(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第4898648号
(P4898648)
(45)発行日　平成24年3月21日(2012.3.21)　(24)登録日　平成24年1月6日(2012.1.6)

(51)Int.Cl.　F I
H04L 12/56 (2006.01)　H04L 12/56　400Z
G06F 13/00 (2006.01)　G06F 13/00　351Z

請求項の数 9　(全 29 頁)

(21)出願番号　特願2007-326789 (P2007-326789)
(22)出願日　平成19年12月19日 (2007.12.19)
(65)公開番号　特開2009-152712 (P2009-152712A)
(43)公開日　平成21年7月9日 (2009.7.9)
審査請求日　平成22年11月26日 (2010.11.26)

(73)特許権者　000004226
日本電信電話株式会社
東京都千代田区大手町二丁目３番１号
(74)代理人　100079049
弁理士　中島　淳
(74)代理人　100084995
弁理士　加藤　和詳
(74)代理人　100099025
弁理士　福田　浩志
(74)代理人　100132115
弁理士　佐久間　顕治
(74)代理人　100151013
弁理士　大古　奈奈

最終頁に続く

(54)【発明の名称】高パケットレートフローのオンライン検出方法およびそのためのシステムならびにそのためのプログラム

(57)【特許請求の範囲】
【請求項１】
設定条件入力手段、パラメタ設計手段、高パケットレートフロー検出手段を具備するコンピュータの処理により、パケットサンプリングによって得られた統計データのみを用いて、所定の測定期間内におけるパケット数Ｘが閾値ｘ＊個以上のフローを高パケットレートフローとして特定する高パケットレートフローのオンライン検出方法において、
前記設定条件入力手段により、パケットレートの閾値Ｒ、所定の測定期間内におけるパケット数Ｘと前記閾値ｘ＊が等しいフローの検出の検出見逃し許容確率ε、およびフローの発生から検出までに要する検出許容時間T_{D_max}を入力するステップと、
前記パラメタ設計手段により、前記所定の測定期間内におけるパケット数Ｘと前記閾値ｘ＊が等しいフローが特定される確率が１-ε以上となるという第１の制約条件で、高パケットフローであることを特定するサンプリングデータのパケット数閾値ｙ＊を最大の整数に設定するステップと、
前記高パケットレートフロー検出手段により、パケットフローを所定の測定期間の間、所定のパケットサンプリングレートｆでサンプリングするステップと、パケットフローごとのサンプルパケット数ｙを計測するステップと、前記パケット数ｙが前記パケット数閾値ｙ＊以上の場合に、前記サンプリングしたパケットフローを高パケットレートフローであると特定するステップを有する
ことを特徴とする高パケットレートフローのオンライン検出方法。
【請求項２】

請求項１に記載の高パケットレートフローのオンライン検出方法において、
前記パケットフローごとのサンプルパケット数を計測する測定期間として、一定時間長T_{SW}のスライディングウインドウ方式を用い、自然数Ｋに対してT_{SW}/K刻みでウインドウをＫ個に分割したベーシックウインドウを設け、T_{SW}/Kごとにベーシックウインドウを単位としたデータ更新を行うステップを有することを特徴とする高パケットレートフローのオンライン検出方法。
【請求項３】
請求項２に記載の高パケットレートフローのオンライン検出方法において、
前記第１の制約条件に加え、これらのフローの発生から検出までに要する時間を上限値T_{D_max}以下とすること、ならびにスライディングウインドウの解析に要する時間をベーシックウインドウの時間T_{SW}/K以下とすること、を第２の制約条件として、低レートフローの誤検出確率P_{WD}を最小化するように、前記T_{SW}，ｆ，Ｋを設計するステップを有することを特徴とする高パケットレートフローのオンライン検出方法。
【請求項４】
請求項３に記載の高パケットレートフローのオンライン検出方法において、
前記低レートフローの誤検出確率を最小化するステップの代わりに、パケットサンプリングレートｆとウインドウサイズT_{SW}との積を最大化することにより、前記T_{SW}，ｆ，Ｋを最適設計するステップを有することを特徴とする高パケットレートフローのオンライン検出方法。
【請求項５】
設定条件入力手段、パラメタ設計手段、高パケットレートフロー検出手段を具備し、パケットサンプリングによって得られた統計データのみを用いて、所定の測定期間内におけるパケット数Ｘが閾値x^*個以上のフローを高パケットレートフローとして特定する高パケットレートフローのオンライン検出システムにおいて、
前記設定条件入力手段は、パケットレートの閾値Ｒ、所定の測定期間内におけるパケット数Ｘと前記閾値x^*が等しいフローの検出の検出見逃し許容確率ε、およびフローの発生から検出までに要する検出許容時間T_{D_max}を入力する手段であり、
前記パラメタ設計手段は、前記所定の測定期間内におけるパケット数Ｘと前記閾値x^*が等しいフローが特定される確率が１-ε以上となるという第１の制約条件で、高パケットフローであることを特定するサンプリングデータのパケット数閾値y^*を最大の整数に設定する手段であり、
前記高パケットレートフロー検出手段は、パケットフローを所定の測定期間の間、所定のパケットサンプリングレートｆでサンプリングし、パケットフローごとのサンプルパケット数ｙを計測し、前記パケット数ｙが前記パケット数閾値y^*以上の場合に、前記サンプリングしたパケットフローを高パケットレートフローであると特定する手段であることを特徴とする高パケットレートフローのオンライン検出システム。
【請求項６】
請求項５に記載の高パケットレートフローのオンライン検出システムにおいて、
前記パケットフローごとのサンプルパケット数を計測する測定期間として、一定時間長T_{SW}のスライディングウインドウ方式を用い、自然数Ｋに対してT_{SW}/K刻みでウインドウをＫ個に分割したベーシックウインドウを設け、T_{SW}/Kごとにベーシックウインドウを単位としたデータ更新を行う手段を有することを特徴とする高パケットレートフローのオンライン検出システム。
【請求項７】
請求項６に記載の高パケットレートフローのオンライン検出システムにおいて、
前記第１の制約条件に加え、これらのフローの発生から検出までに要する時間を上限値T_{D_max}以下とすること、ならびにスライディングウインドウの解析に要する時間をベーシックウインドウの時間T_{SW}/K以下とすること、を第２の制約条件として、低レートフローの誤検出確率P_{WD}を最小化するように、前記T_{SW}，ｆ，Ｋを設計する手段を有することを特徴とする高パケットレートフローのオンライン検出システム。

【請求項８】
請求項７に記載の高パケットレートフローのオンライン検出システムにおいて、
前記低レートフローの誤検出確率を最小化する手段の代わりに、パケットサンプリングレートｆとウインドウサイズT_{SW}との積を最大化することにより、前記T_{SW}，ｆ，Ｋを最適設計する手段を有することを特徴とする高パケットレートフローのオンライン検出システム。
【請求項９】
コンピュータを、請求項５から８のいずれかに記載の高パケットレートフローのオンライン検出システムにおける各手段として機能させるためのプログラム。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、フロー計測から得られたデータを用いて、高パケットレートフローを特定する技術に係り、特に、実際のパケットそのものからの情報は用いないで、パケットサンプリングによって得られた統計データのみを用いることによって、設定した時間内に十分高い確率で高パケットレートフローを検出する高パケットレートフローのオンライン検出技術に関するものである。
【背景技術】
【０００２】
近年、インターネット上で公的機関や企業のサーバを狙ったSenial of Service（ＤｏＳ）攻撃が深刻な問題となっている。ＤｏＳ攻撃とは、サーバがクライアントに対して供給するサービスを不正なパケットを送りつけることによって妨害するという、ネットワークを利用した攻撃のことである。
【０００３】
そのＤｏＳ攻撃の中で現在最も頻繁に発生しているのがＳＹＮ Ｆｌｏｏｄ攻撃である。これは、攻撃者が攻撃対象のサーバに対して、ＴＣＰの接続要求であるＳＹＮパケットのヘッダを改竄して大量に送りつけるというものである。
【０００４】
ＳＹＮパケットを受け取ったサーバは送信元に対してＳＹＮ/ＡＣＫを返す。しかしＳＹＮパケットのヘッダに書かれている送信元のＩＰアドレスが実際には存在しないアドレスに書き換えられているため、サーバからのＳＹＮ/ＡＣＫに対してＡＣＫを返すクライアントは存在せず、サーバは返ってこないＡＣＫをタイムアウトになるまで待ち続けなければならない。
【０００５】
この状態はhalf-openと呼ばれ、half-open状態のコネクションはサーバ内のbacklog queueに蓄積される。backlog queueのサイズはサーバ毎に決められており、このbacklog queueが一杯のときは、サーバはクライアントからの接続要求に応えることができない。
【０００６】
すなわち、ＩＰアドレスを改竄したＳＹＮパケットが大量に送られてくると、サーバのbacklog queueは常に一杯の状態になってしまい、正常なクライアントに対してＴＣＰ接続を確立することができず、サービスを供給できなくなる。
【０００７】
またＳＹＮ Ｆｌｏｏｄ攻撃の他にも、サイズの大きいＵＤＰパケットなどを大量に送り続けサーバ付近の帯域を占有してしまう攻撃なども存在する。
【０００８】
いずれの場合も攻撃者本人のコンピュータからの単一の攻撃ではなく、多くの場合、トロイの木馬などによって攻撃者の支配下に陥ってしまった多数のコンピュータを踏み台とし、一斉に攻撃を仕掛けるDistributed ＤｏＳ（ＤＤｏＳ）攻撃が用いられる。
【０００９】
ＤＤｏＳ攻撃では多数のコンピュータからパケットが送信されるため、個々の送信する

パケットの量がそれほど多くなくても、標的となったホストに到着するパケット数は膨大となる。

【００１０】

ＳＹＮ Ｆｌｏｏｄ攻撃の場合は、すぐにbacklog queueが一杯になってしまい、その後サーバ管理者が気づく、あるいは知らされるまで、サービスが停止した状態が続くことになってしまう。そのためＤｏＳ攻撃が発生したときには、いち早くその発生を検出することが重要である。

【００１１】

ＤｏＳ攻撃の検出に関連する研究としては、下記非特許文献１および非特許文献２に記載されたものがある。

（ａ）非特許文献１において、Sirisらはトラヒックに含まれるＳＹＮパケットの数を計測し、２種類のアルゴリズムを用いて動的に閾値を定め、閾値を超えるＳＹＮパケットが計測された場合にＳＹＮ Ｆｌｏｏｄ攻撃の発生を検出するという手法を提案している。

【００１２】

彼らの手法は、ＳＹＮ Ｆｌｏｏｄ攻撃が発生していることは検知できても、ＳＹＮ Ｆｌｏｏｄ攻撃のフローを特定することができないという点、およびトラヒックのデータを全て計測対象としている点で本発明とは異なる。

【００１３】

また、ＳＹＮ Ｆｌｏｏｄ攻撃のみを対象としている点が、本発明の高パケットレートのＤｏＳ攻撃全てを対象としている点と異なる。

【００１４】

（ｂ）また、非特許文献２において、大下らはＳＹＮ Ｆｌｏｏｄ攻撃を検出する手法を提案している。彼らの手法はbacklog queueのサイズとタイムアウトになる時間を考慮し、サーバがサービス停止状態になる前に検出を行う。

【００１５】

しかし、トラヒックを観測するポイントが、本発明ではネットワークのバックボーンを想定しているのに対し、彼らはサーバ側のネットワークへのインターフェース部分を想定しており、さらにトラヒックの全パケットを観測している点で本発明とは異なっている。

【００１６】

【非特許文献１】V.A.Siris and F.Papagalou、”Application of Anomaly Detection Algorithms for Detecting SYN Flooding Attacks,”Computer Communications,vol.29,issue9,pp.1433--1442,May 2006.

【非特許文献２】Y.Ohsita,S.Ata and M. Murata,”Detecting Distributed Denial of Service Attacks by Utilizing Statistical Analysis of TCP SYN Packets, ”IEICE Technical Report, vol. 103, no. 651, pp. 23－－28, February 2004.

【発明の開示】

【発明が解決しようとする課題】

【００１７】

本発明では、ＤｏＳ攻撃フローのパケットレートが正常なフローのパケットレートに比べて非常に大きいことに注目し、フロー毎のパケットレートを用いてネットワークのバックボーンルータにおいてＤｏＳ攻撃を検出することを試みる。

【００１８】

ＤｏＳ攻撃の検出を行う機能をバックボーンルータに実装させる利点としては、端末や端末に近いルータへ実装させる場合に比べて実装箇所が少なくて済むためコストが抑えられることや、ＤｏＳ攻撃が検出された場合に、すぐに攻撃パケットの破棄や攻撃元の特定などの対抗策が取れることなどが挙げられる。

【００１９】

ここで、ＤｏＳ攻撃のフローが正常なフローよりもパケットレートが大きいことを前提とするが、その値には様々なものが予想される。また、サーバが機能を停止するぎりぎりのレートから、１秒あたりのパケット数が数十万から数百万にも達するような高レートま

でのすべてのＤｏＳ攻撃を、パケットレートの大きさのみを用いて誤りなく検出することは不可能である。

【００２０】

そこで本発明の目的は、ＤＤｏＳ攻撃のように到着レートが非常に大きく、サーバ側での対処が難しい高パケットレートのＤｏＳ攻撃に注目し、単位時間あたりの到着パケット数が予め定めた閾値を越えるような高パケットレートフローをＤｏＳ攻撃等の異常トラヒックの候補として、設定した時間内に十分高い確率で検出できる高パケットレートフローのオンライン検出技術を提供することである。

【課題を解決するための手段】

【００２１】

本発明は、上記目的を達成するために、次のような構成を採用している。

ａ）パケットサンプリングによって得られた統計データのみを用いて、所定の測定期間内におけるパケット数Ｘが閾値x^*個以上のフローを高パケットレートフローとして特定するために、パケットレートの閾値Ｒ、所定の測定期間内におけるパケット数Ｘと前記閾値x^*が等しいフローの検出の検出見逃し許容確率ε、およびフローの発生から検出までに要する検出許容時間T_{D_max}を予め入力しておき、前記所定の測定期間内におけるパケット数Ｘと前記閾値x^*が等しいフローが特定される確率が１-ε以上となるという第１の制約条件で、高パケットフローであることを特定するサンプリングデータのパケット数閾値y^*を最大の整数に設定し、パケットフローを所定の測定期間の間、所定のパケットサンプリングレートｆでサンプリングしてパケットフローごとのサンプルパケット数ｙを計測し、前記パケット数ｙが前記パケット数閾値y^*以上の場合に、前記サンプリングしたパケットフローを高パケットレートフローであると特定するようにしたものである。

【００２２】

さらに、前記パケットフローごとのサンプルパケット数を計測する測定期間として、一定時間長T_{SW}のスライディングウインドウ方式を用い、自然数Ｋに対してT_{SW}/K刻みでウインドウをＫ個に分割したベーシックウインドウを設け、T_{SW}/Kごとにベーシックウインドウを単位としたデータ更新を行うようにしている。

【００２３】

さらに、前記第１の制約条件に加え、これらのフローの発生から検出までに要する時間を上限値T_{D_max}以下とすること、ならびにスライディングウインドウの解析に要する時間をベーシックウインドウの時間T_{SW}/K以下とすること、を第２の制約条件として、低レートフローの誤検出確率P_{WD}を最小化するように、前記T_{SW}，ｆ，Ｋを設計するようにした。

【００２４】

さらに、前記低レートフローの誤検出確率を最小化するステップの代わりに、パケットサンプリングレートｆとウインドウサイズT_{SW}との積を最大化することにより、前記T_{SW}，ｆ，Ｋを最適設計するようにしている。

【００２５】

また、本発明に係るプログラムは、コンピュータを上記の各処理を実行させるためのプログラムである。

【発明の効果】

【００２６】

本発明は、パケットサンプリングによって得られた情報のみを用いて、測定期間のパケット数が閾値以上の高パケットレートフローをリアルタイムに検出できるという効果を有する。

【発明を実施するための最良の形態】

【００２７】

図１は、本発明の高パケットレートフローのオンライン検出方法の実施の形態の一例を示すシステム構成図である。図１において、１０１は設定条件入力手段、１０２はパラメタ設計手段、１０３は高パケットレートフロー検出手段、１０４は高パケットレートフロ

ーリストである。本システムは、通常のコンピュータ構成を有しており、本発明に係る高パケットレートフローのオンライン検出方法を実現するためのプログラムをＣＰＵ、メモリなどのハードウェアを用いて実行することにより、本発明を実現する。

【００２８】

設定条件入力手段１０１により特定閾値パケットレートＲなどの設定条件が入力され、パラメタ設計手段１０２によりウインドウサイズT_{sw}やサンプリングレートｆなどが設計される。設計されたパラメタを用いて高パケットレートフロー検出手段１０３により高パケットレートフローが検出され、検出された高パケットレートフローを高パケットレートフローリスト１０４（例えば、ハードディスクなどの記憶手段に格納）に出力する。

【００２９】

次に、本発明の実施の形態に係る高パケットレートフローのオンライン検出方法について説明する。

【００３０】

（Ａ）＜高パケットレートフローのオンライン検出手法＞

ＤｏＳ攻撃等の異常トラヒックが発生した際、ネットワークのバックボーンにおいて迅速に検知することは、ネットワーク全体を保守管理する上で非常に重要である。

【００３１】

本発明ではＤＤｏＳ攻撃等、特に高パケットレートの異常トラヒックに注目し、パケットレートが予め定められた閾値を超えるフローを異常トラヒックの候補として予め定められた時間内に検出することを試みる。

【００３２】

（Ａ１）＜検出の枠組み＞

まず本発明で扱うフローの定義を行う。ネットワークを流れるトラヒックにおいてフローを定義するものとしてＩＰアドレス，ポート番号，ＳＹＮフラグなどがあるが、本発明では一般のＤｏＳ攻撃を想定し、同一の宛先ＩＰアドレスを持つパケット群をフローと定義する。

【００３３】

フロー毎のパケットレートを求めるためには、測定時間とフロー毎のパケット数という二つの情報が必要である。しかし転送されている全てのパケットを解析して統計量を収集することは、処理能力やメモリなどの面で限界がありスケーラビリティに欠ける。

【００３４】

そこで、これらの問題を解決する手段としてパケットサンプリングを用いる。この手法では、パケットの標本抽出を行い、得られた情報を元にサンプリングの対象となった母集団の統計量を推定するため、処理サイクル、メモリ使用量を押さえることができる。

【００３５】

本発明では、各パケットに対してフローの情報を用いず、独立に一定の確率ｆでサンプリングを行うランダムパケットサンプリングという手法を用いる。これにより処理サイクルを大幅に抑えることができ、バックボーンなどの高速な回線にも適用可能となる。

【００３６】

しかしパケットサンプリングはその性質上、情報の欠如をもたらし、特にサンプルする頻度が少ない場合、サンプリングの対象である母集団の統計量を推定することが困難になる。例えば、一つのパケットもサンプルされないフローについては、母集団における統計量を推定することは非常に困難である。

【００３７】

しかし、通常のトラヒックに含まれるフローに比べてはるかに高いパケットレートをもつフローに対しては、適切なサンプリングレートを用いることによって、十分、母集団を推定できるだけの標本を抽出できると考えられる。

【００３８】

また、トラヒックを監視し続けるためには常にデータの取得，解析，廃棄を行わなければならない。このオンライン処理を実現するために、スライディングウインドウ方式を用

いる。スライディングウインドウ方式とは、解析データを保持するスライディングウインドウをベーシックウインドウと呼ばれるユニットに分割し、そのベーシックウインドウ単位でデータを更新する、オンライン処理アルゴリズムである。

【００３９】

（Ａ２）＜ランダムパケットサンプリングを用いた検出＞

まず、ランダムパケットサンプリングの対象となるｍ個のフローからなるパケット群を想定し、このパケット群を母集団として、各パケットが独立に一定の確率ｆでサンプリングされることを考える。

【００４０】

Ｘ_ｊとＹ_ｊ（ｊ=1,2,・・,ｍ）をそれぞれｊ番目のフローに含まれる、母集団におけるパケット数ならびにサンプルされたパケット数と定義する。ｊ番目のフローにおいてＸ_ｊ=ｘであったとき、ｙ個のパケットがサンプルされる確率を示す数式１は数式２のような二項分布で与えられる。

【００４１】

【数１】

$$q(y \mid x) = \Pr[Y_j = y \mid X_j = x]$$

【００４２】

【数２】

$$q(y \mid x) = \begin{pmatrix} x \\ y \end{pmatrix} f^y (1-f)^{x-y} \quad (y = 0, 1, \cdots, x)$$

【００４３】

異常トラヒックの候補として検出する際の閾値となる母集団におけるパケットレートをＲ[packet/秒]とおくと、母集団の全パケットの測定にかかった時間がＴ[秒]であったとき、それらの積を求めることによって、検出したいフローの母集団における閾値をパケットレートからパケット数に変換することができる。ここでパケット数の閾値をｘ＊とおく。しかし測定時間が実数であるため、自然数であるｘ＊は次の数式３に示す床関数（floor function）により定められる。

【００４４】

【数３】

$$x^* = \lfloor RT \rfloor$$

【００４５】

すなわち、ある測定時間Ｔの間に通過した全パケットで構成される母集団において、ｘ＊個以上のパケットからなるフローを検出することが目的となる。すなわち、ランダムパケットサンプリングによって得られたデータから、母集団においてｘ＊個以上のパケットからなるフローを見つけなければならない。

【００４６】

ここで、ｙ＊個以上サンプルされたフローを異常トラヒックの候補として検出することにする。本発明は、異常トラヒックが発生した際、確実に検出することが目標であるが、パケットサンプリングによって情報が欠如するため、対象フローが検出される確率を1にすることは不可能である。

【００４７】

そこで、母集団におけるパケット数がｘ＊個以上であるフローが、十分高い確率で検出されるようにｙ＊を定める。具体的には、母集団においてｘ＊個のパケットが含まれるフ

ローを、検出できずに見逃してしまう確率を十分小さな確率（検出見逃し許容確率）ε以下にする。言い換えると、パケットレートがR[packet/秒]であるフローが、1-ε以上の確率で検出されるようにｙ＊を定める。よってｙ＊は次式を満たさなければならない。

【００４８】

【数４】

$$\Pr[Y_j \geq y^* \mid X_j = x^*] \geq 1 - \epsilon$$

【００４９】

もし、ｙ＊が数式４を満たすならば、パケットレートがRより大きいフローの検出確率は1-εより大きくなることに注意する。なお、数式４の左辺は数式２を用いて以下の式で表される。

【００５０】

【数５】

$$\Pr[Y_j \geq y^* \mid X_j = x^*] = \sum_{y=y^*}^{x^*} q(y \mid x^*) = 1 - \sum_{y=0}^{y^*-1} q(y \mid x^*)$$

【００５１】

ｙ＊を定める際には、パケットレートがR[packet/秒]未満の低レートフローがｙ＊個以上サンプルされてしまい、誤って検出されてしまう確率も考慮する。異常トラヒックの候補として検出されたフローが、実際に異常トラヒックであるかどうかを確かめる作業はかなりのオーバーヘッドがあるため、低レートフローの誤検出確率はできるだけ小さくしなければならない。この誤検出確率はｙ＊に関して減少関数である。そこで数式４を満たす自然数の中で最大のものをｙ＊とすると、ｙ＊は下記数式６で表わされる。

【００５２】

【数６】

$$y^* = \max_y \{y;\ \Pr[Y_j \geq y \mid X_j = x^*] \geq 1-\epsilon\} = \max_y \left\{ y;\ \sum_{i=0}^{y-1} q(i \mid x^*) \leq \epsilon \right\}$$

【００５３】

サンプルデータにおけるパケット数の閾値ｙ＊は、予め与えられるパケットレートの閾値Ｒ，検出見逃し許容確率εに加えて、サンプリングレートｆと測定時間Ｔが与えられれば一意に定めることができる。

【００５４】

（Ａ３）＜スライディングウインドウ方式によるデータ更新＞

インターネットの回線を流れているトラヒックを常に測定し、対象となるフローの検出を行うにはオンラインのアルゴリズムが必要である。すなわち、データの取得，解析，破棄を常時繰り返し行う必要がある。

【００５５】

データの解析は、測定時間Ｔ[秒]の間にサンプルされた各フロー毎のパケット数がｙ＊個以上であるか否かを判定し、ｙ＊個以上であるフローを検出するという作業を指すので、問題となるのは解析対象となるデータをどのようにして取得および破棄を行い、更新していくかということである。本発明では、解析対象となるデータの測定時間をT_{sw}秒に固定した、スライディングウインドウ方式を用いてデータの更新を行う。

【００５６】

＜スライディングウインドウ方式の概要＞

スライディングウインドウ方式とは、解析対象となるデータを保持するスライディングウインドウをベーシックウインドウと呼ばれる複数のユニットに分割し、解析終了後に最も古いベーシックウインドウのデータを破棄し、新たにサンプルされた1ベーシックウインドウ分のデータを加えることによって解析対象のデータを更新する方式である。

【００５７】

スライディングウインドウ方式には、ウインドウのサイズをパケット数で規定する方法と測定時間で規定する方法の二種類が存在する。前者は一定数のパケットが回線を通過したとき、あるいは一定数のパケットがサンプルされたときにベーシックウインドウを生成し、スライディングウインドウを更新する。

【００５８】

母集団におけるパケット数を一定にすると、トラヒックのフロー毎のパケット数分布などを求めやすくなったり、あるいはサンプル数を一定にすると、メモリの使用量を一定にすることができるなどのメリットがある。

【００５９】

一方、一定時間毎にベーシックウインドウを生成しスライディングウインドウを更新すると、母集団におけるパケット数およびサンプルされるパケット数はスライディングウインドウが更新される度に変わるが、測定時間を一定にすることができる。

【００６０】

本発明では、パケットレートの閾値Ｒ，検出見逃し許容確率ε，サンプリングレートｆ，そして測定時間を用いて、サンプルされたデータにおけるパケット数の閾値ｙ＊を求める。測定時間が一定ならばスライディングウインドウが更新されてもｙ＊は一定であるので、閾値を計算し直す必要がなくなり、オンラインのアルゴリズムとしては都合が良い。

【００６１】

また、トラヒックのフロー毎のパケット数分布なども必要としないため、測定時間を一定にしたスライディングウインドウ方式が適している。

【００６２】

そこで本発明では、スライディングウインドウが保持する解析対象データであるサンプルされたパケットの測定時間が、一定時間T_{SW}秒になるようにスライディングウインドウのサイズを規定する。また自然数Ｋを用いて、スライディングウインドウをT_{SW}/K秒刻みでＫ個のベーシックウインドウに分割する。

【００６３】

すなわち、T_{SW}/K秒毎にサンプルされたパケットで新たなベーシックウインドウを生成し、スライディングウインドウに加えると共に、スライディングウインドウ内の最も古いベーシックウインドウのデータを破棄することによって、解析対象となるデータの更新を行う。以上の手続きを下記にまとめる。

【００６４】

ａ）Srep１：一定の確率ｆで母集団からパケットをサンプルする。

ｂ）Step２：ベーシックウインドウの境界にきたら、T_{SW}秒間に取得したデータ（スライディングウインドウ）に対して、ｙ＊を超えるフローが存在するかどうかを調べる。

ｃ）Step３：新たにT_{SW}/K秒間に取得したデータ（ベーシックウインドウ）をスライディングウインドウに加え、スライディングウインドウ内の一番古いベーシックウインドウのデータを破棄する。

ｄ）Step４：上記Step２とStep３を繰り返す。

【００６５】

図２－１は、図１に示したシステムを用いて実行される高パケットレートフローのオンライン検出方法の処理の流れを示すフローチャートである。

図２－１に示すように、まず、予め設定条件入力手段１０１からパケットレートの閾値Ｒ，検出見逃し許容確率εなどを設定する。次に、パラメタ設計手段１０２により、一定時間T_{SW}（スライディングウインドウのサイズ），サンプリングレートｆ，ベーシックウインドウの数Ｋなどのパラメタを設計する（ステップＳ１０）。

【００６６】
次に、一定のサンプリングレートｆで母集団からパケットをサンプルする（ステップＳ１１）。ベーシックウインドウの境界でなければ（ステップＳ１２：Ｎ）ステップＳ１１に戻り、ベーシックウインドウの境界にきたら（ステップＳ１２：Ｙ）、次にT_{SW}秒間に取得したデータ（スライディングウインドウ）に対して、y^*を超えるフローが存在するかどうかを調べ（ステップＳ１３）、y^*を超えるフローが存在したら（ステップＳ１３：Ｙ）、それらのフローを高パケットレートフローとして検出し、高パケットレートフローリスト１０４に出力する（ステップＳ１５）。
【００６７】
y^*を超えるフローが存在しない場合（ステップＳ１３：Ｎ）、あるいはステップＳ１５の後、新たにT_{SW}/K秒間に取得したデータ（ベーシックウインドウ）をスライディングウインドウに加え、スライディングウインドウ内の一番古いベーシックウインドウのデータを破棄することによってサンプルデータを更新した後（ステップＳ１４）、ステップＳ１１に戻る。
【００６８】
＜スライディングウインドウ方式の利点＞
オンライン処理のためのデータ更新の代表的な手法として、スライディングウインドウ方式の他にジャンピングウインドウ方式が知られている。この方式は、解析が終了したデータは全て破棄し、次に解析するデータを一から取得し直すという方式であり、GR2000などのルータで実装されている。
【００６９】
しかしこの方式は、スライディングウインドウ方式においてベーシックウインドウの数Ｋを１にした場合と等しく、スライディングウインドウの特別な場合として考えることができる。ジャンピングウインドウ方式とベーシックウインドウの数Ｋが２以上のスライディングウインドウ方式を比較したときに最も違いが現れてくるのが、高パケットレートのフローが発生した時点から検出されるまでにかかる時間である。
【００７０】
例えば、ジャンピングウインドウ方式において、パケットレートＲ[packet/秒]のフローがある時点のウインドウの途中から発生したとき、そのウインドウ終了時において当該フローを検出できる確率は$1-\varepsilon$に満たない。$1-\varepsilon$以上の確率で検出されるようになるのは次のウインドウであり、最悪の場合一つのジャンピングウインドウ（スライディングウインドウ）分の時間だけ検出が遅れてしまうことになる。
【００７１】
しかし複数個のベーシックウインドウからなるスライディングウインドウであれば、この遅れを一つのベーシックウインドウ分で済ませることができる。すなわちベーシックウインドウの数Ｋが大きいほど、この遅れは小さくなる。よって、高パケットレートの異常トラヒックが発生してから予め定められた時間内に検出を行うには、ベーシックウインドウの数Ｋは大きいほど都合が良い。
【００７２】
しかし、オンラインアルゴリズムとして正常に動作させるには、新しいベーシックウインドウが生成される前にスライディングウインドウの解析を終了させなければならない。ベーシックウインドウの数Ｋを大きくするとベーシックウインドウの取得にかかる時間が短くなり、スライディングウインドウの解析に割くことができる時間が短くなる。
【００７３】
そのためベーシックウインドウの数Ｋは予め定められた時間内に高パケットレートの異常トラヒックを検出するという条件と、スライディングウインドウ方式を用いた本手法がオンラインのアルゴリズムとして正常に機能するという条件を考慮して、適切に選ぶ必要がある。
【００７４】
（Ａ４）＜パラメタの値による影響＞

本発明では、予め与えられるパケットレートの閾値Ｒ[packet/秒]と検出見逃し許容確率εの他に、スライディングウインドウのサイズであるT_{SW}とサンプリングレートｆ、そしてスライディングウインドウ内におけるベーシックウインドウの数であるＫを定める必要がある。

【００７５】

図３は、これら三つのパラメタ（T_{SW}，ｆ，Ｋ）のうち二つを固定し、パケットレートの閾値Ｒ[packet/秒]のフローを見逃す確率を検出見逃し許容確率ε以下にする条件を満たしたまま、残りの一つを変化させた場合に該パラメタ値が小の場合と大の場合とでそれぞれどのような利点があるかをまとめた図である。

【００７６】

同図に示すように、スライディングウインドウのサイズT_{SW}が小の場合は使用メモリを軽減することができ、また検出までの時間を短縮できるという利点があり、大の場合は低レートフローの誤検出確率を低下させることができるという利点がある。

【００７７】

また、サンプリングレートｆが小の場合はＣＰＵにかかる負荷を低減できるという利点があり、大の場合は低レートフローの誤検出確率を低下させることができるという利点がある。

【００７８】

また、ベーシックウインドウの数Ｋが小の場合はＣＰＵにかかる負荷を低減できるという利点があり、大の場合は検出までにかかる時間を短縮することができるという利点がある。

【００７９】

なお、スライディングウインドウのサイズT_{SW}，サンプリングレートｆ，ベーシックウインドウの数Ｋの３つのパラメタは、予め定められた時間内に検出するという条件と、オンラインのアルゴリズムとして正常に動作するという条件を満たすように選ぶ必要があることに注意する。

【００８０】

（Ｂ）＜制御パラメタの最適化手法＞

パケットレートＲ[packet/秒]をもつフローの発生後、検出許容時間以内に１-ε以上の確率で検出することを保証した上で、パケットレートが閾値未満の低レートフローが誤って検出されてしまう確率を最小化するような制御パラメタの設定法を提案する。その際、検出に要する時間とオンラインアルゴリズムとして機能するためのスライディングウインドウの計算時間に関して制約条件を設ける。すなわち、次のような最適化問題を考察する。

【００８１】

目的関数：低レートフロー誤検出確率→最小

制約条件：パケットレートＲをもつフローの検出確率≧１-ε

パケットレートＲをもつフローの発生から検出までに要する時間≦T_{D_max}

オンラインアルゴリズムとして機能すること

【００８２】

以下では、この問題が近似的に非線形混合計画問題として定式できること、その非線形計画問題が大域的最適解を持つこと、ならびに最適解の導出について述べる。

【００８３】

（Ｂ１）＜制御パラメタと制約条件＞

高パケットレートをもつ異常トラヒックの検出方法は、予め与えられるパケットレートの閾値Ｒ[packet/秒]，検出見逃し許容確率ε，ならびに検出許容時間T_{D_max}[秒]の他に、以下の制御パラメータを設定する必要がある。

【００８４】

ａ）スライディングウインドウのサイズ：T_{SW}[秒]

ｂ）サンプリングレート：ｆ
ｃ）ベーシックウインドウ数：Ｋ
この三つの制御パラメタのうち、スライディングウインドウのサイズＴ$_{SW}$とサンプリングレートｆが与えられると、サンプルされたデータにおけるパケット数の閾値ｙ*は、数式６により一意に定めることができる。
【００８５】
制御パラメタを設定する際に考慮しなければならないのが、定めた時間Ｔ$_{D_max}$以内に対象のフローを検出するという条件と、検出のアルゴリズムがオンラインのアルゴリズムとして正常に動作するという条件である。ここでシステムを規定するために以下のようなパラメタを導入する。
【００８６】
・対象回線の最大パケットレートＣ$_{max}$[packet/秒]
・１サンプルパケット当たりの解析処理に必要な時間Δ$_{1}$[秒]
・１スライディングウインドウの解析に要する、サンプルパケット数とは独立な時間Δ$_{2}$[秒]
【００８７】
スライディングウインドウ方式を用いて検出を行う際に、対象フローが発生してから検出されるまでにかかる時間の最大値Ｔ$_{D}$[秒]は、スライディングウインドウの解析にかかる時間をτ[秒]とおくと、下記数式７で表される。
【００８８】
【数７】

$$T_{\mathrm{D}} = T_{\mathrm{SW}} + T_{\mathrm{SW}}/K + \tau$$

【００８９】
数式７の右辺の各項は順に、１スライディングウインドウ分のパケットの測定時間，１ベーシックウインドウ分のパケットの測定時間，そしてスライディングウインドウの解析にかかる時間を表している。
【００９０】
仮に、パケットレートがＲ[packet/秒]の検出対象フローが発生したとすると、その発生は多くの場合、ベーシックウインドウの途中から始まる。攻撃の開始を含むベーシックウインドウが、スライディングウインドウにおいて最も古いベーシックウインドウになるときを考えると、このときのスライディングウインドウの母集団ではＤｏＳ攻撃フローのパケット数は下記数式８には達しておらず、１-ε以上の確率で検出されることが保証されない。
【００９１】
【数８】

$$x^{*} = \lfloor RT_{\mathrm{SW}} \rfloor$$

【００９２】
１-ε以上の確率で検出されるようになるのは、攻撃の開始を含むベーシックウインドウが破棄されたあとのスライディングウインドウにおいてである。
【００９３】
そのため検出対象のフローが初めて回線を通過したときから検出されるまでには、最悪の場合1スライディングウインドウ分の測定時間Ｔ$_{SW}$と１ベーシックウインドウ分の測定時間Ｔ$_{SW}$/Ｋ、そしてスライディングウインドウの解析にかかる時間の合計時間分だけかかることになる。
【００９４】

また、スライディングウインドウの解析に要する時間の平均値τ[秒]は、トラヒック全体のパケットレートC[packet/秒]を用いて次のような式で表される。

【００９５】

【数９】

$$\tau = fC(T_{SW}/K)\Delta_1 + \Delta_2$$

【００９６】

スライディングウインドウの解析にかかる時間は、サンプルされるパケット数に比例する時間（数式９の右辺第１項）と、サンプルされるパケット数とは無関係な定量的な計算時間（数式９の右辺第２項）に分けて考えることができる。

【００９７】

数式９の右辺第１項は１サンプルパケット当たりの処理時間Δ_1に、ベーシックウインドウの中にサンプルされるパケット数の期待値ｆＣ（T_{SW}/K）をかけたものである。

【００９８】

本発明で用いるスライディングウインドウ方式では、新しく生成されたベーシックウインドウ内に含まれるフロー毎のパケット数をカウントするとともに、宛先ＩＰアドレスに対してハッシュ化を行い、得られたハッシュ値を用いて予め用意しておいた配列にフロー毎のパケット数のデータを格納する。この処理にかかる時間が数式９の右辺第１項にあたる。

【００９９】

新しいベーシックウインドウの統計データが得られると、それをスライディングウインドウ全体の統計データに足し合わせ、同時に最も古いベーシックウインドウの統計データを差し引く。

【０１００】

ベーシックウインドウ数Ｋ個分の配列とスライディングウインドウの統計データを保持する配列を予め十分な大きさで用意しておくことで、処理の簡単化をはかると同時に、計算量のオーダがスライディングウインドウ全体のパケット数とは無関係のアルゴリズムを実現することができる。このデータの更新にかかる時間や、ｙ＊との比較などの処理にかかる時間は、サンプルされたパケット数とは無関係に数式９の右辺第２項として与えられる。

【０１０１】

一方、検出にかかる時間に関して、T_Dは次の条件を満たさなければならない。

【０１０２】

【数１０】

$$T_D \leq T_{D\_max}$$

【０１０３】

また、本発明がオンラインのアルゴリズムとして正常に動作させるためには、スライディングウインドウの解析を次の１ベーシックウインドウ分のパケットをサンプルする前に終わらせなければならない。

そのためτは、以下の数式１１で示す条件を満たす必要がある。

【０１０４】

【数１１】

$$\tau \leq T_{SW}/K$$

【０１０５】

数式７を数式１０へ、数式９を数式１１へそれぞれ代入すると、下記数式１２－１と数式１２－２で示すような二つの数式を得る。

【０１０６】

【数１２－１】

$$T_{\mathrm{SW}} + T_{\mathrm{SW}}/K + fC(T_{\mathrm{SW}}/K)\Delta_1 + \Delta_2 \leq T_{\mathrm{D\_max}}$$

【０１０７】

【数１２－２】

$$fC(T_{\mathrm{SW}}/K)\Delta_1 + \Delta_2 \leq T_{\mathrm{SW}}/K$$

【０１０８】

このとき、トラヒック全体のパケットレートＣ[packet/秒]がどのような値であっても、数式１２－１と数式１２－２が成り立つように、回線の最大パケットレートC_{max}[packet/秒]を用いて下記数式１３のようにする。

【０１０９】

【数１３】

$$T_{\mathrm{SW}} + T_{\mathrm{SW}}/K + fC_{\max}(T_{\mathrm{SW}}/K)\Delta_1 + \Delta_2 \leq T_{\mathrm{D\_max}}$$

$$fC_{\max}(T_{\mathrm{SW}}/K)\Delta_1 + \Delta_2 \leq T_{\mathrm{SW}}/K$$

さらに、これらを変形すると下記数式１４－１と数式１４－２が得られる。

【０１１０】

【数１４－１】

$$(K + 1 + C_{\max}\Delta_1 f)T_{\mathrm{SW}} - K(T_{\mathrm{D\_max}} - \Delta_2) \leq 0$$

【０１１１】

【数１４－２】

$$(C_{\max}\Delta_1 f - 1)T_{\mathrm{SW}} + K\Delta_2 \leq 0$$

【０１１２】

上記の制約が成立していれば、一時的なパケットバーストが起こったとしても、スライディングウインドウの解析が間に合わなくなるという事態を避けられる。この数式１４－１と数式１４－２が、制御パラメタ（スライディングウインドウのサイズT_{SW}，サンプリングレートｆ，ベーシックウインドウの数Ｋ）を設定する際の制約条件となる。

【０１１３】

ここで低レートフロー誤検出確率、すなわちパケットレートがλ（＜Ｒ）[packet/秒]であるフローが誤って検出されてしまう確率P_{WD}（λ）を下記数式１５のようにおく。

【０１１４】

【数１５】

$$P_{\mathrm{WD}}(\lambda) = \Pr[Y \geq y^* \mid X = \lambda T_{\mathrm{SW}}]$$

ただし、λＴ$_{SW}$は整数であると仮定している。このときＸおよびＹは、それぞれ、パケットレートがλ[packet/秒]であるフローの、スライディングウインドウ時間Ｔ$_{SW}$の間に回線を通過した、母集団におけるパケット数およびサンプルされたパケット数を表す確率変数である。

【０１１５】

そして制約条件である上記数式１４－１と数式１４－２の下で、任意のλに対して低レートフロー誤検出確率Ｐ$_{WD}$（λ）が最小になるように制御パラメタを設定する。

【０１１６】

（Ｂ２）＜低パケットレートフロー誤検出確率の最小化問題＞

上述した議論より、（Ｂ）の冒頭で述べた考察すべき最適化問題は以下のように記述できる。

【０１１７】

【数１６】

目的関数：低レートフロー誤検出確率 $P_{\mathrm{WD}}(\lambda) \longrightarrow$ 最小

制約条件：$T_{\mathrm{SW}} > 0$

$f > 0$

K は自然数

$(K + 1 + C_{\max}\Delta_1 f)T_{\mathrm{SW}} - K(T_{\mathrm{D\_max}} - \Delta_2) \leq 0$

$(C_{\max}\Delta_1 f - 1)T_{\mathrm{SW}} + K\Delta_2 \leq 0$

【０１１８】

この最適化問題を解くためには、目的関数である低パケットレートフロー誤検出確率Ｐ$_{WD}$（λ）を陽に表現する必要がある。その手段として、ポアソン分布を用いた近似を考える。二項分布Ｂ（Ｎ，ｆ）において、母集団のパケット数をＮ、サンプリングレートをｆとおいたとき、積Ｎｆを一定に保ちながら、Ｎ→∞，ｆ→０としたときに得られる極限分布はポアソン分布となる。この近似を用いると、パケットレートがＲ[packet/秒]であるフローから、スライディングウインドウサイズＴ$_{SW}$[秒]の間にパケットをｙ＊個以上サンプルする確率は下記数式１７のようになる。

【０１１９】

【数１７】

$$\Pr[Y \geq y^* \mid X = RT_{\mathrm{SW}}] = 1 - \sum_{y=0}^{y^*-1} e^{-RfT_{\mathrm{SW}}} \frac{(RfT_{\mathrm{SW}})^y}{y!}$$

ただし、ＲＴ$_{SW}$は整数であると仮定している。

【０１２０】

さらに、数式１７を用いてパケット数閾値ｙ＊を数式６によって求めると、パケット数閾値ｙ＊はサンプリングレートｆとスライディングウインドウサイズＴ$_{SW}$の積を引数とした関数ｙ＊（ｆＴ$_{SW}$）とみなすことができる。

【０１２１】

また、上記数式１５で定義された、パケットレートがＲ未満のλであるフローを誤って検出してしまう確率を、ポアソン分布による近似を用いて求めると下記数式１８で表される。

【０１２２】

【数１８】

$$P_{\mathrm{WD}}(\lambda)=\Pr[Y\geq y^{*}(fT_{\mathrm{SW}})\mid X=\lambda T_{\mathrm{SW}}]=1-\sum_{y=0}^{y^{*}(fT_{\mathrm{SW}})-1}e^{-\lambda fT_{\mathrm{SW}}}\frac{(\lambda fT_{\mathrm{SW}})^{y}}{y!}$$

ただし、λT_{SW}は整数であると仮定している。

【０１２３】

数式１８より低レートフロー誤検出確率P_{WD}（λ）は、サンプリングレートｆとスライディングウインドウサイズT_{SW}の積を引数とした関数として考えられる。このサンプリングレートｆとスライディングウインドウサイズT_{SW}の値が大きいほどサンプルされるパケット数は多くなり、情報量も増えるので推定の精度が良くなると考えられる。すなわちfT_{SW}を最大化することが、低レートフローの誤検出確率を最小化することになると推測される。

【０１２４】

パケット数閾値y^{*}は自然数の値をとるため、スライディングウインドウサイズT_{SW}に対して階段関数となり、低レートフロー誤検出確率P_{WD}（λ）は単調な減少関数とはならない。これはパケット数閾値y^{*}は自然数の値をとること、ならびに、サンプリングレートｆが固定されている場合、スライディングウインドウサイズT_{SW}が大きいほど低レートフローのサンプル数は多くなり、同一のy^{*}を取る範囲においては誤検出確率が高くなるためである。

【０１２５】

この性質があるため、fT_{SW}が最大のときに、低レートフロー誤検出確率P_{WD}が最小になるとは必ずしも言えない。また単調な減少関数でないため、数学的に証明することは困難であると考えられる。しかし低レートフロー誤検出確率P_{WD}はfT_{SW}の単調減少関数で上から押えられる。

【０１２６】

（Ｂ３）＜制御パラメタの近似最適化問題＞

ここまでの議論から、上記（Ｂ）で述べた最適化問題は、同じ制約下でfT_{SW}の最大化問題（あるいは-fT_{SW}の最小化問題）として書き換えられることが判明した。すなわち、上記（Ｂ）にある最適化問題は下記数式１９の非線形混合計画問題で近似的に定式化できることが明らかとなった。

【０１２７】

【数１９】

$$\begin{aligned}&\text{目的関数：}\ -fT_{\mathrm{SW}}\longrightarrow\text{最小}\\&\text{制約条件：}\ T_{\mathrm{SW}}>0\\&\qquad f>0\\&\qquad K\text{は自然数}\\&\qquad (K+1+C_{\max}\Delta_1 f)T_{\mathrm{SW}}-K(T_{\mathrm{D\_max}}-\Delta_2)\leq 0\\&\qquad (C_{\max}\Delta_1 f-1)T_{\mathrm{SW}}+K\Delta_2\leq 0\end{aligned}$$

【０１２８】

ここでは、Ｋ＞０，C_{max}，T_{D_max}＞０，Δ_1＞０，Δ_2＞０を予め与えられ

た定数とし、T_{D_max}，K，Δ_2の間には、下記数式２０が成立しているという仮定の下で、次の定理１を証明する。

【０１２９】

【数２０】

$$T_{\mathrm{D\_max}} - (K+2)\Delta_2 > 0$$

【０１３０】

＜定理１＞

ベーシックウインドウ数Kが予め与えられたとき、下記数式２１の最小化問題には大域的最小解（数式２２）が存在し、それは下記数式２３で与えられる。

【０１３１】

【数２１】

$$\begin{aligned} &\text{目的関数：} && -fT_{\mathrm{SW}} \longrightarrow \text{最小} \\ &\text{制約条件：} && T_{\mathrm{SW}} > 0 \\ & && f > 0 \\ & && (K+1+C_{\max}\Delta_1 f)T_{\mathrm{SW}} - K(T_{\mathrm{D\_max}} - \Delta_2) \le 0 \\ & && (C_{\max}\Delta_1 f - 1)T_{\mathrm{SW}} + K\Delta_2 \le 0 \end{aligned}$$

【０１３２】

【数２２】

$$(T^*_{\mathrm{SW}}, f^*)$$

【０１３３】

【数２３】

$$T^*_{\mathrm{SW}} = \frac{KT_{\mathrm{D\_max}}}{K+2}, \qquad f^* = \frac{T_{\mathrm{D\_max}} - (K+2)\Delta_2}{C_{\max}\Delta_1 T_{\mathrm{D\_max}}}$$

【０１３４】

（定理１の証明）

上記問題を解くため、$x = T_{SW}$，$y = f$とおき、条件$x > 0$，$y > 0$を取り除いた下記数式２４という緩和された最小化問題Pを考える。

【０１３５】

【数２４】

$$\begin{aligned} &\text{問題}P \quad \text{目的関数：} && g(x,y) = -xy \longrightarrow \text{最小} \\ &\text{制約条件：} && (a+by)x - c \le 0 \\ & && (by-1)x + d \le 0 \end{aligned}$$

ただし、$a = K+1 > 1$，$b = C_{max}\Delta_1 > 0$，

$c = K(T_{D\_max} - \Delta_2) > 0$，$d = K\Delta_2 > 0$

とした。なお、予め与えられているパラメタ間の条件である数式２０より、

ｃ＝ａｄ＞０
が成立することに注意する。
【０１３６】
最初に緩和問題Ｐに大域最適解が存在することを示す。仮に制約条件が等号で成立していると仮定すると、下記数式２５が成立する。
【０１３７】
【数２５】

$$x=\frac{c+d}{a+1}>0,\qquad y=\frac{c-ad}{b(c+d)}>0$$

すなわち、緩和問題Ｐは「-ｘｙ＜０」なる実行可能解をもつため、十分に小さな正定数ζ＞０に対して下記数式２６の最小化問題Ｐ’を考える。
【０１３８】
【数２６】

$$\begin{aligned}\text{問題 } P' \text{ 目的関数：}\ & g(x,y)=-xy \longrightarrow \text{最小}\\ \text{制約条件：}\ & (a+by)x-c\le 0\\ & (by-1)x+d\le 0\\ & -xy\le -\zeta\end{aligned}$$

【０１３９】
まず、２番目と３番目の制約条件より、
０≧（ｂｙ－１）ｘ＋ｄ≧ｂζ－ｘ＋ｄ
となるため、ｘ≧ｄ＋ｂζを得る。
【０１４０】
さらに、１番目と３番目の制約条件より、
ｃ≧（ａ＋ｂｙ）ｘ≧ａｘ＋ｂζ
となるため、（ｃ－ｂζ）/ａ≧ｘ≧ ｄ＋ζである。
【０１４１】
同様に、１番目の制約条件より下記数式２７となるので、下記数式２８が得られる。
【０１４２】
【数２７】

$$a+by\le\frac{c}{x}\le\frac{c}{d+b\zeta}$$

【０１４３】
【数２８】

$$y\le\frac{c}{b(d+b\zeta)}-\frac{a}{b}$$

また、３番目の制約条件より、下記数式２９を得る。
【０１４４】

【数２９】

$$y \geq \frac{\zeta}{x} \geq \frac{\zeta a}{c - b\zeta}$$

【０１４５】

以上の議論より、問題Ｐ'の実行可能集合は有界閉集合になる。連続関数は有界閉集合上で大域的最小値を持つため、問題Ｐ'には大域的最小解（ｘ＊,ｙ＊）が存在する。

【０１４６】

この（ｘ＊,ｙ＊）は緩和問題Ｐの大域的最小解でもある。実際、（ｘ＊,ｙ＊）が緩和問題Ｐの大域的最小解でなければ、下記数式３０となる緩和問題Ｐの実行可能解下記数式３１が存在することになるが、数式３１は問題Ｐ'の実行可能解でもあるので、（ｘ＊,ｙ＊）が問題Ｐ'の大域的最小解であることに矛盾する。

【０１４７】

【数３０】

$$-\tilde{x}\tilde{y} < -x^*y^*$$

【０１４８】

【数３１】

$$(\tilde{x},\tilde{y})$$

【０１４９】

以上の議論により、緩和問題Ｐの大域的最小解（ｘ＊,ｙ＊）は存在し、

$$-x^*y^* < 0$$

となる。すなわち、緩和問題Ｐの局所最小解（数式３２）として、下記数式３３を満たすものが存在する。

【０１５０】

【数３２】

$$(\bar{x},\bar{y})$$

【０１５１】

【数３３】

$$\bar{x}\bar{y} > 0$$

ここでラグランジェ関数Ｌ（ｘ，ｙ，u_1，u_2）を定義する。

Ｌ（ｘ，ｙ，u_1，u_2）

$= -xy + u_1\{(a+by)x - c\} + u_2\{(by-1)x + d\}$

ただし、u_1，u_2はラグランジェ乗数である。数式３３なる局所最小解（数式３２）に対するＫＫＴ条件より下記数式３４を得る。

【０１５２】

【数３４】

$$\left.\frac{\partial L}{\partial x}\right|_{(x,y)=(\bar{x},\bar{y})} = (-1+bu_1+bu_2)\bar{y}+au_1-u_2=0$$

$$\left.\frac{\partial L}{\partial y}\right|_{(x,y)=(\bar{x},\bar{y})} = (-1+bu_1+bu_2)\bar{x}=0$$

【０１５３】

【数３５】

$$\bar{x}\neq 0$$

に注意して整理すると、

$au_1-u_2=0$，　$bu_1+bu_2=1$

が得られ、これよりラグランジェ乗数u_1，u_2は下記数式３６のように一意に定まる。

【０１５４】

【数３６】

$$u_1=\frac{1}{(a+1)b}>0, \qquad u_2=\frac{a}{(a+1)b}>0$$

よって緩和問題Ｐにおける２つの制約条件は共に有効制約となっている。この結果、局所最小解（数式３２）は、数式２５より下記数式３７で与えられ、一意に定まる。

【０１５５】

【数３７】

$$\bar{x}=\frac{c+d}{a+1}>0, \qquad \bar{y}=\frac{c-ad}{b(c+d)}>0$$

よって、数式３７で与えられる緩和問題Ｐの局所最小解（数式３２）は大域的最小解（x^*, y^*）である。さらに$x^*>0$，$y^*>0$より、元の問題の大域的最小解となっていることが分かる。数式２３は数式３７より直ちに得られる。

【０１５６】

（Ｂ４）＜制御パラメタの決定法＞

前節では、fT_{SW}を最大にするf^*と下記数式３８の値が、ベーシックウインドウ数Ｋの関数として数式２３で与えられることを示した。その結果、積fT_{SW}は下記数式３９で与えられる。

【０１５７】

【数３８】

$$T^*_{SW}$$

【０１５８】

【数３９】

$$fT_{SW}(K)=\frac{1}{C_{max}\Delta_1}\left(\frac{K}{K+2}T_{D\_max}-K\Delta_2\right)$$

【０１５９】
ここで自然数Ｋを実数ｚに置き換え、数式３９をｚについて１回微分すると、下記数式４０となる。
【０１６０】
【数４０】

$$\frac{d}{dz}fT_{\mathrm{SW}}(z)=\frac{2T_{\mathrm{D\_max}}}{(z+2)^2}-\Delta_2$$

さらにｚで微分すると、下記数式４１となる。
【０１６１】
【数４１】

$$\frac{d^2}{dz^2}fT_{\mathrm{SW}}(z)=\frac{-4T_{\mathrm{D\_max}}}{(z+2)^3}$$

数式４１は正数ｚの値によらず常に負であるので、数式３９はＫの凹関数であることがわかる。さらに数式４０の右辺が０になるときのｚ＝ｚ＊を求めると、下記数式４２となる。
【０１６２】
【数４２】

$$z^*=\sqrt{\frac{2T_{\mathrm{D\_max}}}{\Delta_2}}-2$$

これから数式３９が最大値をとる実数ｚ＊が求まる。
【０１６３】
Ｋは自然数であるので上記数式４２で与えられた実数ｚ＊から下記数式４３で表わされる二つの値を求める。
【０１６４】
【数４３】

$$K^-=\lfloor z^*\rfloor,\qquad K^+=\lceil z^*\rceil$$

このＫ－とＫ＋のうち、ｆＴＳＷの値が大きくなる方をＫ＊として採用する。すなわち、数式４４としたとき、ベーシックウインドウ数をK=K＊とする。
【０１６５】
【数４４】

$$K^*=\arg\max_{K}\{fT_{\mathrm{SW}}(K)\mid K=K^-,K^+\}$$

なお、数式４２は数式２０の仮定の下で導かれたため、Ｋ＊は数式２０を満たす必要がある。数式４３で表されるＫ＋は関数の性質上、次の条件を満たす。
Ｋ＋＜ｚ＊＋１
下記数式４５と数式２０より、下記数式４６が成立することが、数式２０を満たすための十分条件となる。
【０１６６】

【数４５】

$$K^* < z^* + 1 = \sqrt{\frac{2T_{\mathrm{D\_max}}}{\Delta_2}} - 1$$

【０１６７】
【数４６】

$$T_{\mathrm{D\_max}} - \left(\sqrt{\frac{2T_{\mathrm{D\_max}}}{\Delta_2}} + 1 \right) \Delta_2 \geq 0$$

数式４６を変形すると下記数式４７に示す条件が導かれる。
【０１６８】
【数４７】

$$T_{\mathrm{D\_max}} \geq \left(2 + \sqrt{3} \right) \Delta_2$$

すなわち、数式４７が満たされていれば、本手法によるＫ＊の導出は有効である。
【０１６９】
さらに、得られたＫ＊を数式２３に代入すると、下記数式４８として、ｆとＴ$_{\mathrm{SW}}$も一意に定めることができる。
【０１７０】
【数４８】

$$T^*_{\mathrm{SW}} = \frac{K^* T_{\mathrm{D\_max}}}{K^* + 2}, \qquad f^* = \frac{T_{\mathrm{D\_max}} - (K^* + 2)\Delta_2}{C_{\max} \Delta_1 T_{\mathrm{D\_max}}}$$

【０１７１】
しかし、パケット数閾値ｙ＊が離散値を取るため、パケット数閾値ｙ＊とスライディングウインドウサイズＴ$_{\mathrm{SW}}$が同じであるときは、サンプリングレートｆが小さいほど誤検出確率は低くなる。そこで数式４８により得られるｆ＊と数式３８を用いてｙ＊を定め、その上で下記数式４９を固定したままｙ＊が変化しない最小の値下記数式５０を求め、これを実際に用いるサンプリングレートの最終的な値とする。
【０１７２】
【数４９】

$$T_{\mathrm{SW}} = T^*_{\mathrm{SW}}$$

【０１７３】
【数５０】

$$f = \hat{f}^*$$

【０１７４】
以上の手続きをまとめると以下のようになる。図２－２は、以上の制御パラメタの決定手順を説明するためのフローチャートである。
・Step１：予め与えられた規定パラメタから、数式４２と数式４４を用いてスライディングウインドウに含まれるベーシックウインドウ数Ｋ＊を定める（ステップＳ２１）。

【０１７５】
・Step２：Step１で定めたＫ＊を数式４８へ代入し、サンプリングレートｆの暫定最適解ｆ＊とスライディングウインドウサイズＴ$_{SW}$の最適値（数式３８，数式２３参照）を求める（ステップＳ２２）。
・Step３：ＲとStep２で求めた数式３８で表わされる最適値を用いてｘ＊を下記数式５１より求める（ステップＳ２３）。
【０１７６】
・Step４：Step２で求めたｆ＊とStep３で求めたｘ＊ならびに検出見逃し許容確率εを用いて数式６よりｙ＊を求める（ステップＳ２４）。
・Step５：予め与えられている検出見逃し許容確率ε，Step２で求めた数式３８の値，Step３で求めたｘ＊ならびにStep４で求めたｙ＊に対して、下記数式５２で表わされる最小のサンプリングレートを求め、これをサンプリングレートとする（ステップＳ２５）。
【０１７７】
【数５１】

$$x^* = \lfloor RT_{\mathrm{SW}}^* \rfloor$$

【０１７８】
【数５２】

$$\hat{f}^*$$

【０１７９】
（Ｃ）＜性能評価実験＞
本節では、インターネットを通じて公開されているトレースデータを元に擬似的な高速回線のトラヒックデータを作成し、作成したデータに対して適当な規定パラメタの下で解析実験を行い、本発明の性能評価を行う。
【０１８０】
まず、実験に用いるトラヒックデータについて言及する。続いて性能を評価するための指標について述べ、最後に実験の結果を示し、その考察を行う。
【０１８１】
（Ｃ１）＜トレースデータの概要＞
性能評価を行うトレースデータとして、WIDE ProjectのMAWI Working Groupによって２００６年３月３日の２０:００から２２:３０の間に、１００Mbpsのバックボーンリンクで測定されたトレースデータを用いる。このトレースデータの中には、宛先ＩＰアドレスを持たない、あるいは識別できないパケットデータが含まれているが、これらのパケットは予めサンプリング実験の対象外とした。
【０１８２】
また、バックボーンのような高速な回線を想定し、なおかつ検出対象となる高パケットレートを複数生成するために、１５０分間のトレースデータを時間スケールで１/１０に圧縮し、１５分間のトラヒックデータとして扱うことにする。これによって各フローのパケットレートを実際のトレースデータの１０倍と見なすことができる。
【０１８３】
図４は、本実験に用いたトラヒックデータの概要を示す図である。
同図に示すように、本実験では、測定時間９００秒、パケット数９０８９７９０５個、フロー数１３１３２０４個のトラヒックデータを用いて実験が行われた。
【０１８４】
（Ｃ２）＜性能評価指標＞
本発明の性能を評価するための比較対象として、スライディングウインドウサイズＴ$_{SW}$およびベーシックウインドウ数Ｋは本発明と同じ値であるが、下記数式５３とした場合、すなわち、本発明と同じ条件下で全てのパケットを抽出する場合を考える。

【０１８５】
【数５３】

$$f = 1,\ y^* = x^* = \lfloor RT_{\mathrm{SW}} \rfloor$$

【０１８６】
この比較対象ならびに本発明を用いた際の指標として以下のｔ$_1$，ｔ$_2$を導入する。
【０１８７】
・全てのパケットを抽出する場合の検出対象フローの検出時刻ｔ$_1$
・本発明を用いた場合の検出対象フローの検出時刻ｔ$_2$
なおｔ$_1$，ｔ$_2$には、閾値に達するパケットを抽出したベーシックウインドウの取得終了時間を用いた。
【０１８８】
検出対象フローに関しては検出結果を以下の４通りの場合に分類する。
(i)　ｔ$_1$＞ｔ$_2$，
(ii)　ｔ$_1$＝ｔ$_2$，
(iii)　ｔ$_1$＜ｔ$_2$＜∞，
(iv)　ｔ$_2$＝∞
【０１８９】
場合(i)について補足する。比較対象ではｆ=１かつ下記数式５４であるため、検出対象フローが初めてスライディングウインドウ内でｘ＊個以上のパケットが通過した時のベーシックウインドウの取得終了時間がｔ$_1$となる。
【０１９０】
【数５４】

$$y^* = x^* = \lfloor RT_{\mathrm{SW}} \rfloor$$

【０１９１】
一方、本発明は誤検出確率が０でないため、検出対象フローが初めてスライディングウインドウ内でｘ＊個以上のパケットが通過する以前に（誤って）検出してしまう可能性がある。そのため全パケットを抽出する場合よりも早く検出される、すなわち、ｔ$_1$＞ｔ$_2$となる可能性がある。
【０１９２】
しかし、検出対象のフローに関しては、結果としてｔ$_1$≧ｔ$_2$であれば、本発明がフロー発生時から検出許容時間Ｔ$_{D_max}$以内に検出したことになる。よって、場合（ｉ），（ii）は共に目標時間内に検出されたことを示し、場合（iii）は目標時間は過ぎてしまったが検出はされたことを示す。また場合（iv）は検出対象フローを検出できずに見逃してしまったことを示す。一方、検出対象外のフローを誤検出してしまう割合を評価するため、以下の数式５５で表される誤検出率を用いる。
【０１９３】
【数５５】

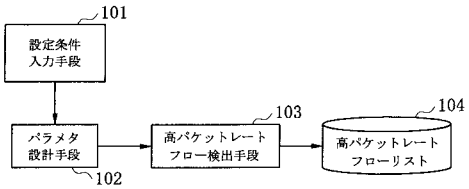
$$\text{誤検出率} = \frac{\text{誤検出された検出対象外フロー数}}{\text{母集団における検出対象外フロー総数}}$$

【０１９４】
（Ｃ３）＜実験結果＞
予め与えられるパラメタである、Ｔ$_{D_max}$[秒]，Δ$_1$[秒]，Δ$_2$[秒]，Ｃ$_{max}$[packet/秒]はそれぞれ図５のように与えた。すなわち、Ｔ$_{D_max}$[秒]＝１０秒，Δ$_1$[秒]＝１０$^{-3}$秒，Δ$_2$[秒]＝１０$^{-2}$秒，Ｃ$_{max}$[packet/秒]＝１０^{6}packet/秒とした

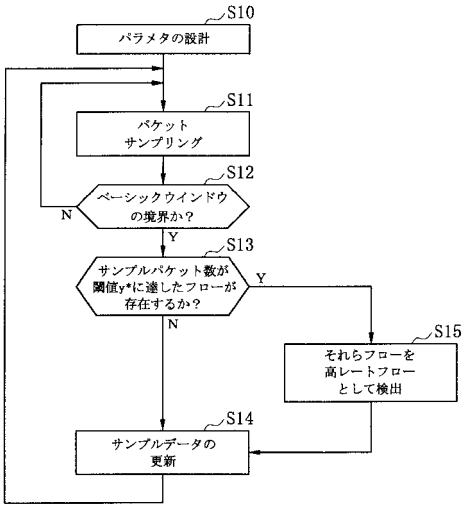
。
【０１９５】
また、検出見逃し許容確率εは０.０５，０.０１の二通りを考え、パケットレート閾値Ｒ[packet/秒]は１０００，２５００，５０００の３通りを考えた。さらに、図５のパラメタおよび３種類のパケットレート閾値Ｒの値を元に、本発明の制御パラメタ設定法およびパケット数閾値ｙ＊の導出法を用いた結果、図６のような制御パラメタ値とパケット数閾値ｙ＊を得た。
【０１９６】
なお、図５のパラメタ値（T_{D_max}[秒]，Δ_1[秒]，Δ_2[秒]，C_{max}[packet/秒]）が与えられると、ベーシックウインドウ数ＫならびにスライディングウインドウサイズT_{SW}は検出見逃し許容確率εならびにパケットレート閾値Ｒの値とは無関係に定まることに注意する。
【０１９７】
これらのパラメタ値を用いて、上記（Ｃ１の項）で示したトラヒックデータに対してサンプリング実験を１００回行った。検出見逃し許容確率εが０.０５ならびに０.０１の場合における検出対象フロー数に関する結果を、図７と図８に示す。
【０１９８】
なお、これらの図７，８に記載されている数値は全て１００回の試行の平均値と９５％信頼区間である。図７および図８より、検出見逃し許容確率εの値に関わらず、検出対象フローの大半において全パケットが抽出される場合（ｆ=１）よりも早く検出されている。
【０１９９】
一般に検出対象となる高パケットレートフローは、フロー発生後、時間と共にパケットレートを増加させていき、いずれ検出目標値を上回るパケットレートに達すると考えられる。
【０２００】
この結果は、検出対象フローが目標値へ向けてパケットレートを増加させる過程において検出されていることを示している。一方、検出許容時間後に検出されるフロー数ならびに検出を見逃したフロー数は一般に非常に少ない。特に大きなレートをもつフローを対象とする場合は、検出を完全に見逃してしまうフローは皆無となっている。また、検出見逃し許容確率を０.０１とすると、検出許容時間後に検出されるフロー数ならびに検出を見逃したフロー数はほとんどなくなっている。
【０２０１】
図９ならびに図１０は、それぞれ、図７ならびに図８の結果を検出対象フロー数で正規化し、結果を割合で示したものである。どのようなパケットレート閾値Ｒについても検出対象フローを１-ε以上の割合で目標時間内に検出できているが、検出に成功する割合は目標値を大きく上回る傾向がある。これはパケットレートが目標値に達するまでの増加過程において（誤って）検出されることに起因しているためである。
【０２０２】
最後に、検出見逃し許容確率εが０.０５ならびに０.０１における誤検出率に関する結果を、図１１ならびに図１２に示す。ただし、誤検出フロー数ならびに誤検出率は１００回の試行の平均値と９５％信頼区間である。これらの結果から、一般に、誤検出されるフロー数は検出対象となるフロー数以上となる場合が多く、目標パケットレートＲが小さくなるとその傾向が顕著になる事が分かる。
【０２０３】
さらに、目標パケットレートが小さい場合、検出見逃し許容確率εを０.０５とすることで、０.０１の場合と比較して誤検出されるフロー数を大幅に減少させることができている。なお、検出対象外のフロー数は非常に多く、検出対象外の低パケットレートフローを誤って検出してしまう割合に関しては検出見逃し許容確率の値に関わらず十分小さく押えられていることがわかる。

【０２０４】
以上より、本発明は設計した通り十分機能していることが確認された。
ａ）目標パケットレートが大きく、検出対象となるフロー数が少ない場合、検出見逃し許容確率εが０.０１であっても０.０５であっても大差はない。
【０２０５】
ｂ）一方、目標パケットレートが比較的小さく、検出対象となるフロー数が増加すると、検出見逃し許容確率によって結果が異なってくるが、検出に成功するフロー数は設定値を大きく上回る傾向があるため、検出見逃し許容確率を多少大きめに設定することによって誤検出フロー数を抑制することが望ましいように思われる。
【０２０６】
なお、本発明の上記各手段によって行われる処理（例えば、図２－１、図２－２参照）を実現するためのプログラムは、ＣＤ－ＲＯＭ、ＤＶＤ、ＦＤやインターネットを介して広く市場に流通させることができる。
【図面の簡単な説明】
【０２０７】
【図１】本発明の実施の形態の一例を示すシステム構成図である。
【図２－１】図１に示したシステムを用いて実行される高パケットレートフローのオンライン検出方法の処理の流れを示すフローチャートである。
【図２－２】本発明における制御パラメタの決定手順を説明するためのフローチャートである。
【図３】パラメタ値の変化による利点を説明するための図である。
【図４】実験に用いたトラヒックデータの概要を示す図である。
【図５】実験に用いた規定パラメタを示す図である。
【図６】図５のパラメタを元に発明の方法で得られた制御パラメタとパケット数閾値ｙ＊を示す図である。
【図７】検出見逃し許容確率ε=０.０５のときの対象フローの検出結果（フロー数）を示す図である。
【図８】検出見逃し許容確率ε=０.０１のときの対象フローの検出結果（フロー数）を示す図である。
【図９】検出見逃し許容確率ε=０.０５のときの対象フローの検出結果（割合）を示す図である。
【図１０】検出見逃し許容確率ε=０.０１のときの対象フローの検出結果（割合）を示す図である。
【図１１】検出見逃し許容確率ε=０.０５のときの対象外フローの誤検出結果を示す図である。
【図１２】検出見逃し許容確率ε=０.０１のときの対象外フローの誤検出結果を示す図である。
【符号の説明】
【０２０８】
１０１：設定条件入力装置
１０２：パラメタ設計装置
１０３：高パケットレートフロー検出装置
１０４：高パケットレートフローリスト

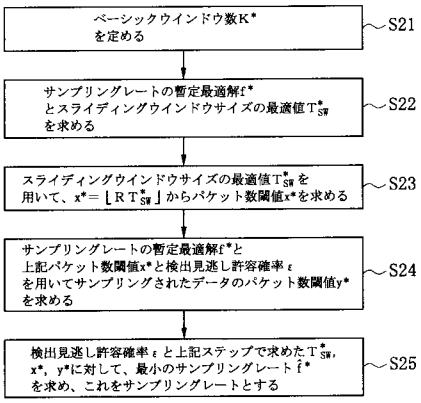
【図１】



【図２－１】



【図２－２】



【図３】

項目	パラメタ値 小	パラメタ値 大
T_{SW}	・使用メモリの軽減 ・検出までにかかる時間の短縮	・低レートフローの誤検出確率の低下
f	・ＣＰＵにかかる負荷の軽減	・低レートフローの誤検出確率の低下
K	・ＣＰＵにかかる負荷の軽減	・検出までにかかる時間の短縮

【図４】

測定時間［秒］	パケット数	フロー数
900	90897905	1313204

【図５】

T_{D_max}	Δ_1	Δ_2	C_{max}
10	10^{-3}	10^{-2}	10^{6}

【図６】

R	K	T_{SW}	$\varepsilon=0.05$		$\varepsilon=0.01$	
			f	y*	f	y*
1000	43	9.5556	8.113×10^{-4}	4	8.794×10^{-4}	3
2500			9.161×10^{-4}	15	8.994×10^{-4}	12
5000			9.473×10^{-4}	35	9.501×10^{-4}	31

【図７】

R	対象フローの総数	$t_1>t_2$	$t_1=t_2$	$t_1<t_2<\infty$	$t_2=\infty$
1000	34	31.06±0.206	2.26±0.095	0.65±0.170	0.03±0.034
2500	11	10.05±0.168	0.77±0.136	0.18±0.081	0
5000	4	1.94±0.047	1.94±0.067	0.12±0.064	0

【図８】

R	対象フローの総数	$t_1>t_2$	$t_1=t_2$	$t_1<t_2<\infty$	$t_2=\infty$
1000	34	31.82±0.085	2.03±0.034	0.14±0.074	0.01±0.020
2500	11	10.69±0.091	0.27±0.087	0.04±0.039	0
5000	4	2.03±0.052	1.96±0.055	0.01±0.020	0

【図９】

R	$t_1>t_2$ or $t_1=t_2$	$t_1<t_2<\infty$	$t_2=\infty$
1000	0.9800±0.0089	$1.912\times10^{-2}\pm5.009\times10^{-3}$	$8.824\times10^{-4}\pm9.883\times10^{-4}$
2500	0.9836±0.0276	$1.636\times10^{-2}\pm7.331\times10^{-3}$	0
5000	0.9701±0.0285	$3.000\times10^{-2}\pm1.600\times10^{-2}$	0

【図１０】

R	$t_1\geq t_2$	$t_1<t_2<\infty$	$t_2=\infty$
1000	0.9956±0.0035	$4.118\times10^{-3}\pm2.171\times10^{-3}$	$2.941\times10^{-4}\pm5.765\times10^{-4}$
2500	0.9963±0.0162	$3.636\times10^{-3}\pm3.509\times10^{-3}$	0
5000	0.9975±0.0267	$2.500\times10^{-3}\pm4.900\times10^{-3}$	0

【図１１】

R	対象外フローの総数	誤検出フロー数	誤検出率
1000	1313170	294.95±1.681	$2.246\times10^{-4}\pm1.279\times10^{-6}$
2500	1313193	19.81±0.345	$1.509\times10^{-5}\pm2.630\times10^{-7}$
5000	1313200	5.92±0.149	$4.508\times10^{-6}\pm1.136\times10^{-7}$

【図１２】

R	対象外フローの総数	誤検出フロー数	誤検出率
1000	1313170	599.74±2.441	$4.567\times10^{-4}\pm1.859\times10^{-6}$
2500	1313193	33.71±0.522	$2.567\times10^{-5}\pm3.976\times10^{-7}$
5000	1313200	6.95±0.151	$5.292\times10^{-6}\pm1.150\times10^{-7}$

フロントページの続き

(72)発明者　上山　憲昭
東京都千代田区大手町二丁目３番１号　日本電信電話株式会社内
(72)発明者　川原　亮一
東京都千代田区大手町二丁目３番１号　日本電信電話株式会社内
(72)発明者　原田　薫明
東京都千代田区大手町二丁目３番１号　日本電信電話株式会社内
(72)発明者　滝根　哲哉
大阪府吹田市山田丘１番１号　国立大学法人大阪大学内
(72)発明者　工藤　隆則
大阪府吹田市山田丘１番１号　国立大学法人大阪大学内

審査官　安藤　一道

(56)参考文献　特開２００６－１９６９８２（ＪＰ，Ａ）
特開２００７－１１６４０５（ＪＰ，Ａ）
特開２００６－１６４０３８（ＪＰ，Ａ）
特開２００７－５９９４（ＪＰ，Ａ）
特開２００７－２２１４１２（ＪＰ，Ａ）
上山憲昭、他２名，フローサンプリングを用いた大量フロー生成ホストの特定，電子情報通信学会技術研究報告，日本，社団法人電子情報通信学会，２００６年　2月２３日，情報ネットワーク 105(628)，pp.165-170
上山憲昭、他２名，大量フロー生成ホスト特定法の性能評価，電子情報通信学会技術研究報告，日本，社団法人電子情報通信学会，２００６年　9月　7日，情報ネットワーク106(237)，pp.97-102

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｈ０４Ｌ　１２／５６
Ｇ０６Ｆ　１３／００