C-028

ネットワーク IDS 向けの Bloom Filter と偽陽性検出回路を用いたパターンマッチング回路
Pattern Matching Using Bloom Filter and False Positive Detection Circuit for IDS

川本美穂 † Miho Kawamoto
小柳滋 ‡ Shigeru Oyanagi

1 はじめに

インターネットを利用する人口の増加に伴い，セキュリティの問題が多く取り上げられている．対応策として，侵入検知システムがある．ソフトウェアによるネットワーク不正侵入検知システムでは，高速化するインターネットに検出のマッチング処理が追いつかないといった事態が発生している．そこで，マッチング処理部をハードウェアで実装し，処理性能を向上する研究が行われている．

2 既存研究

2.1Bloom Filter

Bloom Filter は，複数のハッシュ関数を用いることにより，少ないハードウェアでハッシュ値の衝突を減らすための手法である．BloomFilter の構成には，m ビットの配列と，k 個のハッシュ関数を用いる．ハッシュ関数の出力を m ビット配列のアドレスとし，配列が全て 1 だった場合，match とし，k 個のうち少なくともどこか 1 つに 0 がある場合，unmatch を出力する．しかし，ハッシュ値が衝突し安全なパターンを危険と判断してしまう偽陽性を持つ．

2.2NFA(Nondeterministic Finite Automaton)

NFA の構成には 1 クロックに複数の遷移ステートを持つステートマシンを使用する．全てのルール 1 つ 1 つに対応するステートマシンを生成し，それぞれのステートマシンが独立して処理を行うことにより高速化が可能である．[2] の研究では，ルールを正規表現で表し，それに対応する NFA 回路を構築する方法を利用している．

2.3CAM(Content Addressable Memory)

CAM とは，登録されている特定の値を内容検索することにより高速に読み出すことのできるメモリである．CAM に関する研究では，様々な構成法が提案されている．

2.3.1 比較器を用いた CAM

比較器を用いた CAM は，CAM の入力データと登録パターンのマッチング部に単純な比較器を用いて実現される．この比較器を用いた CAM は複数パターンの文字列のマッチングに対応しており，1 クロックで動作する．

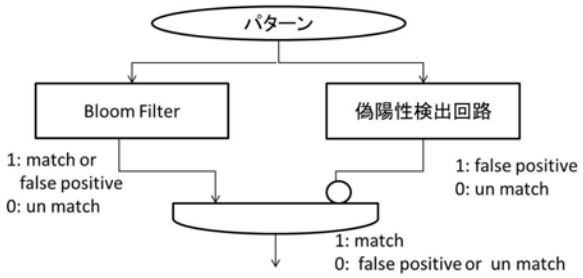

図 1: 提案手法

2.3.2LUT カスケード

LUT(Look-up Table) とは，RAM などのメモリにあらかじめ計算した値を書き込んでおき，その値を必要に応じて参照するためのテーブルである．LUT カスケードとは，複数の LUT を連続して接続したものである．複数の LUT を連続してつなげることによって，少ない回路規模でパターンマッチングを実装可能である．[3] では関数分解によって実現している．

3 提案手法

Bloom Filter でのパターンマッチング回路構成には，ハッシュ関数と RAM を用いる．比較器を用いないため他の手法と比べて少ないハードウェア量で実現できると考えられる．

本研究では，Bloom Filter の短所である偽陽性をなくすために Bloom Filter でパターンマッチングを行うと同時に，偽陽性検出回路でもパターンマッチングを行うことを提案する．

Bloom Filter には，ルールセットに記述されているシグネチャを全て登録する．偽陽性検出回路には，安全ではあるが Bloom Filter でパターンマッチングを行うと偽陽性を起こすパターンのみを登録する．Bloom Filter でパターンが見つかる，かつ偽陽性検出回路でパターンが見つからないといった場合，本当に危険なパターンを検出したことになる．

偽陽性検出回路のアルゴリズムに，比較器を用いた CAM，NFA，LUT カスケードを用いる回路を実装した．しかし LUT カスケードのみで実装した場合，入力数が多くなってしまうので LUT カスケードに NFA を組み合わせたものを実装した．それぞれ，どのパターンマッチングアルゴリズムが適しているかを回路規模と動作周波数から検証を行う．

偽陽性検出回路に必要な容量を求めるため，Bloom

†立命館大学大学院情報理工学研究科
‡立命館大学

表 1: Bloom Filter の偽陽性

n	m	k	偽陽性
1000	16000	10	0.016
10000	160000	10	0.016
100000	1600000	10	0.016

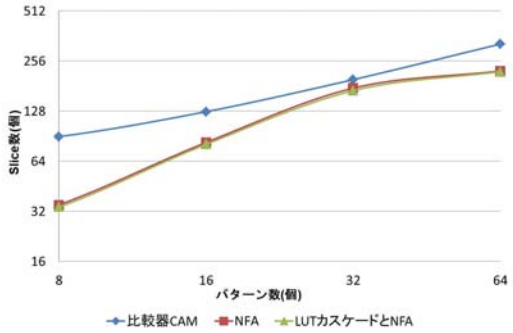

図 2: 各手法の Slice 数

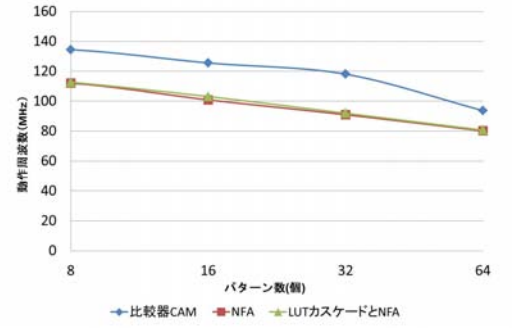

図 3: 各手法の動作周波数

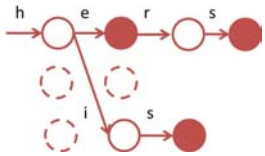

図 4: ステートの共有化

Filter における偽陽性の発生する確率を以下に示す．k 個のハッシュ関数を用いて，n 個の要素を Bloom Filter に追加した後，n+1 個目の要素の m ビット配列の値が 1 である確率，つまり偽陽性である確率 f は式 (1) で表される．表 1 に，m と n を変化させたときの偽陽性発生確率を示す．

$$f = (1 - (1 - \frac{1}{m})^{kn})^k \approx (1 - e^{-nk/m})^k \qquad (1)$$

また，(2) から偽陽性を起こす要素の数 N が求められる．

$$N \quad = n \times f = n(1 - e^{-nk/m})^k \qquad (2)$$

4 評価

Xilinx 社の ISE10.1 を使用し各アルゴリズムのパターンマッチング回路を実装して，シュミレータとして ModelSim XE III 6.3c を利用して回路規模，処理速度の検証を行う．回路規模は Slice の値を参照し，処理速度は周波数の数値を参照する．

図 2，3 より，比較器を用いた CAM が他の 2 つより高速であるが，回路規模は増大する．

5 ハードウェアの削減

NFA と，LUT カスケードの NFA 部はステートを共有化することによりさらに回路規模の削減が可能である．例えば”he”，”hers”，”his”という 3 つのマッチングパターンがあった場合，共通する接頭辞 ”h ”，”he ” は同じ状態を表しているので図 4 のようにステート数を 9 から 6 に削減できる．実用場面においてどの程度のステート数の削減が可能かを検証するため，トライ木を用いて接頭辞の共有化を行うプログラムを作成し，Snort 2 .9 のルール 1000 個を用いて実験した．結果を表 2 に示す．表 2 より，接頭辞に限定した場合 30 %程度の削減が可能であることがわかった．

表 2: ステートの削減

ルール数	共有化前	共有化後
1000	14394	10787

6 おわりに

本研究では，Bloom Filter でパターンマッチングを行うと同時に，偽陽性検出回路でもパターンマッチングを行うことを提案した．今後，接頭辞以外の部分の共有化について検討を行い，全体としてのハードウェア削減について検討する．

参考文献

[1] Reetinder Sidhu. Viktor K. Prasanna. Fast Regular Expression Matching using FPGAs ,Proc.FCCM,pp.227-238,2001

[2] 中原啓貴, 笹尾勤, 松浦宗_：LUT カスケードを用いた CAM エミュレータについて, 電子情報通信学会技術研究報告, ICD-106, pp.91-96 ,2007.

[3] 小野正人：ネットワーク IDS 向けの効率的なパターンマッチング回路の研究, Master’s thesis, 筑波大学,2006.

[4] Yun-Zhao Li：Non-collision Hash Scheme Using Bloom Filter and CAM，Second Pacific-Asia Conference on Web Mining and Web-based Application，pp.55-58, 2009.