもとでの最適フィルタリングルールを求める問題となる．

式 (3) で定義したネットワーク機器の負荷 $L(R, F)$ では，モデルの簡単化のためにルールに含まれるマスクの数が負荷に与える影響を考慮していない．実際のネットワーク機器においては，マスクが少なくなるほど比較ビットが多くなるため，ルール当りの負荷が大きくなる．マスクの数を考慮した負荷を定義し，それを最小にするようなフィルタリングルールの構成法について検討する必要がある．

## 文献


[1] 堀川　茂，今泉貴史，鈴木正人，“アクセスリスト生成システムの設計とその実装に関する研究，” 日本ソフトウェア科学会第 17 回大会予稿集，E4-4, 2000.

[2] D.B. Chapman and E.D. Zwicky（著），歌代和正（監訳），鈴木克彦（訳），ファイアーウォール構築 インターネットセキュリティ，O'REILLY ジャパン，1996.

[3] 柴山　潔，論理回路とその設計，近代科学社，1999.

[4] 髙木直史，論理回路，昭晃堂，1997.


## 付録

以下に，提案した手順によるルールの再構成例を示す．

| 再構成前 | | ルール併合後 | | ルール入換え後 |
|---|---|---|---|---|
| $R_1^A = 0001$ | | $R_1^A = 000\text{-}$ | | $R_1^D = 10\text{--}$ |
| $R_2^D = 11\text{-}0$ | | $R_2^D = 10\text{--}$ | | $R_2^A = 000\text{-}$ |
| $R_3^A = 0000$ | $\Rightarrow$ | $R_3^D = 11\text{-}0$ | $\Rightarrow$ | $R_3^D = 11\text{-}0$ |
| $R_4^A = 11\text{--}$ | | $R_4^A = 1\text{---}$ | | $R_4^A = 1\text{---}$ |
| $R_5^D = 10\text{-}1$ | | $R_5^D = \text{----}$ | | $R_5^D = \text{----}$ |
| $R_6^D = 10\text{-}0$ | | | | |
| $R_7^A = 1\text{---}$ | | | | |
| $R_8^A = 1\text{-}1\text{-}$ | | | | |
| $R_9^D = \text{----}$ | | | | |
| $L(R) = 92$ | | $L(R) = 54$ | | $L(R) = 52$ |

再構成の過程を以下に示す．

(1) 再構成前

| ルール | 評価パケット数 |
|---|---|
| $R_1^A = 0001$ | $\lvert R_1^A \rvert = 1$ |
| $R_2^D = 11\text{-}0$ | $\lvert R_2^D \rvert = 2$ |
| $R_3^A = 0000$ | $\lvert R_3^A \rvert = 1$ |
| $R_4^A = 11\text{--}$ | $\lvert R_4^A \rvert = 2$ |
| $R_5^D = 10\text{-}1$ | $\lvert R_5^D \rvert = 2$ |
| $R_6^D = 10\text{-}0$ | $\lvert R_6^D \rvert = 2$ |
| $R_7^A = 1\text{---}$ | $\lvert R_7^A \rvert = 0$ |
| $R_8^A = 1\text{-}1\text{-}$ | $\lvert R_8^A \rvert = 0$ |
| $R_9^D = \text{----}$ | $\lvert R_9^D \rvert = 6$ |

$L(R) = 92$

(2) 併合

$R_1^A = 0001$　着目
$R_2^D = 11\text{-}0$
$R_3^A = 0000$　移動可能
$R_4^A = 11\text{--}$　$R_2^D$ と重複するため移動不可
$R_5^D = 10\text{-}1$
$R_6^D = 10\text{-}0$
$R_7^A = 1\text{---}$　$R_6^D$ と重複するため移動不可
$R_8^A = 1\text{-}1\text{-}$　$R_6^D$ と重複するため移動不可
$R_9^D = \text{----}$

$\Downarrow$ 移動

$R_3^A = 0000$
$R_1^A = 0001$　着目

$R_2^D = $ 11-0
$R_4^A = $ 11--
$R_5^D = $ 10-1
$R_6^D = $ 10-0
$R_7^A = $ 1---
$R_8^A = $ 1-1-
$R_9^D = $ ----

⇓併合してリナンバ

$R_1^A = $ 000-
$R_2^D = $ 11-0
$R_3^A = $ 11--
$R_4^D = $ 10-1
$R_5^D = $ 10-0
$R_6^A = $ 1---
$R_7^A = $ 1-1-
$R_8^D = $ ----

$R_1^A = $ 000-
$R_2^D = $ 11-0 着目
$R_3^A = $ 11--
$R_4^D = $ 10-1 移動可能
$R_5^D = $ 10-0 移動可能
$R_6^A = $ 1---
$R_7^A = $ 1-1-
$R_8^D = $ ---- $R_7^A$と重複するため移動不可

⇓移動

$R_1^A = $ 000-
$R_5^D = $ 10-0
$R_4^D = $ 10-1
$R_2^D = $ 11-0 着目
$R_3^A = $ 11--
$R_6^A = $ 1---
$R_7^A = $ 1-1-
$R_8^D = $ ----

⇓併合してリナンバ

$R_1^A = $ 000-
$R_2^D = $ 10--
$R_3^D = $ 11-0
$R_4^A = $ 11--
$R_5^A = $ 1---
$R_6^A = $ 1-1-
$R_7^D = $ ----

$R_1^A = $ 000-
$R_2^D = $ 10--
$R_3^D = $ 11-0
$R_4^A = $ 11-- 着目
$R_5^A = $ 1--- 移動可能
$R_6^A = $ 1-1- 移動可能
$R_7^D = $ ----

⇓移動

$R_1^A = $ 000-
$R_2^D = $ 10--
$R_3^D = $ 11-0
$R_6^A = $ 1-1- 移動可能
$R_5^A = $ 1--- 移動可能
$R_4^A = $ 11-- 着目
$R_7^D = $ ----

⇓併合してリナンバ

$R_1^A = $ 000-
$R_2^D = $ 10--
$R_3^D = $ 11-0
$R_4^A = $ 1---
$R_5^D = $ ----

| ルール | 評価パケット数 |
|---|---|
| $R_1^A = $ 000- | $\lvert R_1^A \rvert = 2$ |
| $R_2^D = $ 10-- | $\lvert R_2^D \rvert = 4$ |
| $R_3^D = $ 11-0 | $\lvert R_3^D \rvert = 2$ |
| $R_4^A = $ 1--- | $\lvert R_4^A \rvert = 2$ |
| $R_5^D = $ ---- | $\lvert R_5^D \rvert = 6$ |

$L(R) = 54$

(3) 入換え

| | |
|---|---|
| $R_1^A = $ 000- | $\lvert R_1^A \rvert = 2$ |
| $R_2^D = $ 10-- | $\lvert R_2^D \rvert = 4$ |
| $R_3^D = $ 11-0 | $\lvert R_3^D \rvert = 2$ |
| $R_4^A = $ 1--- | $\lvert R_4^A \rvert = 2$ |
| $R_5^D = $ ---- | $\lvert R_5^D \rvert = 6$ |

$j = 2$ について，$|R_1^A \wedge R_2^D| = 0$ かつ $|R_1^A| < |R_2^D|$ より入換えを行う．

$\Downarrow$ 入れ換えてリナンバ

$R_1^D =$ 10-- $|R_1^D| = 4$
$R_2^A =$ 000- $|R_2^A| = 2$
$R_3^D =$ 11-0 $|R_3^D| = 2$
$R_4^A =$ 1--- $|R_4^A| = 2$
$R_5^D =$ ---- $|R_5^D| = 6$

$j = 3$, 4, 5 については入換えの条件を満たさないため終了．

$L(R) = 52$

（平成 16 年 7 月 7 日受付，11 月 4 日再受付）

**田中　　賢**　（正員）

平元早大・理工・電気卒．平 3 東工大・総合理工・システム科学修士課程了．平 7 東工大・理工・情報工学博士課程単位取得．平 14 新潟大工学部情報工学科講師．博士（情報科学）．情報処理学会会員．

**伊藤　　聖**

平 16 新潟大・工・情報工卒．ネットワーク機器の運用法に関する研究に従事．