L-003

アクセスリストコンパイラにおけるルール最適化手法
Rule optimization technique in access list compiler

加瀬 智博*
Tomohiro KASE

今泉 貴史*
Takashi IMAIZUMI

1 はじめに

近年、ルータなどのネットワーク機器において、ポリシー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事は、安定したネットワークを運用するために必要不可欠である。しかし、パケットフィルタリングで利用するアクセスリストの規模は大きくなる傾向があり、負荷が大きくなりがちである。ACL(AccessListCompiler)を用いるとアクセスリスト作成時の負荷は減らすことができるが、作成されたアクセスリスト全体の最適性には触れられていない。そこで、ネットワーク機器の負荷を軽減できるアクセスリストの最適化手法が求められる。

今までのアクセスリスト作成においては、IP アドレス等の情報が固定的に扱われており、ルールの統合を行なうのは困難となっていた。そこで、IP アドレスを流動的に扱うという新しいアプローチをとり、IP アドレスの再割り当てにより今までは不可能だったルールの集約方法を提案していくことが可能になると考えられる。

本研究は、アクセスリスト全体の総記述量を最適化するために、IP アドレスを変更して、その変更したネットワークからアクセスリストのルール数の総記述を減らす事を目的とし、IP アドレスの変更を考慮に入れた ALC システムを作成する研究である。

2 パケットフィルタリング

図 1 で表されるように、ルータ等のネットワーク機器にパケットが到達すると、R1 から順にルールの条件とパケットの中身を比較していきマッチするものがあった場合にアクセスリストで決められた動作 (通過を許可、禁止) を実行する。

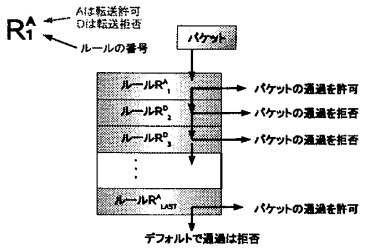

図 1: パケットフィルタリングの動作

パケットフィルタリングにおける、パケットの通過条件をアクセスリストと呼ぶ。さらに、アクセスリストを構成する一つ一つの通過条件をルールと呼ぶ。ルールに記述可能な条件は、一般的に次の通りである。

- パケットの送信元の IP アドレスとサブネットマスク
- パケットの送信先の IP アドレスとサブネットマスク
- 通信に使用するプロトコル
- 送信元の TCP または UDP のポート番号
- 送信先の TCP または UDP のポート番号

パケットフィルタリングにおける主な研究としては、ルール検索回数を減らすためにルール全体を並び替えることを主とした研究 [3] が多いが、これはパケットの種類の発生確率に依存したものであり、ルール数に関する根本的な解決とはいえない。

3 アクセスリスト生成システム

3.1 ALC

ALC(Access List Compiler) は、パケットフィルタリングの設定に関する管理者の負担を軽減することを目的としたシステムである [2]。IPv4 対応の ALC[2] や IPv4 と IPv6 のマルチプロトコルに対応している ALC3[1] は、アクセスリスト作成時の問題となる専門的な知識や記述の複雑さを意識せずに、管理すべきネットワークにおけるルータ等のネットワーク機器全てに、ポリシーを実現したアクセスリストを生成することができる。

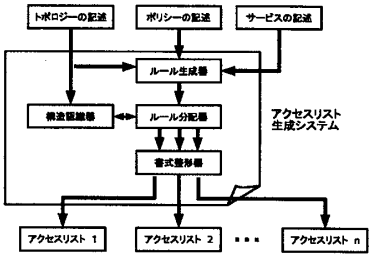

図 2: ALC3 フレームワーク

図 2 は ALC3 システムのフレームワークを表し、ALC は可読性の高い 3 つの記述言語を用い、実際のルータで設定可能なアクセスリストを生成する。

システムで用いる各 3 つの記述言語の内容と具体例を以下にあげる。

*千葉大学大学院融合科学研究科情報科学専攻

- トポロジーの記述
  プログラム 1 に表されるように、ネットワークはネットワーク名とそれを現す IP アドレスと prefix 長の組で指定。ルータはどのネットワークに接続されているかをネットワーク名等を用いて記述。

```
network{
     outside{IPv4 0.0.0.0/0;}
     dmz    {IPv4 133.82.180.19/30;}
     inside {IPv4 192.168.20.0/24;}
}
router {
        R0: portmaster3 {
                outside.* ;
                dmz.* ;
        }
        R1: cisco {
                dmz.* ;
                inside.* ;
        }
}
alias {
        hostA {
              haIPv4global 133.82.180.20/32;
        }
}
```

プログラム 1: トポロジーの記述の例

- サービスの記述
  プログラム 2 のように指定されたサービスに対応したポート番号とプロトコルの情報を与えたり、reverse を用いて応答パケットを自動生成する等の設定が記述されている。

```
service{
     telnet{
         from 1024-65535 to 23/tcp;
         reverse from 20 to 1024-65535/tcp;
     }
     www{
         from 1024-65535 to 80/tcp;
         from 1024-65535 to 8080/tcp;
     }
}
```

プログラム 2: サービスの記述の例

- ポリシーの記述
  ネットワークで実現すべきポリシーを 1 行ずつ記述したものである (プログラム 3 参照)。

```
default deny;
permit from inside;
permit telnet between outside and hostA;
```

プログラム 3: ポリシーの記述の例

ALC システムは大きく分けて 3 つのサブシステムから構成される。1 つは、上記の 3 つの記述言語から中間言語の記述を生成するルール生成器である。

ルール生成器では、ポリシーの記述で許可されている簡易言語の抽象的な記述で書かれた情報を、他の記述の情報を用いて実際のアクセスリストに必要な実数値の情報に展開していく。

2 つ目は、各ネットワーク機器が持つべき部分的なアクセスリストを作成、配布を行なうルール分配器である。ルール分配器では、ルールに対応したパケットの通過する全経路にある機器がそのルールを保持すべきであると定義し、持つべきルール群を判断し各機器に必要なアクセスリストを形成する。

3 つ目は、各ルータに対し中間言語の記述を実際のルータで設定可能なアクセスリストに変換する書式整形器である。

中間言語は 3 つのサブシステム間で情報をやり取りするために用いられ、機種に依存しないアクセスリストとして扱うことができる。

3.2 ルール展開の動作例

ALC3 に実際に図 3 のネットワークとポリシーの記述例を与えて、動作を確認していく。

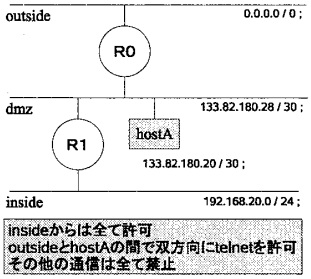

図 3: ALC3 に与えるネットワークとポリシーの例

この、ネットワークに関するトポロジーの記述とサービスの記述、ポリシーの記述はプログラム 1 と 2 と 3 にあたる。これらの情報を元にポリシーの記述からルールを展開していく。まずは、between 等の抽象記述の記述を全て方向性を示す from と to の形にし、サービスに関する応答パケットなどの展開をする (図 4 中段)。

次に、ホスト名を実際の IP アドレスに置換する (4 下段)

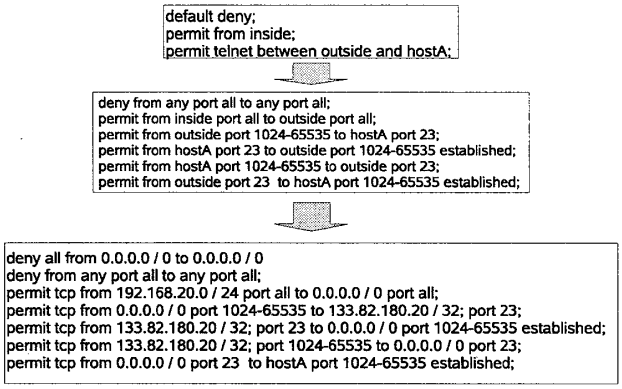

図 4: ルール生成期による展開の例

4 ルール最適化器

本研究では、既存の ALC 3 にルール最適化器を追加する。ルール最適化器は、ルール生成器で出力された中間言語を元にルール数の削減を行ない、最適化された中間言語として出力する機構である。

また、新しいフレームワークを図 5 に示す。

ルール最適化のアプローチとして、図 6 で表されるように 7 台のホストを a,b,c3 つのグループに振り分けていくことを例として考える。

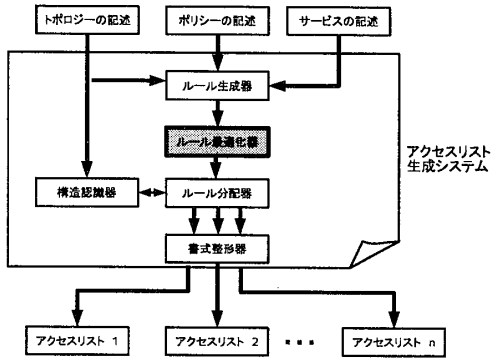

図 5: 新しいフレームワーク

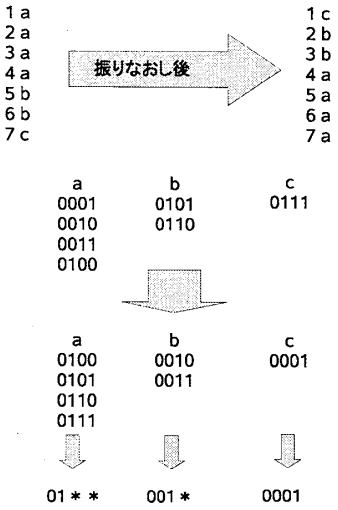

図 6: アプローチの概略図

これらは、最初固定的なアドレス (図 6 上段では頭から順に 1〜7) が割り振られているが、その場合図 6 中断のように a.b.c 各グループにおける共通項がくくりだせない。そこで、図 6 下段のように割り振りなおしてやると、共通部分を括りだして目的とするホストを含むネットワークアドレスを決定できて、a,b,c 各グループを指すならば、IP アドレス領域を 1 つに集約して記述する事が可能になる。このように IP アドレスを再割り当てすることで、複数のルール群を併合していくことで、ルール数を最適化していく。

実際のルール最適化では、アクセスリストに含まれるルール群の中から、送信先 IP アドレス (送信元 IP アドレス) 以外のフィールドが一致するルール群に着目をする。そのルール群が持つ送信先 IP アドレス (送信先 IP アドレス) が、同一ネットワーク内のネットワークアドレスに含まれる場合、それらの IP アドレス群のグループにたいして便宜的なネットワークアドレス (残りの BIT プール内で IP アドレス) を割り振ることで送信先を一つに集約してルールを書き直していく。これらによりグループ内のルールは一つに集約していくことが可能になる。

4.1 アドレス再割り当てアルゴリズム

ここで、便宜的なネットワークにまとめられるべきホスト達のアドレス群 (IP アドレス) をグループと呼ぶ。このグループの分け方としては、まず 1 つのルータとルータに囲まれたネットワークのネットワークアドレスに注目する。

次に中間言語のルール群から、条件にあったルール群を抽出する。そのときのルール群抽出における条件としては以下が上げられる。

- パケットの送信元 (先)IP アドレスが同一
- 送信元・先のポート番号が同一
- パケットの送信先 (元)IP アドレスが同一ネットワークアドレス内に含まれる場合

抽出されたルール群が持つ情報の集合を、後で便宜的な新しいネットワークアドレスで再割り当てを行なう対象としてグループ 1、グループ 2 とグループ分けをする。このとき、最終的にどのグループにも属さないホスト達をまとめたものも 1 つのグループとする。

これらのグループにネットワークアドレスを細分化した領域を割り振っていきアドレス再割り当てを行なう。

以下にプライベート IP アドレス再割り当てのアルゴリズムを述べる。

1 あるネットワーク内のグループ同士を比較していき、情報に重複関係があれば、重複する部分と、そうでない部分としてわけ、複数のグループとして扱う。

2 グループ同士に部分集合の関係があれば、そのグループのサブグループとする。

3 最終的なグループ群 (サブグループ群を除く) を、保持する情報 (IP アドレス) の数の多い順に並べる。

4 ハフマン符号化を行い対象ネットワーク内に許される BIT プール範囲内で便宜的なネットワークアドレスを割り振る。

5 さらに、サブグループを持つグループにたいし 1〜4 を行なう。

6 グループ内のホストに対して、振りなおす IP アドレスを決定する。

7 6 のときの対応表をデータとして保持する。

8 全てのネットワークで 1〜7 を行なう。

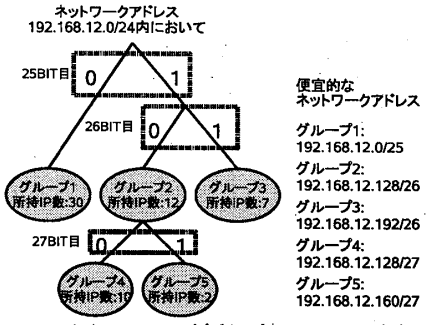

図 7: IP 再割り当てによる例

4.2 簡単なルール集約の例

例えば、図 8 において、hostA から hostC へ,hostA から hostD,hostA から hostE へ IP の通信を許可するとする。この場合、以下のように 1 つのホストから同じネットワークアドレスに対し以下の 3 つのルールが作られてしまう。

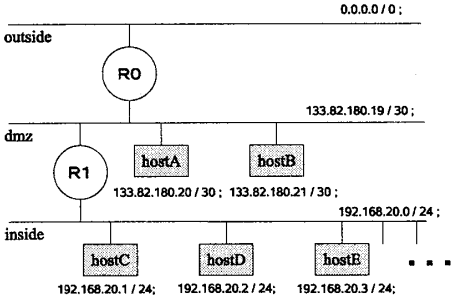

図 8: 論理ネットワーク図の例

```
ipfw -q add permit tcp from 133.82.180.20/30
                        to 192.168.20.1/24
ipfw -q add permit tcp from 133.82.180.20/30
                        to 192.168.20.2/24
ipfw -q add permit tcp from 133.82.180.20/30
                        to 192.168.20.3/24
```

プログラム 4: ipfw 用の出力例 (部分的)

このルール群は、to 以外の情報が一致しているため一つのグループとして扱うことができる (簡単化するために他のグループは存在しないものする)。そして、1 BIT を消費した新しく便宜的なネットワークアドレス 192.168.20.128/25 を割り振ってやることでルールの集約が可能である。

図 8 では、192.168.20.128/25 と 192.168.20.0/25 というネットマスクを用いて図 9 のように新しい IP アドレスを割り振ってやる。これにより、プログラム 4 のルールをプログラム 5 のように集約することができた。

```
ipfw -q add permit tcp from 133.82.180.20/30
                        to 192.168.20.128/25
```

プログラム 5: ipfw 用の集約後の出力例 (部分的)

5 考察

IP アドレスの再割り当てにより、ルール数の削減を実現できたが問題点として、変更後のアドレスの実現性が挙げられる。対応表を元に、ホストに対する DNS のマッピングを行なうことはできるが、DNS サーバーのキャッシュのクリアを行なわなければならない。また、DHCP サーバーにおいては必要不可欠な情報である MAC フィールドに関する情報を保持していないため、再割り当て後のネットワークの実現を行なうのは難しいといえる。トポロジーの記述に MAC アドレスの情報を追加すれば解決はできるが、全ての DNS サーバーのキャッシュをクリアしたり、全てのマシンの MAC アドレス情報を、記述に追加する事はあまり現実的とはいえない。

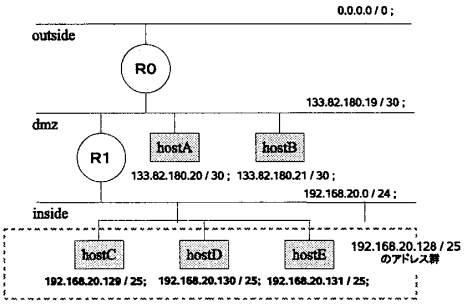

図 9: IP 振りなおしの例

また、グループ化することによりアドレスフィールド内に保持できるホスト数の減少が発生し、フィールド内に無駄が生じてしまう可能性がある。それらを踏まえて IPv6 における有効性は、リナンバリング機能による IP の振りなおしの利便性が高いこと、IPv4 に比べ使用できる BIT フィールドの多さから、IPv6 での有効性は高いといえる。

6 まとめ

IP アドレスの再割り当てによって、アクセスリストのルール数を減らせるという IP アドレス再割り当ての新しい利用方法を提案した。今後は、IP の変更の実現性を高め、現時の同一ネットワーク内のルールの併合だけではなく、ネットワークをまたいだルールの併合も可能にする IP 再割り当ての仕方を考えていく必要がある。

さらには、重複関係で、グループを上手く階層構造に配置できなかったグループ群をいかにして扱うかといった点も考慮して行く必要がある。

また、このアクセスリストを構成するルール群自体を減らす研究と、ルール群の発生確率による並び替えを組み合わせることで、より高いパフォーマンスを発揮できる最適なルール最適化器を作る事が可能になると考えられる。

参考文献

[1] 谷津　文平，今泉　貴史，“マルチプロトコルネットワークにおけるアクセスリスト生成システム”，千葉大学大学院修士論文，2004.0

[2] 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.

[3] 田中 賢，伊藤　聖，“ネット枠機器の負荷を軽減するフィルタリングルール再構成法”，電子情報通信学会論文誌 Vol.J88 － B　No.5，2005.

[4]　石川　拓道，吉浦　紀晃，“ハードウェア処理される AccessControlList の短縮化”，インターネットと運用技術シンポジウム 2009，2009.