L-032

パケットフィルタリング最適化法の有効性について

On Efficiency of Packet Filtering Optimization

野村 圭太* 田中 賢* 三河 賢治†
Keita Nomura Ken Tanaka Kenji Mikawa

1 はじめに

ネットワーク機器の機能の一つであるパケットフィルタリングは，悪意のあるユーザからのアクセスを防ぎ，適切なユーザだけがネットワーク上のリソースへアクセスできるようにトラフィックを制御する．しかし，パケットフィルタリングのルールが増加すると，ネットワークの遅延の原因となる．ここでは，線形探索によってルールの探索を行う iptables を実装した Linux マシンを用いて，パケットフィルタリングによる遅延時間の計測環境を構築する．構築した環境で，昆金 [2] により提案された従属ルールを含むパケットフィルタの再構成法，嶋 [3] により提案されたルール移動による最適配置法の 2 つの手法の有効性を検証する．

2 iptables と rsyslog を用いた遅延計測環境

ルーティングマシンとして設定した Linux マシン 1 台と 2 台のマシンで小規模なネットワークを構築し，フィルタリングによる純粋な遅延時間を計測するため，kernel のログを利用し，パケットがルールを通過した時刻から遅延時間を計測する環境を構築した．ログ出力に μ 秒単位時刻を出力することのできる rsyslog を使用する．MAWI Working Group Traffic Archive[4] からダウンロードしたパケットキャプチャデータを tcpreplay を使用することにより，実際のネットワークトラフィックに近いものを再現する．

2.1 動作環境

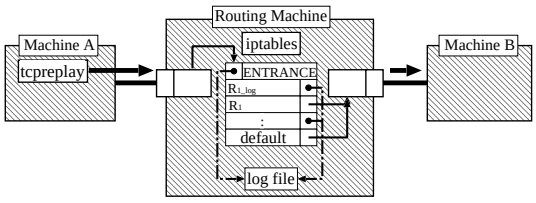

図 1: iptables と rsyslog を用いた遅延計測環境

表 1: マシンスペック

	ルーティングマシン
カーネル	Linux 2.6.32-71.el6.x86-64
CPU	Intel(R)Core(TM)i7 X980 3.33GHz
メモリ	23.6GiB
OS	CentOS 6.0

表 2: Machine A,B のスペック

CPU	Core i5-2400 クロック周波数：3100MHz
メモリ	3.6GB
OS	CentOS 6.2 64bit

2.2 測定の手順

Routing Machine 上で rsyslog が log を出力するように設定する．具体的には rsyslog.conf に kern.debug 任意のファイルの絶対パス RSYSLOG_FileFormat を追記し，iptables の各ルールの前に出力する記述をする．

ルールセットの先頭と各ルールの前にログを出力するルールを配置し，パケットがルールセットの先頭を通過する時刻と各ルールを通過する時刻の差を取り，遅延時間を測定する．

3 従属ルールを含むパケットフィルタの再構成法の検証

検証実験は，第 2 章のネットワーク構成で行った．ダウンロードしたパケットキャプチャデータからルールセットを作成し，iptables-restore コマンドでフィルタリングルールを適用する．さらにそのデータからソースアドレスを抽出し，そのソースアドレス当てのパケットは全て Machine A 側のインタフェースへルーティングするように設定し，その他のパケットは Machine B 側へルーティングするように設定する．MAWI Working Group Traffic Archive のデータは，tcpdpriv と呼ばれるツールによって，パケットから重要なデータを隠しているので，アドレスが書き換えられ，ペイロードがカットされているため，そのままのデータでは使用できない．そのため，tcprewrite によってキャプチャデータの送信元 MAC アドレスと宛先 MAC アドレスをそれぞれ直近の NIC の MAC アドレスに書き換え，パケットヘッダのペイロード長と実際のペイロード長が一致するように不足部分を補完する．

*神奈川大学大学院理学研究科情報科学専攻
†新潟大学学術情報基盤機構情報基盤センター

従属ルールを含むルールセットの最適化前と最適化後の計測結果は表 3 に示す．

表 3: 計測結果

ルール数	最適化前遅延時間 [sec]	最適化後遅延時間 [sec]
1000	3.713026	3.666698
2000	4.080066	3.881571
3000	3.989356	3.963918
4000	4.252231	4.106899
5000	4.333327	4.128496
6000	4.393597	4.197069
7000	4.841843	4.734901
8000	5.035185	4.760993
9000	5.354459	4.855636
10000	6.150942	5.449980

結果より，最適化アルゴリズムはパケットフィルタリングによる遅延時間を減少させるために有効だといえる．

4 ルール移動による最適配置法

検証実験を行うネットワーク環境は，以下の図 2 のようになっている．今回の検証実験は図 2 に示すようなネットワーク環境を用いて，図 3 と表 4 に示すような動作環境で行う．

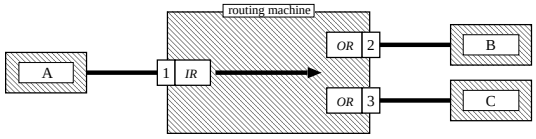

図 2: 実装実験を行うネットワーク環境

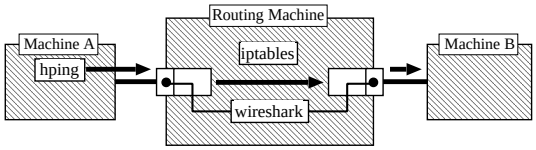

図 3: ルーティングマシンと iptables を用いた環境

表 4: マシンスペック

	ルーティングマシン
カーネル	Linux 2.6.32-71.el6.x86-64
CPU	Intel(R)Core(TM)i7 X980 3.33GHz
メモリ	23.6GiB
OS	CentOS 6.0

図 2 の A，B，C は 3 台のクライアントマシンであり，中央はルーティングマシンである．実装する各ルールセットにおける提案アルゴリズム適用前後のルール数を表 5 に示す．アルゴリズム適用前は，IR^1 に評価型が許可，拒否のルールをそれぞれ含む 300 個のルールを適用する．アルゴリズム適用後は，適用前の 300 個のルールのうち拒否ルールを OR^2 と OR^3 に 100 個ずつ移動する．

表 5: 適用前後のルール数

	IR^1	OR^2	OR^3
適用前のルール数	300	–	–
適用後のルール数	100	100	100

各クライアントへの送信パケット数と，提案アルゴリズム適用前後のフィルタリングの遅延を以下の表 5 に示す．いくつかのルールに合致するようなパケットを，クライアント A から B，C に向けて 1000 個ずつ送信する．計 2000 個の送信パケットはアルゴリズム適用前後で同様のパケットを送信する．パケットロスが発生しないように 1msec 間隔でパケットを送信する．

表 6: 送信パケット数と遅延

	A→B	A→C	合計
送信パケット数	1000	1000	2000
適用前の遅延（μsec）	8573	8901	17474
適用後の遅延（μsec）	7835	8048	15883

提案アルゴリズムによって，フィルタリングの遅延は 17474μsec から 15883μsec になり，約 10%の遅延削減を確認できた．

5 おわりに

昆金 [2] により提案された従属ルールを含むパケットフィルタの再構成法と嶋 [3] により提案されたルール移動による最適配置法の有効性について検証実験を行った．今後は，他の提案された最適化法の有効性について検証する必要がある．また，提案された種々の最適化法の現実のパケットフィルタリングにおける有効性を示すため，Spirent Test Center[5] など大規模なネットワークをシミュレートできる機器で検証を行う必要がある．また，実際のネットワーク環境で実測を行う必要がある．

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 (B),vol.J88-B,No.5,pp.905-912,May.2005.

[2] Ken Tanaka,Kenji Mikawa and Manabu Hikin,”Heuristic Algorithm for Reconstructing a Packet Filter with Dependent Rules”, IEICE Trans.,vol.E96-B,No.01,pp.155-162,Jan.2013.

[3] 嶋良平, 田中賢, 三河賢治,”フィルタリングルール最適配置問題の解法”, 第 10 回情報科学技術フォーラム,pp175-176,2011.

[4] ”MAWI Working Group Traffic Archive”,http://mawi.wide.ad.jp/mawi/

[5] ”Spirent Test Center”,http://www.toyo.co.jp/spirenttestcenter/