

ゲートウェイにおける攻撃パケットに着目したテーブル検索 負荷削減手法の提案

愛甲 達也^{1,a)} 八巻 隼人¹ 三輪 忍¹ 本多 弘樹¹

概要：ゲートウェイでは、テーブル検索に要する消費電力が全消費電力の約4割を占めていることから、省電力化においてテーブル検索処理を改善することが重要である。これまで、ゲートウェイのテーブル検索負荷を削減する手法は様々提案されているが、攻撃パケットのためのテーブル検索負荷の増大を防ぐことはできなかった。そこで本研究では、攻撃パケットに着目した、ゲートウェイのテーブル検索負荷削減手法を検討した。具体的には、テーブル検索に先立ってゲートウェイがパケットの検査を行い、攻撃パケットの可能性が高いと判断したパケットをNIDSへと送信する。そして、NIDSが攻撃パケットと判断したパケットを破棄することによって、ゲートウェイにおけるテーブル検索負荷の削減を実現する。実ネットワークケースを用いたシミュレーションより、提案手法を用いることでゲートウェイのテーブル検索負荷を局所的に最大約70%削減可能となることを示した。

1. はじめに

ネットワーク機器のうち、ルータとスイッチの消費エネルギーの総量は全世界総消費電力量の約1%に達しており、ルータとスイッチの低消費電力化が急務と言える[1]。ルータの中でも、特に外部ネットワークとの節点に配置されるゲートウェイは、大量のトラフィックを処理するため、多大な電力を要する。今後データ通信量が増大していくことから、ゲートウェイの低消費電力化は重要である。

ゲートウェイにおける最大の電力消費要因として、パケット処理におけるテーブル検索が挙げられる[2][3]。ゲートウェイは、ルーティングテーブルやACL (Access Control List), QoS (Quality of Service) テーブルといったパケット処理に必要な情報を格納した複数のテーブルを備えており、パケット毎にこれらすべてのテーブルを検索することで、パケットを処理する。近年のルータは、テーブル検索を高速化するためこれらのテーブルをTCAM (Ternary Contents Addressable Memory) に格納している。

TCAMは、高速にテーブル検索を行うことができるメモリで、ルーティングテーブルやACL, QoS テーブルといった各テーブルを1サイクルで検索できる。しかし、TCAMは同容量のSRAMと比較すると約16倍の電力を消費するため、ゲートウェイの消費電力の約4割を占める[2]。ルータの低消費電力化手法として、高効率なTCAMのテーブル

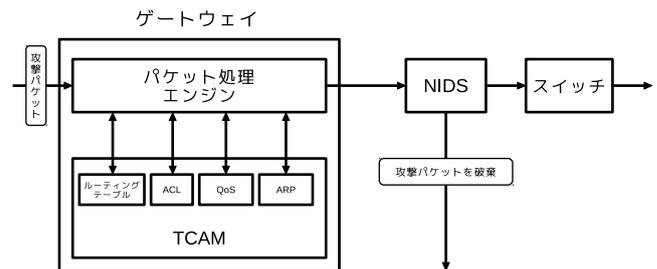


図1 ゲートウェイでの攻撃パケット破棄の様子

検索回路やTCAMのアクセス回数を減らす研究が数多く行われている[4][5]。

しかし、このような既存手法は、攻撃パケットによって発生するTCAMにおけるテーブル検索負荷の増大を防ぐことができない。一般的に、攻撃パケットはゲートウェイからネットワーク内部へと転送された後、NIDSにより破棄される。図1にゲートウェイにおける攻撃パケットの破棄の流れを示す。パケットはゲートウェイでの処理後、侵入検知システムであるNIDS (Network Intrusion Detection System) において攻撃パケットか否かが判断される。攻撃パケットと検知された場合、当該パケットは破棄されるため、ゲートウェイでの攻撃パケットに対するテーブル検索は不要となる。

そこで、本報告ではゲートウェイでのテーブル検索に先立ち攻撃パケットを検知することで、不要なテーブル検索を削減し、テーブル検索の消費電力を削減する手法の提案を行う。

本論の構成を述べる。2章では、関連研究を述べる。3章で

¹ 電気通信大学
The University of Electro-communications
^{a)} aikou@hpc.is.uec.ac.jp

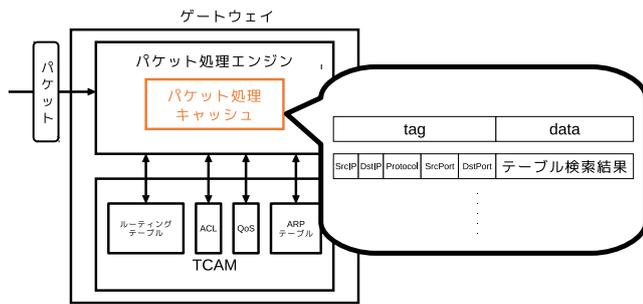


図2 PPCを有するゲートウェイの概要

は、攻撃パケットの特徴とその調査を述べる。4章では、提案手法の詳細を述べる。5章では、提案手法の評価を述べる。6章では、まとめと今後の課題を述べる。

2. 関連研究

TCAM へのアクセス回数を減らすことでテーブル検索消費電力を削減する手法としてパケット処理キャッシュ (PPC) を用いた手法が提案されている。図2に PPC の概要を示す。ゲートウェイ内の各テーブルは、同一の送信元/宛先 IP アドレス、プロトコル番号、送信元/宛先ポート番号を持つパケットに対して、同一のテーブル検索結果を返す。そこで PPC は、パケットヘッダに含まれるこれら5タプルをタグとし、5タプルにより決定される各テーブルの検索結果をキャッシュする。本研究では同一の5タプルの値を有するパケット群をフローと呼ぶ。

Girish ら [4] は、PPC のエントリ置換アルゴリズムの SP を提案した。SP では、キャッシュにデータを登録する際、まず優先度が最も低いエントリに登録し、キャッシュヒットする毎にエントリの優先度をあげる。Girish らは、キャッシュのエントリ置換アルゴリズムに SP を用いた方が、LRU を用いた場合よりもキャッシュヒット率が高くなると述べている。

このような既存手法では、攻撃パケットに対するテーブル検索を行うため、攻撃パケットによるテーブル検索負荷の増大を防ぐことはできなかった。

3. 攻撃パケットの調査

先述したように、攻撃パケットに対するテーブル検索は不要である。攻撃パケットがなにかしらの特徴を持っている場合、その特徴を元に、ゲートウェイでのテーブル検索に先立ち、NIDS に当該パケットを送信し、NIDS で攻撃パケットと検知された場合ゲートウェイでの攻撃パケットのテーブル検索を防ぐことができる。そこで、本章では攻撃パケットの特徴の調査を行う。

3.1 攻撃パケットの特徴

攻撃パケットの例を図3に示す。これらパケットは警察庁の報告で攻撃パケットであることが明らかとなってい

time stamp	Source IP	Destination IP	Protocol	Source Port	Destination Port
577.814638	207.180.176.204	133.243.0.152	TCP	38427	5038
577.814645	207.180.176.204	133.243.0.156	TCP	38427	5038
577.814648	207.180.176.204	133.243.0.158	TCP	38427	5038
577.814651	207.180.176.204	133.243.0.155	TCP	38427	5038
577.814661	207.180.176.204	133.243.0.134	TCP	38427	5038
577.814759	207.180.176.204	133.243.0.138	TCP	38427	5038
577.814762	207.180.176.204	133.243.0.132	TCP	38427	5038
577.814772	207.180.176.204	133.243.0.137	TCP	38427	5038
577.814778	207.180.176.204	133.243.0.130	TCP	38427	5038
577.814782	207.180.176.204	133.243.0.163	TCP	38427	5038
577.814786	207.180.176.204	133.243.0.141	TCP	38427	5038
577.814925	207.180.176.204	133.243.0.172	TCP	38427	5038
577.814932	207.180.176.204	133.243.0.171	TCP	38427	5038
577.814939	207.180.176.204	133.243.0.176	TCP	38427	5038
577.815034	207.180.176.204	133.243.0.178	TCP	38427	5038
577.815057	207.180.176.204	133.243.0.220	TCP	38427	5038
577.815064	207.180.176.204	133.243.0.217	TCP	38427	5038
577.815194	207.180.176.204	133.243.0.211	TCP	38427	5038
577.815198	207.180.176.204	133.243.0.193	TCP	38427	5038
577.815201	207.180.176.204	133.243.0.212	TCP	38427	5038
577.815348	207.180.176.204	133.243.0.229	TCP	38427	5038
577.815361	207.180.176.204	133.243.0.247	TCP	38427	5038
577.815424	207.180.176.204	133.243.0.226	TCP	38427	5038
577.815427	207.180.176.204	133.243.0.246	TCP	38427	5038
577.815441	207.180.176.204	133.243.0.245	TCP	38427	5038
577.815447	207.180.176.204	133.243.0.244	TCP	38427	5038
577.815453	207.180.176.204	133.243.0.240	TCP	38427	5038
577.815560	207.180.176.204	133.243.0.103	TCP	38427	5038
577.815564	207.180.176.204	133.243.0.249	TCP	38427	5038

図3 攻撃パケットの詳細

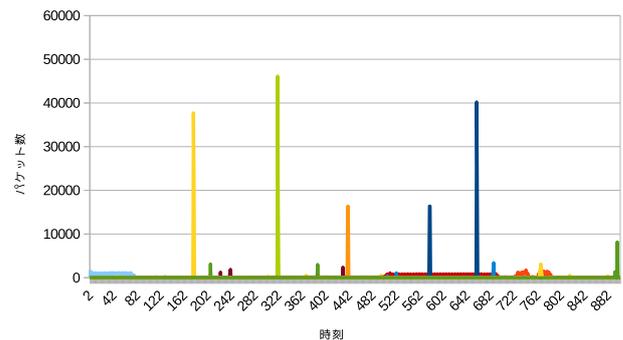


図4 単位時間辺りのフロー数

る [6][7]。図3では、一番左側の列にパケットのゲートウェイの到着時刻 (time stamp)、当該パケットの5タプルの値が表示されている。

図3より、攻撃パケットは次の特徴をもつことが推測できる。

- 特定の送信元 IP アドレスから送信される 1 パケットフローである。ここで 1 パケットフローとは 1 パケットで構成されるフローを指す。
- 1 パケットフローの一部の値を変えて生成される。

3.2 1 パケットフローの調査

このような特徴を元に、実ネットワークに対して、送信元 IP アドレス (ユーザ) に着目した 1 パケットフローの調査を行った。調査に使用したネットワークトラフィックは後述する表2のうち、WIDE を使用した。

調査として、ネットワークトレース中で 1 パケットフローを生成したユーザのうち、上位 100 ユーザに対して、単位時間あたりに生成した 1 パケットフロー数の推移の調査を行った。調査結果を図4に示す。図4は、縦軸が単位時間あたりの生成フロー数、横軸が時刻を示している。また、グラフ中の折れ線は色ごとに異なるユーザが生成した 1 パケットフローであることを表している。

図4より、特定のユーザが短時間に多くの 1 パケットフローを生成していることがわかる。単位時間あたりに大量に 1 パケットフローを生成しているユーザが生成した 1

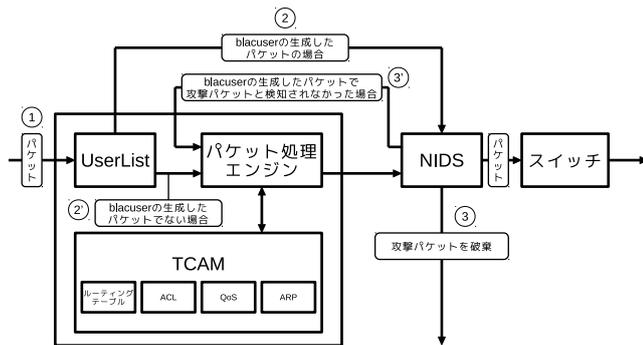


図5 UserList を用いたゲートウェイ及び NIDS の概要

パケットフローの詳細を調べたところ、前節で述べた特徴を有していたため、攻撃パケットの可能性が高いことがわかった。

これより、攻撃パケットの特徴を持つパケットを生成するユーザの特徴として、次の2点がわかった。

- 短時間に大量の1パケットフローを生成するユーザ
 - フローの一部が違う1パケットフローを生成するユーザ
- このような特徴を持つユーザを *blackuser* と呼ぶこととする。

4. 提案手法

前章で述べた分析結果を元に、我々は、*blackuser* が生成する1パケットフローを、ゲートウェイにてテーブル検索を行う前に NIDS に送信し、NIDS によって攻撃パケットでないと判断されたパケットのみゲートウェイにて再びパケット処理を行う手法を提案する。

また、*blackuser* を識別するために、UserList を用いる手法を提案する。UserList を用いて攻撃パケットの特徴を持つ1パケットフローのパケットを同定し、NIDS に送信することでゲートウェイにおけるテーブル検索負荷を削減する手法を UserList 方式と呼ぶこととする。

4.1 UserList の構成

提案手法の UserList は、TCAM とは異なる別の記憶領域で、パケットを生成するユーザの情報を保持する。UserList を用いたゲートウェイ及び NIDS のアーキテクチャを図5に示す。UserList によって、*blackuser* と認定された場合、そのパケットに対するゲートウェイにおけるパケット処理を中止し、NIDS に送信する。

UserList の構成:

UserList の保持する情報を図6に示す。UserList が保持する情報は次の5つの項目である。

- 送信元 IP アドレス (*userip*)
- 重み付き生成フロー数 (*Weighted_flownumber*)
- *blackuser* 識別子 (*isblackuser*)
- *blackuser* として登録された時刻 (*registered_time*)
- 生成したフローのリスト (*Flowlist*)

userip	Weighted_flownumber	Blackuser 識別子	Flow list	registered_time
userip	Weighted_flownumber	Blackuser 識別子	Flow list	registered_time
userip	Weighted_flownumber	Blackuser 識別子	Flow list	registered_time

図6 UserList のデータ構造

表1 UserList の各要素のデータサイズ

要素名	データサイズ
<i>userip</i>	32bit
<i>Weighted_flownumber</i>	7bit
<i>isblackuser</i>	1bit
<i>registered_time</i>	8bit
<i>Flowlist</i>	720bit (10 エントリの場合)

送信元 IP アドレスは、パケットを生成したユーザの IP アドレスを示す。重み付き生成フロー数は、そのユーザが生成したフローの数と、そのフロー数に重みを加えた値を示す。*blackuser* 識別子は、そのユーザが *blackuser* か否かを示す。*blackuser* 識別子が0の場合は *blackuser* ではないこと、1の場合は *blackuser* であることを示す。*registered_time* は、ユーザが *blackuser* と認定された時刻を登録する。生成したフローのリストには、ユーザが生成したフローの情報が登録される。

また、フローのリストに登録されるフローの情報は次の項目である。

- 送信元 IP アドレス以外の4つのパケットヘッダの値
- そのフローのパケットが何パケット生成されているかを表すカウンタ

UserList が保持する5つの値のデータサイズは表1とした。

4.2 提案手法の動作

パケットがゲートウェイに到着すると、UserList での処理が開始される(図5中1)。パケットを生成したユーザが *blackuser* であるかどうか UserList によって判定され、*blackuser* であった場合 NIDS に送信される(図5中2)。

NIDS に送信されたパケットのうち、NIDS で攻撃パケットと検知されたパケットは破棄される(図5中3)。また、攻撃パケットと検知されなかった場合や、UserList によって *blackuser* が生成したパケットでないと判定されたパケットは、ゲートウェイでのテーブル検索処理が行われる(図5中2',3')。

UserList にユーザ情報が登録されたユーザのうち、短時間に多くの1パケットフローを生成する、あるいは酷似するフローを大量に生成するユーザは *blackuser* 識別子が1となり、*blackuser* と認定される。具体的には、短時間に一定の値を超えて大量の1パケットフローを生成する場合、*blackuser*

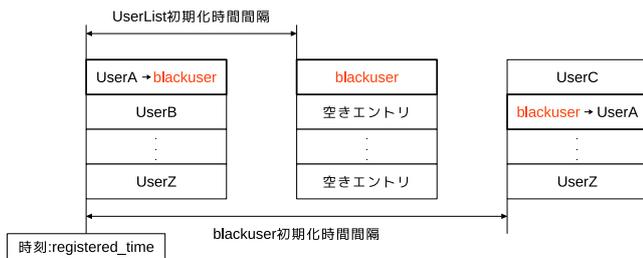


図7 UserList及びblackuserの初期化

と認定する。UserListでは、この一定の値をTHRESHOLDを用いて設定する。

本提案手法では、短時間に多くの1パケットフローを生成するユーザを特定するために、一定時間ごとにUserListに登録されたユーザ情報の初期化を行う。図7にUserList及びblackuserの初期化の概要を示す。このUserListの初期化時間間隔以内にTHRESHOLDの値を超える1パケットフローを生成したユーザをblackuserと認定する。また、初期化を行う際、blackuserであるユーザ以外の初期化を行う。

blackuserと認定されたユーザが生成したパケットはそれ以降ゲートウェイでのパケット処理を一時中断し、NIDSにパケットを送信する。また、blackuserと認定された場合、registered_timeにblackuserと認定された時のパケットのゲートウェイへの到着時刻の値が登録される。

一度blackuserと認定されたユーザのユーザ情報は、UserListの初期化とは非同期に初期化される。これは、UserListを初期化する時間間隔が短い場合、blackuserと認定されたユーザからのパケットが多く生成される前にUserListが初期化されてしまい、再びblackuserとして認定するまでに時間を要するのを防ぐためである。

本提案手法は、PPCを有するゲートウェイに適用することもできる。PPCを有するゲートウェイにUserList方式を適用する場合、PPCにてキャッシュミスしたパケットのみに対してblackuserが生成したパケットか否かの判定が行われる。

5. 評価

UserList方式の評価では、以下に示す2点の評価を行う。

- UserList方式を用いることによるテーブル検索負荷削減量の評価
- ゲートウェイにおける消費電力削減量の評価

5.1 実験方法

本実験は、シミュレーションにて行った。PPCを有するゲートウェイ及び有しないゲートウェイの両方に対して、提案手法を用いた場合及び用いなかった場合の評価実験を行った。シミュレーションでは、ソフトウェアで実装したUserListを表2に示したトレースファイルを入力として与える。UserListによってblackuserと認定されたユーザから

表2 調査に使用したネットワークトラフィックのトレースの詳細

トレース	パケット数	ユーザ数	収集日時	時間
WIDE[8]	22,483,797	309,837	2016年5月1日	900秒
WIDE2[8]	24,702,252	297,833	2016年4月2日	900秒

表3 UserListの各パラメータの値

リスト名	パラメータ	値
ユーザリスト	エン트리数	10
	構成	フルアソシアティブ, LRU
	THRESHOLD	100
	初期化時間間隔	0.01秒
フローリスト	ブラックユーザの生存時間	1秒
	エン트리数	10
	構成	フルアソシアティブ, FIFO

生成されるパケットの単位時間あたりの生成数やTCAMアクセス率を測定した。

UserList方式によってblackuserと認定されたユーザが生成するパケットは、攻撃パケットであることが期待される。提案手法の有効性を評価するためには上記の検証が必要であるが、一般に公開されているネットワークトラフィックのトレースファイルは、パケットヘッダ部分しか含んでいないため、実際にNIDSを用いた攻撃パケットの判定ができない。そこで、単一のユーザが生成した単位時間あたりの1パケットフロー数が単位時間あたりの平均フロー数を超えていた場合、その1パケットフローは攻撃パケットであったと仮定して評価を行う。第3節で述べたように、単一のユーザが単位時間あたりの多くの1パケットフローを生成した場合、生成された1パケットフローは攻撃パケットである可能性が高いためである。また、本評価はNIDSの検知率を100%として行う。

PPCは、エン트리数1024、4wayセットアソシアティブ方式のキャッシュを用い、エン트리置換アルゴリズムはLRUを使用した。また、本評価で使用したネットワークトラフィックのトレース情報を表2に示す。WIDE及びWIDE2は、バックボーンルータで送受信されているネットワークトラフィックをキャプチャしている。トラフィック量が多いことから、ゲートウェイのテーブル検索負荷削減の評価に適切と考えられる。UserListの各パラメータの値は表3のように設定して実験を行った。UserListは後述するようにSRAMで構成されている。エン트리サイズは後述するが、エン트리数や容量から1サイクルでUserList内の全エントリを検索することが可能と考えられる。

5.2 UserList方式を用いることによるテーブル検索負荷削減量の評価

本評価では、TCAMへのアクセス数やアクセス率を調べ、これによりテーブル検索負荷削減率を求める。

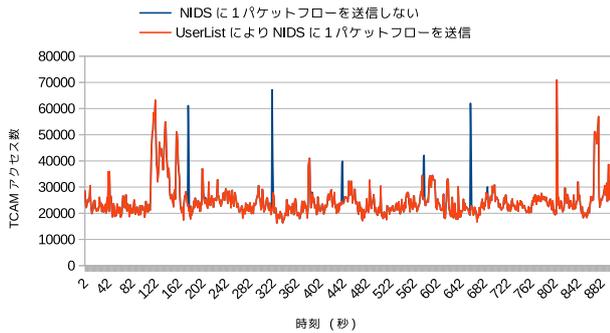


図8 単位時間あたりの TCAM アクセス数の推移 PPC 無し

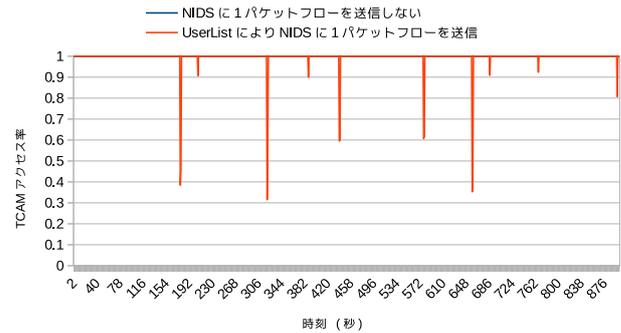


図10 単位時間あたりの TCAM アクセス率の推移 PPC 無し

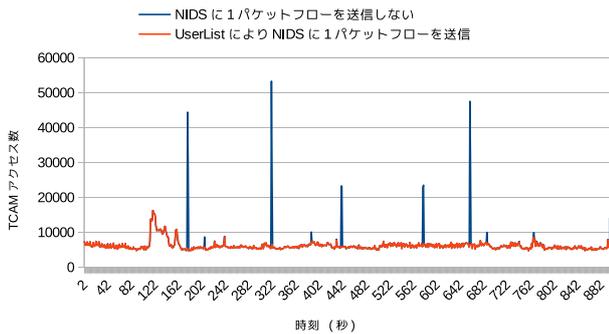


図9 単位時間あたりの TCAM アクセス数の推移 PPC 有り

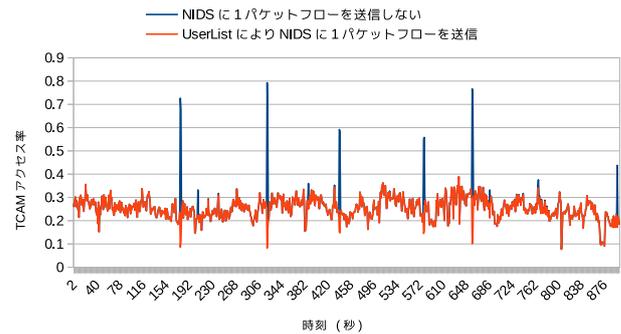


図11 単位時間あたりの TCAM アクセス率の推移 PPC 有り

TCAM へのアクセス数:

UserList 方式を用いたゲートウェイにおける、WIDE の TCAM アクセス数を図 8 及び図 9 に示す。グラフの横軸が時刻、縦軸が TCAM アクセス数を示す。また、図中青線が従来手法、赤線が UserList 方式を示す。図 8 及び図 9 より、TCAM アクセス数が急激に増加する時間帯において、UserList 方式はテーブル検索負荷の増大を防ぐことができたことがわかる。

WIDE で、TCAM アクセス数が急激に増加する時間帯に生成されたパケットは、いずれも単一のユーザが生成したものであった。また、いずれも平均フロー数を超える 1 パケットフローが生成されていた。これより、UserList 方式によって攻撃パケットと推測される 1 パケットフローを同定できたと考えられる。WIDE2 でも、同様の結果となった。

TCAM へのアクセス率:

TCAM へのアクセス率の時間推移を図 10 及び 11 に示す。グラフの横軸が時刻、縦軸が TCAM アクセス率を示す。また、図中青線が従来手法、赤線が UserList 方式を示す。どちらのネットワークトラフィックにおいても、攻撃パケットと推測される 1 パケットフローが多く送信された時間帯においては、TCAM アクセス率が削減した。WIDE では最大で約 70%、WIDE2 では最大で約 60% TCAM アクセス率が削減した。

各ネットワークトラフィックの平均 TCAM アクセス率を表 4 に示す。表 4 より、UserList を用いた場合、平均 TCAM アクセス率は約 1% の削減にとどまった。しかし、攻撃パ

表 4 平均 TCAM アクセス率

トレース	UserList	UserList	UserList	UserList
	無	有	無+PPC	有+PPC
WIDE	100%	98.79%	25.80%	24.50%
WIDE2	100%	98.34%	23.29%	21.27%

ケットと見られる 1 パケットフローが大量に生成されている時間帯において、このようなパケットは NIDS に送信されるため、NIDS の攻撃パケットが検知された場合、UserList 方式によってゲートウェイのテーブル検索負荷は局所的に約 70% 程度削減できることがわかった。

5.3 テーブル検索の消費電力削減

本評価では、ゲートウェイにおける消費電力の削減の度合いを、1 パケットを処理するのに要する消費電力量の平均値より求める。

TCAM, UserList, パケット処理キャッシュにおいて 1 パケットを処理するのに要した消費電力量を求める。ここで、UserList 及びパケット処理キャッシュは SRAM にて構成されているとする。また、TCAM 及び SRAM で消費される電力は動的電力のみを考慮する。TCAM は、4 つのテーブル (ルーティングテーブル, ACL, QoS テーブル, ARP テーブル) の検索を行うこととする。

それぞれ次の場合の消費電力量を算出する。

- (1) 通常のゲートウェイ (P_{conv})
- (2) UserList 方式を用いるゲートウェイ ($P_{ulnoppc}$)
- (3) PPC のみ有するゲートウェイ ($P_{noulppc}$)

表5 変数名と内容

変数名	内容
P_{tcam}	TCAM が 1 つのテーブル検索を行う際の消費電力量
$P_{userlist}$	UserList を構成する SRAM の消費電力量
P_{ppc}	パケット処理キャッシュを構成する SRAM の消費電力量

表6 1パケット処理した際の消費電力量

TCAM	UserList	パケット処理キャッシュ
66.80nJ	0.007361nJ	0.03346nJ

表7 1パケット処理に要する平均消費電力量の削減率

トレース	P_{ulnopp}	P_{noulpp}	P_{ulppc}
WIDE	1.199%	74.15%	75.44%
WIDE2	1.649%	76.66%	78.67%

(4) PPC 及び UserList 方式を用いるゲートウェイ (P_{ulppc})
上記のそれぞれの構成での 1 パケットのテーブル検索処理に要する消費電力量算出式を以下にまとめる。

$$P_{conv.} = P_{tcam} \times 4 \quad (1)$$

$$P_{ulnopp} = P_{userlist} + P_{conv.} \times a_rate \quad (2)$$

$$P_{noulpp} = P_{ppc} + P_{conv.} \times m_rate \quad (3)$$

$$P_{ulppc} = P_{ppc} + P_{userlist} \times m_rate + P_{conv.} \times a_rate \quad (4)$$

式中の変数はそれぞれ表 5 を示す。また、TCAM へのアクセス率を a_rate 、キャッシュミス率を m_rate とする。

本実験では、TCAM において 1 パケットを処理するのに要した消費電力量は文献 [9] に示された値を用いる。本実験では、エン트리サイズ 32Byte、エン트리数 32K エントリの TCAM を用いることとする。

UserList 及びパケット処理キャッシュの 1 パケット処理に要した消費電力量は CACTI[10] にて求める。パケット処理キャッシュはエン트리サイズ 28Byte、エン트리数 1024、4way セットアソシアティブキャッシュとした。また、UserList は表 1 より、1 エン트리 96Byte とし、エン트리数 10 のフルアソシアティブキャッシュとする。表 6 に、TCAM、UserList、PPC において 1 パケットの処理に要した消費電力量を示す。

1 パケットの処理に要した平均消費電力量:

結果を表 7 にまとめた。結果から、PPC を有しないゲートウェイに UserList 方式を用いたとき、1 パケットを処理する際に要する消費電力は WIDE の場合約 1.199%、WIDE2 では約 1.649%削減された。PPC を有するゲートウェイに UserList 方式を用いたとき、1 パケットを処理する際に要する消費電力は WIDE の場合約 75%、WIDE2 では約 78%削減された。また、PPC のみを有するゲートウェイに対する、PPC 及び UserList を有するゲートウェイの消費電力量削減率を求めたところ、WIDE で約 5%、WIDE2 で約 8%削減された。

6. おわりに

本研究ではゲートウェイにおけるテーブル検索負荷削減手法として攻撃パケットの特徴を持つ 1 パケットフローを生成するユーザからの 1 パケットフローをゲートウェイのテーブル検索に先立ち NIDS に送信することで、ゲートウェイにおけるテーブル検索負荷を削減する手法を提案した。

提案手法により、多くの攻撃パケットと思われる 1 パケットフローを NIDS に送信し、ゲートウェイにおけるテーブル検索負荷を削減することができた。提案手法では、テーブル検索に必要な消費電力量は約 1%の削減にとどまった。しかし、攻撃パケットのためのテーブル検索負荷を局所的に約 70%程度削減できることがわかった。

提案した UserList は、短時間に大量に多くの 1 パケットフローを生成するユーザからのパケットを攻撃パケットとしていた。しかし、長時間にわたり少しずつ攻撃パケットを生成するユーザからの攻撃パケットは提案手法では検知されにくい。このようなユーザからの攻撃パケットも検知できるように改良することで、更にゲートウェイにおけるテーブル検索負荷を削減することができる。

参考文献

- [1] Addis, B. et al.: "Energy Management Through Optimized Routing and Device Powering for Greener Communication Networks", *TON*, Vol. 22, pp. 313–325 (2014).
- [2] Nawa, M. et al.: "Energy-efficient High-speed Search Engine Using a Multi-dimensional TCAM Architecture with Parallel Pipelined Subdivided Structure", *Proceedings of the 2016 13th IEEE CCNC*, pp. 309–314 (2016).
- [3] HP: "Energy Efficient Networking Business white paper", <http://h17007.www1.hp.com/docs/mark/4AA3-3866ENW.pdf>.
- [4] Girish, B. and Govindarajan, R.: "Improving Performance of Digest Caches in Network Processors", *Proceedings of the HiPC 2008*, pp. 6–17 (2008).
- [5] 阿多信吾ほか: "低コスト・低消費電力 TCAM における効率的なルーティングテーブル管理法", 電子情報通信学会技術研究報告, Vol. 107, No. 443, pp. 7–12 (2008).
- [6] Security NEXT: "管理不備の「MS SQL Server」狙うアクセスが増加", <http://www.security-next.com/069653>.
- [7] NPA JAPAN Cyber Force Center: "インターネット観測結果等 (平成 28 年上半年 (1 月~6 月))", <https://www.npa.go.jp/cyberpolice/detect/pdf/20160915.pdf>.
- [8] WIDE MAWI Working Group: "MAWI Working Group Traffic Archive - WIDE MAWI Working Group", <http://mawi.wide.ad.jp/mawi/>.
- [9] Agrawal, B. and Sherwood, T.: "Ternary CAM Power and Delay Model: Extensions and Uses", *Trans. on VLSI*, Vol. 16, No. 5, pp. 554–564 (2008).
- [10] HP lab: "CACTI An integrated cache and memory access time, cycle time, area, leakage, and dynamic power model", <http://www.hpl.hp.com/research/cacti/>.