ネットワークへの早期展開を可能とする軽量な異常トラヒック分析手法の確立に関する研究

2012 年 3 月

いそざき ひろおみ

磯﨑裕臣

概要

本論文では，サービス拒否 *(Denial of Service; DoS)* 攻撃やその分散型である分散型サービス拒否 *(Distributed Denial of Service; DDoS)* 攻撃による異常トラヒックの除去やネットワーク管理に必要となるトラヒックの分析技術について,（1）ネットワーク上のトラヒックの統計情報の集計,（2）トラヒックの到着パターンの分類と到着間隔のモデル化,（3）トラヒックの生成元の特定の 3 つの課題について検討し，ルータ同士の連携を必要とせず，自律的に動作し，ネットワークへの早期展開を可能とするシステムを実現する．

トラヒックの統計情報の集計は，パケットサンプリングを利用することで，情報の記録に必要となる資源量を削減可能である．しかし，パケットサンプリングによる情報欠落が生じることで，集計した統計情報の精度が劣化するという問題が発生する．本論文では，異なる間隔のパケットサンプリングで得られた統計情報の差分に着目する．得られた差分情報をフィードバックさせることで，元の統計情報をより正確に推定する手法を新たに提案する．さらに，インターネット上で公開されている実トレースデータに提案手法を適用させることで，計算量のオーダを増加させることなく，従来手法に比べ推定誤差を最大で 85% 程度削減可能であることを示す．

トラヒックの到着パターンの分類と到着間隔のモデル化では，クラウドサービスを利用する多くのアプリケーションにより，ユーザへの応答時間を短縮するために短期間に確立される複数のコネクションについて検討する．これらの短期間に多く生成されるフローは，バースト性を有する．このため，フローのモデル化について多くの研究がなされているが，バースト性を有するフローが考慮されていないため，フローの到着間隔のモデル化が難しい．本論文では，フローの到着間隔に着目し，フローの生成要因を分析することで，２種類のフローの生成要因が存在することを明らかにする．その上で，ユーザ操作に基づいたフローの到着パターンを分析し，ユーザ操作とフローの到着パターンに関係があることを示す．そして，フローの到着間隔分布のモデル化を行い，従来手法であるポアソン分布で近似できないことを示し，ワイブル分布で近似できることを示す．

トラヒックの生成元の特定には，被害ホストにおいてパケットの送信元を特定する IP トレースバックと呼ばれる技術が検討されており，その１つに確率的パケットマーキン

グ *(Probabilistic Packet Marking; PPM)* がある．PPM では，ルータが自律的にパケットにマーキングを行うため，マーキングの上書きが頻繁に発生し，異常トラヒックが通過した経路の再構築に必要となるパケットが増加するという問題が存在する．本論文では，PPM においてマーキングの重複を引き起こす要因を分析し，それらを軽減させることで，パラメータチューニングを必要とすることなくマーキングの重複を減少させる新しいマーキング手法を提案する．シミュレーションを用いて性能評価を行い，提案手法がパラメータチューニングを必要とすることなく，自律的に動作し，攻撃経路の再構築に必要なパケット数を従来手法に比べ最大で 85% 削減可能であることを示す．

目 次

図目次

表 目 次

第1章　まえがき

ネットワークの広帯域化に伴い，様々なアプリケーションやサービスがネットワーク上で提供され，多くのユーザが利用している．クラウドで提供されるアプリケーションは，ネットワークに接続しさえすれば，利用できるという高い利便性がある．しかしながら，ネットワークリソースには限界があるため，特定のユーザがネットワークを使用することで，ネットワークリソースが浪費され，ユーザに安定したサービスが提供できなくなることがある．その代表的なものとして被害ホスト宛に多量の接続要求や無意味なパケットを送信することでネットワークリソースを浪費するとともに，被害ホストがサービスを提供できなくしてしまうサービス拒否 *(Denial of Service; DoS)* 攻撃やその分散型である分散型サービス拒否 *(Distributed Denial of Service; DDoS)* 攻撃がある [1]．これらの攻撃には，

- SYN Flood

  被害ホストに多量の SYN パケットのみを送信し，サーバに接続要求の処理中であるセッションを多量に生成することで，サーバのリソースを浪費する．

- ICMP Flood

  送信元 IP アドレスを被害ホストの IP アドレスに偽装した *ICMP (Internet Control Message Protocol)* echo リクエストパケットをブロードキャストし，その応答パケッ

トを被害ホストが受信するようにすることで，ネットワークの帯域を浪費する．

- Application-Based Bandwidth Attacks

  Web サーバで公開されている検索サービスに対して，多量の検索リクエストを送信することで，サーバのリソースを浪費する．

- HTTP Flood

  WEB サーバにボットネットによって多量の HTTP パケットを送信することで，多量の HTTP セッションを確立させ，サーバのリソースを浪費する．

- SIP Flood

  多量の SIP Invite パケットを送信することで，SIP プロキシサーバのリソースを浪費する．

- Distributed Reflector Attacks

  送信元 IP アドレスを被害ホストの IP アドレスに偽装したパケットを多量のサーバ宛に送信し，サーバからの応答パケットを被害ホストが受信するようにすることで，サーバのリソースを浪費する．

- DNS Amplification Attacks

  送信元 IP アドレスを被害ホストの IP アドレスに偽装した多量の DNS 名前解決リクエストパケットを DNS サーバに送信し，その応答パケットを被害ホストが受信するようにすることで，サーバのリソースを浪費する．

といったものがある．図 1.1 に SYN Flood, ICMP Flood, Application-Based Bandwidth Attacks の例を示す．他の攻撃である HTTP Flood と SIP Flood は，Application-Based

Bandwidth Attacks と攻撃原理は同じであるが，攻撃に使用されるパケットが異なる．同様に Distributed Reflector Attacks と DNS Amplification Attacks は，ICMP Flood と攻撃原理は同じであるが，攻撃に使用されるパケットが異なる．

これらの攻撃は，攻撃用ツールをインターネットから入手することで，誰でも簡単な操作で攻撃を仕掛けることができる．攻撃を受けたホストは，数分から数時間，場合によっては数日間にもわたりサービスを提供できない事態におちいる．このため，緊急に解決すべき課題として社会的に大きな問題となっている．過去に Yahoo や Amazon などに大規模な DoS 攻撃が行われた事例もあり [2–4]，攻撃パケットの送信レートが 10 Gbps におよぶものもある．最近では WikiLeaks やアメリカ政府への攻撃も報告されており [5,6]，攻撃パケットの送信レートは，100 Gbps を超えるものもある．

ユーザに安定したサービスを提供するには，ネットワークリソースを浪費しているトラヒックをネットワーク上から除去する必要がある．これには，トラヒックを分析することで，異常トラヒックを検出し，異常トラヒックをフィルタリングすることが考えられる．ここで，ネットワークリソースを浪費しているトラヒックのうち DoS 攻撃，DDoS 攻撃，ワームの拡散，ポートスキャンによるトラヒックを「異常トラヒック」とする．しかしながら，攻撃に使用されるパケットは，送信元 IP アドレスが偽装されているため，パケットの送信元 IP アドレスに基づいたパケットフィルタリングは，有効に働かない．また，パケットの送信元 IP アドレスから送信元を特定することもできない．このため，パケットを受信したホストにおいてパケットの送信元を特定する IP トレースバック [7] という技術が検討されている．IP トレースバックでは，被害ホストが，異常トラヒックを受信した際にルータから受信したパケットの通過情報をもとにパケットが通過した経路の追跡を行う．これには，ルータ同士の連携が必要となるため，異常トラヒックを完全

に除去するには，ネットワーク上のすべてのルータに IP トレースバックを導入し，異常トラヒックを送出しているネットワーク上のホストをすべて特定する必要がある．しかし，IP トレースバックに対応したルータへの交換が必要となり，一斉にネットワーク上のすべてのルータに導入するのは，現実的ではない．

本論文では，異常トラヒックの除去をはじめ，ネットワーク管理に必要となるトラヒックの分析技術について，

- トラヒックの統計情報の集計

  ネットワーク上のトラヒックの統計情報を集計する．

- トラヒックの到着パターンの分類と到着間隔のモデル化

  ユーザ操作で生成されるフローの到着パターンの分類や到着間隔のモデル化を行う．

- トラヒックの生成元の特定

  トラヒックが通過した経路を追跡することで，トラヒックを送出しているホストを特定する．

の 3 つの課題を検討し，ルータ同士の連携を必要とせずルータが自立的に動作し，ネットワークへの早期展開を可能とするシステムを実現する．

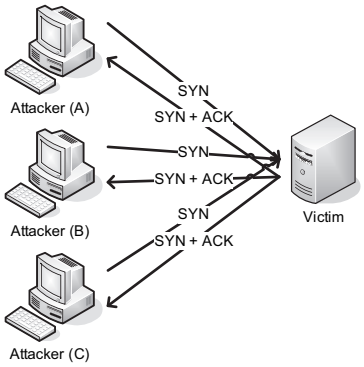
これらの課題についてルータへの負荷軽減には，ネットワークから一部のパケットのみを選択するパケットサンプリング [8,9] と呼ばれる手法が有効であり，数多くの検討がなされている [10–18].

パケットサンプリングを行うことで，パケット情報の記録に必要となる資源量やパケット分析のためのオーバーヘッドを削減することができ，ルータへの負荷軽減が可能となる．

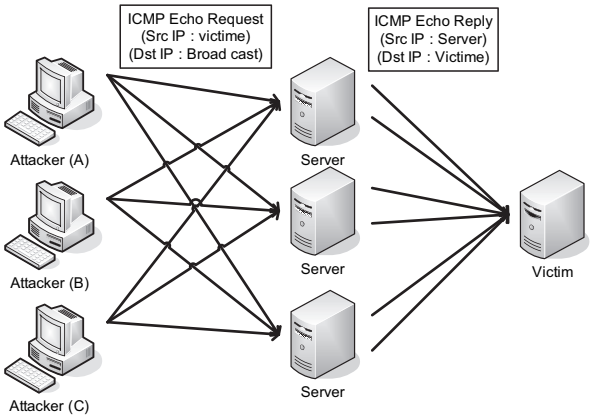
ここで，一般的にトラヒック全体のうち $1/n$ の割合のトラヒックをサンプリングする代表的な方法として，

- ランダム確率により $1/n$ の割合でパケットを選択
- n パケットごとに 1 パケットを選択

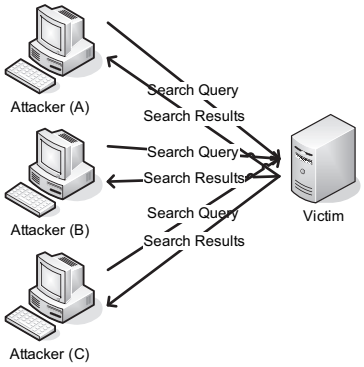
などが考えられる．本論文では，$1/n$ の割合のトラヒックを選択するパケットサンプリングを「$1/n$ パケットサンプリング」と呼ぶ．しかしながら，$1/n$ パケットサンプリングは平均すると n パケットのうち 1 パケットのみの情報しか参照しないため，残りの $n-1$ パケットの情報は用いられない．このため，トラヒック全体を計測する場合と比較して，サンプリングされなかったパケットの情報消失が発生する分だけ誤差が生じることになる [19,20]．一般的には，誤差は n が大きくなるにつれて増大することになる．



(a) SYN Flood



(b) ICMP Flood



(c) Application-Based Bandwidth Attacks

図 1.1: DoS と DDoS 攻撃の例

1.1 トラヒックの統計情報の集計

すでに述べたようにネットワーク上から異常トラヒックを除去するには，まずネットワークからトラヒックを計測し，検出に必要な統計情報を集計する必要がある．

ネットワーク上のトラヒックの統計情報の集計には，ネットワーク上のトラヒックを計測し，フローごとの統計量を計測するのが有効である．しかしながら，コアネットワーク上のルータでパケットをモニタリングし，さらにフローごとの統計量を計測することは，パケット情報や統計情報を記録するための資源量の増加やパケット処理のオーバーヘッドが生じる点で問題がある．ここでフローとは，IP ヘッダのうち送信元（宛先）IP アドレスと送信元（宛先）ポート番号とプロトコル番号が等しいパケットの集合のことである．

このため，トラヒックの計測にパケットサンプリングを利用することが検討されている．一般的にパケットサンプリングによって選択されないトラヒックは，以下のパケットが混合されたものとなる．

- パケット数が多いフローの一部のパケット
- パケット数がとても少ないフローの全パケット

このように，パケットサンプリングでは，2 種類のパケットが選択されないため，サンプリングによってフローのパケット数やフロー総数が減少する．このため，パケットサンプリングで得られたトラヒックから集計した統計情報に誤差が生じることとなる．

そのため，サンプリングによって消失した情報を補間することで元の統計情報を推定する手法が提案されている．これまでに提案されている手法は，推定処理が複雑であり計算量が増加し，推定に長時間を要する点で課題が残る．また元の統計情報は，異常ト

ラヒックの検出に用いることができるが，異常トラヒックが送出されている時間内に統計情報の集計が完了しなければ，その情報は有益でなくなる．このため，統計情報はできるだけ短時間で集計できることが望ましい．

本論文では，異なるサンプリング間隔のフローの統計情報を比較し，統計情報間の差分に着目する．差分情報をフィードバックさせることで，計算量のオーダを増加させることなく，元の統計情報を精度よく推定する *EBM (Equation Based Method)* と呼ぶ新たな手法を提案する．さらに，実際のトレースデータに提案手法を適用した結果，*EM (Expectation Maximization)* アルゴリズム [21] や *MLE (Maximum Likelihood Estimation)* を用いた従来手法に比べ，計算量のオーダを増加させることなく，従来手法に比べ推定誤差を最大で 85% 程度削減可能であることを示す．

1.2 トラヒックの到着パターンの分類と到着間隔のモデル化

すでに述べたように異常トラヒックの除去には，集計した統計情報から異常トラヒックを検出する必要がある．異常トラヒックの検出は，トラヒックのエントロピーを用いた手法 [22–24] をはじめ多く提案されており，異常トラヒックの統計情報が持つ特徴を用いて異常トラヒックを検出するのが一般的である．このため，本論文では，トラヒックの到着パターンの分類と到着間隔のモデル化を検討する．

一般的にフローの到着パターンは，ポアソン分布に従うと考えられている．その一方，近年のクラウドサービスの利用拡大に伴い，ネットワーク上で様々なサービスが提供されている．クラウドサービスを利用する多くのアプリケーションが，通信時間を短縮し，ユーザレスポンスを向上させるために，一度に複数のコネクションを確立する．この結

果，一度に複数のフローが生成されることとなり，フローがバースト性を有する．このため，フローの到着パターンは，ポアソン分布に従わなくなりつつある．

これまでも，一度に複数のパケットが生成される，パケットのバースト性については，多くの検討がなされている．これらの手法では，パケットのバースト性は考慮されているが，フローのバースト性が考慮されておらず，フローの到着プロセスをモデル化するのが難しい [25].

本論文では，フローの到着間隔に着目し，フローの生成要因を分析し，生成要因に基づいた 2 種類のフローが存在することを示す．そして，ユーザ操作に基づいたフローの到着パターンを分析し，フローの到着パターンとユーザ操作の間に関係が存在することを示す．そして，フローの到着間隔分布を関数近似することでモデル化し，ワイブル分布を用いることで近似できることを示す.

1.3 トラヒックの生成元の特定

異常トラヒックの除去には，すでに述べたように IP トレースバックを用いて異常トラヒックを送出しているホストを特定するのが有効である．IP トレースバックにパケットサンプリングを応用した手法として確率的パケットマーキング *(Probabilistic Packet Marking; PPM)* が提案されている．PPM では，ルータが確率にしたがい自律的にマーキングを行うため，マーキング情報の上書きが頻繁に発生する．本論文では，あるルータが複数のパケットに同じ情報をマーキングすることを「マーキング重複」と呼ぶ．また被害ホストにおいて，受信したパケットのうち同じ情報がマーキングされているパケットの数を「マーキング重複数」と呼ぶ．PPM ではマーキング重複が頻繁に発生することで，異常

トラヒックが通過した経路を再構築するのに多くのパケットを必要とし，攻撃経路を短時間で再構築できないといった問題が存在する [26–28].

本論文では，異常トラヒックが通過した経路を「攻撃経路」と呼ぶ．この問題に対して多くの手法が提案されているが，その多くは処理が複雑でありルータ同士の連携が必要である．インターネットが自律的に動作していることを考慮すると，トレースバックシステムもルータ同士の連携を必要とせず，自律的に動作するのが望ましい．

このため本論文では，PPM においてマーキングの重複を発生させる要因を分析し，その要因としてホップ数に起因する要因とトポロジに起因する要因があることを示す．これらの要因を軽減させる 2 つの方式を導出することにより，マーキング重複を削減し，攻撃経路の再構築に必要なパケット数を減少させる *PAPM (Parameter Auto-adjustable Packet Marking)* と *HCPPM (History Cashing based Probabilistic Packet Marking)* の 2 つの手法を提案する．そしてシミュレーションにより，提案手法がパラメータチューニングすることなく自律的に動作し，攻撃経路の再構築に必要なパケット数を従来手法に比べ最大で 85% 削減可能であることを示す．

以下，2 章でトラヒックの統計情報の推定，フローの到着間隔分布のモデル化と IP トレースバックの関連研究を述べる．3 章で，ネットワーク上のトラヒックの統計情報を正確に推定する新たな手法を提案する． 4 章で，ユーザ操作に基づいたフローの到着パータンの分類と到着間隔分布のモデル化について述べる． 5 章でトラヒックの生成元の特定についてパケットマーキングを用いて攻撃経路を効率よく特定する新たな手法を提案する．最後に 6 章でまとめと今後の課題について述べる．

第2章　トラヒック分析手法の関連研究

本章では，トラヒックの統計情報の推定, トラヒックの到着パターンの分類と到着間隔のモデル化，IP トレースバックに関する関連研究について述べる．

2.1　トラヒックの統計情報の推定の関連研究

前章で述べたようにパケットサンプリングを利用することで，情報記録に必要な資源量やパケット処理のオーバヘッドを軽減することができるが，サンプリングされないトラヒックが発生することにより，情報が欠落した統計情報しか得ることができない．

以上の問題を解決するため，トラヒックに含まれるパケットの総数，総バイト数やフロー総数や特定のフローの統計量を推定する手法がいくつか提案されている [29–31]．[29] では，サンプリングにより得られたパケットの総数や総バイト数にサンプリング間隔を乗じることで，統計情報のうちパケットの総数や総バイト数を推定する手法が提案されている．[30] では，パケット数が多いフローに注目し，パケット数が多いフローを正確に抽出することで，ホスト間の総パケット数が多いフローの統計量を推定する手法が提案されている．[31] では，SYN パケットの数にサンプリング間隔を乗じることでフローの総数を推定する手法が提案されている．しかしながら，これらの手法では，トラヒックに含まれるパケットの総数や総バイト数しか取得できないことや特定のフローの統計量

しか推定できない．

このため，サンプリングにより消失した情報を最尤法を用いて補間することで元の統計情報を推定する手法 [32–34] が提案されている．[32] では，*MLE (Maximum Likelihood Estimation)* を用いて元のフロー分布を推定している．本論文では，この手法を MLE と呼ぶ．MLE は，元のフロー分布を分布全体からパラメータを決定した 1 種類のパレート分布を用いて近似している．このため，元のフロー分布のおおまかな形状しか推定できない．さらに，尤度を最大にするパレート関数のパラメータが見つかるまで繰り返し計算するため，推定に長時間を要する．また [33] は，期待値を計算する *E (Expectation)* ステップとパラメータを更新する *M (Maximization)* ステップからなる EM アルゴリズムの各ステップを交互に繰り返すことで元のフロー分布の推定を行う．本論文では，この手法を EM と呼ぶ．EM では，E ステップでの期待値の計算と M ステップでのパラメータの更新において総和を繰り返し計算するため，パケット数が多いフローを推定する場合，計算量が大幅に増加する．さらに，MLE と同様に尤度を最大にする統計情報が見つかるまで 2 つのステップを繰り返すため，推定に長時間を要する．

また，すでに述べた手法とは別のものとして [35–38] が提案されている．[35] では，ハッシュ関数を使用してフローを識別することで，ホスト間の総トラヒック量を推定する手法が提案されている．また，フローのパケット数に基づいてサンプリング確率を変化させることで，オリジナルのフローの特徴を維持したままサンプリングを行うものが [36,37] で提案されている．これらの手法では，フローのパケット数が少なければ，サンプリング確率として高い値が設定され，パケット数が多ければ，低い確率が設定される．[38] では，サンプル・アンド・ホールド技術を用いて元のフロー分布を推定する手法が提案されている．これらの手法では，情報記録に必要な資源量やパケット処理のオー

バヘッドの面で課題が残る．

これまでに提案されている手法は，処理が複雑であり計算量が増加し，推定に長時間を要するといった点で，課題が残る．

本論文では，異なるサンプリング間隔のフローの統計情報を比較し，統計情報間の差分に着目する．差分情報をフィードバックさせることで，計算量のオーダを増加させることなく，元の統計情報を精度よく推定する新たな手法を提案する．

2.2 フローの到着間隔分布のモデル化の関連研究

フローの到着間隔のモデル化について，これまでも，一度に複数のパケットが生成される，パケットのバースト性については，多くの検討がなされている [39–53]

[39] では，Wavelet 解析を用いてパケットの到着プロセスと TCP フローの到着プロセスの間に長期間にわたる依存関係があることを示している．短期間の観測では，一見ばらばらに見えるパケットの到着プロセスと TCP フローの到着プロセスには，長期間の観測では，規則的なパターンが見られることが示されているので，長期間の観測により，パケットや TCP フローの到着プロセスがモデル化可能である．[40] では，インターネットのトラヒックを分析し，パケットの到着間隔分布を秒単位で求めた場合，非定常のスパイクが観測されるが，秒未満で分布で求めた場合，ポアソン過程に従うことを示している．また，時間間隔を長くした場合に，長期間にわたる依存関係があることを示している．[41] では，ポアソンショットノイズ過程を用いてフローベースのバックボーンリンクのモデル化を行っている．このモデルでは，パケットの到着レート，フローの平均サイズ，フローサイズの二乗と継続期間の比の三つのパラメータを用いて，バックボーン

トラヒックのスループットを近似している．[42] では，TELNET と FTP のトラヒックを分析した結果，パケットのバースト性により，ポアソン過程によるモデル化が難しいことが示されている．また，バースト性には，長い期間で見た場合に自己相関があることを示している．[43] では，フローの到着間隔分布やフローのサイズ分布について，ガンマ分布やワイブル分布で近似することで，ポアソン分布で近似する場合に比べ，精度よく近似できることを示している．[44] では，ADSL のトラヒックを分析することで，マイスフローやエレファントフローが白色雑音で摂動されるガウス過程で示せることが述べられている．また，フローの継続時間は，ワイブル分布で示せることが述べられている．[45] では，隠れマルコフ過程を用いて，トラヒックのモデル化を行い，パケットの到着間隔とパケットサイズを近似している．[46] では，フローの平均レート，フローの平均サイズ，フローサイズの 2 乗とフローの継続期間の比の 3 つのパラメータを用いて，フローの総量を推定する手法を提案している．[47] では，TCP フローのパケットの到着プロセスと IP パケットの到着プロセスの間に長期間にわたる依存関係があることが示されている．[48] では，プロキシサーバのログを分析し，ウェブページ閲覧時のトラヒックのモデル化を行っている．[49] では，VoIP トラヒックを分析し，経験則に基づいた VoIP トラヒックのモデル化と理論に基づいた VoIP トラヒックのモデル化を行っている．[50] では，単一のホストからのトラヒックに着目し，トラヒックを分析することで，パケットの到着間隔がワイブル分布で精度よく近似できることを示している．

また，[51] では，セッションレベルでのウェブサーバのトラヒックのモデル化を行い，ウェブサーバのベンチマークを実装している．

さらに，[52,53] では，実トラヒックデータを基にパケットのバースト性を考慮してトラヒックを生成する手法を提案している．

これらの手法では，パケットのバースト性は考慮されているが，フローのバースト性が考慮されておらず，フローの到着プロセスをモデル化するのが難しい．

本論文では，フローの到着間隔に着目し，フローの生成要因を分析することで，２種類のフローの生成要因が存在することを明らかにする．その上で，ユーザ操作に基づいたフローの到着パターンを分析し，ユーザ操作とフローの到着パターンに関係があることを示す．そして，フローの到着間隔分布を関数で近似することでデル化を行う．

2.3 IP トレースバックの関連研究

IP トレースバックは [7] によって初めて提案され，以降さまざまな研究がなされている．既存の IP トレースバックは大きく二つに分類することができる．

一つは，ルータ上でパケットの通過記録を保持する手法である．この手法は，ルータにおいて通過したパケットの情報を記録し，被害ホストが隣接するルータに攻撃パケットが通過したかを問い合わせる．攻撃パケットの通過記録が存在したルータは，さらに隣接のルータに対して同様の問い合わせを行う．これを繰り返すことで攻撃経路の再構築を行なう．この手法は，攻撃パケットが１つでも経路上のルータに記録されていれば，その攻撃経路を再構築できるという利点がある．しかし，ルータにおいてすべてのパケットの通過記録を保持することは，記録するためのストレージ容量や記録処理のオーバーヘッドなどの制限により，特にバックボーンルータにおいては実現することが難しい．このため，より少ない記憶容量やオーバーヘッドで効率的に通過記録を保持する方法が提案されている [54,55]．しかし，依然としてルータへの負荷が高く，記憶容量の設定が難しいという問題や経路上のルータがパケットの通過を記録していなければ，そのルータ

から先の攻撃経路の構築ができないという問題が残っている．

もう一つは，通過するパケット自体にルータの情報を付加する手法である．この手法はパケットマーキング [56] と呼ばれる．パケットマーキングでは，ルータがパケットを転送する際にルータの IP アドレスをパケットに書き込む．攻撃経路の再構築は，受信したパケットに記録されたルータの IP アドレスをもとに行なう．IPv4 では，ルータ情報の記録は IP パケットヘッダのうち通常使用されていない ToS (Type of Service) フィールド（8 ビット），フラグメントフィールド（16 ビット）などを用いて行う [57]．このため，単独パケットでは 32 ビットのルータの IP アドレスを記録できないことから，IP アドレスのハッシュ値を用いる [58–60]，あるいは複数のパケットに分けて記録する [61–63] などの方法が考えられている．パケットマーキングはルータでの通過情報を保持する手法と比較し，ルータへの負荷が低く実装が容易であるといった利点がある．しかし，一方で攻撃経路の再構築に必要なパケットが多くなるという問題がある．

また上記以外にも，Capability System [64,65] と呼ばれる IP トレースバックと異なる手法も提案されている．この手法では，サーバとクライアントが通信を行う前に認証を行う．そして，クライアントはサーバに対して認証済みであるという資格情報を付加したパケットを送信する．ルータでは，パケットに付加されている資格情報の正当性を確認し，正当でないパケットを破棄する．このため，サーバは資格を持ったパケットのみを受信することとなり，攻撃の影響を受けにくい．一方で，サーバの認証チャネルへの DoS 攻撃も報告されている．このため，認証時の手順を複雑にすることで攻撃パケットの影響を軽減し，正当なパケットが資格を得やすくした手法も提案されている [66]．しかし，依然として認証チャンネルへの攻撃という問題が残る．

本論文では，PPM においてマーキングの重複を発生させる要因を分析し，その要因と

してホップ数に起因する要因とトポロジに起因する要因があることを示す．これらの要因を軽減させる 2 つの方式を導出することにより，マーキング重複を削減し，攻撃経路の再構築に必要なパケット数を減少させる新たな手法を提案する．

第3章　フィードバックを用いた元のフロー分布推定法の精度向上手法

3.1　概要

異常トラヒックの検出に必要となるトラヒックの統計情報の集計には，ネットワーク上のすべてのトラヒックを収集し，その統計情報の分析が有効である．しかし，高速回線では，必要とされる資源量や収集オーバーヘッドなどの問題により実現が難しい．このため，確率的に選択されたパケットのみで統計情報を収集する，パケットサンプリングが有効な手法であると考えられている．パケットサンプリングを利用することで，必要となる資源量を削減でき，効率良くフローの収集を行うことが可能である．しかし，サンプリングを行った結果，情報の欠落が発生することで統計情報の精度が劣化するという問題が発生する．本論文では，異なる間隔のパケットサンプリングで得られた統計情報の差分に着目する．得られた差分情報をフィードバックさせることで，元の統計情報をより正確に推定する手法を新たに提案する．さらに，インターネット上で公開されている実トレースデータに提案手法を適用させることで，計算量のオーダを増加させることなく，従来手法に比べ推定誤差を最大で 85% 程度削減可能であることを示す．

3.2 パケットサンプリングによる推定精度の劣化

1 章で述べたようにパケットサンプリングを用いることで，高速なネットワーク上のルータでも効率良く統計情報の収集が可能となるが，サンプリングされないパケットが生じることで，情報劣化が生じる [67,68].

ここでは，パケットサンプリングによる推定精度劣化の具体例を図 1 を用いて示す．この図では，3 種類のフロー (A, B, C) の合計 6 パケットが (A, B, A, C, C, A) の順にルータを通過した場合の例を示している．ここで，1/2 パケットサンプリングを行った場合（ここでは，サンプリングの方法として 2 パケットごとに 1 パケット抽出するものとする），A, A, C のパケットが抽出されることになる．その結果，サンプリングされたトラヒックには (フロー A, フロー B, フロー C) が (2, 0, 1) のパケットが含まれる．ここで，1/2 のサンプリングを行ったのであるから，実トラヒックのフローはサンプルされたフローのパケット数を 2 倍したものであると仮定すると，実トラヒックは (4, 0, 2) であると推定される．しかしながら，実際には元の統計情報は (3, 1, 2) であり，ここで誤差が生じていることが分かる．この理由は，パケットサンプリングによって 1 パケットのみであるフロー B の情報が消失するために，フロー B については存在しないと判断されるためである．一般化すると，1 章で述べた 2 種類のトラヒックが選択されないためである．このようにパケットサンプリングでは，一部のトラヒックがサンプリングされないため，推定誤差が生じる．このため，元のフローの統計情報の推定には，選択されないトラヒックを考慮する必要がある．

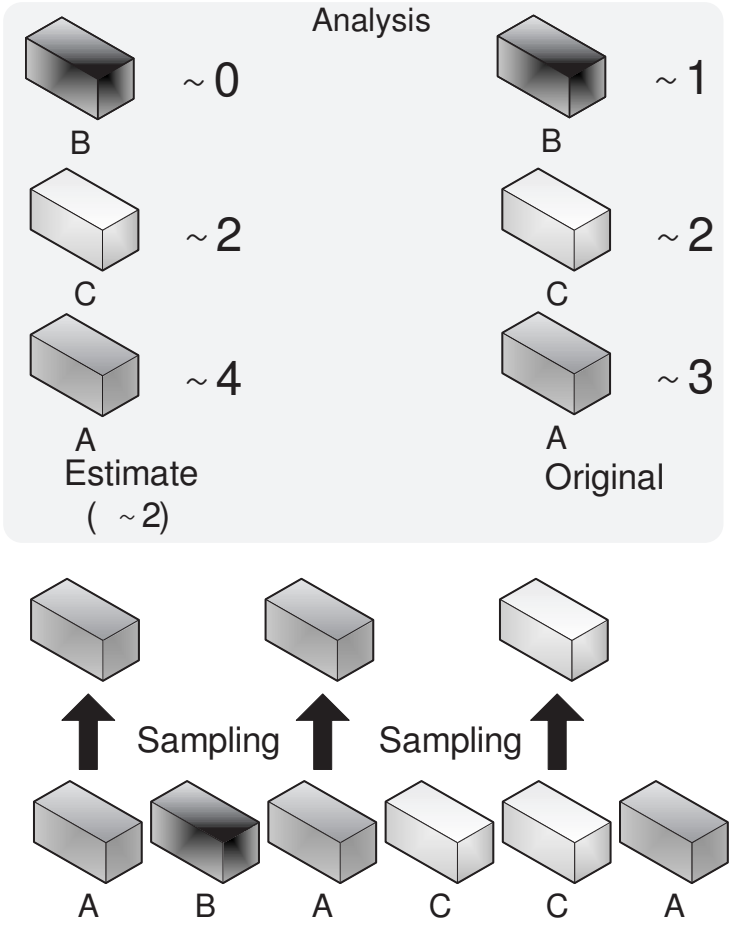


図 3.1: パケットサンプリングを用いた統計情報の推定

3.3 差分情報

本節では，異なるサンプリング間隔のパケットサンプリングで得られた統計情報間の「差分情報」について述べる．統計情報間の差分情報とは，サンプリング間隔が変化した場合に，統計情報の特徴がどのように変化するかを示したものである．1 章で述べたとおり，サンプリングを行うことによって，パケット数が少ないフローの全体あるいはパケット数が多いフローの一部の情報が消失することになる．本論文では，サンプリングによ

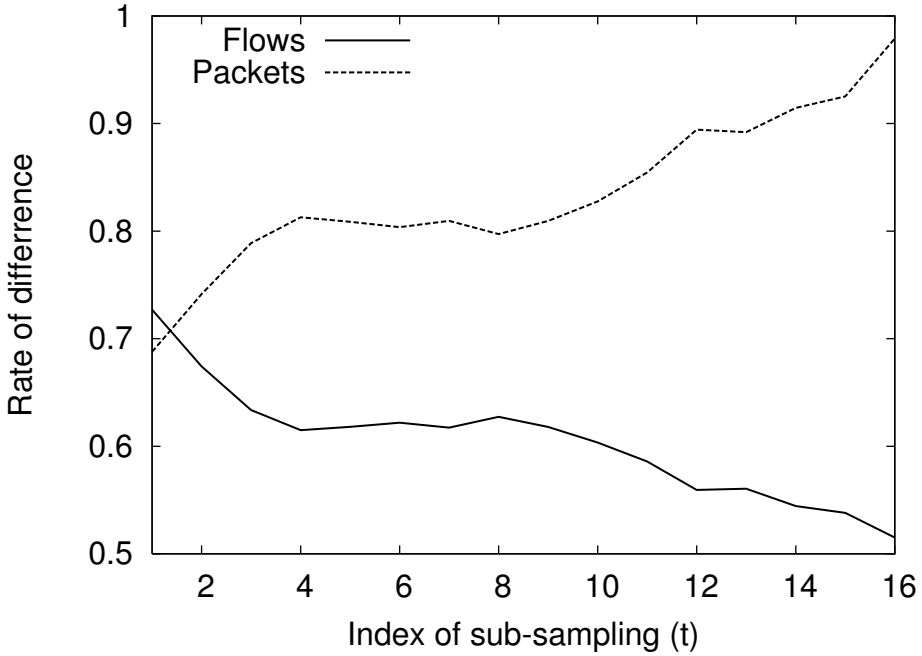


図 3.2: フロー総数とフローごとの平均パケット数の差分率 (Abilene III)

り得られた統計情報のうちフロー総数とフローごとの平均パケット数に着目し，それらがサンプリングによってどのように変化するかを示す指標として「差分率」を定義し，差分情報として用いる．

また，異なる間隔のパケットサンプリングの結果を得るために，サブサンプリングと呼ばれる手法を利用する．サブサンプリングとは，ある間隔のパケットサンプリングによって得られた統計情報を，再度サンプリングすることにより，異なるサンプリング間隔の統計情報を得る手法である．例えば，1/6 パケットサンプリングは，元の統計情報に 1/2 パケットサンプリングを行うことで得られた統計情報に対して，再度 1/3 パケットサンプリングを行うことである．

ここで，異なる間隔のサンプリングで得られた n 個の統計情報を考え，t をサブサン

プリング係数と定義し，$0 \leq t \leq n$ を満たすとする．また，サンプリング間隔を s_t とし，$1 \leq t \leq n$ で $s_{t-1} < s_t$ を満たすとする．このとき，パケット数が i であるフローの数を $f_{t,i}$ とすると，フロー総数 F_t は，

$$F_t = \sum_{i=1}^{\infty} f_{t,i} \tag{3.1}$$

となり，パケット総数 P_t は，

$$P_t = \sum_{i=1}^{\infty} i f_{t,i} \tag{3.2}$$

となる．すると，フローごとの平均パケット数 B_t は，

$$B_t = \frac{P_t}{F_t} \tag{3.3}$$

とおける．ここでフロー総数の差分率 ΔF_t を，

$$\Delta F_t \equiv \frac{F_t}{F_{t-1}} \tag{3.4}$$

と定義し，フローごとの平均パケット数の差分率 ΔB_t を，

$$\Delta B_t \equiv \frac{B_t}{B_{t-1}} \tag{3.5}$$

と定義する．

さらに ΔF_t と ΔB_t は，1 章で述べたように 2 種類のパケットが選択されないことによ

り，ΔF_t と ΔB_t の積は，サンプリング間隔の差分率 Δs_t の逆数と一致し，

$$\Delta F_t \Delta B_t = \frac{P_t}{P_{t-1}} = \Delta {s_t}^{-1} \tag{3.6}$$

で与えられる．ただし，

$$\Delta s_t \equiv \frac{s_t}{s_{t-1}} \tag{3.7}$$

である．式 (3.6) を用いることで 2 つの差分率は，一方の値がわかればもう一方の値を互いに導出することができる．

図 3.2 にトレースデータをサブサンプリングすることによって得られた統計情報間の差分情報のグラフの例を示す．図の横軸はサブサンプリング係数，縦軸はフロー総数の差分率 ΔF_t (Flows) とフローごとの平均パケット数の差分率 ΔB_t (Packets) を示している．ここでは，サンプリング間隔を 2 とし，得られた統計情報をさらにサブサンプリングすることを繰り返すことで異なる間隔のサンプリングによる統計情報を取得している．すなわち，$s_t = 2^t$ である．

図より，サンプリング間隔が長くなるにつれて，それぞれの差分率が変化することが分かる．つまり，サンプリング間隔が短い区間ではフロー総数の差分率が大きく減少するが，サンプリング間隔が長くなるにつれてその減少割合は小さくなり，フローごとの平均パケット数の差分率が大きく増加することが見られる．

これは，インターネットのトラヒックではフローごとのパケット数が一様に分布していないことに起因する．これまでの研究で，インターネットではフローごとのパケット数分布はすその部分がパレート分布であるなど，ヘビーテイル性を持つことが知られてい

る [69]．すなわち，パケット数の少ないフローが数多く存在する一方で，少数のフローが非常に多いパケット数で構成されている．サンプリング間隔が短い区間ではパケット数の少ないフローがより多く消失するため，フロー総数の差分率がより大きく変動する．サブサンプリングを繰り返すことでフロー総数が減少していくが，これにより残されたフローはいずれもパケット数が非常に多いものとなる傾向がある．したがって，サンプリング間隔が長い区間では，パケット数の多いフローのパケットがより多く消失することになり，フローごとの平均パケット数の差分率がより大きく変動する．

3.4 元のフロー分布の推定手法

本節では，差分情報を用いて元のフロー分布を推定する *EBM (Equation Based Method)* と呼ばれる新しい手法を提案する．

ここで，推定に利用する統計情報を $1/s$ パケットサンプリングでネットワークから取得した場合を考え，サブサンプリングのサンプリング確率を q とし，サブサンプリング係数 t を，$t = \log_{\frac{1}{q}} s$ と定義する．サブサンプリング係数 t は，統計情報をネットワークから取得した際のサンプリング間隔 s をサブサンプリング確率 q で表現したものである．パケット数が i であるフローの数を $f_{t,i}$ とすると，フロー総数 F_t は，

$$F_t = \sum_{i=1}^{\infty} f_{t,i} \tag{3.8}$$

となる．$t = 0$ は，サンプリングしない場合（すなわち $s = 1$) の値であり真値を表す．例えば，F_0 は元のフロー総数を表す．$t > 0$ においては，$1/s$ パケットサンプリングを行った場合の値であり，例えば，F_t は，$1/s$ パケットサンプリングされた場合に検出された

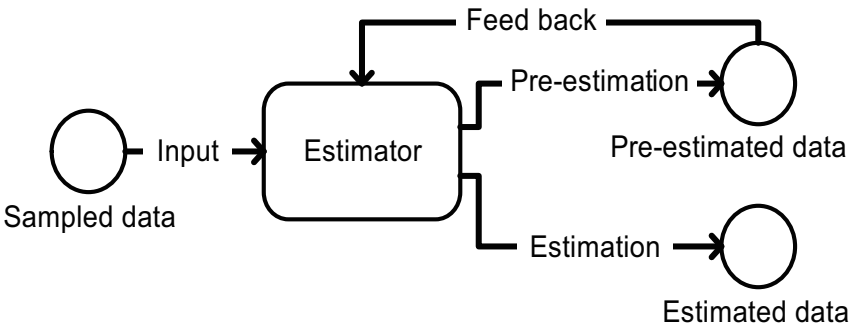


図 3.3: EBM の概要

フロー総数を表している．そして，i パケットからなるフロー数の確率質量関数 (PMF) $d_t(i)$ を

$$d_t(i) \equiv \frac{f_{t,i}}{F_t}, (1 \leq i) \tag{3.9}$$

相補累積分布関数 (CCDF) $c_t(i)$ を

$$c_t(i) \equiv 1 - \sum_{x=1}^{i} d_t(x), (1 \leq i) \tag{3.10}$$

と定義する．

まず，図 3.3 に EBM の概要を示す．EBM は，

- 事前推定 (Pre-estimation) プロセス

  MLE を用いて，元のフロー分布の事前推定を行う．

- フィードバック (Feed back) プロセス

  事前推定で得られたフロー分布に対して擬似的なパケットサンプリングを行う．そ

して，パケットサンプリングによるフローのパケット数の変化をフィードバック情報として取得する．

- 推定 (Estimation) プロセス

  フィードバックプロセスで得られた情報を用いて，元のフロー分布の推定を行う．

の 3 つのプロセスで成り立つ．

次に，各プロセスの具体的な説明を行う．ここでは，$1/s$ パケットサンプリングで得られたフローの数 $f_{t,j}$ から元のフロー分布でのフローの数 $f_{0,i}$ の推定を考える．

3.4.1 事前推定プロセス

事前推定プロセスでは，$f_{0,i}$ の近似値である $f'_{0,i}$ を MLE を用いて $f_{t,j}$ より推定する．

ここで，$d_0(i)$ の近似値を $d'_0(i)$ とすると，[32] より，$d'_0(i)$ は，切断パレート分布と ML パラメータ θ を用いて

$$d'_0(i) = d'_0(i,\theta) = \frac{i^{-\theta} - (i+1)^{-\theta}}{1 - (\nu+1)^{-\theta}} \tag{3.11}$$

と近似できることが示されている．ここで ν は，推定対象のフローの最大パケット数であり，計測で得られたフローの最大パケット数 n とサンプリング間隔 s より，

$$\nu = sn \tag{3.12}$$

となる．さらに，パケット数が i であるフローが，パケットサンプリングによって，パケット数が j に変化する確率 $p_{i,j}$ は，独立試行において，確率 q で i 個のうち j 個選択さ

れる確率と考えることができ，

$$p_{i,j} \equiv \binom{i}{j} q^j \left(1-q\right)^{i-j}, (0 \leq j \leq i) \tag{3.13}$$

である．また q は，サンプリング間隔の逆数 $1/s$ とおける．

そして，尤度関数として，

$$L(\theta) = \prod_{i=1}^{n} \left(\frac{\sum_{y=i}^{\nu} p_{y,i} d'_0(y,\theta)}{1 - \sum_{y=0}^{\nu} p_{y,0} d'_0(y,\theta)} \right)^{f_{t,i}} \tag{3.14}$$

を計算する．MLE を用いて，$f_{t,i}$ から θ を近似することで，$d'_0(i,\theta)$ を推定し，$f'_{0,i}$ の PMF である $d'_0(i)$ を

$$d'_0(i) = d'_0(i,\theta) \tag{3.15}$$

と推定する．

次に，$d'_0(i)$ から $f'_{0,i}$ の導出を考える．まず，$d'_0(i)$ を用いて擬似的なパケットサンプリングを行い，パケットサンプリングによってサンプリングされないフローの数の PMF $d'_t(0)$ を計算する．

$d'_t(0)$ は $d'_0(i)$ より，

$$d'_t(0) = \sum_{i=1}^{\infty} p_{i,0} d'_0(i) \tag{3.16}$$

となる．ここで，パケットサンプリング後のフローの PMF $d'_t(i)$ は，

$$\sum_{i=0}^{\infty} d'_t(i) = 1 \tag{3.17}$$

を満たす．そのため，$1 - d'_t(0)$ は，サンプリングされたフローの PMF の和となる．ここで，F_0 の近似値を F'_0 とすると，F_t は，

$$F_t = (1 - d'_t(0))F'_0 \tag{3.18}$$

となる．よって，F'_0 は，

$$F'_0 = \frac{F_t}{1 - d'_t(0)} \tag{3.19}$$

となり，$f'_{0,i}$ は，$d'_0(i)$ と F'_0 より，

$$f'_{0,i} = d'_0(i)F'_0, (1 \leq i) \tag{3.20}$$

となる．

3.4.2 フィードバックプロセス

フィードバックプロセスでは，パケットサンプリングによって消失する情報であるパケットサンプリングによってフローのパケット数とそのフロー数がどのように変化するかという情報を得るために，事前推定で得られたフロー数 $f'_{0,i}$ を用いて擬似的なパケットサンプリングを行い，パケットサンプリングによってパケット数が $j, (j \leq i)$ に変化したフローの数 $l'_{t,i,j}$ を計算する．

$l'_{t,i,j}$ は $f'_{0,i}$ より，

$$l'_{t,i,j} = p_{i,j}f'_{0,i}, (1 \leq i, 0 \leq j) \tag{3.21}$$

となる．また，擬似的なパケットサンプリングで得られたパケット数が j であるフローの数 $f'_{t,j}$ は，

$$f'_{t,j} = \sum_{i=j}^{\infty} l'_{t,i,j}, (0 \leq j) \tag{3.22}$$

である．ここで $f'_{t,0}$ は，サンプリングされなかったフローの数を示しており，パケットサンプリングによるフロー総数の減少量と等しくなる．そのため $f'_{t,0}$ は，事前推定で得られた F'_0 を用いて

$$f'_{t,0} = F'_0 - F_t \tag{3.23}$$

と近似する．

3.4.3 推定プロセス

推定プロセスでは，フィードバックプロセスで得られた情報を利用する．パケットサンプリングによって i パケットからなるフローから j パケット抽出されたフローの数は，式 (3.21) より得ることができる．本論文では，これらのフローに着目し，これらのフローの分布を計算する．計算した分布を用いて，i パケットからなるフローの数 $f_{0,i}$ を推定する．ここでは， $f_{0,i}$ の近似値として $f'_{0,i}$ を定義し，$f_{t,j}$ を用いて推定する．

まず，i パケットからなるフローから j パケット抽出されたフローの数の分布 $u'_{t,i}(j)$ を

$$u'_{t,i}(j) \equiv \frac{l'_{t,i,j}}{\sum_{i=j}^{\infty} l'_{t,i,j}}, (1 \leq i, 0 \leq j) \tag{3.24}$$

とフィードバックプロセスで得られた $l'_{t,i,j}$ を用いて定義する．得られた $u'_{t,i}(j)$ を用いて

$u'_{t,i}(j)f_{t,j}(0 \leq j \leq i)$ を計算することで，$f'_{0,i}$ を計算できる．したがって，$f_{t,j}$ から $f'_{0,i}$ は，

$$f'_{0,i} = \sum_{j=0}^{i} u'_{t,i}(j)f_{t,j}, (1 \leq i) \tag{3.25}$$

と計算できる．

最後に 事前推定プロセス，フィードバックプロセス，推定プロセスを含めた EBM の具体的な手順を示す．

1. $t \leftarrow \log_{\frac{1}{q}} s$ を計算．

2. $d'_0(i)$ を MLE を用いて計算．

3. $d'_t(0) \leftarrow \sum_{i=1}^{\infty} p_{i,0}d'_0(i)$ を計算．

4. $F'_0 \leftarrow \frac{F_t}{1-d'_t(0)}$ を計算．

5. $f'_{0,i} \leftarrow d'_0(i)F'_0, (1 \leq i)$ を計算．

6. $f'_{t,0} \leftarrow F'_0 - F_t$ を計算．

7. $l'_{t,i,j} \leftarrow p_{i,j}f'_{0,i}, (1 \leq i, 0 \leq j)$ を計算．

8. $u'_{t,i}(j) \leftarrow \frac{l'_{t,i,j}}{\sum_{i=j}^{\infty} l'_{t,i,j}}, (1 \leq i, 0 \leq j)$ を計算．

9. $f'_{0,i} \leftarrow \sum_{j=0}^{i} u'_{t,i}(j)f_{t,j}, (1 \leq i)$ を計算．

上記手順のうち，1. から 5. が事前推定プロセス，6. から 7. がフィードバックプロセス，8. から 9. が推定プロセスに対応する．

表 3.1: トレースデータ

トレースデータ		収集日	計測期間	総パケット数	総フロー数
Abilene III	KSCY to IPLS	2004/06/02	35 seconds	1,909,039	173,549
WIDE	Upstream	2008/03/20	15 minutes	12,906,294	1,734,260
CAIDA	CHIC to SEA	2008/07/17	36 seconds	29,949,060	2,520,710

3.5 推定結果

本節では， *NLANR* [70], *DatCat* [71] や *CAIDA* [72] で公開されているトレースデータに対して提案手法を適用し，元のフロー分布の推定を行う．そして推定結果より，ネットワークの種類に依存せず，提案手法が元のフロー分布を精度よく推定可能であることを示す．Abilene III トレースデータは，アメリカの学術ネットワークである Abilene ネットワークの OC192c バックボーンのうち Indianapolis と Kansas City 間のネットワークで収集されている．WIDE トレースデータは，日本の情報技術研究ネットワークである WIDE に接続されている 150 メガビットイーサネットから収集されている．CAIDA トレースデータは，アメリカの商用ネットワークである Chicago と Seattle 間の Tier 1 ISP をつなぐ OC192 バックボーンリンクから収集されている．表 3.1 にそれぞれのトレースデータの名称，収集日，計測期間，データに含まれるパケットおよびフローの総数を示す．

ここでフローの識別は，1 章で述べたように 5 つの情報が等しいパケットを同一フローとする．また，タイムアウトを 30 秒とし，5 つの情報が等しい場合であっても，ひとつ前のパケットを受信してから 30 秒以内に次のパケットを受信しなかった場合は別フローとみなす．

評価には，元のフロー分布と推定で得られたフロー分布に対する相対誤差や平均相対誤差が一般的に用いられるが，フロー分布のように値が指数的に減少する場合，パケット数の多い区間の誤差がパケット数の少ない区間の誤差に比べ非常に大きくなり，正確な評価が行えない．このため，[33] で提案されている平均相対誤差に元のフロー分布と推定で得られたフロー分布の平均値による重み付けを行うことで，パケット数の多い区間の影響を軽減させた真のフロー分布に対する重み付き平均相対誤差 (WMRD) で行う．WMRD は，

$$\mathrm{WMRD} \equiv \frac{\sum_{i=1}^{\infty} |c_0'(i) - c_0(i)|}{\sum_{i=1}^{\infty} (c_0'(i) + c_0(i))/2} \tag{3.26}$$

で与えられる．ここで，$c_0(i)$ はオリジナルのフロー数の CCDF，$c_0'(i)$ はサンプリングにより得られた統計情報から推定したフロー数の CCDF，i はフローのパケット数である．WMRD は，値が小さいほどより高精度で元のフロー分布を推定可能であることを示す．

図 3.4 から図 3.6 にサンプリング間隔ごとのフロー分布を示す．それぞれの分布は，各データセットを 0.1 秒ごとに区切り，推定を 10 回行ったうちの 1 回分を示している．また，比較のために EM と MLE の結果を示す．図 3.7 にサンプリング間隔が変化した場合のそれぞれの手法における WMRD の値を示す．図の横軸は，サンプリング間隔を示し，縦軸は，WMRD の値を示す．それぞれの値は，各データセットを 0.1 秒ごとに区切り，推定を 10 回行い，その平均を示している．

3.5.1 EBM の推定結果

図 3.4 から図 3.6 より，EBM は，サンプリング間隔に関係なく，EM に比べ精度よく推定可能であることがわかる．また，サンプリング間隔に比例し，推定精度が MLE に近づくが，MLE より精度が悪くなることはなく，推定可能であることがわかる．さらに，パケット数が非常に多いフローを正確に推定できている．これは，EBM では，MLE による事前推定で得られたフロー分布を用いたフィードバックにより，フロー分布の詳細な傾向を反映させることが可能であるためである．

図 3.7 より EBM の WMRD の値は EM の値に対して，サンプリング間隔が 2 の場合，60% から 85% 程度削減できている．さらにこの削減量は，サンプリング間隔に比例している．また，MLE に対して，サンプリング間隔が 2 の場合，18% から 75% 程度削減できている．この削減量は EM の場合と異なり，サンプリング間隔に反比例している．これは，パケットサンプリングによって抽出されるフローの数がサンプリング間隔に反比例し減少するため，サンプリングで得られたフロー分布から傾向を得ることができないためである．EBM の WMRD の値は，最大で 0.09% 程度 MLE の値より増加するものの MLE と同程度の精度で推定可能である．

また，図 3.7 より，EBM と MLE の WMRD の値があるサンプリング間隔以上の区間で大きく増加している．これは，パケットサンプリングで得られるフロー数が非常に少なくなり，フロー分布の傾向を十分に反映できないためである．

推定精度とサンプリングデータの記録容量はトレードオフの関係にあるため，高い推定精度が必要な場合は，サンプリング間隔を短くすることで，必要な記録容量は増加するものの，EBM を用いることで推定精度を向上可能である．

また，図 3.8 にサンプリング間隔が 128 の場合の推定に利用するトラヒックの収集期間ごとの WMRD の比較を示す．図の横軸は，トラヒックの収集期間を示し，縦軸は，WMRD の値を示す．図 3.8 よりネットワークからトラヒックを収集する期間を長くすることで，EBM は，MLE より精度よく推定できている．また MLE は，収集する期間によって，WMRD の値が増減しているが，EBM の WMRD の値は，ほとんど変化しておらず，推定精度にぶれが少なく安定している．これらは MLE では，フロー分布を 1 種類のパレート分布で表現するため，フロー分布の傾向を十分に反映できないが，収集する期間を長くすることで，サンプリングで得られるフローの数が増加し，EBM では，フィードバックプロセスによりフロー分布の傾向を反映させることが可能となるためである．

さらに，秒未満の短い期間のトラヒックを収集し，分析することは，ルータのバッファ設計や遅延に敏感なサービスの提供，輻輳制御に有効であることが [73] で述べられている．トラヒックを収集する期間が短い場合，サンプリング間隔が長いと収集されるフローの数が減少するため，サンプリング間隔を短くする必要がある．このような場合，EM や MLE では，推定精度を向上させるには，サンプリング間隔を 2 と非常に短くする必要があるが，EBM では，サンプリング間隔を 16 程度まで長くしても高い推定精度を維持できる．

以上より EBM は，サンプリング間隔によらず，サンプリングで得られたフロー分布から傾向を把握することができれば，精度よく推定可能である．

次に，提案手法と EM, MLE における計算量のオーダについて評価を行う．ここで，推定するフローの最大パケット数を m，計測されたフローの最大パケット数を n とし，$n \leq m$ を満たすものとする．

EM では，最後に見つかったパラメータの値を反映させた最尤関数の期待値の計算を行

うEステップとEステップで得られた期待値を最大にするパラメータを計算するMステップを交互に繰り返し，パラメータの推定を行う．

EMの各ステップの計算量のオーダを示す．Eステップでは，まず，推定するフローごとに計測された各フローのうち，推定するフローのパケット数以下のフローのパケット数を利用して期待値の計算を行うので，計算量のオーダは，$O(m)$である．そして計算した各フローの期待値の合計を計算するため，必要な計算量のオーダは，$O(m^2)$となる．また，Mステップでは，推定するフローごとのパラメータ更新に推定するフローのパケット数以下の計測されたフローの数と他の推定するフローのパラメータの値を利用するので，計算量のオーダは，$O(m^2)$である．そして，フローごとのパラメータ更新を推定する全フローについて行うので，必要となる計算量のオーダは，$O(m^3)$となる．よって，EステップとMステップを行うのに必要となる計算量のオーダは，$O(m^2 + m^3) = O(m^3)$となる．

MLEでは，最尤関数を定義し最尤関数を最大にするパレート関数のパラメータを計算するために，最尤関数の微分が0となるパレート関数のパラメータを計算する．この時サンプリングされたフローごとに推定するフローの最大数について繰り返し計算を行うため計算量のオーダは，$O(mn)$となる．

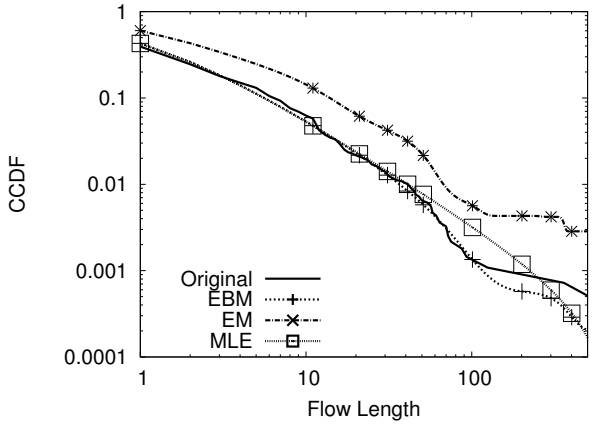
最後にEBMでは，事前推定プロセスにおいて，MLEを用いた推定を行うので，計算量のオーダは，MLEと同様に$O(mn)$となる．フィードバックプロセスにおいて，事前推定で得られたフローに対して擬似的なパケットサンプリングを繰り返し行うので，計算量のオーダは，$O(m)$となる．推定プロセスにおいて，式(16)を利用して繰り返し計算を行う際に，まず，計測されたフローごとに推定する全フローについて計算を行うので，計測されたフローごとの計算量のオーダは，$O(m)$である．そして，フローごとの計算を計測

表 3.2: 計算量のオーダ

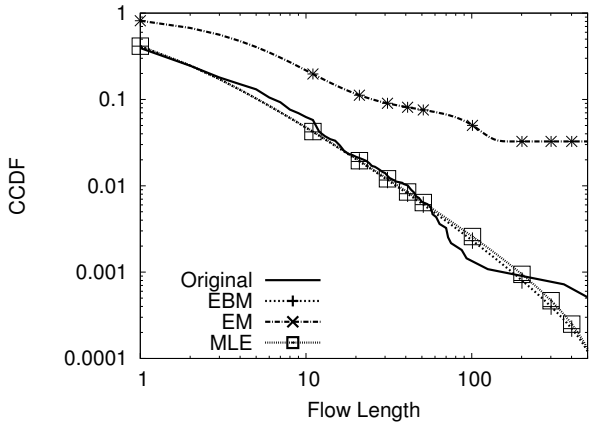
手法	計算量のオーダ
EBM	$O(mn)$
EM	$O(m^3)$
MLE	$O(mn)$

された全フローについて行うので，推定に必要な計算量のオーダは，$O(mn)$ である．よって，全てのプロセスを行うために必要な計算量のオーダは，$O(mn + m + mn) = O(mn)$ となる．表 3.2 に計算量のオーダをまとめたものを示す．以上より，EBM の計算量のオーダは，EM よりも少なく，MLE と同じである．

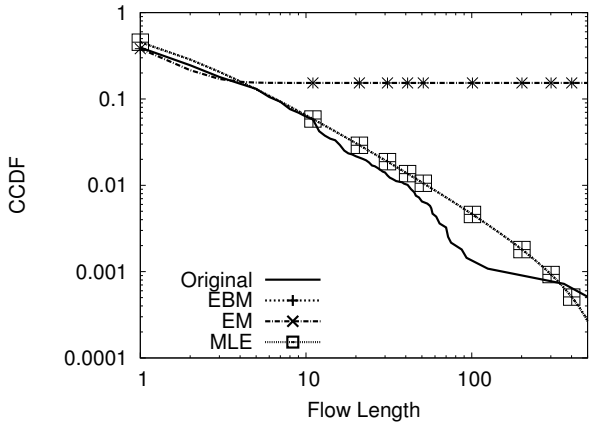
したがって，EBM は，EM に比べ，少ない計算量のオーダで，より高精度な推定を行うことができ，MLE と同程度もしくはより良い精度で推定可能である．高い推定精度が必要な場合は，サンプリング間隔を短くすることで，情報の記録に必要な容量は増加するものの提案手法を用いることで，高精度で元のフロー分布を推定可能である．また，トラヒックを収集する期間を長くすることで，必要となるデータの記録容量が増加するものの，サンプリング間隔が長くなった場合でもフロー分布の傾向を反映させることが可能となり，精度よく推定可能である．さらに，トラヒックを収集する期間を短くした場合，従来手法で高い推定精度を得るにはサンプリング間隔を 2 と非常に短くする必要があるが，提案手法ではサンプリング間隔を 16 程度まで長くしても，精度よく推定可能である．



(a) $s = 8$

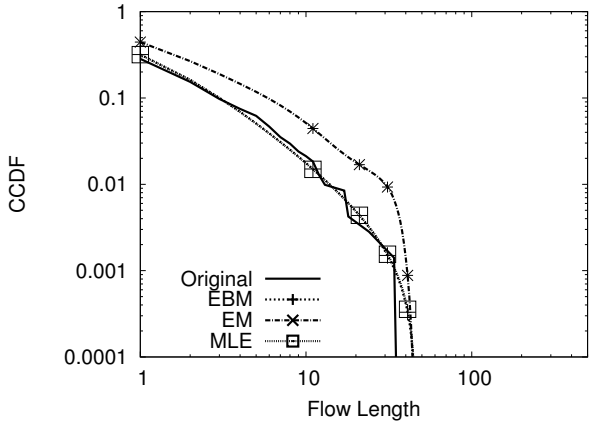


(b) $s = 128$

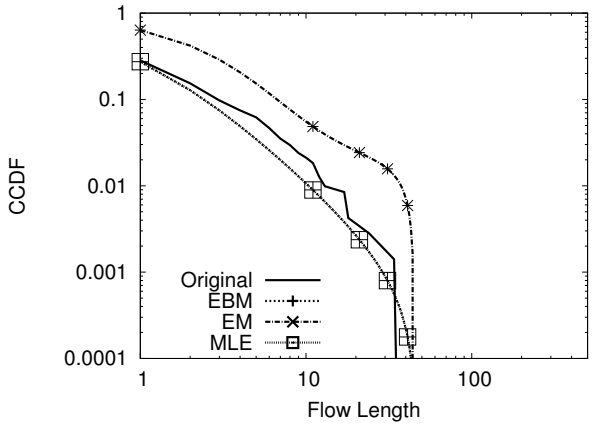


(c) $s = 1024$

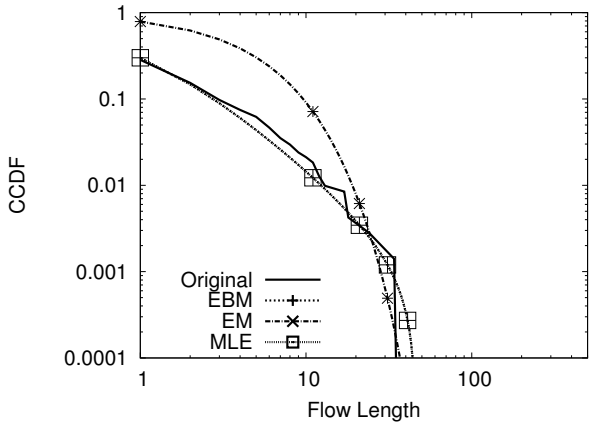
図 3.4: EBM，EM と MLE によるフロー数の CCDF (Abilene III IPLS to KSCY)



(a) $s = 8$

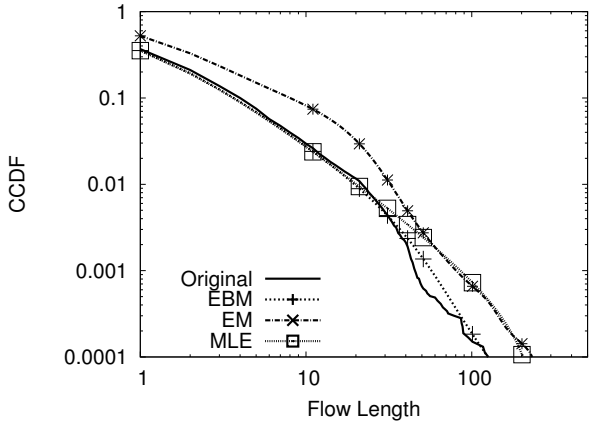


(b) $s = 128$

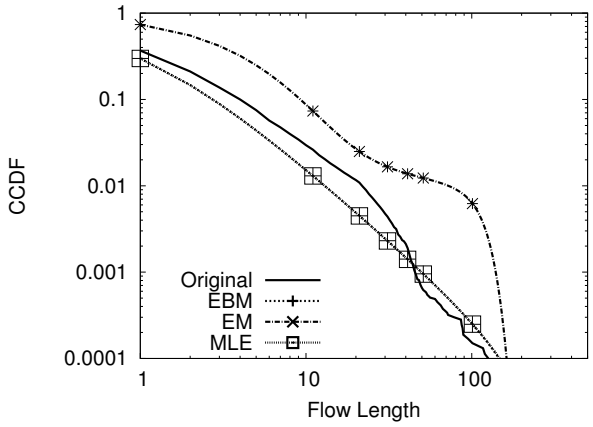


(c) $s = 1024$

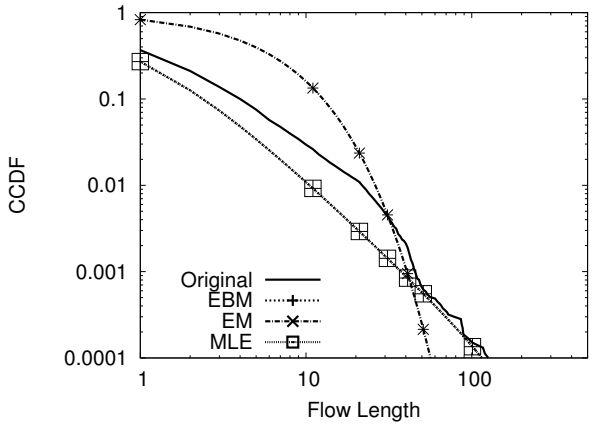
図 3.5: EBM，EM と MLE によるフロー数の CCDF (WIDE Upstream)



(a) $s = 8$

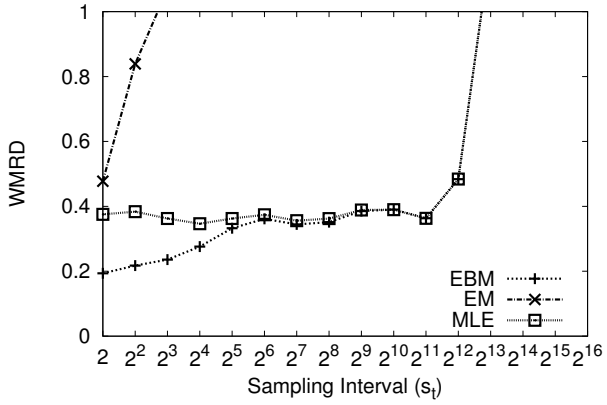


(b) $s = 128$

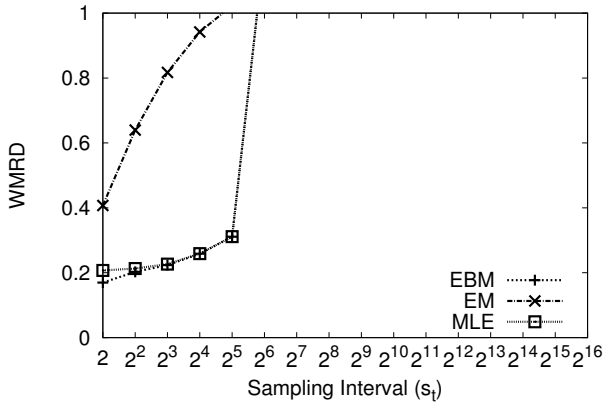


(c) $s = 1024$

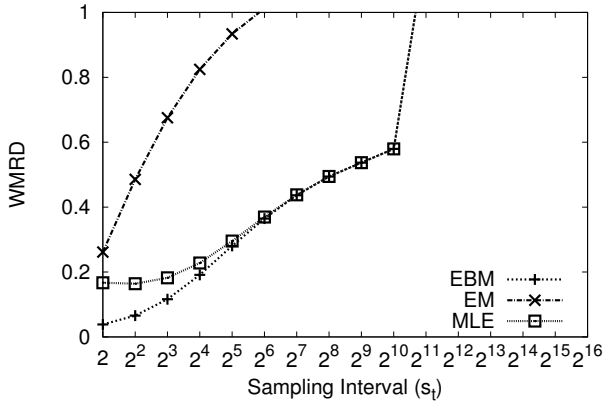
図 3.6: EBM，EM と MLE によるフロー数の CCDF (CAIDA CHIC to SEA)



(a) Abilene III IPLS to KSCY



(b) WIDE Upstream



(c) CAIDA CHIC to SEA

図 3.7: EBM，EM と MLE における WMRD の比較

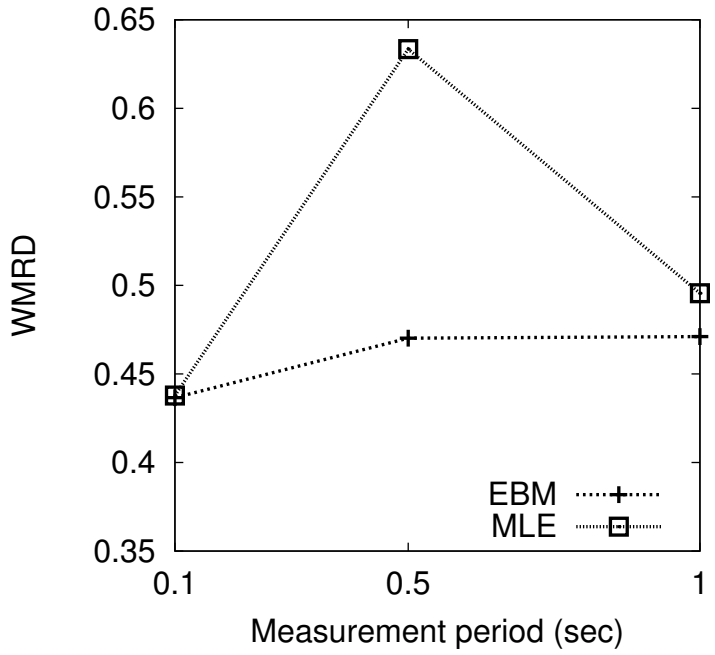


図 3.8: トラヒックを収集する期間ごとの WMRD の比較 ($s = 128$) (CAIDA CHIC to SEA)

3.6 むすび

本章では，異なる間隔のサンプリングフローの統計情報間の差分情報に着目した．パケットサンプリングにより得られた差分情報をフィードバックさせることで，元の統計情報を推定する新たな手法を提案した．そして，トレースデータを分析することにより，提案手法が EM や MLE に比べ計算量のオーダを増加させることなく従来手法に比べ推定誤差を最大で 85% 削減可能であることを示した．

今後の課題としては，推定に利用するデータセットを収集したサンプリング間隔よりパケット数が少ないフローの推定誤差を軽減させるために誤差が生じる区間の傾向を精度よく取得する方法を検討する必要がある．

第4章　アプリケーションの挙動に基づいたフローの到着パターンのモデル化

4.1　概要

異常トラヒックの検出は，異常トラヒックの統計情報の特徴を用いるのが一般的である．そのため，トラヒックの統計情報から特徴を把握する必要がある．クラウドサービスを利用する多くのアプリケーションは，ユーザへの応答時間を短縮するために短期間に複数のコネクションを確立する．この結果，短期間に多くのフローが生成されることとなり，バースト性を有するフローが発生することとなる．

このため，フローのモデル化について多くの研究がなされているが，バースト性を有するフローが考慮されていないため，フローの到着プロセスをモデル化するのが難しい．

本論文では，フローの到着間隔に着目する．フローの生成要因を分析し，2 種類のフローの生成要因が存在することを示す．ユーザ操作に基づいたフローの到着パターンを分析し，ユーザ操作とフローの到着パターンに関係があること示す．そして，フローの到着間隔分布のモデル化を行い，従来手法であるポアソン分布で近似できないことを示し，ワイブル分布で近似できることを示す．

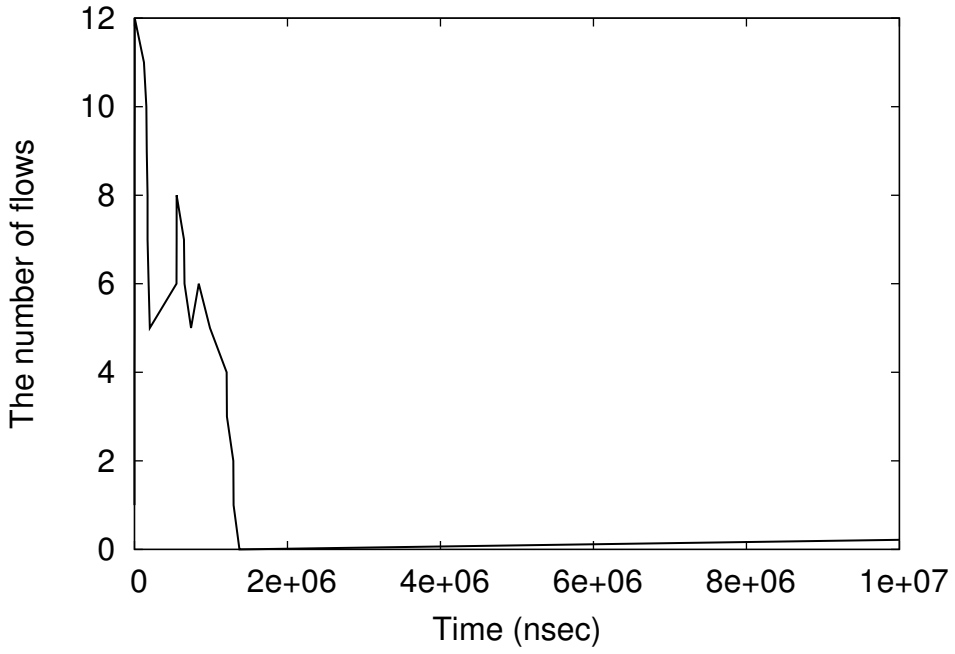


図 4.1: フローの到着間隔

4.2 フローの到着間隔と生成要因

本節では，フローの到着間隔とフローの生成要因について述べる．

図 4.1 にユーザ操作によって生成されたフローの到着パターンの例を示す．図の横軸はフローの到着時間を示し，縦軸は継続中のフローの数を示している．

図 4.1 より，一回のユーザ操作で短期間に多くのフローが生成されていることがわかる．ユーザ操作の後で最初に生成されたフローは，ユーザ操作と直接関係がある．しかし，ユーザ操作の後で生成された 2 つ目以降のフローは，ユーザ操作とは直接関係なく，アプリケーションによって自動生成されたものである．

ここで，

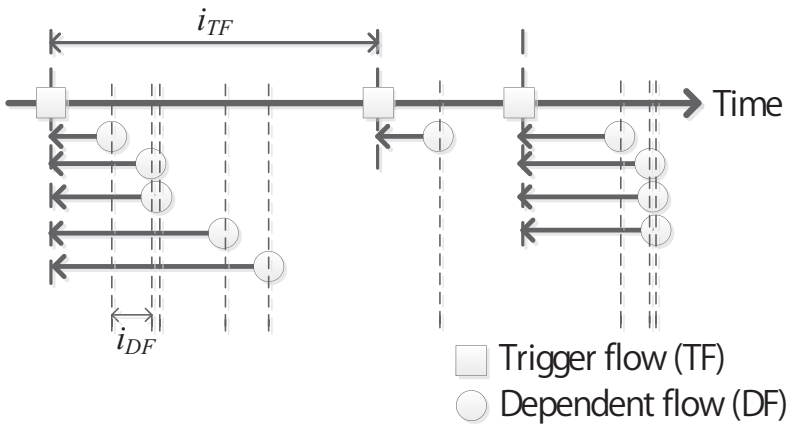


図 4.2: フローの到着プロセス

表 4.1: トリガーフローと従属フローの特徴

フローの種類	フロー数	フローの生成間隔	生成要因
トリガーフロー	1 つ	長い	ユーザ操作
従属フロー	複数	短い	自動生成

- トリガーフロー (TF)

  ユーザ操作により最初に生成される

- 従属フロー (DF)

  ユーザへの応答時間を短くするために，トリガーフローに続いて，アプリケーションにより自動生成される

の 2 種類のフローを定義する．

図 4.2 にトリガーフローと従属フローの関係を示す．図に示すように従属フローは，トリガーフローに起因して生成される．ここで，トリガーフローの生成間隔をトリガーフ

ローインターバル i_{TF} ， 従属フローの生成間隔を従属フローインターバル i_{DF} と定義する．トリガーフローと従属フローは，生成要因に依存して，i_{TF} は，長くなる傾向があり，i_{DF} は，短くなる傾向がある．

表 4.1 にトリガーフローと従属フローについて，ユーザ操作によって生成されるフロー数，フローの生成間隔，フローの生成要因をまとめたものを示す．

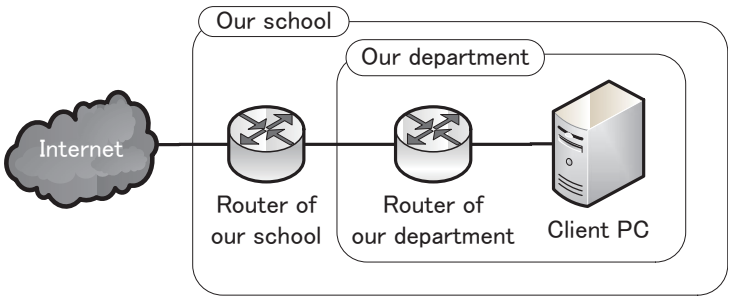


図 4.3: 計測環境

4.3 フローの到着プロセス

本節では，ユーザ操作によるアプリケーションの挙動に基づいたフローの到着パターンを分析する．

ここでは，研究室に設置したクライアント PC で計測したトレースデータを利用する．クライアント PC は，複数の学内のルータ経由して学術ネットワークである SINET [74] に接続されている．図 4.3 に計測環境を示す．さらに OS として OS X と Windows 7 を使用し，ウェブブラウザとして Firefox と Safari を使用し，表 4.2 に使用した OS と Web ブラウザを示す．

ここでフローの識別は，1 章で述べたように 5 つの情報が等しいパケットを同一フローとする．さらに，タイムアウトを 90 秒とし，IP ヘッダの情報が等しい場合でもパケットを受信してから 90 秒以内に次のパケットが届かない場合は，別のフローとする． そして 1 回のユーザ操作について 3 回トラヒックを計測した．

また，計測には，オンラインの地図サービスである Google Maps [75] と Bing Maps [76]，オンラインのオフィスサービスである Google Docs [77] と Sky Drive [78]，

表 4.2: 計測に用いた OS と Web ブラウザ

OS	Web ブラウザ
OS X 10.6	Firefox 6.0.2
OS X 10.6	Safari 5.1
Windows 7 SP1	Firefox 6.0.2

オンラインの動画サービスである Youtube [79] と Ustream [80] を利用した．それぞれのサービスについて，

- Google Maps と Bing Maps (地図サービス)
  1. ズームインボタンを 1 回クリック
  2. ズームインボタンを 3 回クリック
  3. ズームアウトボタンを 1 回クリック
  4. ズームアウトボタンを 3 回クリック
  5. 左ボタンを 1 回クリック
  6. 左ボタンを 3 回クリック
  7. 大阪市立大学を検索
- Google Docs と Sky Drive (オフィスサービス)
  1. メインページの表示
  2. 新規ドキュメントの作成
  3. ドキュメントの編集

4. クライアント PC からドキュメントのアップロード

5. アップロードしたドキュメントのオープン

6. アップロードしたドキュメントをダウンロード

7. アップロードしたドキュメントの削除

- Youtube と Ustream (動画サービス)

1. トップページの表示

2. 動画やライブの試聴

3. 大阪市立大学に関する動画や大阪府に関するライブ動画の検索

の操作を行った．

全ての計測結果より，従属フローの生成数や到着パターンは，サービスの種類に大きく依存せず，ユーザ操作によって更新されるデータの内容や量に依存することが確認できた．また，OS と ウェブブラウザごとでの従属フローの到着間隔分布の相関係数の平均値より，OS X の Fireofox と OS X の Safari での相関係数は，0.100 で，OS X の Firefox と Windows 7 の Firefox では，0.135 と ブラウザ間の相関係数の方がわずかに大きいため，従属フローの到着パターンは，OS よりもウェブブラウザに依存することが確認できる．これは，フローの到着パターンは，ユーザ操作後のフローの生成タイミングに依存しており，ブラウザがユーザ操作によってサーバからどのようなタイミングでデータを取得するかに依存しているためであると考えられる．さらに到着パターンは，

- 急増急減型

  ユーザ操作後にフロー数が急激に増加しその後大きく減少する

表 4.3: 従属フローの瞬間最大フロー数と到着間隔の中央値

	平均瞬間最大フロー数	i_{DF} の中央値 (micro sec)
急増急減型	19	419
不変型	5	761,063
急増緩減型	29	12,708

- 不変型

  フロー数がほとんど増加しない

- 急増緩減型

  フロー数が急激に増加しその後緩やかに減少する

の 3 パターンに分類できる．

それぞれの OS X で Firefox を使用した場合の従属フローの到着パターンの一例を図 4.4 に示す．図の横軸はフローの到着時間，縦軸は，継続中のフローの数を示している．また，表 4.3 に 従属フローの瞬間最大フロー数とフローの到着間隔分布の中央値を示す．

急増急減型である図 4.4(a) は， Google Maps でズームインボタンを 1 回クリックした場合のものである．このパターンは，ユーザ操作によるページ表示のために多くのフローが生成され，ページ表示完了とともに多くのフローが終了するもので，ユーザ操作の多くが該当する．

不変型である図 4.4(b) は，Google Docs でドキュメントを編集した場合のもである．このパターンは，ユーザ操作がクライアント PC のみで完結し，クライアントとサーバ間でほとんど通信が行われない場合のユーザ操作が該当する．

急増緩減型である図 4.4(c) は，Ustream でライブ映像を視聴した場合のものである．

このパターンは，ユーザ操作によるページ表示のために急増急減型と同様に多くのフローが生成されるがページ表示完了後も動画表示のために一部のフローが継続されるもので，クライアントとサーバ間で継続的に通信が行われる場合のユーザ操作が該当する．

4.4 フローの到着プロセスのモデル化

本節では，従属フローの到着間隔分布を近似することで，フローの到着プロセスのモデル化を行う．

全ての計測結果より，i_{DF} の約 50 % は，10 ミリ秒以下であり，約 70 % は，100 ミリ秒以下であった．しかし，フローの到着間隔は，ユーザ操作後に生成されたフローの最大数に依存する．これは，最大フロー数が多いとフローが，短時間に多く生成されるためである．

図 4.5(a) に従属フローの到着間隔分布の一例を示す．図の横軸は従属フローの到着間隔を示し，縦軸は 累積分布関数 (CDF) を示している．

図 4.5(a) より，従属フローの多くが，短時間で生成されていることがわかる．

ここで，以下の数式を用いて従属フローの到着間隔分布の近似を考える．各関数のパラメータは，最小二乗法を用いて求める．

- ポアソン分布

$$f(x) = b\frac{a^x e^{-a}}{x!} \tag{4.1}$$

- ガウス分布

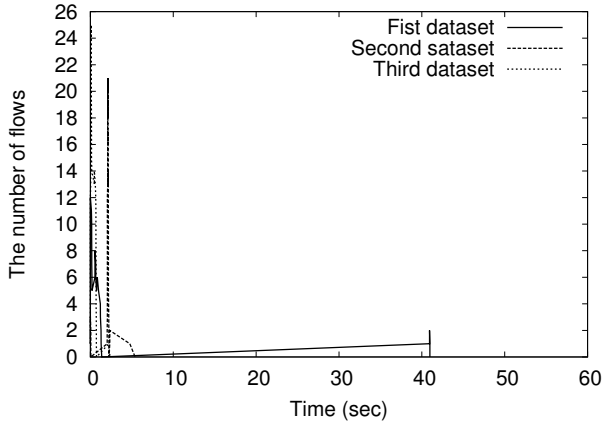
$$g(x) = l\frac{e(\frac{-(x-n)^2}{2m^2})}{\sqrt{2\pi m}} \tag{4.2}$$

- ワイブル分布

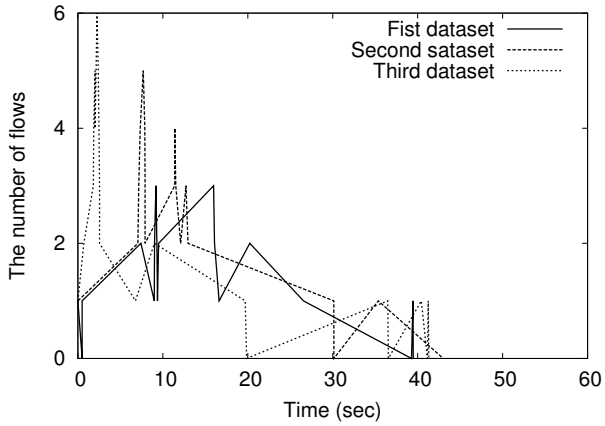
$$h(x) = s\frac{t}{u}\left(\frac{x}{u}\right)^{t-1} e\left(-\left(\frac{x}{u}\right)^{t}\right) \tag{4.3}$$

図 4.5(b) に従属フローの到着間隔分布の近似の一例を示す．図の横軸は従属フローの到着間隔を示し，縦軸は 確率密度関数 (PDF) を示している．.

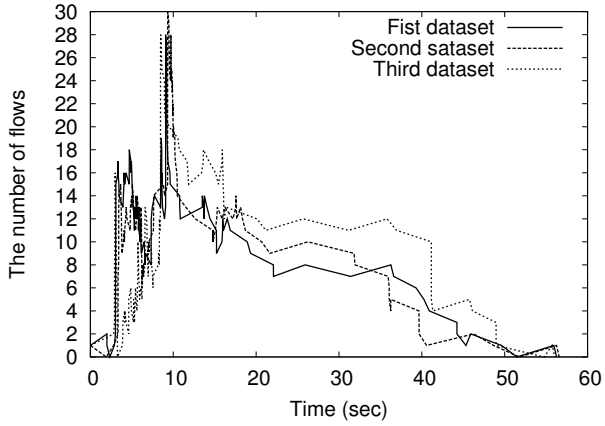
図 4.5(b) より，ポアソン分布では，パラメータが計算できず，グラフが表示されていない．一方，ガウス分布やワイブル分布では，パラメータの計算ができ，グラフが表示されている．このことより，従来手法であるポアソン分布では，近似できていないが，ガウス分布やワイブル分布を用いることで，従属フローの到着間隔分布を精度よく近似できることがわかる．



(a) 急増急減型

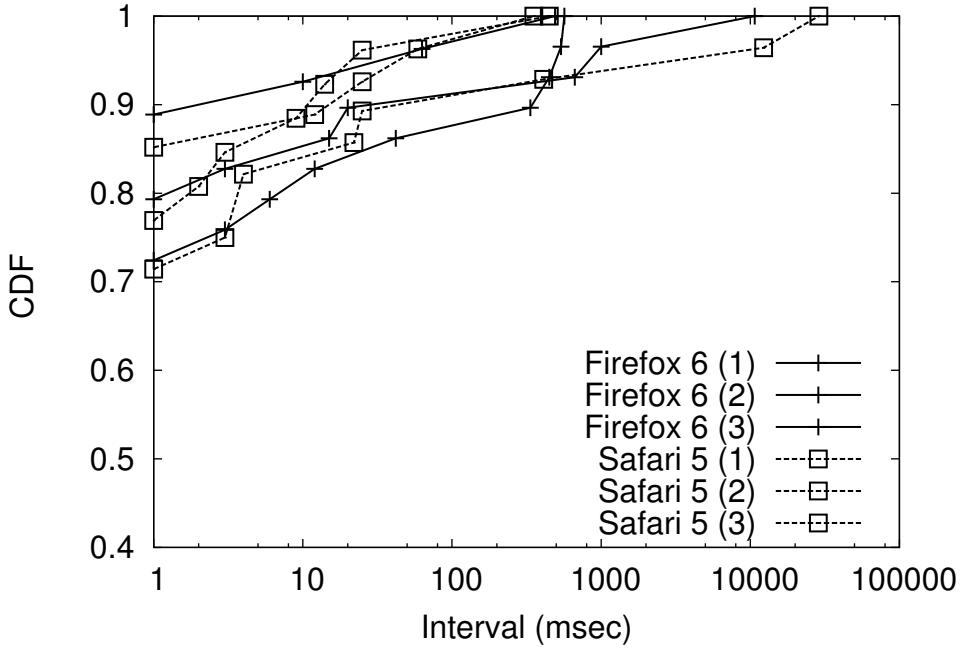


(b) 不変型

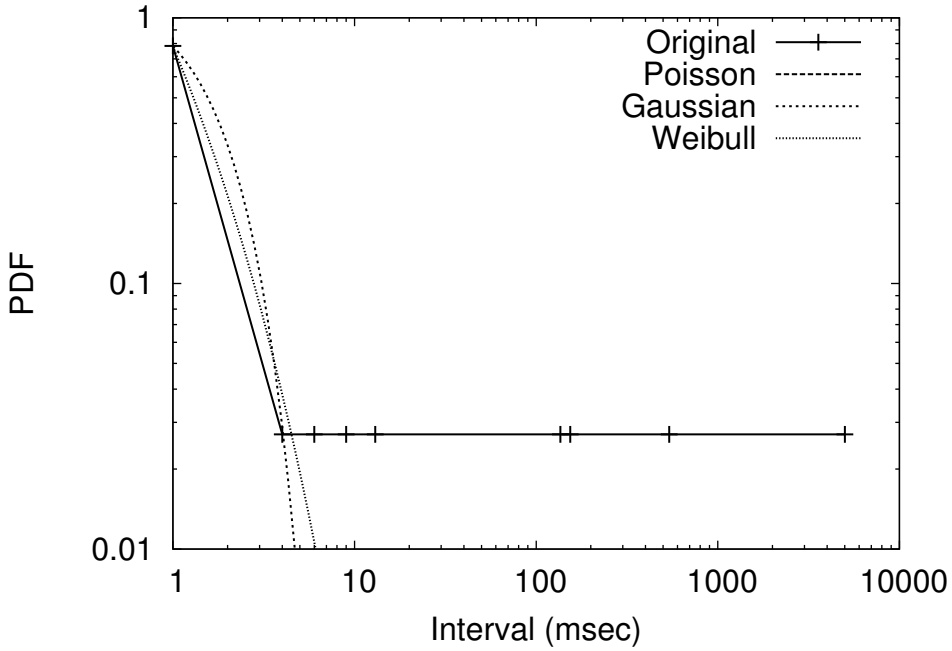


(c) 急増緩減型

図 4.4: フローの到着パターン



(a) CDF（OS X の Firefox と Safari で Google Maps の航空写真モードでズームインボタン 1 回クリックした場合）



(b) PDF（OS X の Firefox で Google Maps の航空写真モードで検索した場合）

図 4.5: 従属フローの到着間隔分布

4.5 むすび

本章では，フローの到着間隔に着目し，フローの生成要因の分析を行った．その結果，生成要因によって 2 種類のフローが存在することを示した．そして，ユーザ操作ごとのフローの到着パターンを分析し，ユーザ操作とフローの到着パターンに関係性があることを示した．さらに，フローの到着間隔の近似を行い，従来手法であるポアソン分布で近似できないことを示し，ワイブル分布で近似できることを示した．

今後は，異なった環境での計測やフローの到着パターンの分類に必要となる．トリガーフローと従属フローを分類する方法を検討する必要がある．

第5章　マーキング数推定による確率的パケットマーキングの高速化手法

5.1　概要

攻撃トラヒックの生成元の特定には，IP トレースバックと呼ばれる技術が検討されており，その 1 つに確率的パケットマーキング *(Probabilistic Packet Marking; PPM)* がある．PPM では，ルータが確率にしたがい自律的にパケットにマーキングを行うため，マーキングの上書きが頻繁に発生し，攻撃パケットが通過した経路の再構築に必要となるパケットが増加するという問題が存在する．

本論文では，PPM においてマーキングの重複を引き起こす要因を分析し，それらを軽減させることで，パラメータチューニングを必要とすることなくマーキングの重複を減少させる新しいマーキング手法を提案する．シミュレーションを用いて性能評価を行い，提案手法がパラメータチューニングを必要とすることなく，自律的に動作し，攻撃経路の再構築に必要なパケット数を従来手法に比べ最大で 85% 削減可能であることを示す．

5.2 PPM におけるパケットマーキングの重複

一般的に PPM では，被害ホストに近いルータほどマーキング重複数が多くなるという傾向があることがこれまでの研究で示されている [81]．この傾向の要因として，[82] で示されているように主に次の 2 つが考えられる．

- ホップ数に起因するマーキング重複

  PPM では，ルータはパケットがすでに他のルータによってマーキングされているかどうかに関わらず，確率的にパケットのマーキングを行うことから，ルータを経由するたびにマーキングが上書きされる可能性がある．このため，攻撃ホストと被害ホスト間のホップ数が多いほど，パケットがより多くのルータを経由することになるため，マーキング情報が上書きされやすい．その結果，マーキング重複数は被害ホストに近いルータほど指数的に増加することになる．

- トポロジに起因するマーキング重複

  上述の通り，PPM では被害ホストに近いルータほどマーキング重複数が多くなるという傾向があるが，被害ホストへのホップ数が同一のルータ間においても，被害ホストから下流に向かうリンク数によってマーキング重複数が異なる．一般的に DDoS 攻撃は，世界中に広く分散された攻撃ホストから，攻撃パケットを同時に被害ホストへ送信することで，被害ホストのサービスを停止させる．このため，多くのリンクが接続されたルータほど被害ホストへのトラヒックは増大することから，マーキング重複数が多くなる．

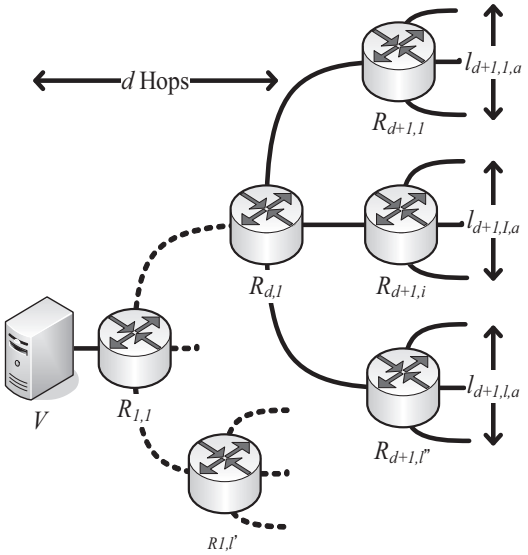


図 5.1: 攻撃ホストまでの経路

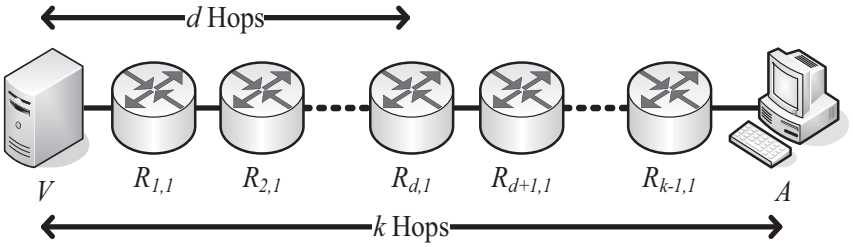


図 5.2: タンデムネットワークモデル

一般的に DDoS 攻撃の経路は，被害ホストを根としたツリー構造で表現することができる．そこで本論文では，一般的な攻撃経路として図 5.1 で示されるツリーを考える．ここで，被害ホスト (V) から d ホップ離れたルータの集合を $\mathrm{R_d}$ とし，その要素を $\mathrm{R_{d,i}}(1 \leq i \leq |\mathrm{R_d}|)$ とする．さらに，ルータ $R_{d,i}$ において，下流にアドレスが a である被害ホストを攻撃対象としている攻撃ホストが存在するエッジ数を $l_{d,i,a}$，マーキング確率を $p_{d,i}$ とする．

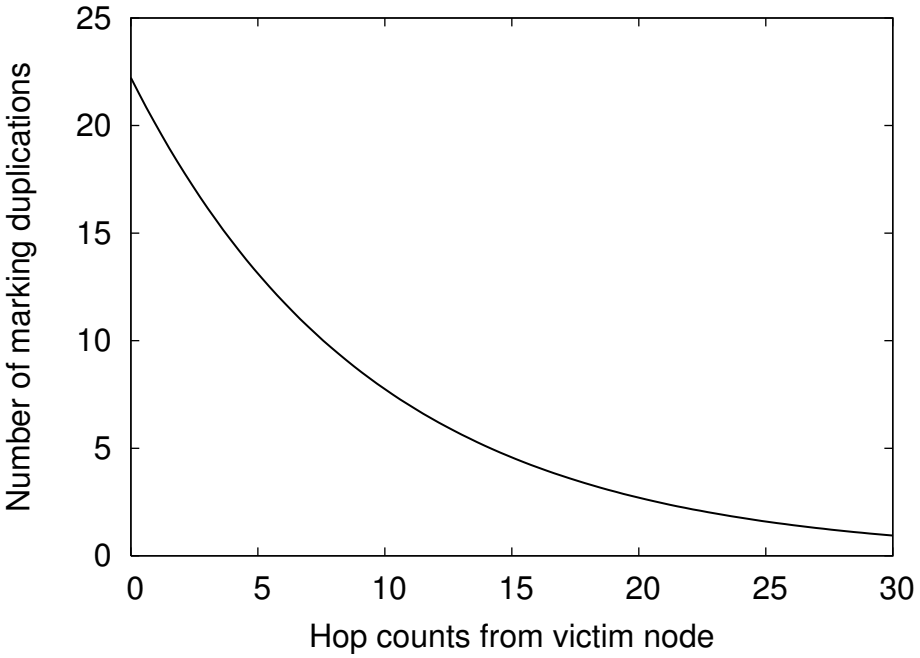


図 5.3: 各ルータにおけるマーキング重複数

5.2.1 ホップ数に起因するマーキング重複

本節ではホップ数によるマーキングの上書きによる影響を分析するため，図 5.2 で示された k ホップ離れた単一の攻撃ホスト (A) から被害ホスト (V) までが線形に接続されたモデルを用い，PPM によるマーキング重複数がどのように変化するかを述べる．したがって，各ルータの下流ルータへの接続リンク数はすべて 1 である．また PPM ではすべてのルータが同じマーキング確率 p によりマーキングを行うものとする．このとき，被害ホストにおいて，$R_{d,1}$ のマーキング重複が発生する確率を $v_{d,1}$ とすると，

$$v_{d,1} = p(1-p)^{d-1} \tag{5.1}$$

である．さらに攻撃ホスト (V) が N 個の攻撃パケットを送信した場合，ルータ $R_{d,1}$ のマーキング重複数の期待値 $n_{d,1}$ は，

$$n_{d,1} = Nv_{d,1} = Np(1-p)^{d-1} \tag{5.2}$$

となる．図 5.3 に $k = 30$ の場合の各ルータのマーキング重複数を示す．この図より，マーキング重複数は被害ホストに近いホストほど増加し，その増加数は指数的に変化することがわかる．

攻撃経路を特定するためには，必ずしも被害ホストに近いルータほど指数的に重複数を大きくする必要はない．例えば線形に重複数を変化させることによって，マーキング重複数を減少させることができる．これらは，被害ホストに近いルータほどマーキング確率を減少させることによって実現できる．

5.2.2 トポロジに起因するマーキング重複

これまでに述べたとおり，被害ホストからの攻撃経路は被害ホストを根とするツリーによって構成されると考えることができる．すなわち，被害ホストに近いルータほど下流からのトラヒックが集約され，被害ホスト宛のトラヒックが多くなる．したがって，等しい確率でパケットマーキングを行った場合，より多くの下流トラヒックを集約したルータほどより多くのパケットをマーキングすることになる．このため，被害ホストまでのホップ数が同一のルータ間においても，下流からのトラヒックが多く集約されるルータほど，マーキング重複数が多くなる．攻撃ホストがネットワークに多数かつ広く分散されている場合，被害ホストまでのホップ数が同じルータでは，マーキング重複数はルー

タの持つ下流への接続リンク数に比例すると考えられる．

攻撃経路を再構築するために必要となるマーキング重複数の増加量を維持できれば，それ以上のマーキング重複は必ずしも必要でないため，これらの不要なマーキング重複については，下流への接続リンク数が多いほどマーキング確率を小さくすることによって削減できる．

5.3 マーキング重複の削減

すでに述べたとおり，マーキング重複が発生する要因は主に 2 つ存在する．本節ではこれらの要因を考慮し，不必要なマーキング重複を削減することで，攻撃経路特定のために必要となるパケット数を削減する手法を提案する．

5.3.1 ホップ数に起因するマーキング重複の削減

図 5.3 より，マーキング重複数は被害ホストに近いルータほど指数的に増加することから，ルータのマーキング確率を被害ホストからのホップ数に応じて与えることで，ルータにおける不必要なマーキング重複を軽減する方法を考える．この場合，各ルータにおいて被害ホストまでのホップ数 d を取得する必要がある．しかしながら，ルータにおいて被害ホストまでのホップ数を計測することはルータに負荷がかかるため望ましくない．そこで本論文では，下流ルータのマーキング確率からルータのホップ数 d を推定する方法を考える．ここで，被害ホストにおいて，$R_{d,i}$ のマーキング重複が発生する確率を $v_{d,i}$

とすると，$v_{d,i}$ は，

$$v_{d,i} = p_{d,i} \prod_{x=1}^{d-1} (1 - p_{x,i}) \tag{5.3}$$

となる．すでに述べたようにマーキング重複数は，攻撃ホストに近いルータほど少なくなる．このことに着目し，被害ホストに近いルータのマーキング重複数が遠いルータの重複数に等しくなるように，被害ホストに近いルータのマーキング確率を低く設定する．これには $v_{d,i}$ が，

$$v_{d,i} = v_{d+1,i} \tag{5.4}$$

を満たせばよい．このことにより，ルータのマーキング確率 $p_{d,i}$ は，

$$p_{d,i} = \frac{p_{d+1,i}}{1 + p_{d+1,i}} \tag{5.5}$$

のように被害ホストまでのホップ数 d を取得することなく，隣接ルータのマーキング確率から計算できる．ここで，ホップ数に起因するマーキング確率を $p_{d,i}^{(H)}$ とすると，

$$p_{d,i}^{(H)} = \frac{p_{d+1,i}^{(H)}}{1 + p_{d+1,i}^{(H)}} \tag{5.6}$$

となる．

ここで図 5.4 に $k = 30$ の場合の各ルータのマーキング重複数を解析により計算したもの (Analysis) とシミュレーションにより計算したもの (Simulation) を示す．図の横軸はルータ，縦軸はマーキング重複数を示す．比較のために軽減手法を使用しない場合

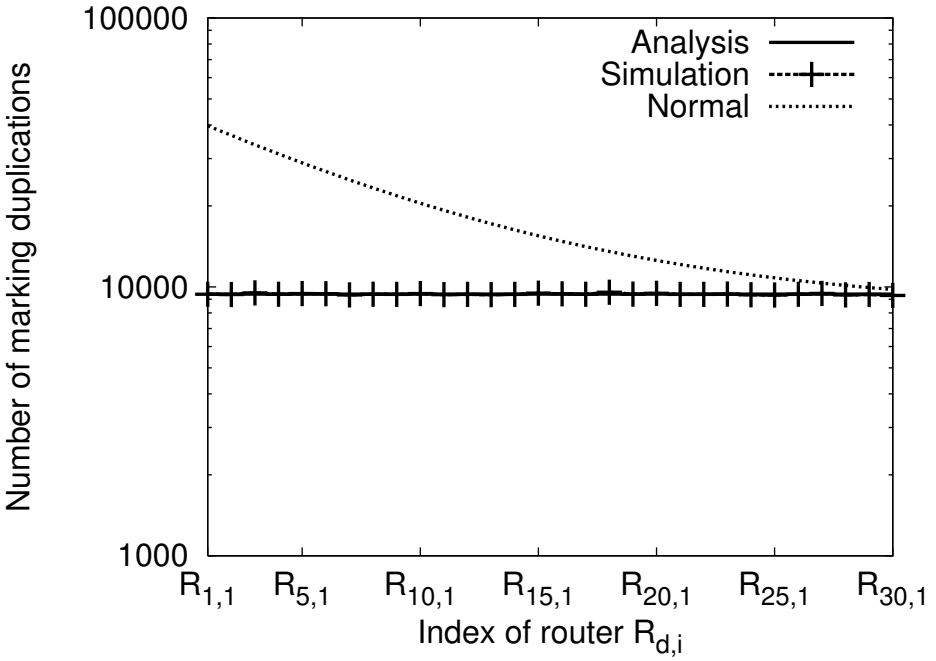


図 5.4: ホップ数によるマーキング重複の削減

(Normal) も示す．図より Normal では被害ホストに近づくにつれマーキング重複が頻繁に発生しているが，Analysis では全ルータにおけるマーキング重複の発生率が一定になっていることがわかる．一方 Simulation でも Analysis に対する相対誤差の平均が 0.0043 であることより，シミュレーションの場合でも解析で計算した場合と同様にマーキング重複を削減できていることがわかる．

5.3.2 トポロジに起因するマーキング重複の削減

被害ホストに近いルータは，より多くの下流からの攻撃トラヒックが集約されるため，同じマーキング確率を用いた場合，マーキング重複数が相対的に多くなる．理想的には，

マーキング重複数は被害ホストからのホップ数によって決定されればよく，ルータを経由するトラヒック量によって生じたマーキング重複を削減することによって，必要となるパケット数を削減することができる．ルータ $R_{d,i}$ がエッジルータではない，すなわち $R_{d,i}$ に直接攻撃ホストが接続されていない場合，$R_{d,i}$ において受信される宛先が a であるパケットの数 $N_{d,i,a}$ は，下流ルータからのトラヒックの合計値であるから，

$$N_{d,i,a} = \underbrace{\sum_{t_{k-d}=L_{d,i,a}}^{J_{d,i,a}} \cdots \sum_{t_0=L_{k,t_1},a}^{J_{k,t_1,a}}}_{k-d} N_{k,t_0,a} \tag{5.7}$$

で与えられる．ここで $L_{d,i,a}$ と $J_{x,y,a}$ は，それぞれ

$$L_{d,i,a} = \sum_{t=1}^{i-1} l_{d,t,a} \tag{5.8}$$

$$J_{x,y,a} = L_{x,y,a} + l_{x-1,y,a} \tag{5.9}$$

である．ここで，集約されたトラヒックの一部をマーキング対象とすることにより，マーキング重複数はトポロジが線形，すなわち下流ルータへの接続数が 1 の場合におけるマーキング重複数まで削減することができる．すなわち，ルータ $R_{d,i}$ の下流に存在するアドレス a のホストを攻撃対象としている攻撃ホスト数を $e_{d,i,a}$ とすると，マーキング確率は，

$$p_{d,i} = \frac{1}{e_{d,i,a}} \tag{5.10}$$

を満たせばよい．ここで $e_{d,i,a}$ は，

$$e_{d,i,a} = \underbrace{\sum_{t_{k-d}=L_{d,i,a}}^{J_{d,i,a}} \cdots \sum_{t_0=L_{k,t_1,a}}^{J_{k,t_1,a}}}_{k-d} l_{k,t_0,a} \tag{5.11}$$

である．しかし，$e_{d,i,a}$ を計算するには，ルータ $R_{d,i}$ から攻撃ホストまでのホップ数 $(k-d)$ と下流に攻撃ホストが存在する隣接ルータへのエッジ数 $l_{d,i,a}$ が必要である．しかし，通常 DDoS パケットは送信元アドレスが偽装されているため，計測ツールを用いても正しい $(k-d)$ を取得できない．また，$l_{d,i,a}$ についても下流ルータに問い合わせる必要があり，値を取得するのが困難である．ここで全攻撃ホストが生成するパケット数が等しい，つまり $N_{k,t_0,a}$ が一定であると仮定すると，式 (5.7) は，

$$N_{d,i,a} = l_{d,i,a} N_{d+1,i,a} \tag{5.12}$$

となり，$e_{d,i,a}$ も同様に

$$e_{d,i,a} = l_{d,i,a} e_{d+1,i,a} \tag{5.13}$$

となるため，ルータは隣接ルータの $e_{d+1,i,a}$ から $e_{d,i,a}$ を計算できる．しかし，依然としてルータが中継するパケットが攻撃パケットであるかそうでないかを判断できないため，正確な $l_{d,i,a}$ の値を取得することが困難である．そのため $l_{d,i,a}$ を下流に攻撃ホストおよびクライアントホストが存在する隣接ルータへのエッジ数，つまり，宛先が a であるパケッ

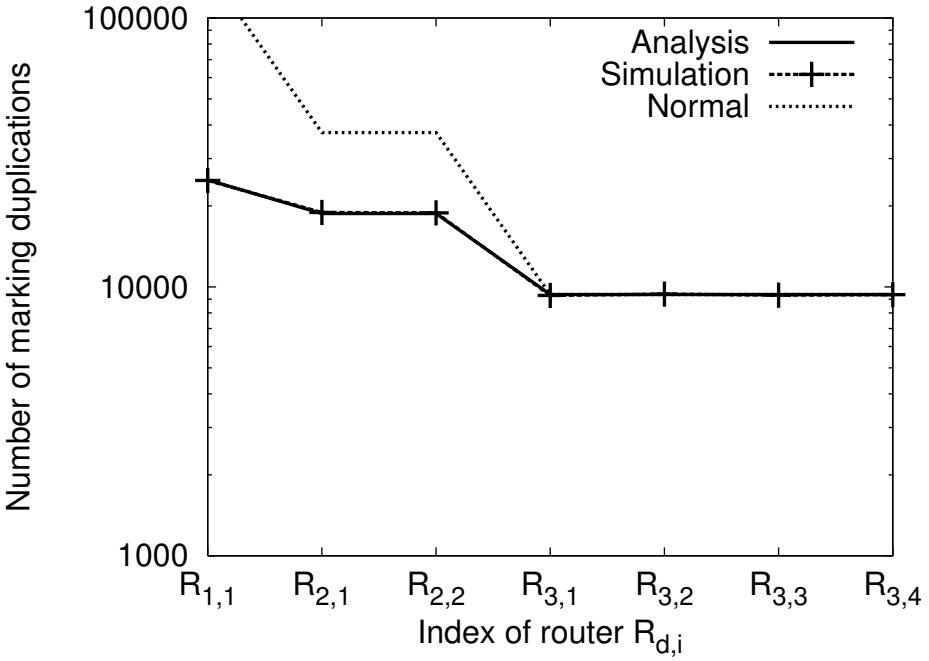


図 5.5: トポロジによるマーキング重複の削減

トが到着するエッジ数 $l'_{d,i,a}$ とする．$l'_{d,i,a}$ で近似した場合の $e_{d,i,a}$ を $e'_{d,i,a}$ とすると，

$$e'_{d,i,a} = l'_{d,i,a} e'_{d+1,i,a} \tag{5.14}$$

となる．ここで，ルータにおけるトポロジに起因する要因を考慮したマーキング確率を $p^{(T)}_{d,i}$ とすると，$p^{(T)}_{d,i}$ は，

$$p^{(T)}_{d,i} = \frac{1}{l'_{d,i,a} e'_{d+1,i,a}} \tag{5.15}$$

となる．

ここで図 5.5 に被害ホストを根，4 台の攻撃ホストを葉とする 2 分木構造で被害ホス

トから攻撃ホストまで 4 ホップのトポロジにおける各ルータのマーキング重複数を解析により計算したもの (Analysis) とシミュレーションにより計算したもの (Simulation) を示す．図の横軸と縦軸は，図 5.4 と同じである．比較のために削減手法を使用しない場合 (Normal) も示す．図より Normal では，被害ホストに近いルータほどマーキング重複数が非常に多くなっているが，Analysis では重複数を大きく削減できている．また Simulation でも Analysis に対する相対誤差の平均が 0.0053 であることより，シミュレーションの場合でも解析で計算した場合と同等にマーキング重複を削減できていることがわかる．

5.4 提案手法

本節では，不必要なマーキング重複を削減することで攻撃経路の再構築に必要なパケット数を軽減する二つの新しい手法を提案する．

一つ目は，PAPM (Parameter Auto-adjustable Packet Marking) であり，5.3.1 項と 5.3.2 項で述べたホップ数とトポロジに起因する要因を考慮した手法である．ここで，図 5.6 に PAPM のデータ構造を示す．図より，ルータは，パケットのマーキングフラグとして，フラグフィールド (1 ビット) を使用し，マーキング情報を記録するために IP ヘッダのうち 24 ビット (TOS フィールド (8 ビット) と 識別子フィールド (16 ビット)) を使用する．また，マーキング情報としてルータの IP アドレスを使用する．PAPM ではネットワークの状況に応じて最適なパラメータ値が自動的に設定されるため，PPM と異なり最適化する必要がない．しかしながら，自動的に最適化することにより，隣接ルータのマーキング確率を推定する必要がある．このため，ルータを経由するすべてのパケッ

トを処理する必要があり，ルータへの負荷が高くなる．

そこで，ルータへの負荷が低い手法として HCPPM (History Cashing based Probabilistic Packet Marking) を提案する．HCPPM でもマーキング情報を記録するために IP ヘッダのうち PAPM と同じものを使用する．HCPPM は PAPM と異なり，マーキング確率を計算する際に隣接ルータのマーキング確率を推定しないため，ルータを通過するすべてのパケットを処理する必要がない．

5.4.1 PAPM

ホップ数に起因するマーキング重複とトポロジに起因するマーキング重複は，互いに独立して生じるので，PAPM ではこれらの要因を考慮したマーキング確率を個別に計算し，計算した二つのマーキング確率の積をマーキング確率とする．ここで，マーキング確率を計算するには隣接ルータのマーキング確率が必要となる．このため，各ルータにおいてマーキング確率を計算する際に隣接ルータにマーキング確率を問い合わせることが考えられる．しかし，問い合わせ先ルータの負荷が高くなるため望ましくない．そのため PAPM では，マーキング確率を問い合わせることなく，各ルータを通過するパケットから隣接ルータのマーキング確率を推測する．具体的には，ルータにおいて宛先アドレスごとにパケットの通過数とマーキングフラグがセットされているマーキング済パケット数をカウントする．それにより，通過したパケットがすでに他の下流ルータによりマーキングされている確率を計算する．ここで a をパケットの宛先，$u_{d,i,a}$ を隣接ルータでマーキングされたパケット数，$s_{d,i,a}$ をルータを通過するパケット数とすると，隣接ルータの

マーキング確率 $p'_{d+1,i,a}$ は，

$$p'_{d+1,i,a} = \frac{u_{d,i,a}}{s_{d,i,a}} \tag{5.16}$$

と推測できる．しかし，式 (5.16) で得られるマーキング確率は，ホップ数に起因する要因を考慮した確率 $p^{(H)}_{d+1,i,a}$ とトポロジに起因する要因を考慮した確率 $p^{(T)}_{d+1,i,a}$ の積であり，個々の確率を推測できない．そこで，推測した $p'_{d+1,i,a}$ から $p^{(H)}_{d+1,i,a}$ と $p^{(T)}_{d+1,i,a}$ を別々に取得した場合のマーキング確率を計算することを考える．

ここで，隣接ルータより $p^{(H)}_{d+1,i,a}$ と $p^{(T)}_{d+1,i,a}$ を個別に取得できたと仮定することにより，ルータで計算されるマーキング確率は，

$$p_{d,i,a} = \frac{p^{(H)}_{d+1,i,a} p^{(T)}_{d+1,i,a}}{l'_{d,i,a}\left(1 + p^{(H)}_{d+1,i,a}\right)} \tag{5.17}$$

となる．一方，推定したマーキング確率から計算したマーキング確率 $p'_{d,i,a}$ は

$$p'_{d,i,a} = \frac{p'_{d+1,i,a}}{l'_{d,i,a}\left(1 + p'_{d+1,i,a}\right)} \tag{5.18}$$

となる．ここで $p'_{d+1,i,a}$ は，$p^{(H)}_{d+1,i,a} p^{(T)}_{d+1,i,a}$ の近似値であるので，

$$p'_{d,i,a} \simeq \frac{p^{(H)}_{d+1,i,a} p^{(T)}_{d+1,i,a}}{l'_{d,i,a}\left(1 + p^{(H)}_{d+1,i,a} p^{(T)}_{d+1,i,a}\right)} \tag{5.19}$$

となる．次に $\frac{1}{p_{d,i,a}}$ と $\frac{1}{p'_{d,i,a}}$ の差分を考えると，

$$\frac{1}{p_{d,i,a}} - \frac{1}{p'_{d,i,a}} = \frac{l'_{d,i,a}}{p^{(T)}_{d+1,i,a}} - l'_{d,i,a} \tag{5.20}$$

となるので，$p_{d,i,a}$ は，

$$p_{d,i,a} = \frac{1}{\frac{1}{p'_{d,i,a}} + \frac{l'_{d,i,a}}{p^{(T)}_{d+1,i,a}} - l'_{d,i,a}} \tag{5.21}$$

となり，$p'_{d,i,a}$ から計算することができる．

ただ依然として，式 (5.21) で計算するには，$p^{(T)}_{d+1,i,a}$ が必要となってしまう．そこで，

$$p''_{d,i,a} = \frac{1}{\frac{1}{p'_{d,i,a}} - l'_{d,i,a}} \tag{5.22}$$

と近似することを考える．このとき $\frac{l'_{d,i,a}}{p^{(T)}_{d+1,i,a}} > 1$ であるため，$p''_{d,i,a} > p_{d,i,a}$ となる．式 (5.21) のようにマーキング確率を低く設定することで，マーキング情報の上書きを防ぐことが可能になる．しかしながら，攻撃者がマーキング情報を偽装した場合，偽装されたマーキング情報が被害ホストに届きやすくなってしまう．このため，マーキング偽装の影響を軽減するためには，式 (5.22) のようにマーキング確率を高めに設定することが有効である．図 5.7 にマーキング確率を式 (5.22) で近似した場合 ($p''_{d,i,a}$) とマーキング確率を式 (5.21) で計算した場合 ($p_{d,i,a}$) に攻撃ホストと正常なクライアントホストが送信した総パケット数ごとの攻撃経路上のルータの発見率 (5.5 節で詳しく述べる) を示す．図 5.7(a) より，$p''_{d,i,a}$ の場合の全ルータを発見するのに必要なパケット数は $p_{d,i,a}$ の場合の $1,920$ 個に比べ約 8% 増加していることがわかる．しかしながら，図 5.7(b) より全ルータを発見

するのに必要なパケット数が $p''_{d,i,a}$ では $4,170$ 個，$p_{d,i,a}$ では $4,744$ 個であり，近似を行うことにより必要なパケット数を約 12% 削減できていることがわかる．以上より，式 (5.22) で近似することで攻撃者がマーキングを偽装しなかった場合，必要となるパケット数が増加するものの，一般的な攻撃では攻撃ホストの検出を妨げるようマーキングを偽装すると考えれられるため，式 (5.22) で近似することでマーキング偽装を効果的に削減することが可能である．

また式 (5.22) において $p'_{d,i,a} = 1, l'_{d,i,a} = 1$ となる場合，$p''_{d,i,a}$ が発散してしまう．このことは，隣接ルータが存在しないにもかかわらず式 (5.22) を使用してマーキング確率の近似をおこなうことにより生じる．そこで $p''_{d,i,a}$ が発散してしまう場合には近似を行わず，

$$p''_{d,i,a} = p'_{d,i,a} \tag{5.23}$$

とする．また，隣接ルータでマーキングされたパケットが一つも通過しなかった場合，$p'_{d+1,i,a} = 0$ となる．このため，パケットがすでにマーキングされているかを確認し，マーキングされていなければ

$$p'_{d+1,i,a} = 1 \tag{5.24}$$

とし，マーキングされていないパケットを削減する．一方，すでにマーキングされていれば，他の宛先に対する隣接ルータのマーキング数とパケットの通過数から $p'_{d+1,i,a}$ をそれぞれに計算し，その平均値 $\overline{p'_{d+1,i,a}}$ を使用し，

$$p'_{d+1,i,a} = \overline{p'_{d+1,i,a}} \tag{5.25}$$

とする．しかしながら，ルータが他の宛先にパケットを転送していない場合，平均値が計算できず，$p'_{d+1,i,a} = 0$ となる．ここでもマーキング偽装の影響を軽減するために，$\overline{p'_{d+1,i,a}} = 0$ となる場合には式 (5.24) により $p'_{d+1,i,a}$ を計算する．

各ルータは，パケットの宛先 a をキーとするマーキングカウンタテーブル，パケットカウンタテーブルとエッジカウンタテーブルを持つ．それぞれのテーブルには，値としてサンプリングしたパケット数 $s_{d,i,a}$，隣接ルータでマーキングされたパケット数 $u_{d,i,a}$ とパケットが到着したエッジ数 $l'_{d,i,a}$ を持つ．また，宛先が a のマーキング確率を $p_{d,i,a}$ とする．以下に PAPM のマーキング手順を示す．

1. マーキングカウンタテーブル，パケットカウンタテーブル，エッジカウンタテーブルから 宛先が a である $s_{d,i,a}, u_{d,i,a}$ と $l'_{d,i,a}$ を取得する．
2. パケットが隣接ルータでマーキングされていれば，$u_{d,i,a} \leftarrow u_{d,i,a} + 1$ を計算．
3. $p'_{d,i,a} \leftarrow \frac{\frac{u_{d,i,a}}{s_{d,i,a}}}{1.0+\frac{u_{d,i,a}}{s_{d,i,a}}}$ を計算．
4. $p'_{d,i,a} = 0$ でパケットがマーキングされていれば，$p'_{d,i,a} \leftarrow \overline{p'_{d,i,a}}$ を計算．
5. $p'_{d,i,a} = 0$ であれば，$p'_{d,i,a} \leftarrow 1.0$ を計算．
6. $p'_{d,i,a} \leftarrow \frac{p'_{d,i,a}}{l'_{d,i,a}}$ を計算．
7. $p_{d,i,a} \leftarrow \frac{1}{\frac{1}{p'_{d,i,a}} - l'_{d,i,a}}$ を計算．
8. $p_{d,i,a} > 1.0$ であれば，$p_{d,i,a} \leftarrow p'_{d,i,a}$ を計算．
9. $p_{d,i,a} = 1.0$ でパケットがマーキングされていれば，$p_{d,i,a} \leftarrow p'_{d,i,a}$ を計算．
10. マーキング確率 $p_{d,i,a}$ にしたがって，ルータの IP アドレスのハッシュ値を記録する．

5.4.2 HCPPM

HCPPM は，被害ホスト宛パケットの通過数が被害ホストに近いルータほど多くなる点に着目し，各ルータのマーキング確率をパケットの通過数をもとに動的に変化させ，被害ホストからのホップ数に応じてマーキング確率を指数的に増加させる．ただし，通過するパケットをすべてカウントするとルータへの負荷が高いため，パケットサンプリングにより抽出されたパケットの数を宛先アドレスごとにカウントする．このとき，サンプリングされたパケットに他のルータのマーキングがされていなければ，マーキング確率に関係なくマーキングを行う．

ルータ $R_{d,i}$ は，パケットの宛先 a をキーとするパケットカウンタテーブルを持つ．テーブルには値としてこれまでにサンプリングしたパケット数 $s_{d,i,a}$ を保持する．さらに，マーキング確率の最大値 $h_{d,i}$ とし，T をマーキングトリガ数とする．以下に HCPPM のマーキング手順を示す．

1. サンプリング確率 $q_{d,i}$ でパケットを抽出する．
2. パケットカウンタテーブルから宛先が a である $s_{d,i,a}$ を取得する．
3. $s_{d,i,a} \leftarrow s_{d,i,a} + 1$ を計算．
4. $p_{d,i,a} \leftarrow \frac{T}{s_{d,i,a}}$ を計算．
5. $p_{d,i,a} > h_{d,i}$ であれば，$p_{d,i,a} \leftarrow h_{d,i}$ を計算．
6. パケットがマーキングされていれば，$p_{d,i,a} \leftarrow 1.0$ を計算．
7. マーキング確率 $p_{d,i,a}$ にしたがって，ルータの IP アドレスのハッシュ値を記録する．

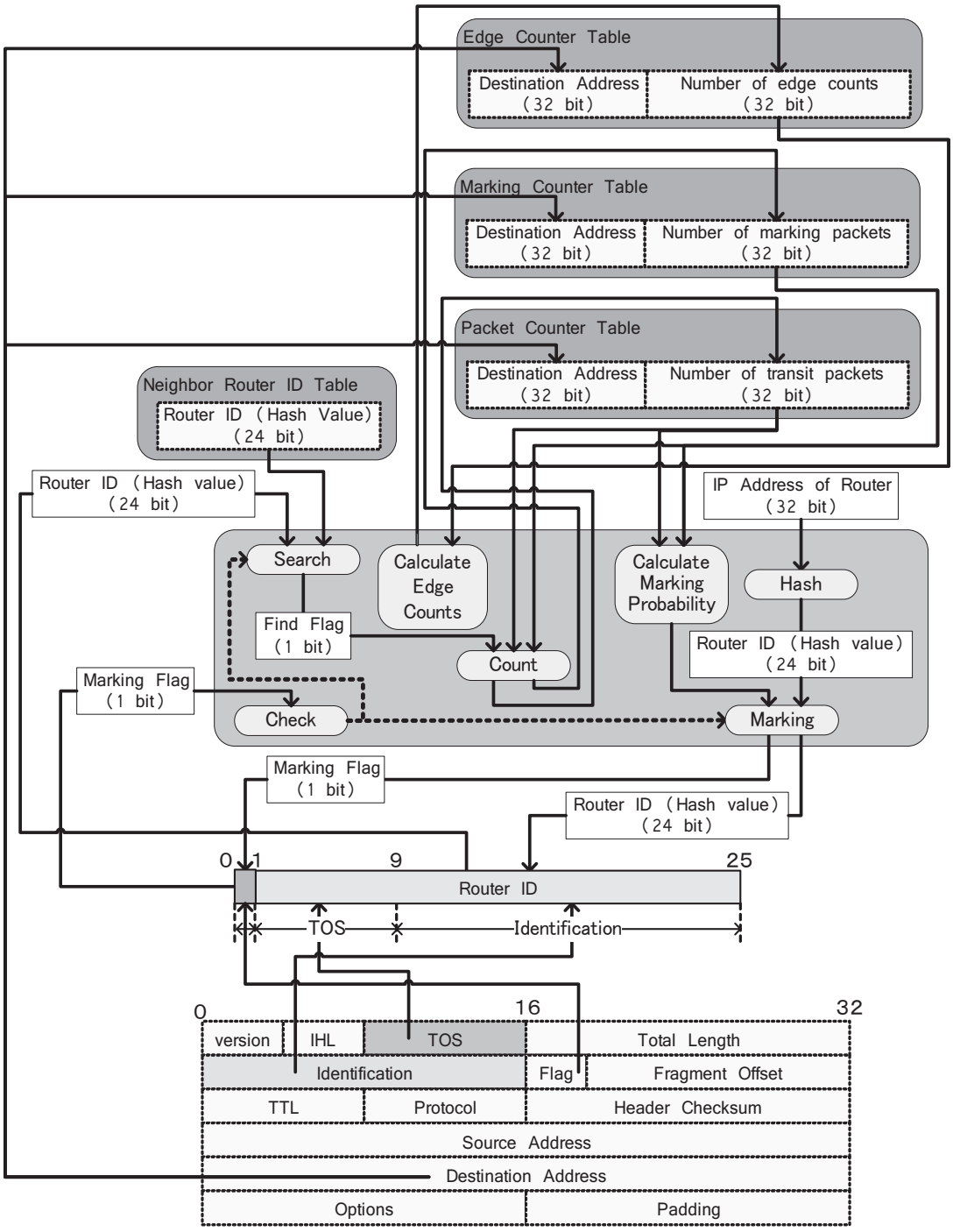
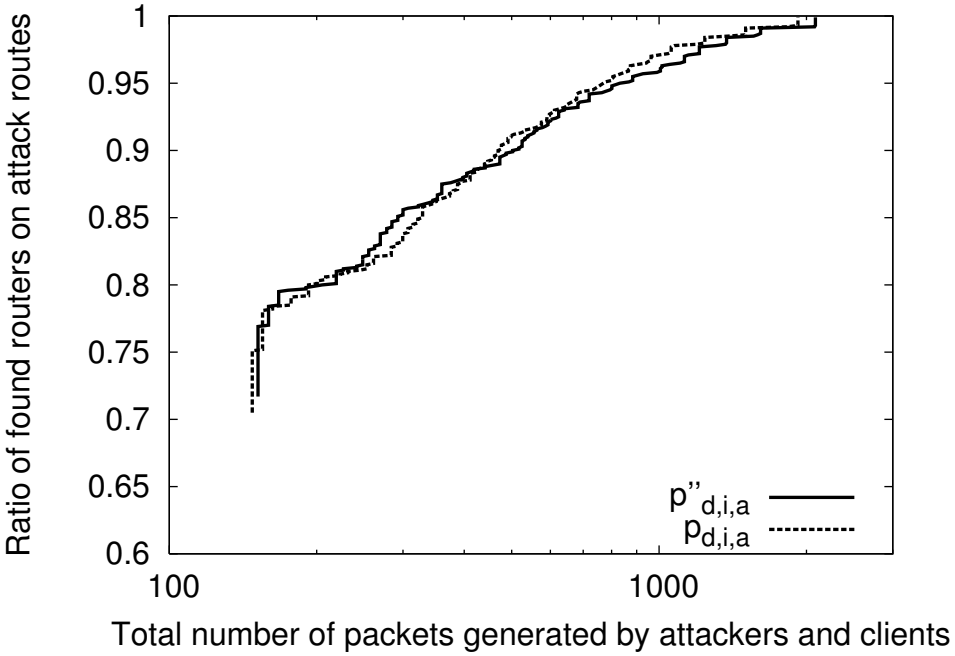
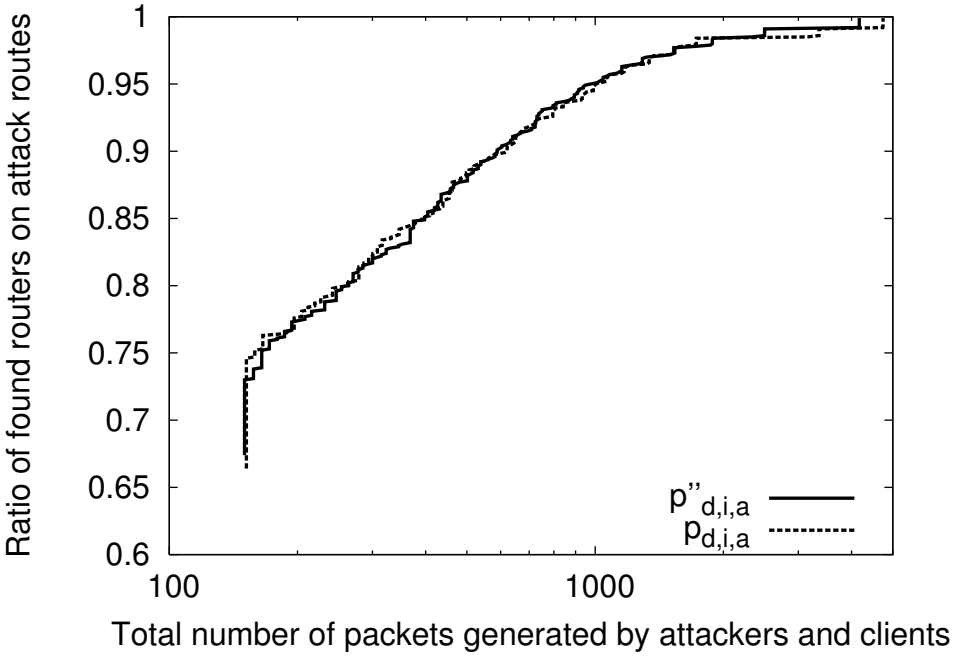


図 5.6: PAPM のデータ構造



(a) マーキング偽装なし



(b) マーキング偽装あり

図 5.7: 式 (5.22) による近似の影響

5.5 シミュレーションによる評価

本節では，提案手法の評価を行う．事前に攻撃ホストと攻撃を行わない正常なクライアントホストから合計 $50,000$ 個のパケットを送信し，その後，同様に $100,000$ 個のパケットを送信する期間におけるルータ発見率と攻撃経路上の全ルータを発見するのに必要なパケット数を評価指標とする．ルータ発見率とは，攻撃経路上の全ルータ数に対する被害ホストが受信したパケットから発見できたルータ数の割合であり，攻撃経路上の全ルータ数を b，送信されたパケット数が x の時点での被害ホストが受信したパケットから発見できたルータ数を c_x とするとルータ発見率 z_x は，

$$z_x = \frac{c_x}{b} \tag{5.26}$$

となる．発見率による評価では，より少ないパケット数で発見率が高いほど，効率良く攻撃経路の再構築できていることとなる．同様にパケット数による評価では，パケット数が少なければ少ないほど効率的にマーキングが行えていることとなる．

5.5.1 シミュレーション環境

トポロジには，トポロジジェネレータである Brite [83] で生成したノード数 $1,000$ の GLP モデルを使用し，トポロジ上のノードをルータ $R_{d,i}$ とする．トポロジ上の $1,000$ 台のルータに対して，ランダムに被害サーバホスト 1 台，攻撃を受けていない正常なサーバホスト 10 台，攻撃ホスト 100 台，攻撃を行わない正常なクライアントホスト 100 台を接続する．

また，以下の条件を仮定する．

- 攻撃ホストは IP アドレスを偽装する．
- 攻撃ホストは被害ホスト宛のパケットのみ送信する．
- 攻撃ホストはマーキングの偽装を行う．
- パケットはルータにおいて遅延や消失しない．
- パケットのフラグメンテーションは起こらない．

ここで，一般的なインターネットのトラヒックを考慮して，正常なクライアントホストは，

$$f(x) = \frac{e^{-1}}{x!} \tag{5.27}$$

を満たす平均パケット生起率 1 のポアソン分布 $f(x)$ にしたがいランダムに生成したパケットの集合 (フロー) を宛先ごとに生成し，フローごとにランダムに宛先サーバとして被害ホストと正常なホストのどちらかを選択し，パケットを送信するものとする．

5.5.2 提案手法の効果

まず，各手法が攻撃中にどれだけ効率良く攻撃経路上のルータを発見可能であるかを調べる．図 5.8 に攻撃ホストとクライアントホストが送信した総パケット数ごとのルータ発見率を示す．PPM のパラメータは $P_{d,i} = 0.2$，HCPPM は，$h_{d,i} = 0.4, q_{d,i} = 0.7, T = 150$ である．一方 PAPM では最適なパラメータ値が自動的に設定されるためパラメータの設

定は不要である．図の横軸は攻撃ホストとクライアントホストから送信されたパケット数，縦軸はルータ発見率を示している．

図 5.8(a) より PPM は，提案手法である HCPPM と PAPM に比べ，高いルータ発見率を得るには，非常に多くのパケットが必要であることがわかる．攻撃経路上のルータをすべて発見できた時点での各手法のパケット数が，PPM では 6,566 個，HCPPM では 947 個，PAPM では，2,085 個であることより，提案手法は PPM に比べ，約 70% から 85% のパケット数を削減できていることがわかる．HCPPM では，受信したパケットが他のルータでマーキングされていない場合，必ずマーキングを行うことで，攻撃経路上のどのルータでもマーキングされていないパケットを被害ホストが受信する数を削減している．このことにより，HCPPM は もっとも少ないパケット数で攻撃経路上のすべてのルータを発見することができると考えられる．また HCPPM が攻撃経路上の全ルータを発見した時点のパケット数での PAPM の発見率は 96% であり，PAPM も効率良く攻撃経路上のルータを発見できていることがわかる．

また，HCPPM と PAPM では，パケットがすでにマーキング済みであるかを確認するため，攻撃者がマーキング情報を偽装することにより，その影響を受けルータ発見率が低下することが考えられる．図 5.8(b) より HCPPM と PAPM において，攻撃ホストがマーキング情報を偽装した場合，各手法ともマーキング偽装が行われない場合と比べ，攻撃経路上のルータを発見するのに必要なパケット数は増加している．また，パケット数が 150 のとき PAPM は，HCPPM に対して発見率が 50% 以上向上している．パケット数が 1,000 の場合も同様に 15% 以上向上している．一方 HCPPM では，パケット数が 150 のとき，マーキングが偽装されていない場合に比べ，発見率が 50% 以上低下している．これは，マーキング偽装により，すべてのパケットがマーキングされている状態にな

り，マーキングされていなければ必ずマーキングするという処理が行えないためであると考えられる．このことより， PAPM はマーキングが偽装された場合でも，攻撃開始から短時間で多くのルータを発見することが可能であり，効率良くマーキング可能であることがわかる．

HCPPM と PAPM はネットワーク上の全ルータに導入することができれば，もっとも効率良く攻撃経路上のルータを発見することが可能になるが，一度にすべてのルータに導入することは困難である．このためこれらの手法の導入は，段階的に行われることとなり，従来手法である PPM と混在することが予想される．そこで，図 5.9 に $P_{d,i} = 0.2$ である PPM を導入しているルータで構成されるネットワークのうち，一部のルータのみが HCPPM と PAPM を導入した場合に攻撃ホストとクライアントホストから送信されたパケット数が $1,000$ である時点でのルータ発見率を示す．HCPPM と PAPM を導入するルータの割合は，$0\%, 25\%, 50\%, 75\%, 100\%$ とする．図 5.9(a) よりマーキングが偽装されなかった場合，導入した割合が 75% の場合を除き HCPPM の方が発見率が高いことがわかる．しかしながら全ての導入割合において，HCPPM と PAPM の発見率の差は 5% 以下となっている．このことより，HCPPM と PAPM は，導入割合が少なくなるにしたがい発見に必要となるパケット数は増加するが，従来手法である PPM が混在した環境でも動作可能であることがわかる．PAPM を導入したルータの割合が 25% の場合，マーキング偽装されていなければ，ルータ発見率は 100% 導入した場合に比べ約 25% 低下しているものの，PPM を 100% 導入した場合より約 13% 向上している．図 5.9(b) よりマーキングが偽装された場合，偽装されていない場合に比べ PAPM での発見率は 5% 程度しか低下していないが，HCPPM は最大で 20% も低下している．このことより HCPPM は，マーキング偽装の影響を受けることがわかる．また PAPM を 全体の 25% のルータに導

入することにより，導入しない場合に比べ発見率を 13% 向上させることが可能である．

以上より HCPPM は，パケットサンプリングを使用するので，ルータへの負荷が少ない．しかし，攻撃経路上の全ルータのパラメータを最適化する必要があり (5.5.3 項で詳しく述べる)，マーキング偽装の影響も受ける．一方 PAPM は，ネットワークの状況によってパラメータチューニングを行う必要がなく，マーキング偽装の影響をほとんど受けないため，実用的である．

5.5.3 パラメータによる影響

HCPPM は，PPM と比較して設定可能なパラメータが増加している．ネットワークのトポロジやトラヒックの特性によってパラメータが大きく変化すれば，状況に応じたパラメータの最適化が必要となるおそれがある．そこで本節では，提案手法の各パラメータによって必要パケット数にどのような影響が現れるのかを調べる．

はじめに HCPPM のサンプリング確率 $q_{d,i}$，およびマーキング確率の上限値である最大マーキング確率 $h_{d,i}$ の影響について調べる．ただし マーキングトリガ数 T は固定 ($T = 150$) とする．図 5.10 にサンプリング確率を 0.01 から 0.9 まで変化させた場合の必要パケット数の変化について示す．比較のため，$p_{d,i} = 0.2$ の PPM の結果もあわせて示す．図より，サンプリング確率は必要パケット数に大きな影響を与えることがわかる．特にサンプリング確率が非常に小さいとサンプル数が少なくなり，マーキング確率が高く設定される．そのため，マーキング確率が高く設定されてしまい，マーキングの重複が頻繁に発生し，必要なパケット数が急激に上昇する．一方，サンプリング確率を上昇させた場合も必要パケット数が増加するが，この傾向はサンプリング確率を低く設定した

場合と比較して穏やかである．この理由としてサンプリング確率が上昇すると，サンプル数が多くなり，マーキング確率が低く設定されるので，どのルータにもマーキングされないパケットが増加していることが考えられる．したがって，本シミュレーションの環境でもっとも最適な値は $q_{d,i} = 0.7$ のときであるが，ネットワーク状況の変化を考慮した場合，若干高めの値を設定する方が望ましいと考えられる．

次に，図 5.11 に最大マーキング確率を 0.1 から 0.9 まで変化させた場合の結果を示す．先ほどの結果とは異なり，マーキング確率を変化させても必要パケット数はあまり変化せず，パラメータによる影響が少ないことがわかる．一方，PPM に着目すると，マーキング確率によって必要パケット数が急激に変化しており，PPM のパラメータ設定が非常に難しいことが読み取れる．図 5.10 を参照すると HCPPM もサンプリング確率による影響が無視できないが，最適な場合の必要パケット数に対して 50% 程度までの増加を許容する場合，HCPPM は最適な値を含むように 0.4 から 0.9 までのサンプリング確率の変化を許容できるが，PPM ではマーキング確率の許容範囲は 0.1 から 0.3 までと非常に狭い．このことからも提案手法が PPM と比較してパラメータの設定が容易であると考えることができる．

次に，HCPPM におけるマーキングトリガ数 T の影響について調べる．図 5.12 に T を 50 から 200 まで 50 ずつ変化させた場合のサンプリング確率による必要パケット数の変化を示す．ここでは，$h_{d,i} = 0.4$ とする．図の結果よりサンプリング確率による影響は T にも依存することがわかる．ただし，このことは逆に言えば T を適切に設定することによって，HCPPM ではサンプリング確率の影響を抑えることも可能であると考えることもできる．

表 5.1: 提案手法の特徴

	PPM	HCPPM	PAPM
ルータへの負荷	とても低い	低い	高い
実装	とても容易	容易	容易
パラメータ数	少ない	多い	多い
パラメータチューニング	必要	必要	不要
必要パケット数	多い	とても少ない	少ない
マーキング偽装の影響	受ける	受ける	受けない

5.6 提案手法の比較

5.5 節で述べたように HCPPM と PAPM は，PPM に比べ効率的に攻撃経路上のルータを発見可能であることを示した．しかしながら，HCPPM はパラメータの最適化が必要であったり，PAPM は，処理が複雑であったりといった特徴がある．本節では，PPM と提案手法の特徴の比較を行い，どの手法がより実用的であるかを述べる．表 5.1 に各手法の特徴を示す．

表 5.1 より PPM は，単純に確率にしたがってマーキングを行うだけなので，ルータへの負荷は低い．HCPPM では 5.4 節で述べたように，マーキングを行うためにパケット数のカウントやマーキングの有無を調べることが必要となるため，ルータへの負荷は少し高くなる．一方 PAPM は，HCPPM と同等の処理に加え，受け取ったパケットのマーキング情報が隣接ルータによってマーキングされたものであるかを調べる必要がある．このため，ルータへの負荷は高くなる．

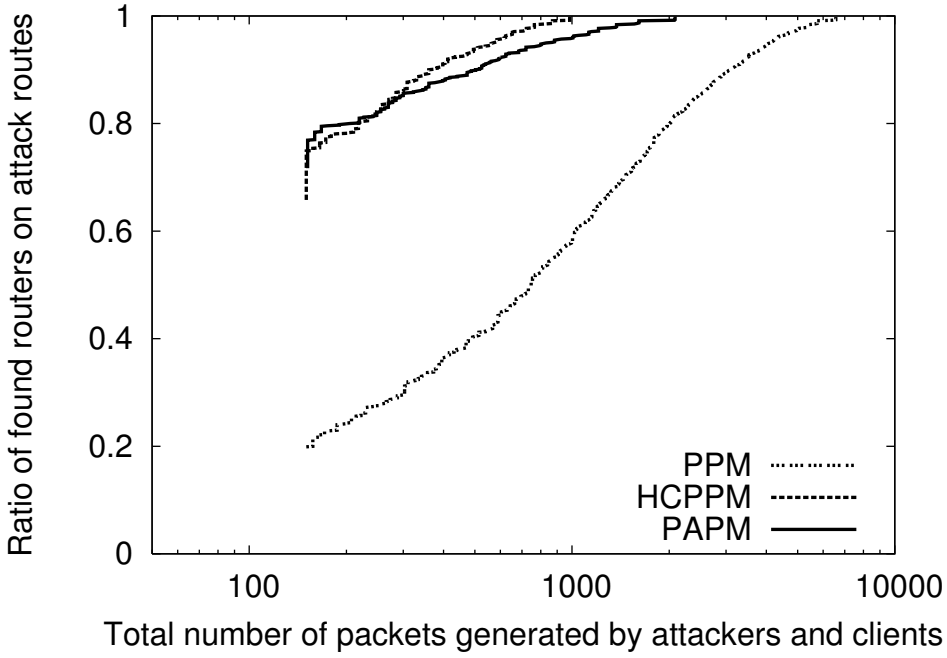
PPM はマーキング手順が簡単であるために，実装が容易である．HCPPM と PAPM は，基本的にマーキング手順が PPM と同じであり，マーキング確率を動的に計算する処

理を追加するだけでよい．

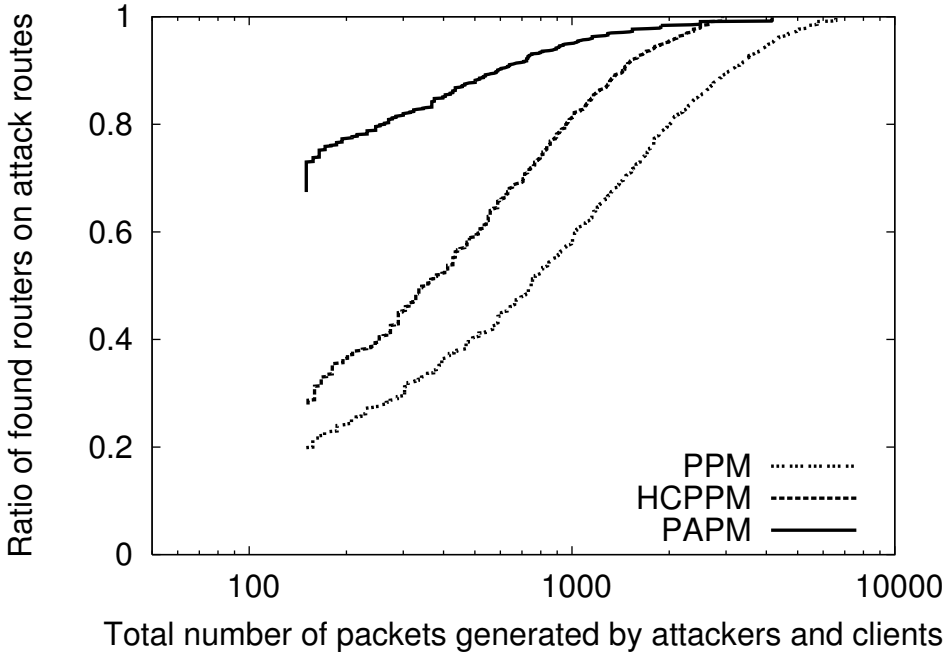
PPM のパラメータ数は 1 個であり，提案手法のパラメータ数は，5.4 節で示したように HCPPM が 4 個，PAPM が 6 個である．しかしながら PPM と HCPPM では，値の設定が必要となるパラメータが存在するが，PAPM は，値を設定する必要がなくすべて自動的に設定される．また HCPPM は 5.5.3 項で述べたように，パラメータの値によってルータの発見に必要なパケット数が大きく変化するので，パラメータチューニングが必要である．一方 PAPM は，パラメータが自動的に最適化されるため，パラメータチューニングが不要である．

攻撃経路のルータを発見するのに必要なパケット数は，マーキング偽装が行われていない場合 5.5.2 項で示したように，HCPPM がもっとも少なく PAPM，PPM の順になっている．また，HCPPM がマーキング偽装による影響を受ける一方で，PAPM はわずかながら影響を受けるが，より少ないパケット数で多くのルータを発見することが可能である．

以上より PAPM は，ルータを通過するすべてのパケットを処理する必要があるため，ルータへの負荷が高くなる．しかしながら，パラメータチューニングを行う必要がなく，マーキング偽装の影響を軽減でき，少ないパケット数でより多くのルータを発見可能である．このため，ルータへの負荷を考慮する必要がなければ，有効な手法である．また HCPPM は，パラメータチューニングが必要であり，マーキング偽装の影響を受けるが，ルータを通過する一部のパケットのみを処理するためルータへの負荷は低い．このため，コアネットワークに存在するハイエンドルータにおいてルータへの負荷の点で PAPM が実装できない場合に有効な手法である．

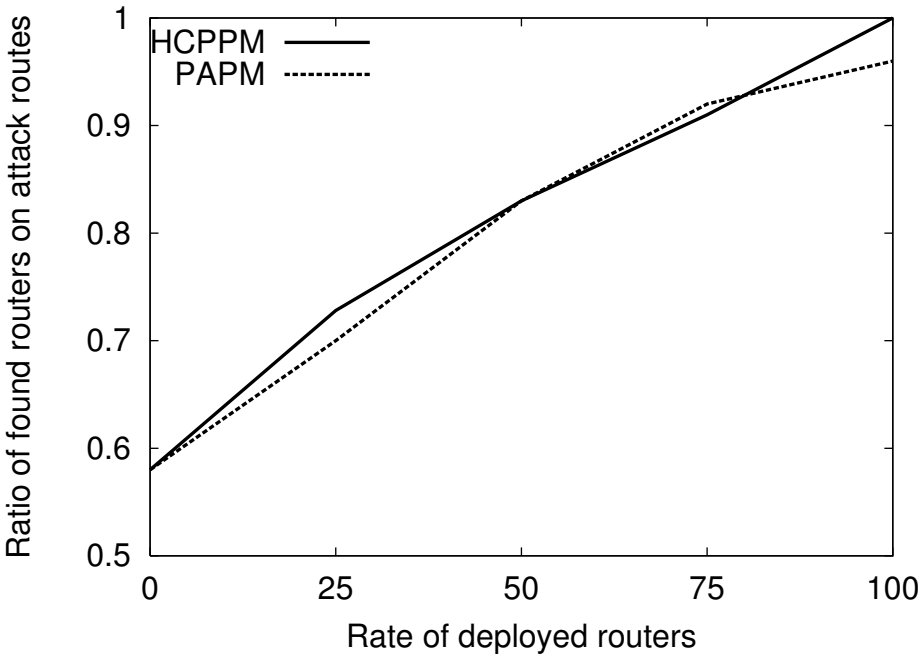


(a) マーキング偽装なし

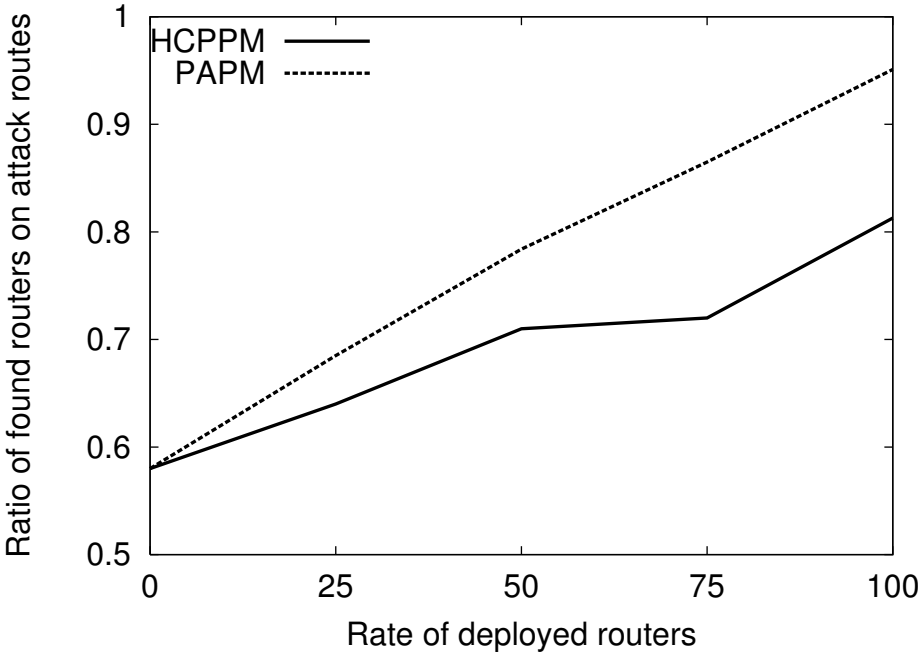


(b) マーキング偽装あり

図 5.8: 攻撃ホストと正常なクラインとホストが送信した総パケット数ごとのルータ発見率



(a) マーキング偽装なし



(b) マーキング偽装あり

図 5.9: 提案手法を導入したルータの割合によるルータ発見率

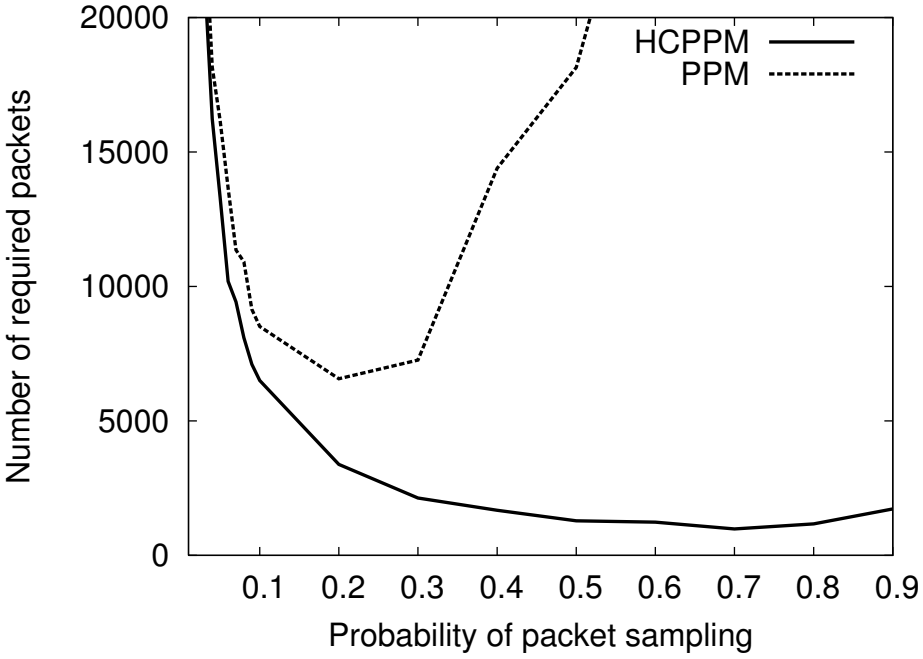


図 5.10: サンプリング確率 $q_{d,i}$ による影響 (HCPPM; $T = 150$)

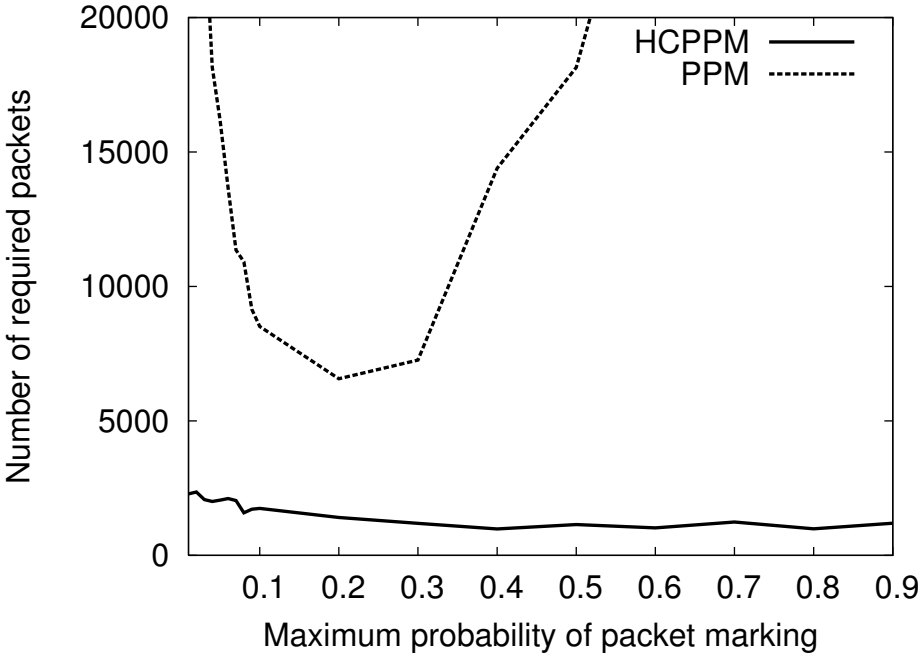


図 5.11: 最大マーキング確率 $h_{d,i}$ による影響 (HCPPM; $T = 150$)

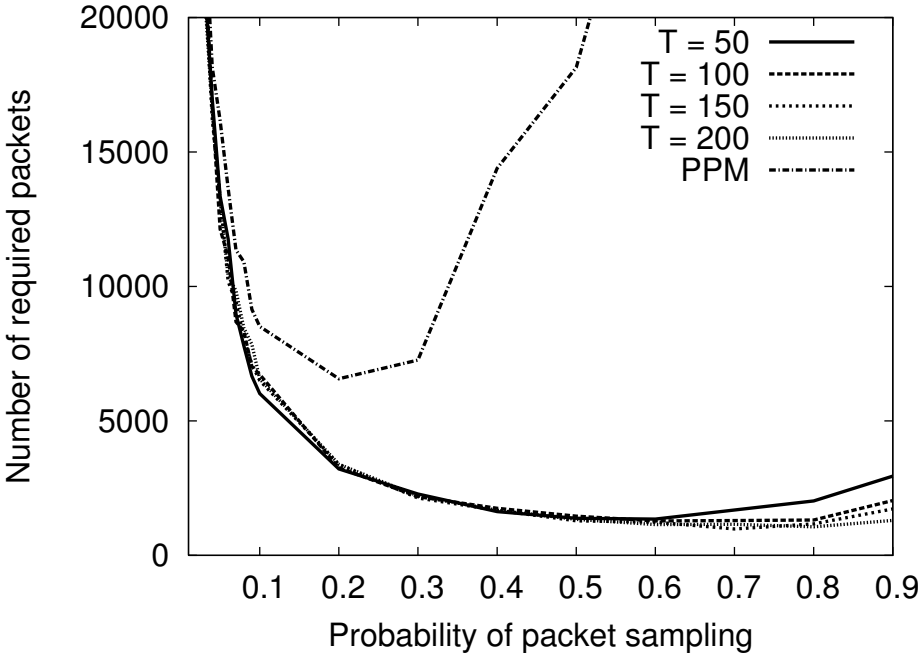


図 5.12: マーキングトリガ数 T による影響 (HCPPM; $h_{d,i} = 0.4$)

5.7 むすび

本章では，パケットマーキングにおいてマーキング情報の重複を発生させる要因の分析を行った．そして，これらの要因を軽減し，パラメータチューニングが不要で自律的に動作する手法を提案した．また，シミュレーションによりパラメータチューニングを行うことなく，攻撃経路の再構築に必要なパケット数が大幅に減少可能で，再構築に必要なパケット数を PPM に比べ最大で 85% 削減可能であることを示した．

今後は，PAPM のルータへの負荷軽減，マーキング確率の推定精度の向上，HCPPM のマーキング偽装に対する対策などを行っていく必要がある．

第6章　むすび

本論文では，近年頻繁に報告がなされ問題となっているサービス拒否 *(Denial of Service; DoS)* 攻撃やその分散型である分散型サービス拒否 *(Distributed Denial of Service; DDoS)* 攻撃による異常トラヒックの除去をはじめ，ネットワークの管理に必要となるトラヒックの分析技術について,（1）トラヒックの統計情報の集計,（2）トラヒックの到着パターンの分類と到着間隔のモデル化,（3）トラヒックの生成元の特定の 3 つの課題に対して，ルータ同士の連携を必要とせずルータが自立的に動作し，ネットワークへの早期展開を可能とするシステムを実現について検討し，以下のことを行った．

3 章では，トラヒックの統計情報の集計について，異なる間隔のパケットサンプリングで得られたフローの統計情報の差分情報に着目し，パケットサンプリングにより得られた差分情報をフィードバックさせることで，元の統計情報を推定する新たな手法を提案した．そして，トレースデータを分析することにより，提案手法が EM や MLE に比べ計算量のオーダを増加させることなく従来手法に比べ推定誤差を最大で 85% 削減可能であることを示した．

4 章では，トラヒックの到着パターンの分類と到着間隔のモデル化について，フローの到着間隔に着目し，フローの生成要因の分析を行った．その結果，生成要因によって 2 種類のフローが存在することを示した．そして，ユーザ操作ごとのフローの到着パターンを分析し，ユーザ操作とフローの到着パターンに関係性があることを示した．さらに，

フローの到着間隔の近似を行い，従来手法であるポアソン分布で近似できないことを示し，ワイブル分布で近似できることを示した．

5 章では，トラヒックの生成元の特定について，パケットマーキングにおいてマーキング情報の重複を発生させる要因の分析を行った．そして，これらの要因を軽減し，パラメータチューニングが不要で自律的に動作する方式を提案した．また，シミュレーションによりパラメータチューニングを行うことなく，攻撃経路の再構築に必要なパケット数が大幅に減少可能で，再構築に必要なパケット数を PPM に比べ最大で 85% 削減可能であることを示した．

今後は，EBM の事前推定プロセスにおいて近似精度を向上可能な近似式や差分率の近似精度を向上可能な手法，推定に利用するデータを収集する際のサンプリング間隔以下のパケット数のフローの分布の傾向を正確に把握し事前推定プロセスおよび推定プロセスでの推定誤差を軽減可能な手法，トリガーフローと従属フローを分類する方法，PAPM のルータへの負荷軽減，マーキング確率の推定精度の向上，HCPPM のマーキング偽装に対する対策を検討する必要がある．

謝辞

本論文を終えるにあたり，ご指導・ご教授いただいた大阪市立大学大学院工学研究科岡育生教授に深く感謝致します．特に本研究を進めるにあたり，終始直接ご指導いただきました阿多信吾准教授に深く感謝致します．また，技術的なアドバイスを頂いた鳥生隆教授，大阪市立大学大学院理学研究科小松孝教授に感謝致します．そして，本論文の査読とご助言をいただきました原晋介教授に感謝いたします．最後に，多くのご協力とご助言をいただきました情報ネットワーク工学研究室のみなさま，北九州工業高等専門学校電気電子工学科の教員のみなさまに心から御礼申し上げます．

参考文献

[1] T. Peng, C. Leckie, and K. Ramamohanarao, "Survey of network-based defense mechanisms countering the dos and ddos problems," *ACM Computing Surveys*, vol. 39, pp. 1–42, April 2007.

[2] CERT, "Denial-of-service developments." `http://www.cert.org/advisories/CA-2000-01.html`, February 2000.

[3] D. Dittrich, "Distributed denial of service (DDoS) attacks/tools resource page." `http://staff.washington.edu/dittrich/misc/ddos/`.

[4] D. Moore, C. Shannon, D. Brown, G. M. Voelker, and S. Savage, "Inferring Internet denial-of-service activity," *ACM Transactions on Computer Systems*, vol. 24, pp. 115–139, May 2006.

[5] Netcraft, "WikiLeaks attacked during launch of cablegate." `http://news.netcraft.com/archives/2010/11/29/wikileaks-attacked-during-launch-of-cablegate.html`.

[6] F-Secure, "Lyzapo DDoS attack on US and south korean websites." `http://www.f-secure.com/weblog/archives/00001720.html`.

[7] S. Savage, D. Wetherall, A. Karlin, and T. Anderson, "Network support for IP traceback," *IEEE / ACM Transactions on Networking*, vol. 9, pp. 226–237, June 2001.

[8] IETF PSAMP Working Group, "Packet sampling (psamp) charter." `http://www.ietf.org/html.charters/psamp-charter.html`.

[9] N. Duffield, "Sampling for passive Internet measurement: A review," *Statistical Science*, vol. 19, pp. 472–498, August 2004.

[10] N. Duffield, C. Lund, and M. Thorup, "Charging from sampled network usage," in *Proceedings of ACM SIGCOMM Internet Measurement Workshop*, pp. 245–256, November 2001.

[11] N. Duffield, C. Lund, and M. Thorup, "Flow sampling under hard resource constraints," in *Proceedings of International Conference on Measurements and Modeling of Computer Systems*, pp. 85–96, June 2004.

[12] N. Duffield, C. Lund, and M. Thorup, "Learn more, sample less: Control of volume and variance in network measurement," *IEEE Transactions on Information Theory*, vol. 51, pp. 1756–1775, May 2005.

[13] C. Guang, G. Jian, and D. Wei, "A traffic sampling model for measurement using packet identification," in *Proceedings of IEEE International Conference on Networks*, pp. 409–413, August 2002.

[14] T. Zseby, "Comparison of sampling methods for non-intrusive SLA validation," in *Proceedings of International Workshop on End-to-End Monitoring Techniques and Services*, pp. 46–53, October 2004.

[15] J. Quittek, T. Zseby, G. Carle, and S. Zander, "Traffic flow measurements within IP networks: requirements, technologies, and standardization," in *Proceedings of Applications and the Internet Workshops*, pp. 97 – 98, January 2002.

[16] H. V. Madhyastha and B. Krishnamurthy, "A generic language for application-specific flow sampling," *ACM SIGCOMM Computer Communication Review*, vol. 38, pp. 5–16, April 2008.

[17] G. He and J. C. Hou, "An in-depth, analytical study of sampling techniques for self-similar internet traffic," in *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems*, pp. 404 – 413, April 2005.

[18] B. Y. Choi, J. Park, and Z. L. Zhang, "Adaptive packet sampling for flow volume measurement," *ACM SIGCOMM Computer Communication Review*, vol. 32, p. 9, July 2002.

[19] H. Isozaki, S. Ata, and I. Oka, "Estimating flow distribution by using difference information of multiple packet samplings," in *Proceedings of 23rd International Conference on Information Networking*, pp. 1–5, January 2009.

[20] 磯崎 裕臣, 阿多 信吾, 岡 育生, "フィードバックを用いた元のフロー分布推定法の精度向上手法," **電子情報通信学会和文論文誌**, vol. J95-B, April 2012.

[21] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society, Series B*, vol. 39, pp. 1–38, May 1977.

[22] W. Wang and W. WU, "Online detection of network traffic anomalies using degree distributions," *International Journal of Communications, Network and System Sciences*, vol. 3, pp. 177–182, February 2010.

[23] A. Lakhina, M. Crovella, and C. Diot, "Mining anomalies using traffic feature distributions," in *Proceedings of ACM SIGCOMM Internet Measurement Conference*, pp. 217–228, August 2005.

[24] G. Nychis, V. Sekar, D. G. Andersen, H. Kim, and H. Zhang, "An empirical evaluation of entropy-based traffic anomaly detection," in *Proceedings of ACM SIGCOMM Internet Measurement Conference*, pp. 151–156, September 2008.

[25] H. Isozaki, S. Ata, and I. Oka, "Modeling of flows based on behavior of applications," in *Proceedings of 26th International Conference on Information Networking*, pp. 447–552, February 2012.

[26] 磯崎 裕臣, 阿多 信吾, 岡 育生, "履歴キャッシングを用いた確率的パケットマーキングの性能向上," **電子情報通信学会技術研究報告** *(NS2004-249, IN2004-249)*, vol. 104, pp. 35–40, March 2005.

[27] 磯崎 裕臣, 阿多 信吾, 岡 育生, “マーキング重複の軽減によるパケットマーキングの高速化手法,” **電子情報通信学会技術研究報告** *(IN2005-151)*, vol. 105, pp. 49–54, February 2006.

[28] 磯崎 裕臣, 阿多 信吾, 岡 育生, “マーキング数推定による確率的パケットマーキングの高速化手法,” **電子情報通信学会和文論文誌**, vol. J92-B, pp. 840–852, May 2009.

[29] N. Duffield, C. Lund, and M. Thorup, “Properties and prediction of flow statistics from sampled packet streams,” in *Proceedings of ACM SIGCOMM Internet Measurement Workshop*, pp. 159–171, November 2002.

[30] B.-Y. Choi, J. Park, and Z.-L. Zhang, “Adaptive random sampling for traffic volume measurement,” *Telecommunication Systems*, vol. 34, pp. 71–80, February 2007.

[31] Q. G. Zhao, A. Kummar, and J. J. Xu, “Data streaming algorithms for accurate and efficient measurement of traffic and flow matrices,” in *Proceedings of International Conference on Measurements and Modeling of Computer Systems*, pp. 177–188, June 2005.

[32] T. Mori, M. Uchida, R. Kawahara, J. Pan, and S. Goto, “Identifying heavy-hitter flows from sampled flow statistics,” *IEICE Transactions on Communications*, vol. E90-B, pp. 3061–3072, November 2007.

[33] N. Duffield, C. Lund, and M. Thorup, “Estimating flow distributions from sampled flow statistics,” *IEEE / ACM Transactions on Networking*, vol. 13, pp. 933–946, October 2005.

[34] L. Yang and G. Michailidis, "Sampled based estimation of network traffic flow characteristics," in *Proceedings of Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 5, pp. 1775 – 1783, May 2007.

[35] N. Hohn and D. Veitch, "Inverting sampled traffic," *IEEE / ACM Transactions on Networking*, vol. 14, pp. 68–80, February 2006.

[36] A. Kumar and J. J. Xu, "Sketch guided sampling using on-line estimates of flow size for adaptive data collection," in *Proceedings of IEEE International Conference on Computer Communications*, pp. 1–11, April 2006.

[37] D. Wang, G. Xie, J. Yang, Z. Li, and W. Jiang, "Feedback and resources guided mechanism for adaptive packet sampling," in *Proceedings of IEEE Global Communications Conference*, pp. 2586–2590, November 2007.

[38] R. Clegg, R. Landa, H. Haddadi, M. Rio, and A. Moore, "Techniques for flow inversion on sampled data," in *Proceedings of Conference of the IEEE Computer and Communications Societies Workshops*, vol. 4, pp. 1 – 6, April 2008.

[39] N. Hohn, D. Veitch, and P. Abry, "The impact of the flow arrival process in Internet traffic," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 6, pp. 37–40, April 2003.

[40] T. Karagiannis, M. Molle, M. Faloutsos, and A. Broido, "A nonstationary poisson view of Internet traffic," in *Proceedings of the 23th AnnualJoint Conference of the IEEE Computer and Communications Societies*, pp. 1558–1569, March 2004.

[41] C. Barakat, P. Thiran, G. Iannaccone, C. Diot, and P. Owezarski, "A flow-based model for internet backbone traffic," in *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*, pp. 35–47, November 2002.

[42] V. Paxson and S. Floyd, "Wide-area traffic: the failure of poisson modeling," in *Proceedings of the conference on Communications architectures, protocols and applications*, pp. 257–268, October 1994.

[43] P. Olivier and N. Benameur, "Flow level IP traffic characterization," in *Proceedings of the International Teletraffic Congress*, September 2001.

[44] N. B. Azzouna, F. Clerot, C. Fricker, and F. Guillemin, "A flow-based approach to modeling ADSL traffic on an IP backbone link," *Annals of Telecommunications*, vol. 59, pp. 1260–1299, November 2004.

[45] A. Dainotti, A. Pescape, P. S. Rossi, F. Palmieri, and G. Ventre, "Internet traffic modeling by means of hidden markov models," *Computer Networks*, vol. 52, pp. 1260–1299, October 2008.

[46] C. Barakat, P. Thiran, G. Iannaccone, C. Diot, and P. Owezarski, "A flow-based model for Internet backbone traffic," in *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*, pp. 35–47, November 2002.

[47] N. Hohn, D. Veitch, and P. Abry, "The impact of the flow arrival process in internet traffic," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pp. 37–40, April 2003.

[48] J. J. Lee and M. Gupta, "A new traffic model for current user web browsing behavior." http://blogs.intel.com/research/2007/09/13/a_new_traffic_model_for_curren/, September 2007.

[49] B. Xi, H. Chen, W. S. Cleveland, and T. Telkamp, "Statistical analysis and modeling of Internet VoIP traffic for network engineering," *Electronic Journal of Statistics*, vol. 4, pp. 58–116, 2010.

[50] J. V. P. Gomes, P. R. M. Inacio, B. Lakic, M. M. Freire, H. J. A. D. Silva, and P. P. Monteiro, "Source traffic analysis," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 6, pp. 58–116, AUgust 2010.

[51] Z. Liu, N. Niclausse, and C. Jalpa-Villanueva, "Traffic model and performance evaluation of web servers," *Performance Evaluation*, vol. 46, pp. 77–100, October 2001.

[52] C. Rolland, J. Ridoux, and B. Baynat, "Litgen, a lightweight traffic generator: Application to P2P and mail wireless traffic," in *Proceedings of the 8th international conference on Passive and active network measurement*, pp. 52–62, April 2007.

[53] K. V. Vishwanath and A. Vahdat, "Swing: Realistic and responsive network traffic generation," *IEEE/ACM Transactions on Networking*, vol. 17, pp. 712–725, June 2009.

[54] A. C. Snoeren, C. Partridge, L. A. Sanchez, C. E. Jones, F. Tchakountio, S. T. Kent, and W. Strayer, "Single-packet IP traceback," *IEEE / ACM Transactions on Networking*, vol. 10, pp. 721–734, December 2002.

[55] T.-H. Lee, W.-K. Wu, and T.-Y. W. Huang, "Scalable packet digesting schemes for IP traceback," in *Proceedings of IEEE International Conference on Communications*, June 2004.

[56] M. T. Goodrich, "Probabilistic packet marking for large-scale IP traceback," *IEEE / ACM Transactions on Networking*, vol. 16, pp. 15–24, February 2008.

[57] D. Dean, M. Franklin, and A. Stubblefield, "An algebraic approach to IP traceback," *ACM Transactions on Information and System Security*, vol. 5, pp. 119–137, May 2002.

[58] D. X. Song and A. Perrig, "Advanced and authenticated marking schemes for IP traceback," in *Proceedings of 20th Annual Joint Conference of the IEEE Computer and Communications Societies*, April 2001.

[59] A. Yaar, A. Perrig, and D. Song, "Pi: a path identification mechanism to defend against DDoS attacks," in *Proceedings of 24th IEEE Symposium on Security and Privacy*, May 2003.

[60] A. Yaar, A. Perrig, and D. Song, "Fit: fast Internet traceback," in *Proceedings of 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, March 2005.

[61] A. Belenky and N. Ansari, "Tracing multiple attackers with deterministic packet marking (DPM)," in *Proceedings of 2003 IEEE Pacific Rim Conference on Communications and Computers and Signal Processing*, August 2003.

[62] A. Belenky and N. Ansari, "IP traceback with deterministic packet marking," *IEEE Communications Letters*, vol. 7, pp. 162–164, April 2003.

[63] A. Belenky and N. Ansari, "On deterministic packet marking," *The International Journal of Computer and Telecommunications Networking*, vol. 51, pp. 2677–2700, June 2007.

[64] T. Anderson, T. Roscoe, and D. Wetherall, "Preventing Internet denial-of-service with capabilities," *ACM SIGCOMM Computer Communication Review*, vol. 34, pp. 39–44, January 2004.

[65] X. Yang, D. Wetherall, and T. Anderson, "A DoS-limiting network architecture," *ACM SIGCOMM Computer Communication Review*, vol. 35, pp. 241–252, October 2005.

[66] B. Parno, D. Wendlandt, E. Shi, A. Perrig, B. Maggs, and Y.-C. Hu, "Portcullis: protecting connection setup from denial-of-capability attacks," *ACM SIGCOMM Computer Communication Review*, vol. 37, pp. 289–300, October 2007.

[67] 磯崎 裕臣, 阿多 信吾, 岡 育生, "サンプリングフローの差分情報を用いたフロー分布の推定," **電子情報通信学会技術研究報告** *(IN2006-174)*, vol. 106, pp. 71–76, February 2007.

[68] 磯崎 裕臣, 阿多 信吾, 岡 育生, “異なる間隔を用いたパケットサンプリングにおける差分情報のモデル化,” **電子情報通信学会技術研究報告** *(TM2007-13)*, vol. 107, pp. 71–76, May 2007.

[69] S. Ata, M. Murata, and H. Miyahara, “Analysis of network traffic and its application to design of high-speed routers,” *IEICE Transactions on Information and Systems*, vol. E83-D, pp. 988–995, May 2000.

[70] NLANR Measuament and Network Analysis Group, “Nlanr pma: Special traces archive.” `http://pma.nlanr.net/Special/`.

[71] K. Cho, “WIDE-TRANSIT 150 megabit Ethernet trace.” `http://imdc.datcat.org/collection/1-05L8-9=WIDE-TRANSIT+150+Megabit+Ethernet+Trace+2008-03-18+(Anonymized)`.

[72] C. Shannon, E. Aben, K. claffy, and D. E. Andersen, “CAIDA anonymized 2008 Internet traces dataset.” `http://www.caida.org/data/passive/passive_2008_dataset.xml`.

[73] Z.-L. Zhang, V. J. Ribeiro, S. Moon, and C. Diot, “Small-time scaling behaviors of Internet backbone traffic: an empirical study,” in *Proceedings of Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 3, pp. 1826–1836, March 2003.

[74] “SINET.” `http://www.sinet.ad.jp/`.

[75] "Google maps." http://maps.google.com/.

[76] "Bing maps." http://www.bing.com/maps/.

[77] "Google docs." http://docs.google.com/.

[78] "Sky drive." http://skydrive.live.com/.

[79] "Youtube." http://www.youtube.com/.

[80] "Ustream." http://www.ustream.tv/.

[81] H. Isozaki, S. Ata, I. Oka, and C. Fujiwara, "Performance improvement on probabilistic packet marking by using history caching," in *Proceedings of the Asia-Pacific Symposium on Information and Telecommunication Technologies*, pp. 381–386, November 2005.

[82] H. Isozaki, S. Ata, and I. Oka, "Methods for improving performance on packet marking by reducing marking duplicates," in *Proceedings of 9th Asia-Pacific Network Operations and Management Symposium*, pp. 609–616, September 2006.

[83] A. Medina, A. Lakhina, I. Matta, and J. Byers, "Brite: an approach to universal topology generation," in *Proceedings of 9th International Workshop on Modeling, Analysis and Simulation of Computer and Telecommunications Systems*, August 2001.

発表文献

論文誌

1. 磯崎裕臣, 阿多信吾, 岡育生,“フィードバックを用いた元のフロー分布推定法の精度向上手法,”電子情報通信学会和文論文誌, vol. J95-B, (掲載予定), April 2012.

2. 磯崎裕臣, 阿多信吾, 岡育生,“マーキング数推定による確率的パケットマーキングの高速化手法,”電子情報通信学会和文論文誌, vol. J92-B, pp. 840-852, May 2009.

国際会議

3. H. Isozaki, S. Ata, and I. Oka,“Modeling of Flows based on Behavior of Applications,”in Proceedings of 26th International Conference on Information Networking, pp. 447-452, February 2012.

4. H. Isozaki, S. Ata, and I. Oka,“Estimating flow distribution by using difference information of multiple packet samplings,”in Proceedings of 23rd International Conference on Information Networking, pp. 1-5, January 2009.

5. H. Isozaki, S. Ata, and I. Oka,“Methods for improving performance on packet marking by reducing marking duplicates,”in Proceedings of 9th Asia-Pacific Network Operations and Management Symposium, pp. 609-616, September 2006.

6. H. Isozaki, S. Ata, I. Oka, and C. Fujiwara,“Performance improvement on probabilistic packet marking by using history caching,”in Proceedings of the Asia-Pacific Symposium on Information and Telecommunication Technologies, pp. 381-386, November 2005.

研究会

7. 磯崎裕臣, 阿多信吾, 岡育生,“異なる間隔を用いたパケットサンプリングにおける差分情報のモデル化,”電子情報通信学会技術研究報告 (TM2007-13), vol. 107, pp. 71-76, May 2007.

8. 磯崎裕臣, 阿多信吾, 岡育生,“サンプリングフローの差分情報を用いたフロー分布の推定,”電子情報通信学会技術研究報告 (IN2006-174), vol. 106, pp. 71-76, February 2007.

9. 磯崎裕臣, 阿多信吾, 岡育生,“マーキング重複の軽減によるパケットマーキングの高速化手法,”電子情報通信学会技術研究報告 (IN2005-151), vol. 105, pp. 49-54, February 2006.

10. 磯崎裕臣, 阿多信吾, 岡育生,“履歴キャッシングを用いた確率的パケットマーキングの性能向上,”電子情報通信学会技術研究報告 (NS2004-249, IN2004-249), vol. 104, pp. 35-40, March 2005.

その他

11. 磯崎裕臣, 阿多信吾, 岡育生, ”異なるパケットサンプリングの差分情報を用いたフロー分布の推定,” 待ち行列シンポジウム「確率モデルとその応用」報文集, pp. 139-148, January 2008.

12. 磯崎裕臣, 阿多信吾, 岡育生, ”差分情報のモデル化に基づいたフロー統計の推定の精度向上手法,” 電子情報通信学会 2007 年ソサイエティ大会講演論文集 (B-14-14), p. 343, September 2007.

13. 磯崎裕臣, 阿多信吾, 岡育生,“異なる周期のサンプリングを用いたオリジナルフローの推定,”電子情報通信学会総合大会 (B-7-19), March 2007.