社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2005－DPS－121 (11)
2005－GN－ 54 (11)
2005／1／20

トラヒック特性に基づく近似機能を有する高速パケットキャプチャ

大須賀 怜* 片山 喜章 高橋 直久 (名古屋工業大学大学院)

本稿では，空間分割型パケット分類器に対して以下の機能を特徴とする近似方式を提案し，その実現法について述べる．フィルタが与えるパケット照合条件を元の式から変化させることにより，パケット分類空間の分割数を，指定された値より小さくする機能．空間分割の過程において，パケット分類空間の各部分空間に属する格子点を計数し，その結果に従って，近似のためにどのパケット照合条件を変化させるか決定する機能．照合条件を変化させた部分空間に含まれる格子点に対するパケット出現頻度を実トラヒックから推定し，パケット出現頻度の高い格子点をその部分空間に含まないように近似を行う機能．本稿ではまた，提案方式に基づき開発したパケットキャプチャ実験システムによる評価実験について述べる．

High-Speed Packet Caputuring with Approximation based on Characteristic of Traffic

Satoshi Osuga* Yoshiaki Katayama Naohisa Takahashi
(Graduate School of Engineering, Nagoya Institute of Technology)

In this paper, the approximation method characterized by the following functions to the Space-Division Packet Classification is proposed, and the realizing method is stated. The function made smaller than the value which had the number of division of packet classification space specified by changing the packet matching rules which a filter gives from the original formula. Which packet matching rules are changed according to the result for approximation by carrying out calculation of the lattice point belonging to each partial space of packet classification space in the process of space division, and the function for which it opts. The function which approximates so that the packet frequency of appearance to the lattice point included in the partial space to which matching rules were changed may be presumed from real traffic and the lattice point that packet frequency of appearance is high may not be included in the partial space. Moreover, this paper describes the evaluation experiment by the packet capture experiment system developed based on the proposal system.

1 はじめに

ネットワークの監視・診断ではネットワークを流れるパケットを収集して，その中身から状態を把握するパケットキャプチャシステムが広く使われている．しかし，高速なネットワークでは膨大な量のパケットが短時間で流れてくる場合がある．このような場合には，パケットキャプチャシステム内部での転送速度やディスクの書き込み速度，記憶装置の容量といったハードウェア上の制限や，キャプチャ後にそれらの処理が追いつかないといったソフトウェア上の制限により，流れるパケット全てをキャプチャし監視・診断を行うのは困難である．このような問題を解決するために，ネットワークのインターフェースでフィルタリング技術を用いて監視・診断に必要なパケットだけを選択してキャプチャする方法が有効になる．この方法を用いればパケットキャプチャシステムが処理するトラヒック量を減らす事が可能になる．しかし，高速なネットワークでフィルタリングを行うにはフィルタリング速度も高速でなければならない．一方，多数のフィルタに対しても高速なフィルタリングが行える方式として空間分割型のパケット分類方式がある[2]~[6]．空間分割型のパケット分類方式では，事前に分類のための表(**分類表**)を作成しておき，パケットを受け取ると，分類表を参照して合致するフィルタを高速に選択する．本稿ではフィルタはキャプチャすべきパケットが満たすべき条件を記述したものであるとし，分類の結果合致するフィルタが無かったパケットは廃棄する．分類空間の任意の点 p に対応するパケットが照合に成功するフィルタの集合を M(p) とする．このとき，空間分割型パケット分類器では，事前計算において，分類空間を以下のような部分空間に分割する．

1. 部分空間に属するすべての点 p の M(p) は等しい
2. S1 と S2 を異なる部分空間とし，p を S1 の任意の点，q を S2 の任意の点とすると，M(p) と M(q) は異なる．

パケットが到着すると，パケット分類器は，そのパケットに対応する分類空間上の点 p に対する M(p) を求めて，M(p) が空でないとき，そのパケットをキャプチャし，空のときには廃棄する．本稿ではこのような空間分割型パケット分類方式に対して以下の特徴をもつ近似方式を提案し，その実現法について述べる．

特徴 1 フィルタが与えるパケット照合条件を元の式から変化させることにより，パケット分類空間の分割数を，指定された値より小さくする機能 (近似機能と呼ぶ) を実現する．パケット分類用テーブルの大きさは，上記分割数により決まるため，近時機能により，この大きさを，与えられたメモリ容量以下に抑えることが可能になる．この結果，パケット分類器をオンチップ化できるようになる．一方，照合条件を変化させると，パケットが照合に成功するフィルタの集合 M(p) が変化するような点 p が発生する．近似機能では，本来照合に成功するパケットは近似しても必ず照合に成功することを保証するため，M(p) は要素が増大するように変化させる．M(p) が増大した点 p からなる部分空間を**誤差領域**と呼び，p に対応するパケットを**ノイズ**と呼ぶ．

特徴 2 空間分割の過程において，パケット分類空間の各部分空間に属する点を計数し，その結果に従って，近似のためにどのパケット照合条件を変化させるか決定する機能を実現する．これにより，近似機能により発生する誤差領域に属する点の数を減少させることが可能になる．その結果，ノイズの低減が期待できる．

特徴 3 近似により生じる誤差領域に含まれる点に対するパケット出現頻度を実トラヒックから推定し，パケット出現頻度の高い点を誤差領域に含まないように近

似を行う機能

本稿の構成は次の通り．まず第 2 章でパケット分類問題を定義する．第 3 章では関連研究について述べ，従来の手法を概観する．第 4 章では提案方式の実現法について述べ，第 5 章では評価実験について述べる．

2 パケット分類器

2.1 機能概要

本稿では，パケットのヘッダの値に従い，IP パケットを識別する分類器を考える．ここで，ヘッダは送信先 IP アドレス，送信先ポート番号など，それぞれ長さ一定の複数のフィールドからなる．これらのうち，分類器でパケットの識別に用いるフィールドを**キー**と呼び，各キーが満たすべき条件の記述を**フィルタ**と呼ぶ．フィルタの集合を**フィルタセット**と呼び，フィルタの識別子の集合で表す．本稿ではパケット分類器を次のように動作するものとする．すなわち，フィルタセット F とパケット P が与えられた時，次節で述べる照合法によりパケット P の各キーとフィルタ集合の各フィルタとを照合する．そして，あるパケットに対し，全てのキーが対応する条件との照合に成功するフィルタが 1 つ以上あればそのパケットをキャプチャする．

2.2 照合法

本稿で扱うパケット分類器は，フィルタセット F とパケット P が与えられた時，以下に述べる照合法によりキーとフィルタとを照合し，全てのキーが対応する条件との照合に成功するフィルタがあればキャプチャする．1 つのキーに複数の照合を適用することも許され，その場合は 1 つでも成功する照合が有ればそのキーは照合に成功したとする．

- 完全一致照合 (exact matching) フィルタの条件でキーの値を v と指定した時，パケットのキーの値が v と一致すれば照合に成功
- プレフィクス照合 (prefix matching) フィルタの条件でキーの値を v/p と指定した時，パケットのキーの値の上位 p ビットが v の上位 p ビットと一致すれば照合に成功
- 領域照合 (range matching) フィルタの条件でキーの値を v1-v2 と指定した時，パケットのキーの値が v1 以上かつ v2 以下であれば照合に成功

2.3 空間による表現

パケット分類問題は，計算幾何学的なアプローチにより多次元空間における点位置決定問題として捉える事ができる [4]~[7]．すなわち，キーの数を N とした時，それぞれのキーを軸とする N 次元空間を考える．つまり，全てのパケットはこの N 次元空間の 1 点で表される．このような，任意のパケットがある 1 点で表される離散空間を本稿では**分類空間**と呼ぶ．尚，分類空間は離散空間であるが，本稿では視覚的に分かりやすくするために，ある区間 $[a,b](a \leq b)$ を図示する際には a-1 と a の間から b と b+1 の間までの区間を描く．例として，$SrcIFAddress(0 \leq SrcIFAddress \leq 2^{32}-1)$ と $DstIFAddress(0 \leq DstIFAddress \leq 2^{32}-1)$ をキーとする分類を考える．このような分類において次の 2 つのフィルタ F1,F2 がある時，分類空間は図 1 のようになる．

F1: $4 \leq SrcIFAddress \leq 16$ かつ $10 \leq DstIFAddress \leq 16$

F2: $7 \leq SrcIFAddress \leq 16$ かつ $3 \leq DstIFAddress \leq 12$

このとき，図 1 で F1,F2 の矩形で示されるように，フィルタは分類空間の部分空間を指定することになり，これを本稿では**フィルタ領域**と呼ぶ．フィルタ領域は 1 つ以上の格子点を包含する．本稿ではフィルタ領域に包含される格子点の数をフィルタ領域の**容量**と呼ぶ．分類空間において照合に成功するとは，パケットを示す点が 1 つ以上のフィルタ領域に包含されることである．図 1 ではパケット (Src IP Address=13, Dst IP Address=14) が点で表されており，F1 のフィルタ領域に包含されているので，このパケットは F1 との照合に成功するパケットである．Src IP Address の軸では，F1 の区間は [4,16] で F2 の区間は [7,16] となる．F1,F2 の区間の重なり具合により全区間 $[0,2^{32}-1]$ を次の 5 区間に分割できる．重なるフィルタの集合が等しい区間の集合を**等影区間**と呼ぶ．すなわち，F1 のみの区間 [4,6]，F1,F2 の区間 [7,16]，空の区間 [0,3][16,$2^{32}-1$] の 4 つの等影区間ができる．

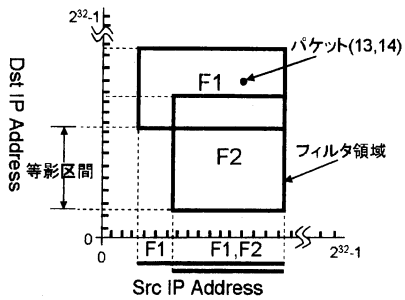

図 1: 分類空間の例

2.4 関数による表現

パケット分類器は，与えられたパケットがどのフィルタ領域に含まれるか決定する．以下に示す関数 sieve を導入し，フィルタセットとキーの値と位置に対して次のように繰り返し sieve を適用する関数 SIERRA(SIEve aRRAy) を用いるとパケット分類器は次の様に定義できる [5][6][7]．

SIERRA(F, k1, k2, k3, ..., kn)
= sieve(...(sieve(sieve(sieve(F, k1, 1), k2, 2), k3, 3), ...), kn, n)

- フィルタ篩関数 sieve(F, ki, i)
  i 番目のキーの値が ki のとき，i 番目の条件が満たされるフィルタの集合を入力フィルタ集合 F から選択する関数
- フィルタ照合関数 SIERRA(F, k1, k2, k3, ..., kn)
  n 個の sieve を用いて，n 個のキーの値が全て対応する条件を満たすフィルタを入力フィルタ集合 F から選択する関数

上の方式は各 sieve 関数をそれぞれ 1 つのプロセッサで実行させるとパイプライン構成により実現される．ここで，以下のように複数のフィルタの積集合を求める関数 product を導入する．

- フィルタ集合の積関数 product(F1, F2)
  入力フィルタ集合 F1，F2 の積集合を求める関数

関数 product と sieve を用いると，sieve 関数を実行するプロセッサを並列に動作させる並列型 SIERRA を定義できる．例えば n=4 の場合は次の様になる．

SIERRA(F, k1, k2, k3, k4)
= product(product(sieve(F, k1, 1), sieve(F, k2, 2)), product(sieve(F, k3, 3), sieve(F, k4, 4)))

分類器の実現方式として，1つ1つのキーに対して順に照合を行っていくパイプライン型と，複数のキーに対して同時に照合を行う並列型が考えられる．パイプライン型では，照合の途中で照合に成功するフィルタが無い事を判定できる．しかし，成功するフィルタが有る場合はキーの数と同じ回数の照合を要する．一方パイプライン型では，複数のキーに対して同時に照合を行うため，あるキーで照合に成功するフィルタが無い場合でも他のキーも照合が行われる．しかし，複数のキーに対して同時に照合を行う場合も，1つのキーに対して照合を行う場合と同じ時間で照合を行うことが可能である．

3 関連研究

パケット分類について，高速ルーティングテーブル探索のために連想メモリ (CAM) を用いた方式 [1] がある．この方式では，簡単なハードウェアで高速なパケット分類が可能であるが，フィルタ数が多い場合やキーが長い場合には高価な連想メモリを並列動作させる等の必要が有り高コストとなる．

ソフトウェアによる実現方式として V.Srinivasan らの方式 [2] や Pankaj Gupta らの方式 [3] がある．これらの手法ではどちらも次元ごとに分割してキーごとに照合を行い，その照合結果の積を求めるという空間分割によるパケットの分類を行っている．

文献 [1]〜[4] の方式は，前述の並列型 SIERRA の1つの実現法と位置づけられる．文献 [3] の方式では，分割した空間の並列処理により 11 バイト分のキーを持つ 3000 個のフィルタでも 300KB 程度のメモリを用いて数回のテーブル参照のみで分類している．しかし，近年の高速ネットワークに対応するにはテーブルを極めて高速にアクセスできるメモリ (プロセッサ内部のキャッシュメモリ等) に収める必要があり，高コストとなる．またメモリ容量の制限により利用できない場合がある．

文献 [5][6] の方式はパイプライン型 SIERRA である．この方式では最適化によりメモリ量の削減を行っているが，オンチップ化した場合，メモリ容量の制限により利用できない場合がある．

文献 [7] では SIERRA の計算で近似を導入し，1チップに収められる程度にテーブルサイズを小さくすることにより，キーの数とフィルタの数が多い場合でも低コストで高速な分類を可能にしている．また，分類条件の変化をできるだけ抑えることに着目した近似手法を実現することによりノイズを低減している．本稿ではこの方式に特徴 2 と特徴 3 を新たに提案し，さらなるノイズの低減を実現する．

4 提案方式の実現法

本章では，第1章で挙げた3つの特徴についてそれぞれ実現法を述べる．

4.1 フィルタ領域の近時法

パケット分類器では，パケットヘッダの N 個のフィールドを，照合に用いるキーとする．本稿では $N = 2^n$ とする．空間分割方式では分類条件を N 次元空間上の領域として表現し，パケット分類を N 次元における点位置決定問題と捉えて実現する．提案方式ではハードウェア化による高速化のため 2 章で述べた並列型 SIERRA を採用している．また，分類空間のある軸の各座標と，その座標を含む等影区間を示す識別子との対応表 (分類表) をオンチップのメモリに格納し，パケット到着時にはその表を探索するのみで高速にパケットを分類する．この分類表を少ない容量のメモリに確実に格納するために，後述する近似を用いて表のエントリ数を一定以下に抑えている．

4.1.1 空間のマッピング

並列型 SIERRA では，空間を N/2 次元,N/4 次元,N/8 次元と次元数が半分の空間にマッピングする product のフェーズを繰り返し，1 次元まで次元数を減らすことによりパケットの分類を実現している．例えば，図 2 では送信元 IP アドレス (以下 SrcIP) と送信先 IP アドレス (以下 DstIP) を軸とする分類空間上に分類条件 F1,F2 が表現されている．これは N 次元空間の，SrcIP と DstIP の次元への射影である．この射影の空間内にある格子点も N 次元空間に存在する格子点の SrcIP と DstIP の次元への射影である．同様に，図 3 では送信元ポート番号 (以下 SrcPort) と送信先ポート番号 (以下 DstPort) について F1,F2 が表現されている．ここで，図 4, 図 5 に示すように，各軸について等影区間に対して識別子 (以降単に ID という) を付与し，ID の組の値に従って分類空間を部分空間に分割する (以下，ID の値を示す際には軸名に (ID) と付けて記述)．さらに，ID の組の値が同じ部分空間を一次元の直線上の一点にマッピングし，フィルタの区間を表示すると図 6 を得る．図 6 の座標は次の式により決定されている．

$$4 * SrcIP(ID) + DstIP(ID)$$

式中の 4 は DstIP 軸の ID 数である．このように ID の組で表される 1 つの部分空間を 1 つの座標とし，部分空間に含まれる全格子点を 1 つの格子点にマッピングする操作を**アサイン**と呼ぶ．図 6 の座標 (2,1) は図 4 の SrcIP(ID) が 2 かつ DstIP(ID) が 1 で示される部分空間 (2,1) がアサインされた座標である．この分類空間の軸は SrcIP と DstIP の組合わせの軸 (以下 IP 軸) である．同様に図 5 の分類空間も一次元にマッピングすると SrcPort と DstPort の組み合わせの軸 (以下 Port 軸) が作成できる．こうしてできた 2 つの 1 次元の分類空間を軸とすることで新たに図 7 の分類空間を作成できる．この部分空間の格子点は N/2 次元の格子点の IP 軸と Port 軸で形成される平面への射影である．この分類空間も同様に一次元にマッピングすることができ，このような操作を図 8 のように Phase[0] から Phase[n] まで繰り返す．Phase[0] から Phase[n] では，Phase[i] の部分空間を Phase[i+1] の点 (i=0,1,2,...n-1) にマッピングしている．これにより，N 次元 ($N = 2^n$) を一次元にマッピングする．

4.1.2 分類条件の近似

マッピング後の分類空間から作られる分類表のエントリ数はマッピング前の等影区間の数に依存する．そこで，近似的空間分割型パケット分類器では等影区間の数が減るようにフィルタ領域を変形することで分類表のデータ量を任意の大きさまで小さくする．ある軸で等影区間の組 A と B を選んだ時，それらを 1 つの等影区間にするためには，それぞれが持つフィルタ集合を等しくすれば良い．そこで，まず A が持っていて B が持っていないフィルタの集合を求める．このフィルタ集合を B の等影区間に拡大すれば A と B は等しいフィルタ集合を持つことになる．この時，他の軸の等影区間に影響を与えないようにするため，拡大しようとしているフィルタ領域が他の各軸でどこにあるかを

調べ，それと同じ位置に拡大する．同様にBが持っていてAが持っていないフィルタの集合もA側に拡大する．このようにして拡大した各フィルタ領域の拡大部分を**変化領域**と呼び，2つの等影区間を1つの等影区間とするようにフィルタ領域を拡大する操作をフィルタ領域の近似と呼ぶ．例えば，図2の分類空間はSrcIP軸では{ },{F1},{F1,F2}のフィルタ集合を持つ3つの等影区間がある．図9のように斜線部のF2の領域を左の波線部に拡大することにより{ },{F1,F2}のフィルタ集合を持つ2つの等影区間になる．このようにフィルタ領域を変形させると，パケット分類器は与えられたフィルタセットと異なる分類の条件を持つことになる．フィルタ領域の近似により等影区間の数を常に一定以下に抑えることができる．フィルタ領域の近似を行うと，もともとフィルタ領域が無かった場所にフィルタ領域が拡大されることがあるため，本来廃棄すべきパケットをキャプチャしてしまうフィルタ領域(**誤差領域**)が生じるという問題がある．例えば，図9でSrcIP(ID)が1でDstIP(ID)が1の誤差領域に来るパケットは本来破棄されるはずであるが，近似によりフィルタ領域を拡大したためキャプチャされる．このようなパケットを**ノイズ**と呼ぶ．

4.2 部分空間の容量を用いたノイズ低減法

提案方式では，次のように3種類の容量を定義する．

点の容量 その点にマッピングした部分空間の容量(初期値1)

部分空間の容量 部分空間を各軸に射影してできる区間の容量の積

区間の容量 区間に属する点の容量の和

これによりマッピング過程のPhase[i]でも部分空間の容量が計算できるようになる．特徴1の近似において，変化領域の容量が最小となるように近似するフィルタ領域を選択することで，ノイズを低減できる．例として，図6の座標(2,1)が持つ点の容量を求める．これは，図4でSrcIP(ID)が2，DstIP(ID)が1で示される部分空間をマッピングしたものである．図2を見ると，この部分空間をSrcIP軸へ射影した区間の容量は10で，DstIP軸へ射影した区間の容量は7である．よってこの部分空間の容量は$10\times7=70$であり，これが図6で座標(2,1)の点の容量となる．同様に他の座標や図3をマッピングした空間についても容量を求めることで，図10でフィルタ領域の斜線部を波線部で示される部分空間に拡大する場合の変化領域の容量を求めることができるようになる．例えば，図10の波線部をIP軸へ射影した区間に属する点は座標(2,1)のみなので区間の容量は上記のように70を持ち，同様にPort軸へ射影した区間に属する点は座標(1,2)の点のみなので区間の容量は図5より$64512\times3=193536$と求まる．これにより，波線部にF1を拡大した場合の変化領域の容量は$193536\times70=13547520$となる．点の容量の初期値は1であるため，この変化領域の容量はPhase[0]で拡大されたフィルタ領域に含まれる点の数を表している．

4.3 パケット出現頻度によるノイズ低減法

各格子点におけるパケットの出現頻度を推定し，出現頻度が高い格子点を誤差領域に含むような近似を避けることでノイズが低減できる．出現頻度を推定する手法として，トラヒックをキャプチャし，その中での出現頻度を求めるという手法がある．ネットワークの構成やサービスに変化が無ければ，キャプチャされたトラヒックが持つ傾向が今後も続くと考えられる．提案方式はトラヒック全てをキャプチャするのが困難なネットワークでの使用を想定しているため，部分的にキャプチャしたトラヒックを用いたパケットの出現頻度の推定を繰り返すことで全体のパケット出現

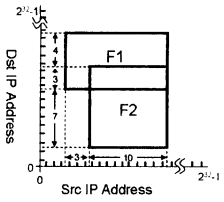

図2: 二次元分類空間における分類条件の例(IP address)

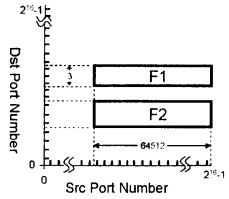

図3: 二次元分類空間における分類条件の例(Port Number)

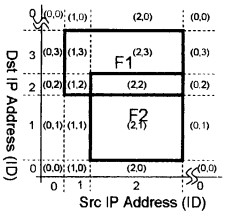

図4: IDを割り当てた等影区間の例(IP address)

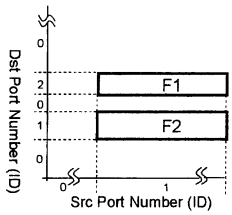

図5: IDを割り当てた等影区間の例(Port Number)

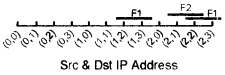

図6: 二次元から一次元へのマッピングの例(IP Address)

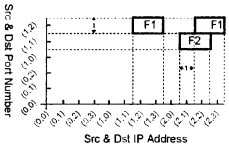

図7: 四次元から二次元へのマッピングの例

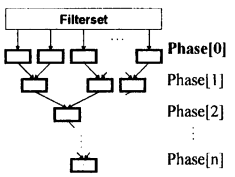

図8: 分類空間のマッピングの流れ

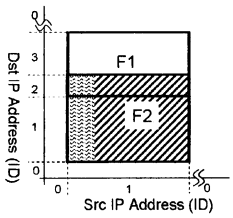

図9: 二次元での近似の例

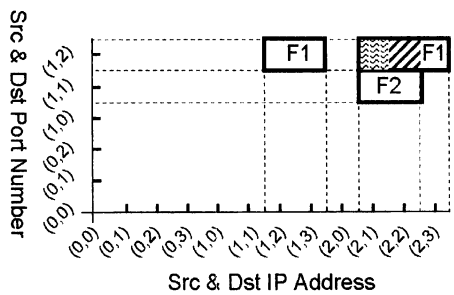

図10: マッピング後の二次元での近似の例

頻度を推定する．これは次の2つの機能を用いて実現する．

機能1 誤差領域に含まれる点に対するパケットの到着数と観測時間を実トラヒックから求め出力する．

機能2 機能2を初めて実行する場合には機能1の出力より推定したパケット出現頻度をそのまま機能2の出力とする．2回目以降は機能1の出力より推定したパケット出現頻度と，前回の機能2により推定した出現頻度とをマージした結果を出力する．

ここでパケット出現頻度は，次の式で推定される単位時間辺りのパケット到着数とする．

$$\text{パケット出現頻度} = \frac{\text{パケット到着数}}{\text{観測時間}}$$

特徴1,2で述べた方式で分類表を作成してキャプチャを行った後，キャプチャしたトラヒックを元に機能1,2を適用しパケットて出現頻度を求める．その結果を各点の容量の初期値として再び分類表を作成してキャプチャを行い，再度キャプチャしたトラヒックを元に機能1,2を適用する．このように分類表の作成→トラヒックの観測→機能1→機能2という操作を繰り返すことでキャプチャし得る格子点でのパケット出現頻度が推定できる．この節では上記2つの機能について説明する．

4.3.1 機能1

各格子点でパケット出現頻度を求めるためには，各格子点についてキャプチャを行った観測時間とパケット到着数を記録する必要がある．しかしながら全ての格子点について記録を行うとデータ量が膨大になるため，提案方式では次元ごとに独立して記録する．また，近似前にフィルタ領域が元々ある部分空間が誤差領域になることは無いため，誤差領域によってキャプチャされたパケットのみ記録する．ここで，パケット分類器はパケットをキャプチャするか否かの判断をするのみなので，パケットが誤差領域によってキャプチャされたものかどうかを判別する手法が必要となる．このため，提案方式ではキャプチャしたパケットを近似前のフィルタセットと改めて照合し，合致しなかったものは誤差領域によってキャプチャされたものであるとして，各キーごとに合致しなかった時の値をカウントする．全次元のパケットが出現した各座標のうち，パケット出現頻度が最も低い座標を1とした出現頻度の比を各座標で求める．この比をその座標の点の容量の初期値として設定する．ただし，パケットが1度も出現していない座標の点の容量の初期値は1とする．初めて近似を行う際には記録されたデータが無いため，全ての座標の点の容量は1である．点の容量の初期値が高く設定されると，その座標の点は多数の格子点をマッピングしたものに相当するとして計算するため近似対象に含まれ難くなる．このように点の容量の初期値を設定して近似を行うことで，パケット出現頻度の高い格子点を避けるように近似が行われる．例えば図11の分類空間において，送信元IPアドレスaかつ送信先IPアドレスbのパケットが，出現頻度が最小のパケットの100倍出現する場合を考える．送信元IPアドレスの次元で近似をする場合，F2を左に拡大する近似は変化領域の容量が $5\times10=50$ であるのに対し，F1を右に拡大する近似は $3\times7=21$ で済む．しかし，F1を右に拡大してしまうとパケットが多量に到着する点に誤差領域が作られてしまうためノイズが増大してしまう．そこで送信元IPアドレスの次元で座標aに点の容量として初期値100を与えるとF1を右に拡大する近似は変化領域の容量が $(2+100)\times7=714$ になると計算され，F2を左に拡大する近似の方が選ばれる．

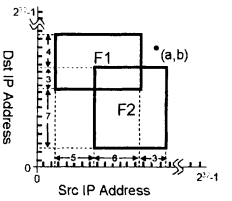

図11: 多量にパケットが来る点のある二次元分類空間の例

4.3.2 機能2

点の容量の初期値を変更し近似を行うと，誤差領域に含まれる格子点が変更前とは異なるものになるため，再度キャプチャを行うとそれまでとは異なる特性のトラヒックがキャプチャされる．ここで再び各格子点について観測時間とパケットの到着数を記録する．これを前回の結果に加えることで，それまでにキャプチャしたことのある格子点について出現頻度を求めることができる．

5 評価実験

実験により次の事を確認する．

- 部分空間の容量を用いた近似によるノイズ低減効果
- パケット出現頻度の値がノイズ率に与える影響の評価

5.1 実験システム

評価実験で使用するサンプルフィルタはsnort[8]のルールを元に作成したフィルタから無作為に100個を選んだ．パケットキャプチャシステムは文献[9]の提案に基づいて作成したものを用いる．このシステムはフィールドを8ビットごとに区切ったものからキーを選ぶ．サンプルフィルタに提案方式を適用して分類表を作成するフィルタコンパイラを作成した．但し，特徴3については機能1のみ実装が完了している．サンプルトラヒックはMAWI[10]で公開されているトラヒックアーカイブの中からsamplepoint-Bの2003年09月07日のものを使用する．このデータは15分間のトラヒックであり，平均14.23Mbpsで3836636個のIPパケットを含んでいる．主なパケットはicmpが41.6%，httpが16.5%，nntpが9.8%である．

5.2 実験項目

実験1 部分空間の容量を用いた近似によるノイズ低減効果

この実験では，部分空間の容量を用いて近似を行うことでノイズが低減できることを確認する．．分類表の大きさがキャプチャシステムに要求されるメモリ量となり，近似する量を多くすると分類表は小さくなる．この実験では，近似により生じる誤差領域によって余分にキャプチャされるパケット(ノイズ)の割合と分類表の大きさとの関係を求めて，特徴2のノイズ低減効果について考察する．ノイズ量を示す値として"ノイズ率"を次の様に定義する．

$$\text{ノイズ率} = \frac{\text{ノイズの数}}{\text{破棄すべきパケットの数}} \times 100[\%]$$

以下の4つの方式について実験した．

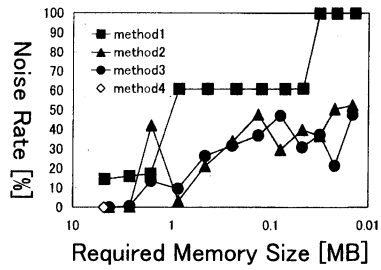

図 12: ノイズ率の比較結果

method1:特徴 1 のみを適用した場合 (近似するフィルタ領域は無作為に選択)

method2:特徴 1 と 2 を適用した場合

method3:特徴 1 と 2 と 3 を適用した場合

method4:近似を行わない場合

図 12 は実験 1 の結果を表す．横軸は要求されるメモリの大きさで縦軸はノイズ率である．

実験 2 パケット出現頻度の値がノイズ率に与える影響の評価

この実験では，あるパケット出現頻度の高い点に初期値として大きな容量を与えることで，その点が誤差領域に含まれにくくなるように近似を行うと，その点に到着するパケットがノイズとして出現する割合が減ることを確認する．ここでは点の容量の初期値による影響を分かりやすくするため，1 つのキーの値のみを点の初期値を変更する対象として選び，比較的大きいと思われる値 1000000 を与えてみる事にする．点の容量の初期値を大きくするキーの値は，実験 1 でキャプチャされたパケットを近似前のフィルタと比較し，正確な照合が行われなかった回数が多いキーの値を選ぶ．実験 1 でキャプチャされたパケットを近似前のフィルタと比較した結果，例えばファイルサイズ 0.02MB 付近の点では回数の多いものから順に 10 個挙げると表 1 のようであった．他の点でも同様に比較を行ったところ，全体的に宛て先ポート番号の上位バイト (DstPort(High)) が 0 の時が正確な照合が行われなかった回数が多い事が分かった．この結果より，点の容量の初期値を大きく与えるキーの値として DstPort(High) の 0 を選び，実験では図 12 の method3 の結果を得た．また，正確な照合が行われなかった回数が多いキーの値を多いものから順に 9 個と，DstPort(High) が 0 の時を挙げるとファイルサイズ 0.02MB 付近の点では表 2 の結果を得た．

表 1: 正しい照合が行われなかった回数 (点の容量の初期値は全て 1)

KEY	Value	Count
Protocol	6	6392699
SrcPort(High)	0	1689880
DstPort(High)	12	1458822
DstPort(Low)	80	1364229
SrcPort(Low)	80	1337917
DstPort(High)	**0**	**1212685**
DstPort(Low)	135	1027926
DstIPaddr1	213	900782
DstIPaddr1	212	742519
DstIPaddr2	37	728690

表 2: 正しい照合が行われなかった回数 (DstPort(High)=0 の点の容量の初期値は 1000000)

KEY	Value	Count
Protocol	6	5411679
DstPort(High)	12	2150162
SrcPort(High)	0	1220532
DstIPaddr1	212	1138676
SrcPort(Low)	80	1095143
SrcPort(High)	18	963697
SrcIPaddr1	83	776982
SrcPort(Low)	158	751947
DstIPaddr2	32	739260
DstPort(High)	**0**	**466962**

5.3 結果と考察

実験 1 Phase[0] の空間での分類条件の変化量を計算可能にしたことによるノイズ低減効果

図 12 より method2 のノイズ率を method1 と比較すると次のことが分かる．

1. 特徴 2 を適用するとノイズ率が大幅に下がり，特徴 1 だけの場合より 10%〜60%程度抑制できた
2. ファイルサイズ 2M 程度の時に提案方式ではノイズ率が目だって高くなっている

1 より提案方式の特徴 2 はノイズ量の抑制に有効な方式であると言える．また，2 は 4.3 節で述べた"パケットの出現頻度が高い格子点"が誤差領域に多く含まれたものと考えられる．このようなノイズはそのキーの値に対し点の容量として大きな初期値を与える事で解消できる．

実験 2 特定のキーの値に点の容量として大きな初期値を与えた場合のノイズの変化

図 12 と表 2 よりノイズを比較すると次の事が分かる．

1. DstPort(High) のキーで値が 0 の時に正しい照合が行われなかった回数は提案方式では 1/3 程度に減っている
2. 幾つかのファイルサイズの時にはノイズ率が大幅に抑えられている

1 より提案方式の特徴 3 は特定の種類のノイズを低減するのに有効な方式であると言える．また，2 より全体的なノイズを低減できるような近似の選び方があることがわかる．これにより，今後特徴 3 の機能 2 を実装することで全体的なノイズの低減が期待できる．

6 おわりに

提案方式に基づくパケットキャプチャシステムを作成し，ソフトウェアシミュレーションによりノイズ低減効果を確認した．

謝辞

本研究の一部は，平成 16 年度科学研究補助金 (基盤研究 (C)16500028) の研究助成によるものである．

参考文献

[1] Anthony J. McAuley and Paul Francis. Fast routing table lookup using CAMs. *INFOCOM*, Vol. (3), pp. 1382–1391, 1993.

[2] V. Srinivasan, Subhash Suri, and George Varghese. Packet classification using tuple space search. *SIGCOMM*, pp. 135–146, September 1999.

[3] Pankaj Gupta and Nick McKeown. Packet classification on multiple fields. *SIGCOMM*. pp. 147–160, August 1999.

[4] T. V. Lakshman and Dimitrios Stiliadis. High-speed policy-based packet forwarding using efficient multi-dimensional range matching. *SIGCOMM*, pp. 203–214, 1998.

[5] 高橋直久. フィルタ sieve 関数の部分計算に基づく実時間パケット分類器. 日本ソフトウェア科学会 インターネット技術ワークショップ WIT 99, pp. 190–197, 1999.

[6] N. Takahashi. A systolic sieve array for real-time packet classification. *Journal of IPSJ*, Vol. 42, No. 2, pp. 146–166, 2001.

[7] 大須賀怜, 加藤和夫, 片山喜章, 高橋直久. 高速パケットキャプチャのための選択近似機能を有する空間分割型パケット分類器の実現と評価. 第 31 回分散システム/インターネット運用技術研究会, 2003.

[8] Sourcefire. The open source network intrusion detection system. http://www.snort.org/.

[9] 加藤和夫, 大須賀怜, 高木淳史, 片山喜章, 高橋直久. スケーラブルパケットキャプチャシステムの実現. 日本ソフトウェア科学会 インターネット技術ワークショップ WIT 2003, 2003.

[10] WIDE Project. Mawi working group traffic archive. http://tracer.csl.sony.co.jp/mawi/.