平成 25 年度卒業論文

論文題目

仮想マシンを用いた分散ファイアウォールの負荷軽減

神奈川大学 工学部 電子情報フロンティア学科
学籍番号　201002678
大畑　嵩

指導担当者　木下宏揚　教授

目 次

図 目 次

第1章　序論

近年、通信インフラとしてのインターネットの急速な発展に伴い、誰でも容易にインターネットを利用できるようになってきた。一方で、インターネットに接続時は、正規の受信者以外の第三者に対しても不正アクセスの機会を与えている。そのため誰もが悪意のあるものによる攻撃（不正侵入、データの改竄、データ盗聴、なりすまし、過負荷攻撃、経路の追跡な）の対象者になりうる。このような問題に対してネットワーク利用者がセキュリティ対策を実施する場合、IP パケット対策として、ファイアウォールが一般的である。ファイアウォールの目的は、必要な通信のみを通過させ、不要な通信を遮断することであり、通常内部のネットワークにアクセスができないような制御が一般的である。

ファイアウォールの設置方法として、外部ネットワークと内部ネットワークの境界のみに設置する方法（集中型）と、境界のみでなく内部ネットワークの部署単位でも設置する方法（分散型）が考えられる。今回は、部署単位ごとにファイアウォールを設置する分散型ファイアウォールに注目する。

分散型ファイアウォールでは、部署ごとに異なるセキュリティポリシーが設定でき、柔軟で強固なセキュリティを実現できる。しかし、今日の大規模ネットワークに対応するため、日々ルールチェーンの数は増え続けている。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（遅延）と考えられる。分散ファイアウォールのセキュリティポリシー（ルール）をほかのファイアウォールと連携させることにより、全体としてのルールを減らし、負荷の要因となるマッチング回数軽減によるファイアウォールの最適化について、従来示されている特定の場合と、新しく不特定の場合、についてアルゴリズムを考える。

本論文の流れを以下に示す。

第 2 章では、研究についての基礎知識を述べる。

第 3 章では、特定の場合、不特定の場合の提案したアルゴリズムを行い、実験結果を示す。

第 4 章では、考察を述べる。

第 5 章では、本研究で行ったことをまとめ、今後の課題について述べる。

第2章　基礎知識

2.1　ファイアウォール

ファイアウオールとは、インターネットなどの信頼できないネットワークからの攻撃や、不正アクセスから系邸哉内部のネットワークを保護するためのシステムである。[1] ファイアウオールは、ネットワークとネットワークを分離し、特定のプロトコルや宛先パケットしか通過させないパケットフィルタリングと、インターネット上のサーバに代わって内部のネットワークにサービスを提供するプロキシサーバの機能により、内部ネットワークと外部ネットワーク問のパケット転送を行う。

ファイアウォールの設置方法には、以下の二通りが考えられる。

- 外部ネットワークと内部ネットワークの境界のみに設置する方法（集中型）
- 外部ネットワークと内部ネットワーク境界のみでなく、内部ネットワークの部署単位でも設置する方法（分散型）

上記二通りについて利点および問題点について述べる。

2.1.1　集中型ファイアウォール

集中型のファイアウォールでは外部ネットワークと内部ネットワークのみに設置するため以下のような利点問題点が考えられる。

利点

・管理が一元化できる。

問題点

・部署ごとの異なるセキュリティポリシーに対応できない。

・内部からの攻撃に対処できない。

・ウイルスに感染したパソコンの持込による被害拡大に対処出来ない。

集中型ファイアウォールの例を図に示す。

2.1.2 分散型ファイアウォール

集中型ファイアウオールに対して分散型ファイアウオールでは内部ネットワークの部署単位でもファイアウオールを設置するので以下の利点が考えられる。

利点

・部署ごとの異なるセキュリティポリシーが設定可能。

・外部からの攻撃だけでなく、内部からの攻撃に対する防御がくる。

・ウイルスに感染したパソコンの持ち込みによる被害拡大の防止。

分散型では上記のような利点があるため、集中型よりも頑固で柔軟なセキュリティを実現できる。

分散型ファイアウォールの例を図に示す。

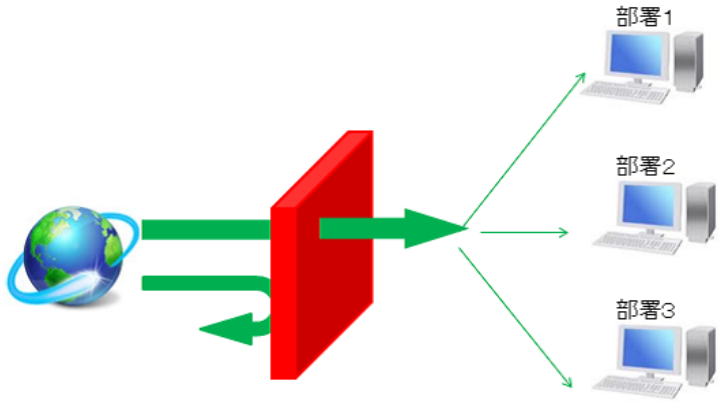

図 2.1: 集中型ファイアウォール

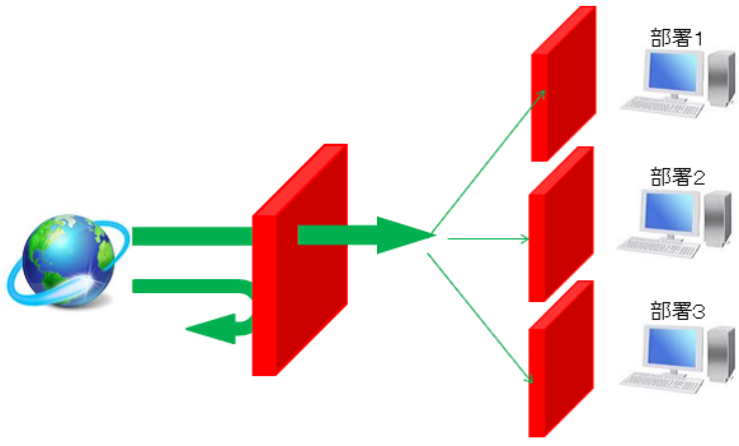

図 2.2: 分散型ファイアウォール

上流層と下流層

ルールの移動は、分散環境下に属するファイアウォール間でのルールの受け渡しにより行う。負荷という観点からマッチング回数に注目し、部署の負荷を軽減するためにルールの移動を行っていた。しかし、部署間での整合性の問題が起こると考えられる。たとえば、下図のように部署 B での負荷が大きいので、マッチング回数の多いルールを部署 C に移動をする。こうすると、負荷は軽減されるが、部署 B のセキュリティポリシーが変わってしまう。この問題を解決するため今回は上流層と下流層でのルールの受け渡しにより整合性の問題を解消する。共通のルールを上流層に持ってくることで、負荷を軽減し、かつセキュリティポリシーも変わらずに済む。

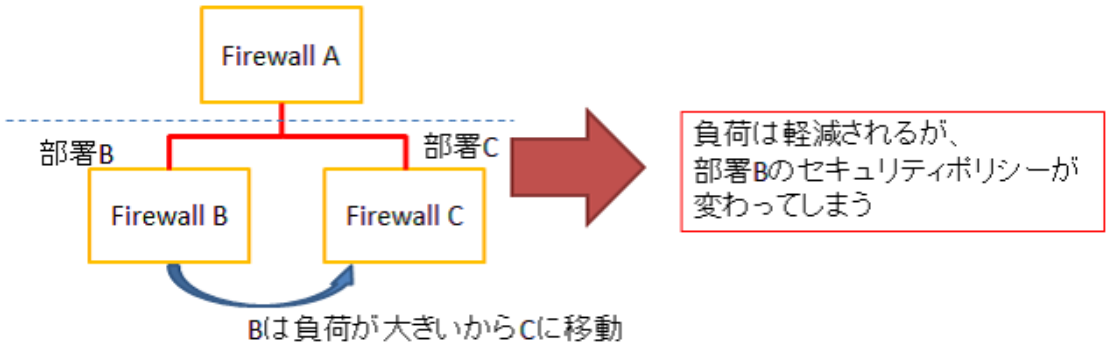

図 2.3: 整合性の問題

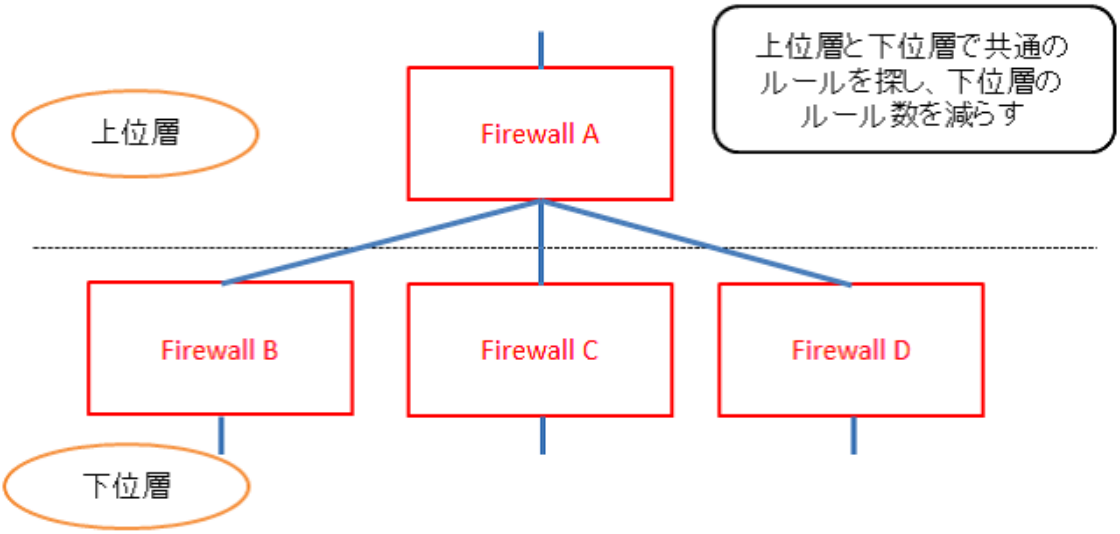

図 2.4: 上流層と下流層

2.2 パケットフィルタリング

IP パケットのヘッダに含まれている情報をもとに，通信を制御するファイアウオールのもっとも基本的な機能となっている.IP ヘッダに書き込まれているプロトコル，送信先 IP アドレス，発信元 IP アドレス，その他オプションの情報を読みとって，ユーザーが指定したルールにのっとってパケットを通過あるいは遮断する.一般的には，明示的に許可されていないアクセスはすべて拒否され・るので、通過させるルールにあてはまらないパケットは，受信しでも破棄されることになる.

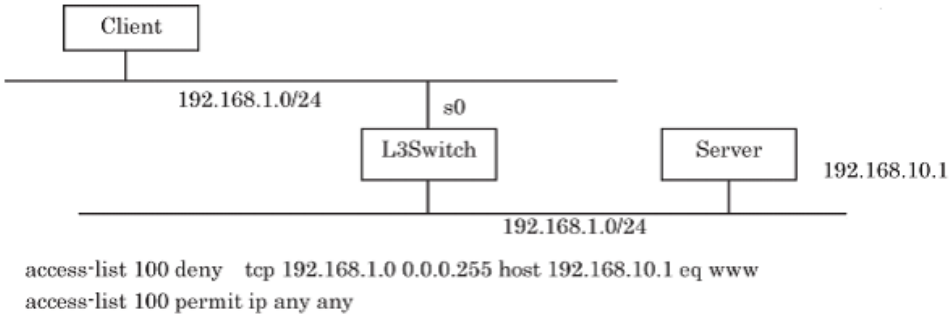

図 2.5: パケットフィルタリング (1)

2.2.1 ルータ

ルータの機能とは、パケットの最終目的地をもとに複数のネットワークインターフェイスの間でパケットを転送することである。通常パケットの転送はカーネルの機能である。パケットフィルタリングは転送の際に特定のパケットのみ通過させ、それ以外のパケットは転送を阻止することによって内部ネットワークを保護する。通過させるかどうかの判断基準をルールリストと呼ぶ。送られて来たパケットの IP ヘッダ情報を元に通信を許可するか、または拒否するかをルールとのマッチングを行い判断する。

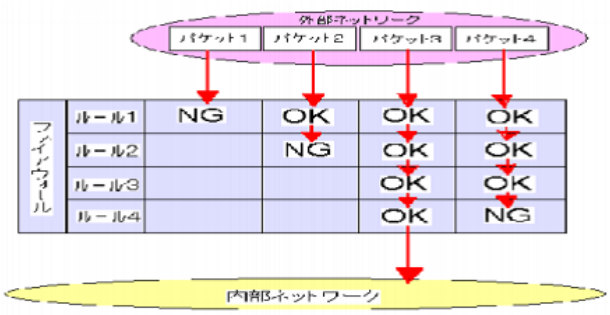

図 2.6: パケットフィルタリング (2)

2.2.2 パケットフィルタリングによる制御

発着 IP アドレス、ポート番号、接続を開始する方向性、プロトコルなどに基づくパケットフィルタリングを行なう。通常、ルータなどの経路制御機能を有する装置に利用する。

2.2.3 パケット

パケットとは、ネットワーク上を流れるデータの小さなまとまりのことです. パケットは，あて先情報などのヘッダー，本体，エラー検出コードの 3 つにより構成されています. ネットワークではうデータをいくつかのパケットに分割して送信し，受信した側で結合作業をすることにより，データの転送中のエラー、， などを防止しています. また，送受信時にエラーが発生してもパケット単位で送受信し直すことが可能である．ここで，本研究で使用するパケットフィルタリングの一つ， IPFW において制御できるプロトコルについて触れておく.

2.2.4 IPFW

ルータに IPFW ルールの設定を行うことでファイアウォールを実現できる。IPFW フィルタリングではパケットごとに、マッチするルールが見つかるまでルールリストを調べる。IPFW のルールの例を図に示す。

```
add pass tcp from any to 133.72.88.10 22
add deny udp from any to any
```

図 2.7: IPFW のルールの例

上記のルールの各コマンドは以下の役割を持つ。

- 操作
  - ・add: ルールを追加する
  - ・delete: ルールを削除する
- 処理方法
  - ・pass: パケットを通過させる
  - ・deny: パケットを破棄する
  - ・count: 通過パケットのカウントをする、制御は行わない

- プロトコル
  - ・tcp: tcp プロトコルのみ制御する
  - ・udp: udp プロトコルのみ制御する
  - ・icmp: icmp プロトコルのみ制御する
  - ・ip: すべてのプロトコルにおいて制御する

2.2.5 パケットフィルタリングのモデル化

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 2 のようにモデル化することができる。図中、Ri 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 Ri のルール番号とよぶ。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あらわす。ルールが与える評価型を明示する際は Ri に評価型を添え、RiA によりパケットの転送を許可するルールを、RiD により転送を拒否するルールをあらわす。

ルータにパケットが到着すると、R1 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 Rn 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。

送信元アドレス、送り先アドレス、ポート番号、プロトコルといった条件をすべてビット列として 1 つの論理式として扱う。式中のーはドントケアである。

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よび、—Ri—と表す。ルールは R1 から順に適用されるので、—Ri—は Ri 以前のすべてのルールに依存する。1 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、一組みのルール R によって決まるフィルタリングの負荷 L(R) は、—Ri(F)—の重み付の総和として以下のように定義できる。

L(R) は、フィルタリングがネットワーク機器に与える負荷を表す。フィルタリングルールの効率を最適化することは、L(R) を最小にするようなルール R を構成する問題と定義できる。この問題をフィルタリングルール最適問題とよぶ。[4][5]

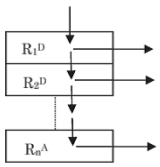

図 2.8: パケットフィルタリングのモデル

2.3 処理能力に影響を及ぼす負荷の要因と改善方法

ファイアウォールの処理能力に影響を及ぼす負荷の要因として、受信パケットのサイズ、送られてきたパケットと設定したルールとのマッチング処理、経路情報の増加、などが考えられる。[2] ファイアウォールのメモリの量や処理能力には限界がある。したがって、負荷の増加は通信速度の低下を招き、その環境下にある部署にも影響を及ぼす。ルール数が増えるとマッチングを行う際に負荷がかかりスループットに影響を与える。すなわち、「ルール数を減らす」＝「負荷の軽減」といえる。

よって負荷を軽減させる方法として以下のことが考えられる。

・操作とプロトコルが同じルールが連続になるように並び替える。

・ルールの統合を行う

・ファイアウォール間でのルールの受け渡し、受け取ったルールの並び替え。

2.3.1 ルール統合

マッチング処理回数を減少させるため、ルールの統合によりルールチェーン内に設定されているルールを削減する。スループットはあるルール数行（K）までは一定のスループットを保つことが確認できるが、（K）行目以上のルールでマッチング処理が行われるとスループットは大きく低下する。ルールの統合を行うことは、ファイアウォールにかかる負荷を軽減するとともにスループット低下の防止に繋がる。

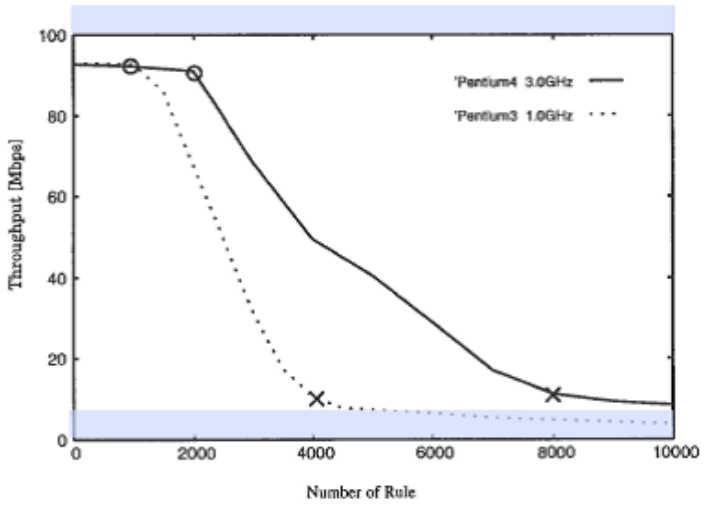

図 2.9: スループットとルール数の関係

2.3.2 Quine-McClaskey の方法

Quine-McClaskey の方法、与えられた論理関数の積和形論理式の最小化を求める方法である。論理式の加法標準形 IP アドレスと考えると、この手法を用いることで IP アドレスを基にルール数を削減できる。[3]

以下に Quine-mcClaskey 法の手順を記す。

1. 論理式を加法標準形で表す。
2. 各最小項 (IP アドレス) を 2 進数表示の 1 の数によってグループにわける.1 の数の同じ項を同一グルーフとし，小さい順に並べる.
3. 1 の数が一つだけ違うグループの項同士を比較し，差が 2 のべき乗の項を取り出す
4. 取り出した項同士で再び比較を行う (比較できなくなった項は主項となる)
5. 全ての項が主項となるまで比較を繰り返す
6. 縦軸に主項，横軸に最小項を記した表を作成する
7. 各最小項を少なくとも一回は含むような主項の組み合わせを求める

図に項の数が 6 個から 3 個に減少した例を示す。

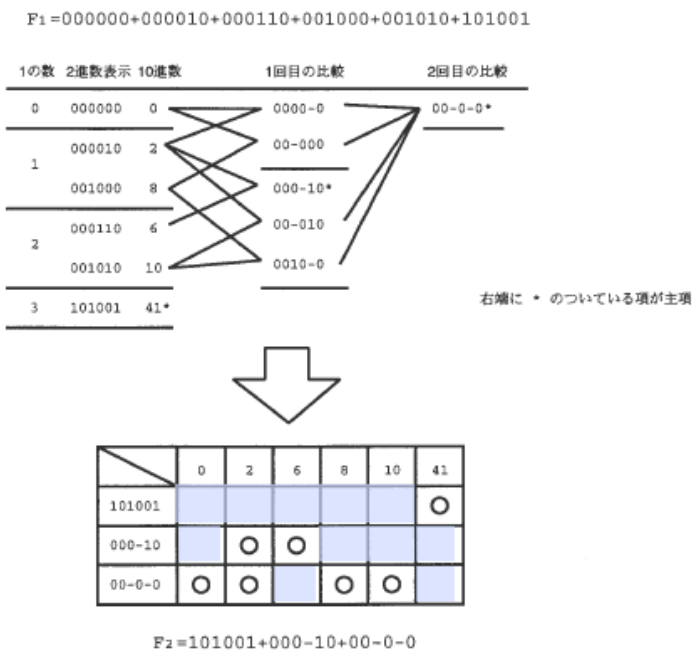

図 2.10: Quine-McClaskey の方法

2.3.3 マトリックス分解によるルール削減

マトリックスとは、ルール集合で作る多次元空間を、各ルールの各条件属性の範囲指定されたすべての境界点で区割りしてできる最小の多次元立方体である。ルール集合からマトリックスを作成し、ルールとマトリックス対応付けを行うことをマトリックス分解と呼ぶ。[7]
下図に 8 個のルールを持つルール集合の例とそれを 2 次元平面で表現する。優先順位は R 1 が最も高く、R 8 がデフォルトルールである。ルールの書式は（第 1 属性の範囲、第 2 属性の範囲、アクション）とする。

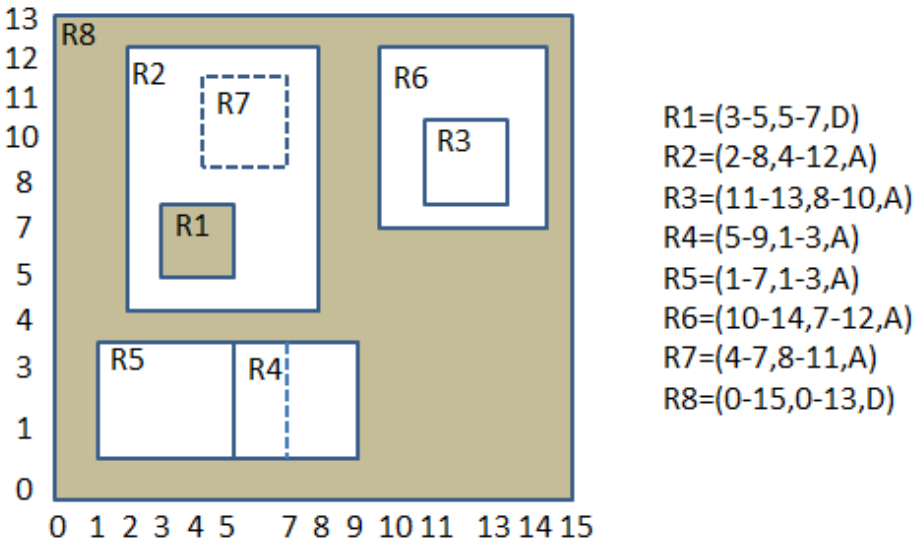

図 2.11: ルール集合の例

これにより様々なルール分析が可能になる。時間の経過とともにルール数が増加したルール集合から、不要なルールおよび冗長な条件をもつルールを検出するルール分析について述べる。このようなルールを検出・削除することで、ルール数を削減させ、ルール集合のメンテナンス性を向上させることが可能になる。

2.3.4 特定、不特定の場合の負荷の軽減

firewall を 4 台用意し 3 台を並列に、もう 1 台を 3 台に対して縦列に接続することで分散型ファイアウォールを実現する。

・特定の場合

セキュリティポリシーが a〜m までの計 13 個のセキュリティポリシーが下図のように各ファイアウォールに設置されていた場合、ファイアウォール B,C,D はどれも同じ a,d というセキュリティポシリーを持つことになる。この a,d というセキュリティポシリーをファイアウォール A に設定すれば 3 台別々に行う必要がなくなる。個々の firewall の負担を減らすことでシステム全体としての負荷の軽減が実現される。

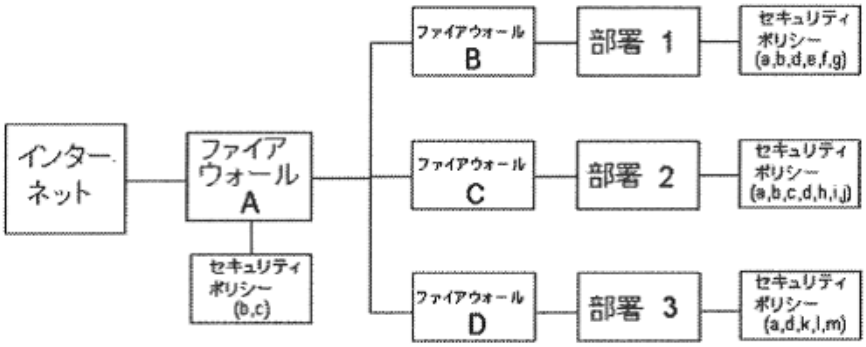

図 2.12: 実装モデル（最適化前）

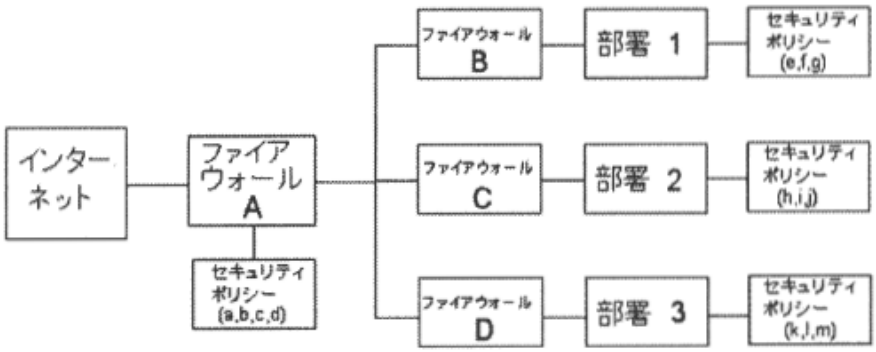

図 2.13: 実装モデル（最適化後）

・不特定の場合

特定の場合のように環境分散下のファイアウォールに共通のセキュリティポリシーがあれば、統合しまとめることができる。しかし、実際のネットワーク上での動作では、さらに複雑なネットワークになるため、セキュリティポリシーの数も増し、あらゆる攻撃に対してせいぎょできるようにしなければならない。さまざまなセキュリティポリシーを持った分散型のファイアウォールでも、ファイアウォールの連携による負荷の軽減が見込めるようなシステムの検討が必要である。

第3章　提案システム

3.1　提案システム

ファイアウォールの処理能力に影響を及ぼす負荷の要因として、受信パケットのサイズ、送られてきたパケットと設定したルールとのマッチング処理、経路情報の増加、などが考えられる。本稿ではルールのマッチング処理に注目した。各ファイアウォールに設定するルール数を減らすことにより、ルールのマッチングの最適化に努め、ファイアウォールのメモリ使用量の軽減、パケットが通過する際の遅延時間減少が見込める。分散環境下にあるファイアウォールとその上にあるファイアウォールとの間でルールの受け渡しをすることによって、各ファイアウォールでのルールのマッチング回数を減少させる事ができる。ルール数とスループットは反比例の関係にあるため、ルール数が減少することで、各ファイアウォールのマッチング回数が減り、負荷の軽減につながる。

ルール統合の手順

部署ごとのセキュリティポリシーを集める。

集めたセキュリティポリシーに基づいてルールの統合し最適化を行う。

最適化の論理式に基づいて各ファイアウォールにルールを返し、設定する。

部署ごとのセキュリティポリシーに変更があった場合は　～　をやり直す。

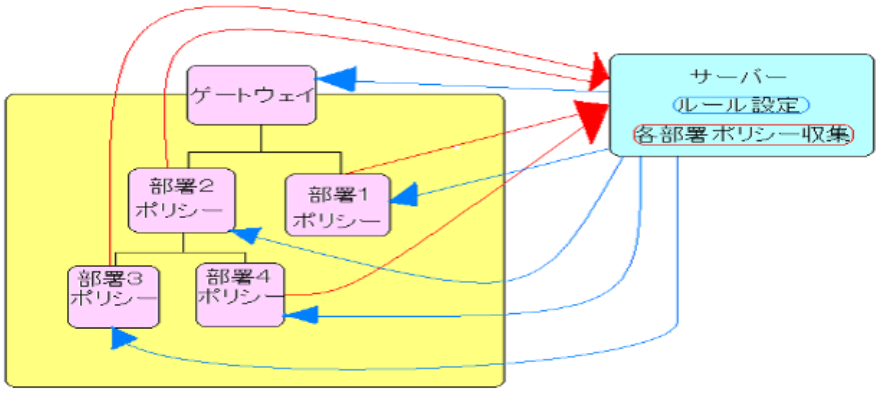

図 3.1: システムモデル

3.2 システム詳細

本研究では3. 1で述べたような手順が自動的に行われるようプログラムに書いて実装する。まず、分散環境下にあるファイアウォールにそれぞれのセキュリティポリシー（ルール）を設定する。プロトコルやポート番号に応じてどのようなアクションをとるか記入し、次にセキュリティポリシーを自動で回収できるようにさせるために、すべてのファイアウォールに共通のカギを持たせて認証なしでアクセスできるようにする。このような流れで回収したカクファイアウォールのセキュリティポリシーをプログラムに読み込ませる。このプログラムでは、プロトコルやポート番号やアクションの違いに応じてルールを並び替える動作を行う。これによってプロトコルやポート番号やアクションからなるルールが似たものから順番に並び替わることになる。並び替わったルールの上下を比べることで、同じルール、同意義のルールを見つけられる。ここでルール統合の処理を行ってルール数自体を減らす。統合されたルールと統合されなかったルールを最適化の論理式に基づいて各ファイアウオールに返し、設定する。上記のようにして分散ファイアウオールの連携を実装する。

3.3 実装実験

仮想マシンを用いた実装実験により、分散型ファイアウォールにおいて、各ファイアウォール間での連携を持たせることでルールの最適化を実現する。最適化前と最適化後の処理速度の計測を行い、負荷の軽減が実現されたか示す。

3.4 仮想マシン

電子情報実験 C(「経路制御と防火壁」の実験) で使われている仮想マシンを参考に仮想マシンを実現する。ホストOS（ゲストOS）をLinuxのUbuntu13.04でインストールし、gateway ruter 1台、host 3台、計4台を実行する。外部ネットワークとgatewayをNATで接続し、gatewayと各々のhostはホストオンリーネットワークで接続する。それぞれのgateway,hostで設定されているファイアウォールのパケットフィルタリングルールを削減する手順を、自動的に行われるプログラミングを書いて実装する。[8]

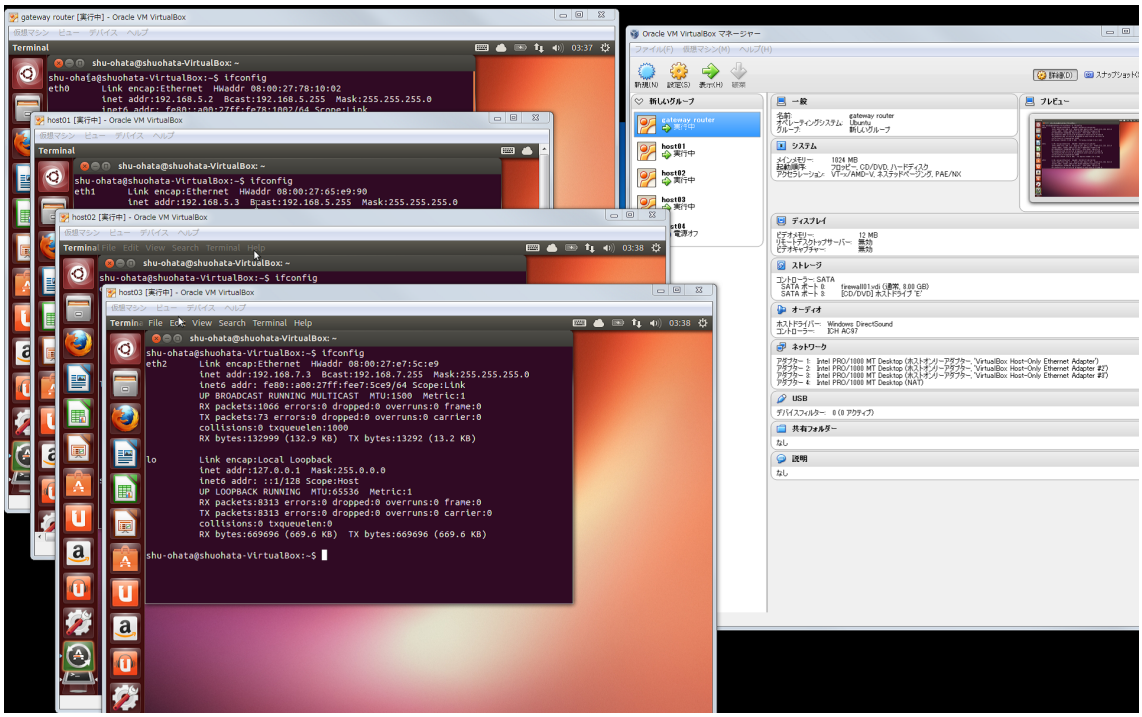

図 3.2: 仮想マシンモデル

第４章　質疑応答

Q：上流層と下流層の共通のルールをみつけた場合、なぜその共通のルールを上流層に持ってこなければいけないのか。下流層ではダメなのか。
A：分散環境下において下流層内で共通のルールは上流層に持っていって良いが、上流層と下流層のみの共通のルールの場合、実験をしてみて検討する必要がある。

参考文献

[1] 国崎大地，:”分散ファイアウォールの連携”，

[2] 田中賢，:　”ファイアウォールにおけるパケットフィルタリングの最適化”，

[3] 宮地信晴，:”分散ファイアウォールにおけるルールの配置と処理の最適化”，

[4] 宝木和夫，小泉稔，寺田真敏，萱島信 :”ファイアウォール　-インターネット関連技術について-，昭晃堂（1998）

[5] 石渡智明，:”分散ファイアウォールの最適化”，

[6] 山崎朗，:”パケットフィルタリングのルールによる処理速度の向上”，

[7] 松田勝志 :”マトリックス分解によるパケットフィルタリングルール分析”，社団法人　情報処理学会　研究報告

[8] ”Ubuntuの日本語環境　インストール”，http://www.ubuntulinux.jp/japanese