パケットフィルタにおける ルールの最適配置

宇都宮大学

○井上 裕司　　趙 亮　　山本 英雄

YAMAMOTO LAB.
Utsunomiya University

発表の構成

- ○はじめに
- ○パケットフィルタ
- ○提案方法
  - ルール再配置
  - デフォルトルールからの新ルールの作成
- ○検証実験
  - 実験方法
  - 実験結果と考察
- ○おわりに

1.はじめに

研究の背景

インターネットが一般的な通信手段となり

セキュリティ対策が必須

→ 不正なアクセスを防止できる
パケットフィルタを使用

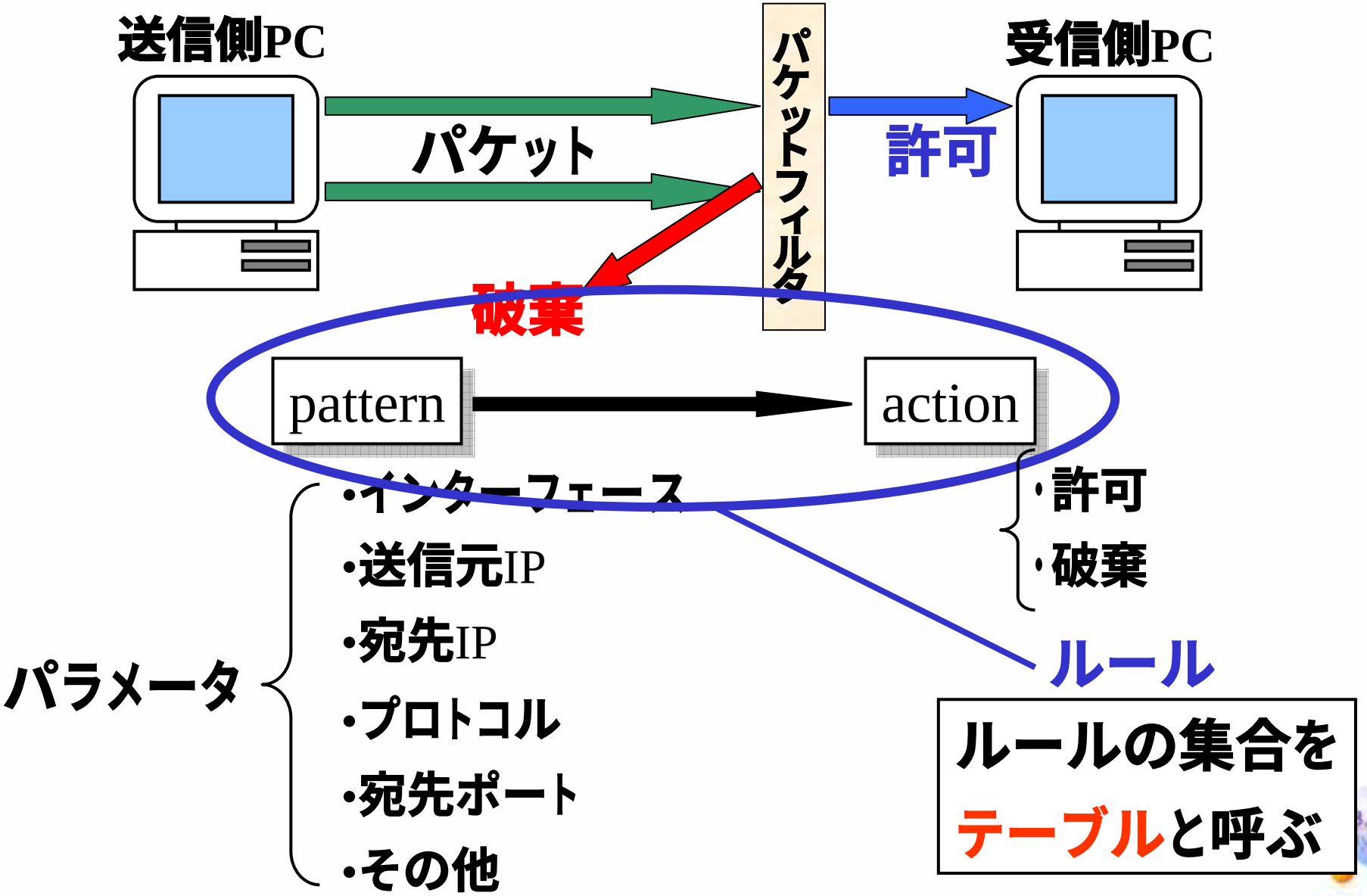
問題点：パケットそれぞれに処理を行うため
遅延時間が発生する

研究の目的

ルールの最適配置により遅延時間の短縮を図る

→
- **ルールの再配置**
- **デフォルトルールからの新ルールの追加**

2.パケットフィルタ


送信側PC
パケット
パケットフィルタ
許可
受信側PC
破棄
pattern
action
・インターフェース
・送信元IP
・宛先IP
・プロトコル
・宛先ポート
・その他
パラメータ
・許可
・破棄
ルール
ルールの集合を
テーブルと呼ぶ

3.パケットフィルタの動作

	送信元IP	プロトコル	宛先ポート	・・・	処理
Rule1	192.168.12.131	tcp	7066		許可
Rule2	ANY	tcp	433		破棄
Rule3	ANY	udp	54		破棄
・ ・ ・					

通常パケットはテーブルの上から順に比較される

IP192.168.34.61ポート54宛て
UDPパケット

Rule1 ×
Rule1に従って許可
Rule2 ×
Rule3に従って破棄

テーブルの下方で処理されるパケットほど比較回数が多い
＝よく使われるルールがテーブルの下にあると
平均の比較回数が多くなる（処理時間が長い）

4.ルール再配置

よく使うルールほどテーブルの上にあれば
平均的な比較回数は少なくなる

ルールごとの使用頻度Pを調べ

$$P_1 \geq P_2 \geq P_3 \geq \cdots \geq P_n$$

を満たすテーブルになるように
再配置を行えばよい

問題点

パケットフィルタの動作が変わってしまう場合がある

5.パケットフィルタ動作の変更

	送信元IP	プロトコル	宛先ポート	処理
Rule1	192.168.12.131	tcp	7066	許可
Rule2	ANY	tcp	7066	破棄

使用頻度：Rule2>Rule1

ルールの使用頻度降順に並べると

	送信元IP	プロトコル	宛先ポート	処理
Rule2	ANY	tcp	7066	破棄
Rule1	192.168.12.131	tcp	7066	許可

送信元IPが192.168.12.131**プロトコルが**tcp
宛先ポートが7066**のパケットも破棄してしまう**

6.動作が変わらない並び替え 1-ACTION

	プロトコル	宛先ポート	・・・	処理
Rule1	tcp	8080	・・・	許可
Rule2	ANY	8080	・・・	許可

	プロトコル	宛先ポート	・・・	処理
Rule2	ANY	8080	・・・	許可
Rule1	tcp	8080	・・・	許可

基準① パケットへの処理(ACTION)が同じ場合交換可能

7.動作が変わらない並び替え 2-PROTOCOL

ACTIONが異なる場合

	プロトコル	宛先ポート	・・・	処理
Rule1	tcp	8080	・・・	許可
Rule2	ANY	8080	・・・	破棄

	プロトコル	宛先ポート	・・・	処理
Rule2	ANY	8080	・・・	破棄
Rule1	tcp	8080	・・・	許可

	プロトコル	宛先ポート	・・・	処理
Rule1	tcp	8080	・・・	許可
Rule2	udp	8080	・・・	破棄

	プロトコル	宛先ポート	・・・	処理
Rule2	udp	8080	・・・	破棄
Rule1	tcp	8080	・・・	許可

基準② 使用プロトコルが重複しない場合は交換可能

8.動作が変わらない並び替え 3-DPORT

ACTIONが異なり、プロトコルが重複している場合

	プロトコル	宛先ポート	・・・	処理
Rule1	tcp	8080	・・・	許可
Rule2	ANY	8080	・・・	破棄

	プロトコル	宛先ポート	・・・	処理
Rule2	ANY	8080	・・・	破棄
Rule1	tcp	8080	・・・	許可

	プロトコル	宛先ポート	・・・	処理
Rule1	tcp	8080	・・・	許可
Rule2	tcp	1080	・・・	破棄

	プロトコル	宛先ポート	・・・	処理
Rule2	tcp	1080	・・・	破棄
Rule1	tcp	8080	・・・	許可

基準③ 宛先ポート番号が重複しない場合は交換可能

9.動作が変わらない並び替え 4-まとめ

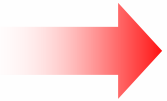
ACTIONが異なり、プロトコルと宛先ポート番号が重複している場合

	プロトコル	宛先ポート	・・・	処理
Rule1	tcp	8080	・・・	許可
Rule2	tcp	ANY	・・・	破棄

	プロトコル	宛先ポート	・・・	処理
Rule2	tcp	ANY	・・・	破棄
Rule1	tcp	8080	・・・	許可

基準① パケットへの処理(ACTION)が同じ場合交換可能
基準② 使用プロトコルが重複しない場合は交換可能
基準③ 宛先ポート番号が重複しない場合は交換可能

この3つで交換できなかった場合は交換しないことにする



送信元IP,宛先IPなどの他のパラメータについても
比較を行えば交換可能な場合もある

10.ルール交換のアルゴリズム

ルールの使用頻度：Rule5>Rule4>Rule3>Rule2>Rule1

	送信元IP	プロトコル	宛先ポート	処理
Rule4	192.168.12.254	tcp	420	破棄
Rule3	ANY	udp	54	破棄
Rule2	ANY	tcp	433	破棄
Rule1	192.168.12.131	tcp	7066	許可
Rule5	ANY	tcp	7066	破棄
default_Rule	ANY	ANY	ANY	破棄

前提　　ルールの使用頻度を比べる・・・頻度が高いものを上に配置
基準①　パケットに対する処理を比較する・・・同じならば交換
基準②　使用プロトコルを比較する・・・重複していなければ交換
基準③　宛先ポート番号を比較する・・・重複していなければ交換
以上で交換できない場合は交換しない

11.デフォルトルールとは

	送信元IP	プロトコル	宛先ポート	処理
Rule1	192.168.12.131	tcp	7066	許可
Rule2	ANY	tcp	433	許可
Rule3	ANY	udp	54	破棄
Rule4	192.168.12.254	tcp	420	破棄
Rule5	ANY	tcp	7066	破棄
default_Rule	ANY	ANY	ANY	破棄

デフォルトルールとは設定したルールに当てはまらない
パターンのパケットを処理するためのルール

デフォルトルールで処理されるパケットは
比較回数が多い(=処理時間が長い)

12.デフォルトルールからの新ルール作成-1

デフォルトルールによって
処理されたパケットの
宛先ポート番号について調べる

port1は N_1 回
port2は N_2 回
.
.
.
port(i)は N_i 回
.
.
.
port65535は N_{65535} 回

一番回数が多かった
宛先ポート番号を
パラメータとしたルールを
作成する

例) ポート6900が一番多かった
→ ポート6900宛てのパケットを破棄するルールを作る

13.新ルールの追加位置

1.デフォルトルールのすぐ上に追加
2.ルール位置交換のアルゴリズムに従って並び替える

パケットフィルタの動作を変えないで追加可能

	送信元IP	プロトコル	宛先ポート	・・・	処理
Rule1	・・・	・・・	・・・	・・・	・・・
Rule2	・・・	・・・	・・・	・・・	・・・
Rule3	・・・	・・・	・・・	・・・	・・・
Rule4	・・・	・・・	・・・	・・・	・・・
・・・	・・・	・・・	・・・	・・・	・・・
~~Rule_NEW~~	~~ANY~~	~~ANY~~	~~6900~~	~~ANY~~	~~破棄~~
default_Rule	ANY	ANY	ANY	ANY	破棄

位置が変更されなかったら削除する

YAMAMOTO LAB.
Utsunomiya University

14.新ルールの追加例

	プロトコル	宛先ポート	・・・	処理
Rule1	tcp	80	・・・	許可
Rule2	tcp	433	・・・	許可
default_Rule	ANY	ANY	ANY	破棄

Rule1の使用回数1000回
Rule2の使用回数500回
デフォルトで一番多く処理された
宛先ポート番号520で700回
その他の宛先ポートが計50回

総処理回数
1000+500×2+(700+50)×3
＝4250　(回)

	プロトコル	宛先ポート	・・・	処理	使用回数
Rule1	tcp	80	・・・	許可	1000
Rule_NEW	ANY	520	ANY	破棄	700
Rule2	tcp	433	・・・	許可	500
default_Rule	ANY	ANY	ANY	破棄	50

1000+700×2+500×3+50×4＝4100　(回)

15.新ルール追加をしない場合

	プロトコル	宛先ポート	・・・	処理
Rule1	tcp	80	・・・	許可
Rule2	tcp	433	・・・	許可
default_Rule	ANY	ANY	ANY	破棄

Rule1の使用回数1000回
Rule2の使用回数700回
デフォルトで一番多く処理された
宛先ポート番号520で800回
その他の宛先ポートが計600回

総処理回数
1000+700×2+(800+600)×3
＝6600　(回)

	プロトコル	宛先ポート	・・・	処理	使用回数
Rule1	tcp	80	・・・	許可	1000
~~Rule_NEW~~	~~ANY~~	~~520~~	~~ANY~~	~~破棄~~	~~800~~
Rule2	tcp	433	・・・	許可	700
default_Rule	ANY	ANY	ANY	破棄	600

総処理回数が増える場合は新ルールを追加しない

16.検証実験

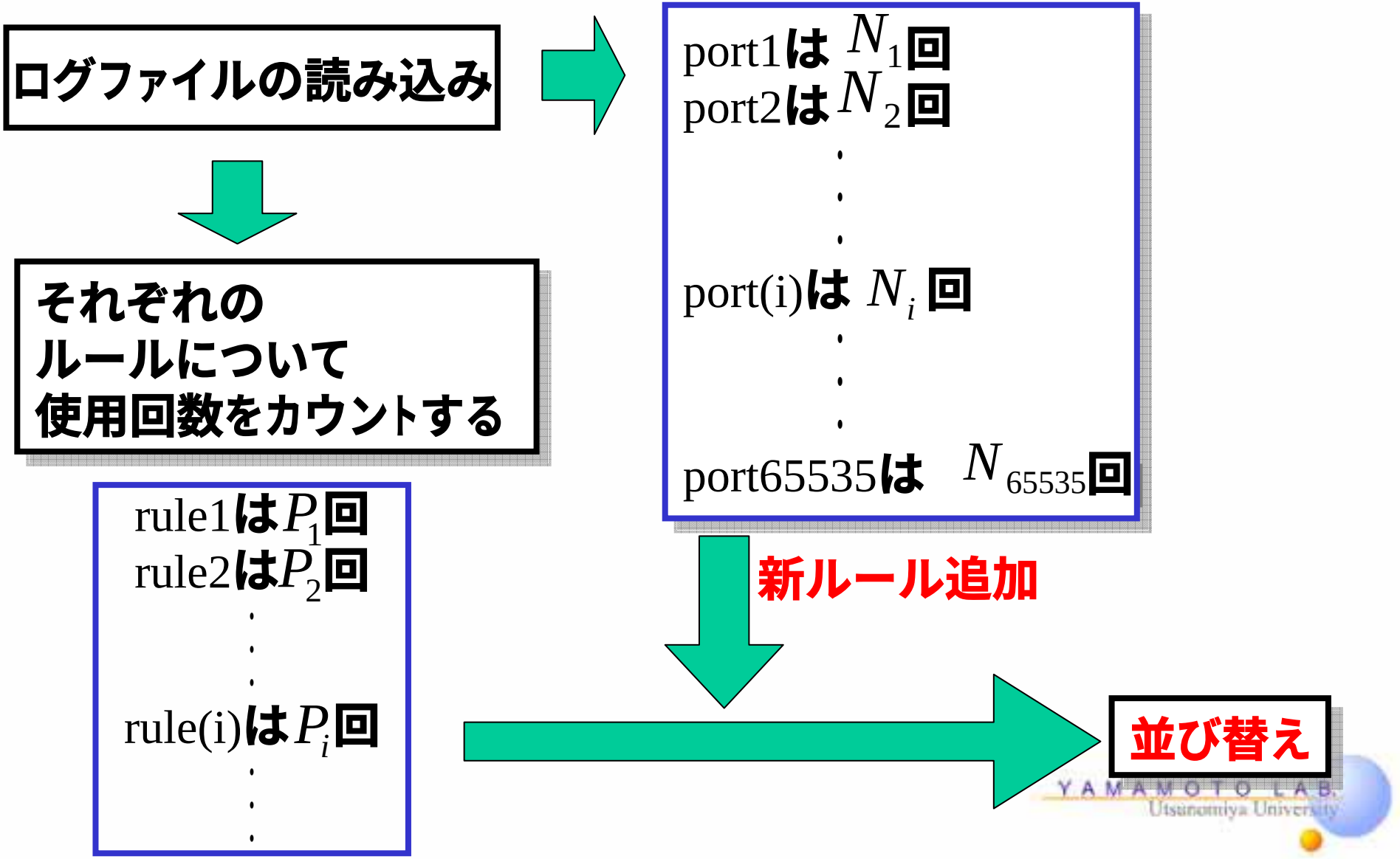
目的

提案方法により処理回数がどれくらい
削減できるのか検証する

実験方法

1. 実際のテーブル(ルール数53個)と
   ログファイル(総パケット数165552個)をもとに
   提案方法によって新しいテーブルを作成する
2. 2つのテーブルに対しログと同一のパケットを
   通したときの処理回数を算出して比較する

17.実験の流れ



18.実験結果と考察

	もとのテーブル	提案方法で作成したテーブル
処理回数合計 (回)	1,151,583	528,709

総処理回数が約54%削減できた

提案方法を用いることで
遅延時間を大幅に短縮することが可能

19.おわりに

まとめ

ルールの再配置とデフォルトからの新ルールの追加を行った

→ パケットフィルタの遅延時間を短縮できた

今後の課題

- テーブル内の冗長性の削除
- ルール再配置の際にルールごとの処理時間の違いについての考慮

パケットフィルタにおける ルールの最適配置

宇都宮大学

○井上 裕司　　趙 亮　　山本 英雄

YAMAMOTO LAB.
Utsunomiya University