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

2006－ＡＶＭ－55（14）
2006／12／15

送受信データの構成変化を利用したネットワークアプリケーション弁別方式

和泉　勇治†　　根元　義章†

† 東北大学大学院情報科学研究科〒980–8579 宮城県仙台市青葉区荒巻字青葉 6–6–05
E-mail: †{wai,nemoto}@nemoto.ecei.tohoku.ac.jp

あらまし　サーバを利用しない効率的なデータ通信を可能にした P2P ネットワークが盛んに利用されるようになり，個人対個人での簡便な情報共有が実現している．しかし，P2P ネットワーク上で蔓延する不正アクセスによる情報流出などが新しい社会問題となっている．情報流出の対策として，P2P ソフトウエアが発生した通信を検知し遮断する必要があるが，P2P ネットワークを構築するソフトウエアでは，通信に利用するポート番号を改竄することが可能であるため，ポート番号による P2P アプリケーションの特定は不可能である．また，それらのソフトウエアのプロトコルが未公開であったり，暗号通信の利用などによって，効果的な対策が更に困難となっている．そこで本稿では，パケットペイロード部のデータの構成変化を定量的に評価することにより，ポート番号を利用しないアプリケーションの弁別方式を提案する．本稿で提案するアプリケーション弁別方式は，アプリケーションが送受信するデータ内容の変化に何らかの法則性があることを仮定し，その法則性を，パケットペイロードのコード分布の変化により評価するものである．幾つかの P2P ソフトウエアの発生したトラヒックを利用し，提案方式の弁別性能について報告する．
キーワード　アプリケーション弁別，ポート番号改竄，P2P

A Discriminating Method of Network Applications Based on Structural Transformations of Communication Data

Yuji WAIZUMI† and Yoshiaki NEMOTO†

† Graduate School of Information Sciences, TOHOKU University　Aramak Aza Aoba 6–6–05, Aoba-ku, Sendai, Miyagi, 980–8579 Japan
E-mail: †{wai,nemoto}@nemoto.ecei.tohoku.ac.jp

Abstract　A person-to-person imformation sharing is easily realized by P2P networks that serves are unnecessary to do so. Leakages of information, which are caused by malicious accesses for P2P netwroks, has become new social issues. To prevent information leakage, it is necessary to detect and block traffics of P2P software. Since some P2P softwares can spoof port numbers, it is difficult to detec the traffics sent from P2P softwares by using port numbers. It is more difficult to devise effective countermeasures for detecting the software because their protocol are not public. In this report, we propose a discriminating method of network applications structural transformations of communication data without port numbers. The proposed method is based on an assumption that there are any rules among structural transformations of communication data of applications. By extracting the rule from payloads of packets, the proposed method can discriminate applications without port numbers. An easy and simple way to person-to-person share imformation
Key words　Application Discrimination, Port Number Interpolation, P2P

1. まえがき

サーバを利用しない効率的なデータ通信を可能にした P2P ネットワークが盛んに利用されるようになり，個人対個人での簡便な情報共有が実現している．しかし，P2P ネットワーク上で蔓延する不正アクセスによる情報流出などが新しい社会問題となっている．2005 年に起きたネットワークを経由した情報漏洩は，報道されているだけでも約 130 件程度であるとの報告も

ある [1]．同報告書では，ネットワーク以外の流出経路を含めた場合の情報漏洩に対する損害賠償額が 7000 億円を超えると試算している．情報流出は，経済的な被害だけではなく，個人のプライベートに関わる問題でもあり，速やかに取り組むべき問題であると言える．

ネットワークを経由する情報流出としては，メールソフトの設定ミスなどの原因もあるが，近年では，P2P ソフトウエアの不適切な利用が原因となっている事故が増えている．P2P ソフトウエアによる情報流出の場合，情報が外部へ洩れたことを利用者が気付かない場合があり，更に問題が深刻となり得る．P2P ソフトウエアによる情報流出を防ぐには，そのトラヒックを検知し遮断する必要があるが，P2P ネットワークを構築するソフトウエアでは，通信に利用するポート番号を改竄することが可能であるため，ポート番号による P2P アプリケーションの特定は不可能である．また，それらのソフトウエアのプロトコルが未公開であったり，暗号通信の利用などによって，効果的な対策が更に困難となっている．

この問題に対し，パケットサイズやタイプなどの遷移や接続要求の発生状況を統計的にモデル化し，アプリケーションの弁別を試みる研究が行われている [2]〜[5]．これらの研究は，パケットのヘッダの情報を利用する通信の秘匿性を考慮した方法であり，実用的観点から有様な研究であると言える．しかし，アプリケーションを特定するために多数のパケットやフローを観測する必要があることや，観測対象の情報が限定されているために，手法単体での弁別の精度が不十分であるなどの課題もあると考えられる．

一方，我々は，パケットのヘッダ情報ではなく，ペイロードに含まれる文字コードの出現頻度のヒストグラムを利用した不正アクセスの検知方式を提案している [6]〜[10]．この方式は，前述のヘッダ情報のみを利用する手法と同様に，通信の秘匿性を考慮し，ペイロード部のデータを直接評価せず，文字コードの出現確率のヒストグラムを用いてペイロード部のデータを数値化するものである．これらの研究から，ペイロード部の文字コードの出現確率のヒストグラムの形状に不正アクセスの特徴が保存されていることが判明している．

そこで，本稿では，パケットペイロード部のデータの構成変化をペイロード部の文字コードの出現確率のヒストグラムの変化として評価するポート番号を利用しないアプリケーションの弁別方式を提案する．本稿で提案するアプリケーション弁別方式は，アプリケーションが送受信するデータの変化に何らかの法則性があることを仮定し，その法則性を，パケットペイロードのコード分布の変化により評価するものである．観測対象にパケットのペイロード部も含まれているが，ペイロードの内容を直接保存せず，文字コードの出現確率のヒストグラムとして扱うため，通信の秘匿性を守ることが可能な手法でもある．幾つかの P2P ソフトウエアの発生したトラヒックを利用した実験で，提案方式のアプリケーションの弁別性能について明らかにする．

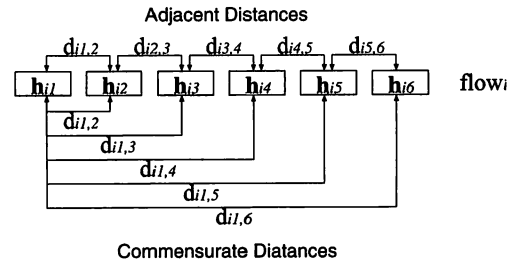

図 1　パケットのヒストグラム間距離の算出方法

2. パケットペイロードの構造変化の数値化方式

ポート番号を利用しないアプリケーション弁別における，本稿の仮定は，アプリケーションによって送受信されるデータの内容の変化には，その種別毎に異なる法則性がある，ということである．この仮定を定量的に評価するためのパケットペイロードの構造変化の数値化方式を提案する．ここで，提案手法は，送受信 IP アドレス，送受信ポート番号，プロトコルの 5-tuple が一致するパケットの集合として定義されるフローを評価の単位とする．ただし，送受信 IP アドレスと送受信ポート番号が逆転したパケットも同一のフローであるとして扱う．つまり，送受信双方向のパケットを一つのフローとして見做し，以下に提案する評価手法を適用する．

2.1　パケットペイロードの構成変化の定量評価手法

パケットペイロードの構成を定量的に評価するために，本稿では，フローを構成するパケットのペイロード部を 8 ビット毎のコードに分割し，それらの出現確率をヒストグラムとして表現する．ヒストグラムを算出するパケットは，1 バイト以上のペイロードを有しているもののみとし，フラグのみなどのパケットは，処理対象から除外する．抽出されるヒストグラムは，256 階級により構成され，フロー内の解析の対象となる個々のパケットから一つずつのヒストグラムを算出する．このヒストグラムを 256 次元のベクトルと見做し，ベクトル間の距離によりパケットペイロードの構成変化を評価する．

あるフロー i を構成するパケット j のヒストグラムを $\mathbf{h}_{i,j}$ とし，パケット j，k 間の距離を $d_{j,k}$ を次式のように定義する．

$$d_{j,k} = \sum_{l=0}^{255} (h_{i,j,l} - h_{i,k,l})^2 \qquad (1)$$

ここで，$h_{i,j,l}$ はヒストグラム $\mathbf{h}_{i,j}$ の第 l クラスを表す．距離の算出は，1 バイト以上のペイロードを持つ最初のパケットとそれ以降のパケット間の距離 (Commensurate Distances，以下，Cdist) と隣接するパケット間の距離 (Adjancent Distances，以下，Adist) の 2 種類を考慮する (図 1)．パケットの発生順は，TCP の sequence number に基づき 1 バイト以上のペイロードを持つパケットを配置する．

図 2〜5 に，パケットのヒストグラム間の距離の遷移を示す．グラフ中一つの線が一つのフローに対応している．横軸は，1 バイト以上のペイロードを持つパケット発生順を示しており，図

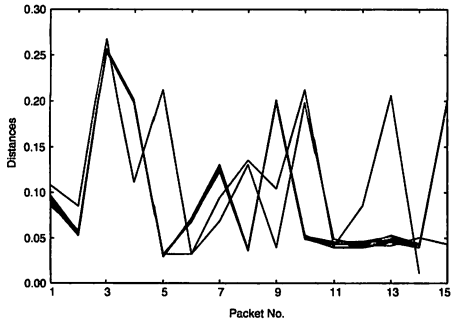

図 2　pop3 通信のヒストグラム間の距離 (Cdist)

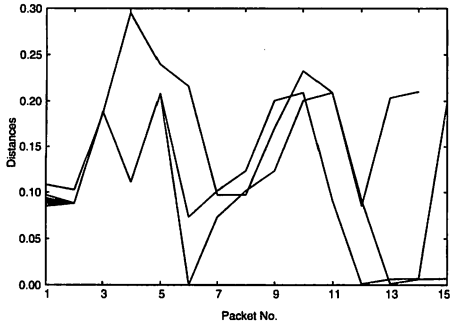

図 3　pop3 通信のヒストグラム間の距離 (Adist)

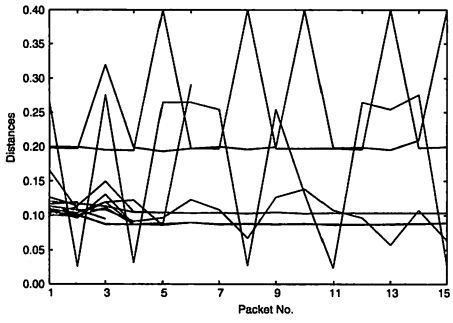

図 4　winny 通信のヒストグラム間の距離 (Cdist)

は，pop3 を利用したメールの受信と winny による通信の例である．winny に関しては，コンテンツの検索を行わず，winny のネットワークに中継ノードとしてのみ参加した場合のトラヒックを観測したものである．また，観測開始以前に通信を始めていたトラヒックも含まれている．

図より，最初の 3〜5 パケット目までは，複数のフローで類似した距離の変化が確認でき，アプリケーションに対応した何らかの法則性を，この距離の変化により抽出出来ていることが予想される．

2.2　距離系列ベクトルによるデータの構成変化のモデル化

前節で算出したヒストグラム間の距離の系列をベクトルと見做し，次式により定義する．

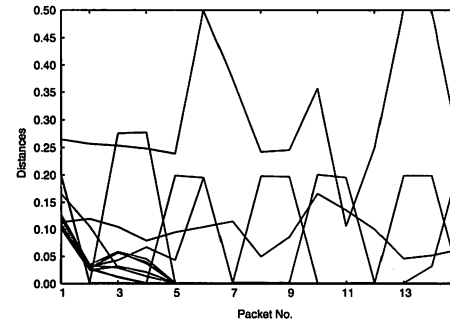

図 5　winny 通信のヒストグラム間の距離 (Adist)

$$\mathbf{v}_{Ai} = \{d_{1,2}, d_{2,3}, \cdots, d_{N-1,N}\} \quad (2)$$

$$\mathbf{v}_{Ci} = \{d_{1,2}, d_{1,3}, \cdots, d_{1,N}\} \quad (3)$$

ここで，$\mathbf{v}_{A,i}$ と $\mathbf{v}_{C,i}$ は，それぞれ，フロー i の Adist と Cdist の系列を表す．また，N は，ベクトルを構成する際，利用するパケット数である．

図 6，7 に $N = 5$ とした場合のベクトルの分布を Self-Organizing Map(SOM) [11] によって可視化したものを示す．SOM は，対象データを参照ベクトルによって量子化し，クラスタリングを行うモデルである．参照ベクトル間には，通常 2 次元での隣接関係 (位相) が定義されており，多次元空間での隣接関係をある程度保持し，2 次元への写像が可能である．SOM を利用することにより，任意次元の空間内のクラスタの分布を 2 次元のマップとして可視化可能となる．本実験では，SOM の参照ベクトルを 16x16 に配置したモデルを利用した．解析対象には，表 1 に示すアプリケーションを用いた．

表 1　解析フローの構成

アプリケーション	フロー数
http	752
pop3	71
winny	12
skype	9
netbios	5
ntp	3

図から，それぞれのアプリケーション毎にクラスタを生成して分布していることが分かる．http トラヒックは，通信開始後に GET メソッドなどの URL を指定するペイロード部のバリエーションが多くなる通信を行うため，クラスタが広範囲に分布していると思われる．一方，pop3 においては，ユーザ名やパスワードなどの不定な文字列が含まれるが，認証結果の送信など定型文も多く含まれることから，http などに比べ狭い範囲で分布していると考えられる．winny のトラヒックにおいても同様に，暗号通信を確立するための情報交換が行われているため，局所的にベクトルが分布していると推定出来る．

図 8 には，P2P ソフトウエアである Winny と Share の $N = 5$ での Cdist の分布を示す．Winny と Share の分布間にグレーの領域が生成されていることから，5 次元空間において，異な

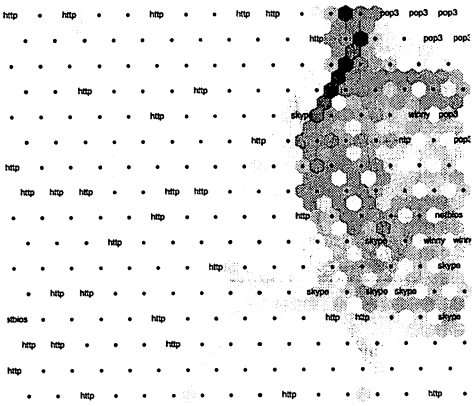

図 6　ベクトル $\mathbf{v}_{C,i}$ の分布 ($N = 5$)

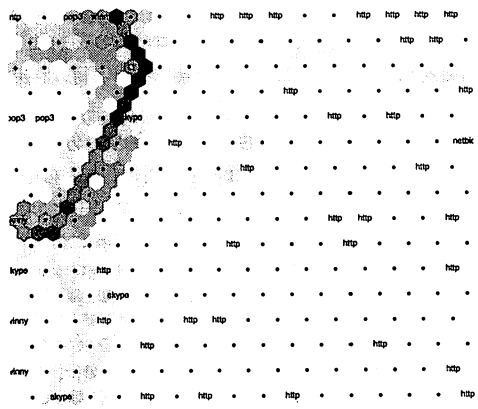

図 7　ベクトル $\mathbf{v}_{A,i}$ の分布 ($N = 5$)

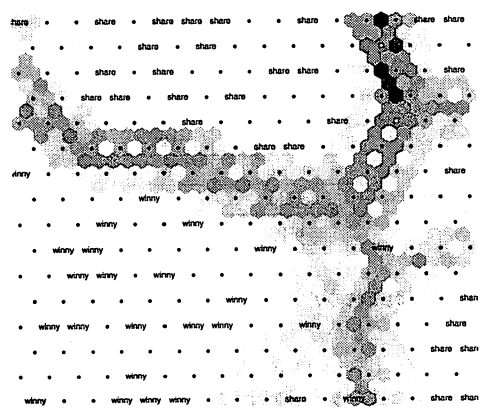

図 8　Winny と Share のベクトルの分布 ($N = 5$)

る領域に，それぞれのトラヒックから抽出したベクトルが分布していることが分かる．

これらから，適切なクラスタリングやルールを適用することにより，このベクトルからアプリケーションの弁別が可能であると考えられる．

3. データ構成の遷移を利用したアプリケーション弁別

前章では，あるフローを構成する最初の数パケットのヒストグラム間の距離の遷移をベクトルとして定義し，その分布状況を SOM を利用し解析を行った．解析の結果，異なる種別のアプリケーションのトラヒックから生成したベクトルは，異なる領域にクラスタを生成し分布していることが明らかとなった．通信データのバリエーションが多いと考えられる http は広範囲に分布していると考えられるが，pop3 などの定型文を多くやり取りするアプリケーションは，比較的狭い範囲に分布しているため，高精度な弁別が期待できるこが分かった．そこで本章では，教師あり学習モデルである Learning Vector Quantization(LVQ) [12] を利用したアプリケーション弁別手法を提案する．

3.1 Learning Vector Quantization を利用したアプリケーション弁別方式

本稿で提案する弁別方式は，LVQ の中でも比較的高い精度を得ることが可能な LVQ2.1 を用いアプリケーションの弁別を行う．LVQ は，SOM と同様にベクトル量子化を行うが，SOM とはことなり，参照ベクトル間の位相の保存は行わず，投入されたベクトルのカテゴリを考慮した教師あり学習により判別境界を設定するモデルである．予め収集したラベル付きのデータを利用し，LVQ により参照ベクトルを学習し，新たに観測されたトラヒックと参照ベクトル間のユークリッド距離を評価することにより弁別を行う．

本提案手法においては，トラヒックを発生させたアプリケーション名を前節で提案したヒストグラム間距離の遷移ベクトルのラベルとして学習用データセットを作成する．このデータセットを利用し，LVQ2.1 により参照ベクトルを学習する．実験では，インターネットに接続した 3 台の PC のうち 1 台で Winny，Share を稼働させ，その他のアプリケーションとしては，Web アクセス，pop3 によるメールの送受信，skype を利用した．Winny と Share に関しては，利用可能帯域を 1kbps, 3kbps, 5kbps, 7kbps, 無制限と設定し，通信を行った．Winny と Share のトラヒックは，ファイルの検索を行わず，各ソフトウエアの起動のみを行い観測したものであるため，ファイル転送の中継のみのトラヒックが対象となっている．ヒストグラム間の距離の算出方法は，フローのペイロードを持つ最初のパケットからの距離 (Cdist) を利用し，距離の系列の長さを $N = 3$, $N = 5$, $N = 10$ とした場合について弁別を行った．観測されたフローの構成は表 2 のようになった．

3.2 弁別結果

表 3，4，5 に，それぞれ，$N = 3$, $N = 5$, $N = 10$ とした場合の学習データとテストデータの弁別結果を示す．全体の正解率としては，$N = 5$ と $N = 10$ の場合に 99%以上の高い弁別性能を得ることが出来た．P2P ソフトウエアである Share に関しては，テストデータにおいても 98% 以上の高い正解率で検知が可能であるが，Winny は正解率が低下してしまっている．特に，$N = 10$ の場合，Winny の検知精度 65.52%と他に比べ低

表 2　観測フローの構成

	学習データ	テストデータ
アプリケーション	フロー数	フロー数
http	4626	3753
share	723	614
pop3	421	334
winny	145	77
netbios	24	18
skype	18	23
ntp	14	9
dns	5	2

下の幅が大きくなっている．これは，利用する距離の系列が長くなると，通信開始時の定型文のやり取りのみでなく，コンテンツの通信に利用されているパケットも距離の算出対象になるため，8bit コードのバリエーションが増加し，アプリケーション間の有意な差位が減少することが原因であると考えられる．一方，一部のアプリケーションに対しては，100%の正解率を得ているものがあり，本提案手法が良好に機能していることが分かる．

表 3　アプリケーションの弁別結果 ($N = 3$)

	学習データ	テストデータ
アプリケーション	正解率	正解率
winny	89.63%	83.33%
share	100.0%	98.86%
http	99.87%	99.95%
pop3	91.63%	90.78%
netbios	44.12%	48.00%
skype	73.91%	75.00%
ntp	100.00%	100.00%
dns	18.75%	100.00%
TOTAL	98.38 %	98.40 %

表 4　アプリケーションの弁別結果 ($N = 5$)

	学習データ	テストデータ
アプリケーション	正解率	正解率
winny	93.10%	85.71%
share	99.31%	98.70%
http	99.72%	99.84%
pop3	96.20%	97.31%
netbios	79.17%	88.89%
skype	72.22%	78.26%
ntp	100.00%	100.00%
dns	40.00%	100.00%
TOTAL	99.05%	99.15 %

3.3　参照ベクトルの形状

LVQ2.1 の学習によって得られた参照ベクトルを図??に示す．LVQ2.1 は，教師あり学習を行い，異なるカテゴリ間の差位を強調する学習を行うことが可能であるため，学習により得られた参照ベクトルの形状からカテゴリ間の観測量の違いを確認することが可能である．

表 5　アプリケーションの弁別結果 ($N = 10$)

	学習データ	テストデータ
アプリケーション	正解率	正解率
winny	75.00%	65.52%
share	99.43%	99.34%
http	99.86%	99.91%
pop3	99.52%	100.00%
netbios	78.95%	73.33%
skype	92.31%	94.12%
ntp	100.00%	100.00%
TOTAL	99.55%	99.45 %

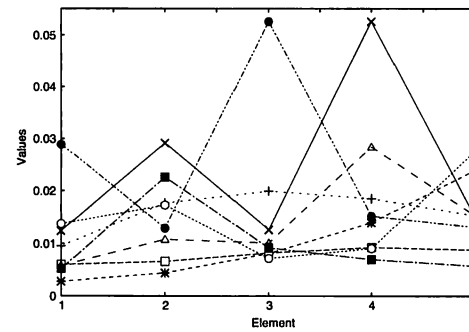

図 9　http トラヒックから得られた参照ベクトル

図 9 の http トラヒックから得られた参照ベクトルは，各要素が 0.0 から 0.05 付近までしか分布しておらず，Winny と Share の参照ベクトルと比べて，各要素の絶対値が小さくなる傾向がある．これは，パケットのペイロードを構成する文字コード分布が似ていることを示しており，html で記述されたコンテンツの送受信では，タグなどの類似したトークンが頻繁にやり取りされていることを示していると考えられる．しかし，http では，多様な情報のやり取りが行われていることから，更に多くのトラヒックを利用した検証が必要であると言える．

図 10 の Winny のトラヒックに関しては，通信開始直後のパケットペイロード間の距離は比較的小さくなっていることが分かる．これは，暗号通信を行うための鍵交換が行われている様子を示していると考えられる．観測したトラヒックのパケットサイズの調べたところ，100 バイト程度のパケットが連続しており，バリエーションの少ない情報の交換が行われている箇所を参照ベクトルが抽出していると推定出来る．

Share(図 11) のトラヒックに関しては，詳細が不明ではあるが，通信開始直後から http や Winny と比較してサイズの大きいパケットの送受信が行われていた．そのために，ペイロードにバリエーションの大きいデータが記録され，パケットのヒストグラム間の距離が大きくなり，参照ベクトルの値も他のアプリケーションと比べて大きくなっていると考えられる．

4.　まとめと今後の課題

本稿では，ポート番号を利用しないアプリケーションの弁別

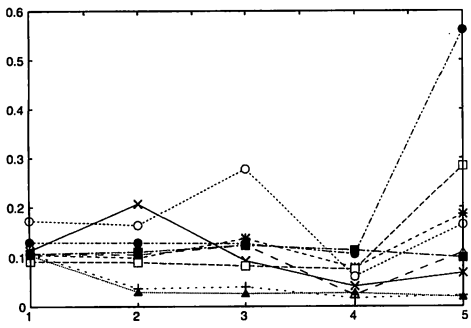

図 10　Winny トラヒックから得られた参照ベクトル

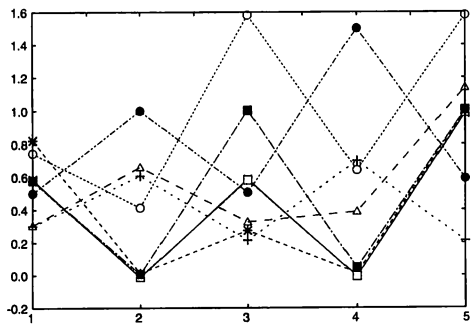

図 11　Share トラヒックから得られた参照ベクトル

を目的とし，パケットペイロードのコードのヒストグラム間の距離を利用し，ペイロード部のデータの遷移に基づいた弁別方式を提案した．提案手法は，個々のアプリケーション種別に対応した定型文などのデータのやり取りが存在することを仮定し，その法則性をヒストグラム間の距離の遷移により評価する手法である．本手法は，個々のネットワークフローの最初の数パケットを観測すれば弁別が可能であるという点が最大の特徴である．P2P トラヒックを含む検証実験においては，約 99%程度の正解率でアプリケーションの弁別を実現することが確認され，本手法の高い弁別性能と本手法の仮定の正当性が証明されたと考えられる．

しかし，本稿で報告した実験結果は，フロー数 10000 程度，アプリケーション種別数 8 種類と，非常に小規模で限られた環境で得られたものであり，本手法の適用範囲を検討するには不十分である．特に，暗号通信では，定型文などのやり取りが，どの程度観測可能であるか不明であるため，更にアプリケーション種別を増やした実験が必要であると考えられる．また，本稿では，弁別の基準として LVQ2.1 によって得られた参照ベクトルを利用しているが，これは，対象データが存在する空間をボロノイ領域に分割し，新たな観測量がその領域のどれに属するかによって，アプリケーション種別を特定するものである．ボロノイ領域には，隣接する参照ベクトルの位置関係により相対的に変化するものであり，学習データの取得状況によっては，大きく弁別性能が低下する恐れもあると考えられる．この問題に対応するために，パケットのヒストグラム間の距離の遷移のルールをアプリケーション毎に自動的に抽出するアルゴリズムの開発が今後の課題でもある．

文　献

[1] 2005 年度 情報セキュリティインシデントに関する調査報告書. 日本ネットワークセキュリティ協会セキュリティ被害調査ワーキンググループ, 2005.

[2] 太井優樹, 阿多信吾, 岡育生. フロー統計情報によるバルク・リアルタイムトラヒック分別法. 信学技報, Vol. NS2006-28, pp. 29–32, 5 2006.

[3] 北村強, 静野隆之, 岡部稔哉. パケットタイプ遷移パタン分析を用いたトラフィック識別手法. 信学技報, Vol. NS2006-27, pp. 25–28, 5 2006.

[4] 松田 崇, 中村 文隆, 若原 恭, 田中 良明. CP セッション統計による P2P トラヒック弁別手法. 電子情報通信学会総合大会,pp.121-121,B-6-121,Mar.,2005

[5] 中村文隆, 松田崇, 若原恭, 田中良明. トラヒック特徴量解析とアプリケーション弁別. 信学技報, Vol. NS2007-80, pp. 57–62, 9 2006.

[6] 和泉勇治, 辻雅史, 根元義章. パケットペイロードのクラスタリングによる拡散型不正アクセス検知方式に関する一考察. 信学技報, Vol. CS2005-19, pp. 19–24, 9 2005.

[7] 辻雅史, 和泉勇治, 角田裕, 根元義章. フローペイロードの類似性に基づく拡散型ワーム検出方式に関する一検討. 信学技報, Vol. NS2005-112, pp. 9–12, 11 2005.

[8] K. Simkhada, H. Tsunoda, Y. Waizumi, and Y. Nemoto. Differencing worm flows and normal flows for automatic generation of worm signatures. In *Proc. of the Seventh IEEE International Symposium on Multimedia*, 12 2005.

[9] K. Simkhada, T. Taleb, Y. Waizumi, A. Jamalipour, N. Kato, and Y. Nemoto. An efficient signature-based approach for automatic detection of internet worms over large-scale networks. In *Proc. of IEEE Int. Conf. Commun. (ICC 2006)*, 6 2006.

[10] K. Simkhada, T. Taleb, Y. Waizumi, A. Jamalipour, N. Kato, and Y. Nemoto. A multi-level security based autonomic parameter selection approach for an effective and early detection of internet worms. In *Proc. of IEEE Globecom'06*, 11 2006.

[11] T.Kohonen. The self-organizing map. In *Proc.IEEE*, Vol. 78, pp. 1464–1480, 1990.

[12] T.Kohonen. *Self-organization and Associate Memory (2nd Edition)*. Spring-verlag, 1998.