博士学位論文

パケットサンプリングを用いた
異常トラヒックのオンライン検出に関する研究

工藤 隆則

2014年1月

大阪大学大学院工学研究科

内容梗概

本論文は，筆者が大阪大学大学院工学研究科電気電子情報工学専攻在学中に行ったパケットサンプリングを用いた異常トラヒックのオンライン検出に関する研究成果をまとめたものである．

インターネット上を流れるトラヒックには，多くのホストにとって，あるいはネットワークの管理者にとって不都合な異常トラヒックが存在する．DoS 攻撃のトラヒックは標的となったホストのサービスを停止に追い込むだけでなく，そのトラヒックと回線を共有する他の正常なトラヒックにも影響を与える．このときホストは攻撃を受けていることに気づいたとしても，攻撃元を突き止めたり，正常なトラヒックのために上流で怪しいパケットを意図的に廃棄したりといった積極的な対応は難しく，攻撃が止むまでホストを切り離してネットワークの管理者に報告することが現実的な対応となる．ワームやボットの感染拡大にも用いられるポートスキャンは，感染可能かどうかを確認する探査パケットを多数のホストとポート番号の組み合わせに対して送るが，ポートスキャンを行っているホストの管理者自身が感染していることに気づいていないケースが多い．そのため発見が遅れ，気づいた頃にはすでに広範囲に感染が拡大しているということが起こり得る．ワームやボットなどのマルウェアに対してホスト側で行えることは，感染しないようにセキュリティを高めておき，OS やソフトウェアに脆弱性が発見された時にはすぐに対策を取るなどの自衛手段が挙げられるが，未知のマルウェアに対しては対処が難しく，例えば感染してセキュリティソフトを含めて他者の制御下に置かれてしまえば，感染していることに気づくことは難しい．

ネットワーク攻撃や感染拡大活動に対して，セキュリティ意識を高く持つことでホスト側でも防衛策を取ることは可能だが，一旦異常トラヒックが発生している状況にまで至ると，ホスト側で対処できることには限界がある．そこで，ネットワークの内部において異常トラヒックを検出することを考える．その場合，異常トラヒックの検出後はそれに属するパケットを転送せずに廃棄したり，あるいはフォワーディング元を順次辿ることで発生源の ISP やホストを特定するといったこともエンドホストで行うことと比較すると断然行いやすい．

本論文では，DoS 攻撃のトラヒックに代表されるような非常にパケットレートが高いフロー，パケットレートが高い状態が一定時間以上持続するようなフロー，ポートスキャンで使用される一つのホストから多数の宛先に送られる探査用のフロー群，以上三つを異常トラヒックの候補としてネットワーク内部で検出することを考える．ネットワーク内部では回線速度が高速なため，処理能力やメモリ領域の観点からスケーラビリティを確保する必要がある．本論文では一貫してラン

ダムパケットサンプリングを用いたデータ取得を行う．またオンラインの検出を行うために，一定時間ごとにデータを更新するウィンドウ型のアルゴリズムを使用する．

本論文は，以下に示す 5 章により構成する．

第 1 章では，まずインターネットの異常トラヒックを中心とした本論文の研究背景について述べる．さらにネットワーク内部でトラヒックを計測する際に必要となる主要技術として技術として，パケットサンプリング，フロー集約，スライディングウィンドウ方式によるオンラインデータ処理の三つの技術を既存の技術の紹介も含めて述べる．

第 2 章では，高パケットレートフローのオンライン検出におけるパラメータ決定手法について述べる．まずパケットサンプリングによるトラヒックデータの取得とスライディングウィンドウ方式によるデータ更新を採用した高パケットレートフローのオンライン検出手法について述べ，誤検出確率や未検出確率，オンライン検出の実行可能性などを考慮したパラメータ決定手法を提案する．そして実トレースデータを用いた数値実験により評価を行う．

第 3 章では，一定時間以上高いパケットレートが続くような，持続的高パケットレートフローの検出手法について提案する．本検出手法もパケットサンプリングとスライディングウィンドウ方式を用いている．その後，第 2 章で提案したパラメータ決定手法を踏襲した，本検出手法に対するパラメータ決定手法を提案し，実トレースデータを用いた提案手法の評価を行う．

第 4 章では，ポートスキャントラヒックのオンライン検出について述べる．まず TCP ポートスキャンの検出手法について述べたのち，誤検出確率と未検出確率の両方を考慮したパラメータ決定手法を提案する．ここでも実トレースデータを用いた数値実験を行い提案手法の評価を行う．

最後に第 5 章において本論文の結論を述べる．

謝辞

本論文は筆者が大阪大学大学院工学研究科の博士後期課程において研究した成果をまとめたものであり，研究過程においてお世話になった方々にここで御礼申し上げます．

本論文の主査として，また本論文に関する研究の全過程を通じ，懇切丁寧なる御指導，御鞭撻を賜った 大阪大学大学院工学研究科電気電子情報工学専攻 滝根 哲哉 教授に深甚なる感謝の意を表し，心より厚く御礼申し上げます．研究面のみならず，生活面でも常に暖かく見守っていただき，また，ときに叱咤激励していただくことによって精神的にも大きく支えていただきました．

本論文の執筆にあたり，副査として査読をしていただいた 大阪大学工学研究科電気電子情報工学専攻 馬場口 登 教授に心より御礼申し上げます．筆者の文字通りの拙稿に対して，的確なご指摘，丁寧な御助言，そして暖かい励ましのお言葉をいただきました．

本論文の執筆にあたり，同じく副査として査読していただいた 大阪大学工学研究科電気電子情報工学専攻 松田 崇弘 准教授に厚く御礼申し上げます．松田 准教授には本論文に関わる研究の過程においても数多くの御助言をいただきました．

大阪大学大学院工学研究科において，御指導，御教授を賜った 大阪大学大学院工学研究科電気電子情報工学専攻 北山 研一 教授，三瓶 政一 教授，井上 恭 教授，鷲尾 隆 教授を始めとする各教官の方々には，様々な御助言，御提案，御指摘を頂戴いたしました．衷心より御礼申し上げます．

本研究を遂行するにあたり，熱心な御指導，的確な御助言を頂いたロバストネットワーク工学領域研究室の 笹部 昌弘 助教に深甚なる感謝の意を表します．また，同研究室において様々な御支援を頂いた 下屋敷 優美 事務補佐員，ならびに 橋本 幸子 前事務補佐員，そして公私に渡って暖かい御助言，御支援を賜った 後藤 嘉代子 元技官に心より御礼申し上げます．

卒業生を含め，ロバストネットワーク工学領域の諸氏には，日頃より多くの御助言，御協力を頂き，種々の面でお世話になりました．ここに深謝申し上げます．

ここに記して，以上の方々に深甚なる感謝の意を捧げます．

目次

内容梗概 iii

謝辞 v

第 1 章　序論 1
1.1 研究背景と目的 1
1.1.1 DoS 攻撃 2
1.1.2 DDoS 攻撃 3
1.1.3 マルウェアとポートスキャン 3
1.1.4 目的 4
1.2 トラヒックのオンライン計測技術 4
1.2.1 パケットサンプリング 6
1.2.2 フロー集約 7
1.2.3 スライディングウィンドウ方式 8
1.3 構成 8

第 2 章　高パケットレートフローの検出におけるパラメータ決定手法 11
2.1 まえがき 11
2.2 高パケットレートフローのオンライン検出手法 11
2.3 パラメータ決定手法 13
2.3.1 制約条件 13
2.3.2 目的関数 14
2.3.3 大域的最適解 15
2.4 数値実験 17
2.4.1 実験準備 18
2.4.2 性能評価指標 18
2.4.3 実験結果 19
2.4.4 最適パラメータの有効性 23
2.5 まとめ 27

第 3 章　持続的高パケットレートフロー検出手法とそのパラメータ決定手法 29
3.1 まえがき 29
3.2 持続的高パケットレートフローの検出手法 29

3.2.1 持続的高パケットレートフローとスライディングウィンドウ方式 29
3.3 ランダムパケットサンプリングを用いた検出手法 31
3.4 閾値の設定方法 32
3.5 パラメータ決定手法 35
3.6 数値実験 37
3.6.1 トレースデータと評価指標 38
3.6.2 実験結果 39
3.7 まとめ 42

第 4 章 TCP ポートスキャンの検出におけるパラメータ決定手法 45
4.1 まえがき 45
4.2 TCP ポートスキャン検出手法 45
4.2.1 TCP ポートスキャンのオンライン検出手法の概要 46
4.2.2 検出手法の実行可能性 47
4.2.3 ホストごとの SYN-only フロー数の計測方法 49
4.3 パラメータ決定手法 50
4.3.1 ランダムパケットサンプリングにおける検出確率 50
4.3.2 パラメータ決定 52
4.4 数値実験 53
4.4.1 実験準備 53
4.4.2 実験結果 54
4.5 まとめ 58

第 5 章 結論 61

付 録 A 定理 1 の証明 65

付 録 B 定理 2 の証明 67

付 録 C 検出確率の上界と下界 71

付 録 D 近似的閾値フロー特定のためのヒューリスティック法 73

付 録 E 問題 P^* の最適解 77

付 録 F 外部からの TCP ポートスキャン検出手順 79

参考文献 81

研究業績 85

図 目 次

1.1 トラヒックの計測地点 5

2.1 f の関数とみなした y^* ($R = 4,000$, $\epsilon = 0.01$) 12
2.2 f の関数とみなした誤検出確率 ($R = 4,000$, $r = 2,000$, $\epsilon = 0.01$) . . 15
2.3 fT_{SW} の関数とみなした誤検出確率 ($R = 4,000$, $r = 2,000$, $\epsilon = 0.01$) 16
2.4 フローの最高パケットレートに基づくランキング ($T_{\mathrm{D\_max}} = 10$) . . 20
2.5 フローの最高パケットレートに基づくランキング ($T_{\mathrm{D\_max}} = 20$) . . 20
2.6 典型的な検出対象フローのパケットレートの変化 ($k = 61$, $R = 1,000$) 22
2.7 誤検出されたフローのパケットレートの累積分布 ($T_{\mathrm{D\_max}} = 10$) . . 26
2.8 誤検出されたフローのパケットレートの累積分布 ($T_{\mathrm{D\_max}} = 20$) . . 26
2.9 検出確率の平均と 95%信頼区間 ($f = f^*/c$, $T_{\mathrm{D\_max}} = 20$, $\epsilon = 0.05$) . 27

3.1 BW, SW, HW の例 30
3.2 持続的高パケットレートフローの検出手順 31
3.3 HW, SW, BW のパケット数の関係 32
3.4 サンプルパケット数の閾値 θ^* の設定手順 36
3.5 一様分布フローと検出確率の関係 (バックボーン) 40
3.6 一様分布フローと検出確率の関係 (バックスキャッタ) 41
3.7 検出対象と検出結果 (バックボーン). 43
3.8 検出対象と検出結果 (バックスキャッタ). 44
3.9 バックボーントレースのフロー 7 のパケットレート 44

4.1 TCP コネクション成立 46
4.2 TCP コネクション不成立 46
4.3 正常なホストとポートスキャン実行ホストの SYN-only フロー数 . . 48
4.4 SYN-only フロー数の確率関数 49
4.5 ホストごとの SYN-only フロー数計測手順 50
4.6 正常なホストとポートスキャン実行ホストの sampled-SYN-only フロー数 51
4.7 $f = 1$ としたときの FPR と閾値の関係 55
4.8 $f = 1$ としたときの FNR と閾値の関係 56
4.9 閾値 $\hat{\theta}_{\mathrm{FPR}}$ および $\hat{\theta}_{\mathrm{FNR}}$ とサンプリングレートの関係 57
4.10 交点付近における閾値 $\hat{\theta}_{\mathrm{FPR}}(T_{\mathrm{JW}} = 5)$ および $\hat{\theta}_{\mathrm{FNR}}(N_{\mathrm{S}} = 1000)$ とサンプリングレートの関係 58

4.11 $f = \hat{f}_{\mathrm{opt}}$ としたときの FPR と閾値の関係 **59**
4.12 $f = \hat{f}_{\mathrm{opt}}$ としたときの FNR と閾値の関係 **60**

D.1 ランダムに生成した MTF の検出確率 ($s = 2$, $m = 3$, $z^* = 30,000$, $f = 9.8 \times 10^{-4}$, 試行回数 $= 10^6$, 生成パターン数 $= 10^5$, $h = 4, 5, 6, 7, 8, 9$) **76**

F.1 ホストごとの RST-only フロー数計測手順 **80**

表 目 次

1.1 ネットワーク攻撃に対する立場ごとの検出しやすさ/検出後の対応のしやすさ (◎:しやすい, ○:比較的しやすい, △:しにくい) **2**

2.1 制御パラメータと閾値 ($T_{\mathrm{D\_max}} = 10$) **18**
2.2 制御パラメータと閾値 ($T_{\mathrm{D\_max}} = 20$) **19**
2.3 検出対象フローの検出数と検出率 ($T_{\mathrm{D\_max}} = 10$) **21**
2.4 検出対象フローの検出数と検出率 ($T_{\mathrm{D\_max}} = 20$) **22**
2.5 誤検出されたフロー数と FPR ($T_{\mathrm{D\_max}} = 10$) **24**
2.6 誤検出されたフロー数と FPR ($T_{\mathrm{D\_max}} = 20$) **25**
2.7 サンプルパケット数の閾値 y^* と誤検出フロー数 N_{WD} の関係 **25**

3.1 トレースデータの情報 **38**
3.2 実験に用いた入力パラメータ **39**
3.3 制御パラメータ f^* および m^* **39**
3.4 多段閾値の値 **40**
3.5 検出されたフローの分類範囲および誤検出確率の上限 **41**
3.6 各指標の 100 回の平均と 95%信頼区間 **42**

4.1 N_{S} が取り得るの最小値と $f = 1$ のときの FPR と FNR **54**
4.2 パケットレートの最適解 $\hat{f}_{\mathrm{opt}}$ と閾値の最適解 $\hat{\theta}_{\mathrm{opt}}$ **59**
4.3 誤検出確率と未検出確率 **59**

D.1 検出確率 ($s = 1$, $h = 3$, $m = 3$, $f = 0.5$, $z^* = 9$, $\theta^* = 2$) **74**
D.2 閾値フローの検出確率 ($s = 1$, $h = 3$, $m = 3$, $f = 0.5$, $\theta^* = 2$) **74**
D.3 閾値フローの検出確率 ($s = 7$, $m = 1$, $f = 0.5$, $z^* = 7$, $\theta^* = 2$) . . . **75**

略語一覧

DoS	サービス拒否 (Denial of Service)
DDoS	分散型サービス拒否 (Distributed Denial of Service)
LAN	ローカルエリアネットワーク (Local Area Network)
NIC	ネットワークインターフェースカード (Network Interface Card)
SW	スライディングウィンドウ (Sliding Window)
BW	ベーシックウィンドウ (Basic Window)
HW	ヒストリウィンドウ (History Window)
JW	ジャンピングウィンドウ (History Window)
FP	誤検出 (False Positive)
FPR	誤検出確率 (False Positive Ratio)
FN	未検出 (False Negative)
FNR	未検出確率 (False Negative Ratio)
MTF	最少パケット数検出対象フロー (Minimum Target Flow)
AS	自律システム (Autonomous System)

変数一覧

登場が 1 度，ないしは定義された箇所の近辺でしか登場しない変数は省略した．

T_{SW}	スライディングウィンドウ (SW) の時間長
T_{BW}	ベーシックウィンドウ (BW) の時間長
T_{HW}	ヒストリウィンドウ (HW) の時間長
T_{JW}	ジャンピングウィンドウ (JW) の時間長
f	パケットサンプリングにおけるサンプリングレート
f^*	提案手法によって決定された f の値
f^{**}	提案手法によって f^* に調整を加え決定された f の最適値
k	HW を用いない SW 方式での 1SW 内の BW 数
k^*	提案手法によって決定された k の最適値
h	HW を用いた SW 方式でのウィンドウサイズ規定パラメータの一つ
m	HW を用いた SW 方式でのウィンドウサイズ規定パラメータの一つ
s	HW を用いた SW 方式でのウィンドウサイズ規定パラメータの一つ
R	検出対象フローを定義するパケットレートの閾値
x^*	SW における母集団でのパケット数の閾値 (第 2 章)
y^*	SW におけるサンプルパケット数の閾値 (第 2 章)
ϵ	最大許容未検出確率
$T_{\mathrm{D\_max}}$	最大許容検出遅延
τ	SW 内のデータ (HW があるときは HW 内のデータも) の処理時間
$C_{\max}$	計測している回線全体のパケットレートの最大値
$P_{\mathrm{WD}}(r)$	パケットレートが r のフローの誤検出確率
$G(\cdot)$	SW 内のデータの処理時間 τ を見積もる関数
z^*	SW における母集団でのパケット数の閾値 (第 3 章)
H_S	一つの HW 内に含まれる SW の数
θ^*	SW におけるサンプルパケット数の閾値 (第 3 章)
θ_i	検出されたフローをクラス分けするための多段の閾値 (第 3 章)

α	誤検出確率 FPR の最大許容値 (第 4 章)
β	未検出確率 FNR の最大許容値 (第 4 章)
S_{N}	正常なホストが生成する SYN-only フロー数
S_{A}	ポートスキャン実行ホストが生成する SYN-only フロー数
S_{B}	ポートスキャンによって生成される SYN-only フロー数
N_{S}	検出確率を保証する，S_{B} の下限値
F_{SYN}	SYN パケット用のブルームフィルタ
F_{Others}	SYN パケット以外のパケット用のブルームフィルタ
$\hat{S}_{\mathrm{N}}$	正常なホストからの sampled-SYN-only フロー数
$\hat{S}_{\mathrm{A}}$	ポートスキャン実行ホストの sampled-SYN-only フロー数
$\hat{S}_{\mathrm{B}}$	ポートスキャンによる sampled-SYN-only フロー数
$\hat{N}_{\mathrm{S}}$	サンプリング後の N_{S} からの sampled-SYN-only フロー数
$\hat{\theta}_{\mathrm{FPR}}$	FPR の制約を満たす最大の sampled-SYN-only フロー数
$\hat{\theta}_{\mathrm{FNR}}$	FNR の制約を満たす最小の sampled-SYN-only フロー数
$\hat{\theta}_{\mathrm{opt}}$	提案手法で決定された sampled-SYN-only フロー数の検出閾値
$\hat{f}_{\mathrm{opt}}$	提案手法で決定されたサンプリングレート

第1章

序論

1.1 研究背景と目的

近年，サイバー攻撃 (サイバーテロ) という言葉が一般的になってきている．サイバー攻撃はネットワークを介して行われる攻撃であり，攻撃を受けたホストが機能低下あるいは機能停止に追い込まれたり，ホストが他者に侵入されてデータを改竄されたりと，大きな被害をもたらすことがあるため非常に脅威となっている [3]．サイバー攻撃は多くの場合，国などの公的機関や企業，大学などのいわゆるサーバに対して行われることが多い．一方，一般のユーザにとっての脅威としては，ウィルスなどの不正なプログラムにホストが感染してしまうことや，偽物のウェブサイトやメールを偽物と気づかずに個人情報を流出してしまうフィッシング詐欺などが挙げられる．

こういった脅威に対して，ホストのユーザはセキュリティソフトを利用するなどして，既知のウィルスなどに感染していないかどうかチェックしたり，あるいはファイアウォールを設けて危険性やその疑いがある通信を制限したりする．ホストの OS やインストールされているソフトウェアのバージョンを常に最新のものに更新し，プログラムに含まれる脆弱性を放置しないことも，外部からの侵入などを防ぐ上で重要である．しかしながら，どれだけセキュリティに対して高い意識を持ち，万全の準備をしていても，ホストでは対処しきれない脅威が存在する．例えば，未知のウィルスに感染してしまった場合，セキュリティソフトでは感染を見つけることは難しい．また，OS やソフトウェアに脆弱性が見つかった場合に，修正パッチが配布されるまでの間にその脆弱性を突いてホストを機能停止に追い込んだり，侵入したりするゼロデイ攻撃と呼ばれる攻撃も対応が難しい．さらに，最初に挙げたサイバー攻撃の中でも，ホストが行っているサービスを妨害する，通称 DoS (Denial of Service) 攻撃と呼ばれる攻撃を受けた場合，ホストでは攻撃を受けていることを検出することはできても，攻撃をやめさせるといった根本的な解決は難しい．

表 1.1 にいくつかのネットワーク攻撃に対して，攻撃側のネットワークの管理者，攻撃側と標的側のネットワークを結ぶバックボーンネットワークの管理者，標的側のネットワーク管理者，標的ホストのそれぞれの立場での検出しやすさと検

表 1.1: ネットワーク攻撃に対する立場ごとの検出しやすさ/検出後の対応のしやすさ (◎:しやすい, ○:比較的しやすい, △:しにくい)

	攻撃側 NW	中継 NW	標的側 NW	標的ホスト
DoS 攻撃	◎/◎	◎/○	◎/△	◎/△
DDoS 攻撃	△/○	◎/○	◎/△	◎/△
ポートスキャン	○/◎	△/○	○/△	◎/△
ウィルスメール	△/◎	△/○	○/○	◎/○
フィッシング	△/◎	△/○	△/○	◎/○

出した後の対応のしやすさを主観によりまとめる．検出のしやすさは，攻撃の特徴をとらえやすい箇所ほど検出しやすいと考えている．検出後の対応のしやすさは，攻撃元の特定のしやすさと被害を出さずにすむかどうかで判断している．基本的にネットワークの管理者はパケットのペイロードは見れない (見ない) ため，ウィルスメールやフィッシングなどを検出することは難しく，ホスト側で対処するしかない．一方で，上の三つの攻撃は，ホスト側では検出したとしてもその後の対処がしにくく，ネットワークの管理者の立場で検出することにメリットがある．

本論文ではホストのユーザではなく，ネットワークの管理者の立場においてルータなどでトラヒックを計測することで，ホストでは対処が困難なネットワーク上の脅威を検出することを考える．以下では，まず DoS 攻撃，分散型の DoS 攻撃 (DDoS 攻撃)，マルウェアとポートスキャンについてそれぞれ解説する．その後，本論文の目的を述べる．

1.1.1 DoS 攻撃

サービス拒否攻撃，通称 DoS (Denial of Service) 攻撃は攻撃元ホストから何らかのパケットを標的となるホストへ送ることで，標的ホストが通常行っているサービスを妨害するネットワークを介した攻撃である．SYN フラッド攻撃は代表的な DoS 攻撃であり，TCP 接続の確立要求である SYN パケットを自分自身のアドレスを改竄して標的ホストに向けて大量に送りつける [25]．受け取ったホストは通信用のソケットを用意して SYN/ACK パケットを送り返すが，宛先が改竄されているためいつまで経っても次のパケットは返ってこず，用意したソケットをタイムアウトになるまで保持しておかなければならない．このとき，同時に保持できるソケットの数には上限があるため，SYN フラッド攻撃によって同時使用可能なソケットを全て専有されてしまい，正常なユーザへのサービスが行えなくなってしまう．

Web サーバに対する攻撃の手段として，ブラウザで閲覧中のページの更新を頻繁に繰り返すことで負荷を掛ける方法がある．更新のためのショートカットがキーボードの F5 キーに割り当てられているため F5 攻撃と呼ばれる．人の手による連打やツールを使って行われるこの攻撃は，一人で行おうとしてもさほど威力はな

いが，次小節で説明するように分散型で行われる場合には非常に脅威となる．

この他にも回線の帯域を浪費するような攻撃がある．例えば UDP のパケットは TCP のような複雑な制御をしなくてよいため，容易に宛先まで大量のパケットを送りつけることができる．また DNS サーバに対して送信元アドレスを標的のアドレスに偽ったパケットで問い合わせを行うことで，DNS サーバからの返答パケットを標的に向かわせる攻撃がある．これは DNS リフレクタ攻撃と呼ばれ，DNS のパケットは問い合わせよりも返答パケットの方がパケットサイズが大きいため，攻撃元からのトラヒックが増幅される形で対象へと届くことになる．

1.1.2 DDoS 攻撃

複数の攻撃元ホストから同時に行われる DoS 攻撃は DDoS (Distributed DoS) 攻撃と呼ばれ，近年大きな問題となっている [19]．DDoS 攻撃は攻撃元ホストが意図的に参加しているか否かで分類することができる．意図的な場合，攻撃者が示し合わせ一斉に標的ホストに攻撃をしかけることになり，F5 攻撃を使った例がある．攻撃元ホストが意図せず参加する例として，そのホストがウィルスやワーム，ボットなどのマルウェアに感染し，悪意のあるホストの制御下に置かれていることが挙げられる．マルウェアについては次小節で述べる．感染時に仕掛けられたタイマーであったり，攻撃を指示するホストからの合図などによって，複数の感染したホストから攻撃対象ホストに対して一斉に攻撃を仕掛けるため，トラヒックの量や攻撃元の分散性などから攻撃されているホストでの対応が非常に困難となる．

1.1.3 マルウェアとポートスキャン

悪意のあるソフトウェアは総称してマルウェアと呼ばれる [16]．マルウェアはホストに侵入，感染することで，ファイルの削除や個人情報の漏洩，あるいは無自覚のうちに犯罪に加担させられるなどの様々な被害をホストに与える可能性がある．マルウェアの例としてはウィルスやワーム，ボットなどが挙げられる．ウィルスがホストに感染するためにはユーザの何らかの行動を必要とするのに対し，ワームはそれ自身がネットワークなどを通じてホストに入り込み，感染する．感染することによって，そのデバイスを遠隔操作可能にするウィルスやワームはボットとして区別され，この遠隔操作可能なホストで構成されるネットワークはボットネットと呼ばる．ボットネットは前小節の DDoS 攻撃にも利用される．

ワームやボットに感染したホストは感染の拡大や情報の共有などのために，感染可能な未感染ホストや，すでに感染しているホストを探そうとする．このときにしばしば用いられるのがポートスキャンである [17]．Code Red [9] と Conficker [24] はそれぞれポートスキャンを使用するワームおよびボットとしてよく知られている．ポートスキャンはポートが開いているホストを探す行為である．具体的には，宛先 IP アドレス，宛先ポート番号，そして TCP や UDP といったプロトコルの三つを

指定した，相手の状態に応じて反応が変わるようなパケットを送りつける．例えばTCP の SYN パケットを送ると，相手がそのポートを開いていれば SYN/ACK パケットを返すが，閉じていれば RST/ACK パケットを返してくる．宛先 IP アドレスを固定し，宛先ポート番号を変化させたスキャンは垂直スキャン，その逆で特定のポート番号に対して開いている IP アドレスは探すスキャンを水平スキャンと呼ばれる．

1.1.4 目的

本論文では，DoS 攻撃や DDoS 攻撃のトラヒックに代表されるような非常にパケットレートが高いフロー，パケットレートが高い状態が一定時間以上持続するようなフロー，ポートスキャンで使用される一つのホストから多数の宛先に送られる探査用のフロー群，以上三つを異常トラヒックとして定義し，ネットワーク内部で検出することを目的とする．

フローとは，指定する指標あるいはその組み合わせに，同じ値を有するパケットの集合として定義される．インターネットトラヒックの計測においては一般に，送信元 IP アドレス，宛先 IP アドレス，送信元ポート番号，宛先ポート番号，プロトコルの五つ組がしばしば用いられる．ここで用いられるプロトコルとは，IP ヘッダで指定される上位層のプロトコルのことで，トランスポート層の TCP や UDP が代表的である．以降本論文では五つ組と言った場合はこの 5 種類を指すものとする．

ネットワーク管理者が異常トラヒックをオンラインで検出できれば，当該トラヒックのパケットを意図的に破棄したり，発生源を突き止めるなどの積極的な対応が可能となる．次節ではオンライン検出のためのトラヒックの計測技術について述べる．

1.2 トラヒックのオンライン計測技術

文献 [10] にあるように，インターネットトラヒックの計測には様々な物理的要素や技術的要素が存在する．ここでは異常トラヒックのオンライン検出に必要となる，オンライン計測に関わる技術的要素を紹介していく．図 1.1 に示すように，ネットワークのトラヒックを計測する地点として，ホストのネットワークインターフェース，LAN の境界ルータ，バックボーンネットワーク内部のルータなどが挙げられる．ホストにおいて，ネットワークインターフェースを通過するトラヒックは tcpdump [26] や wireshark [30] などを利用するとパケットレベルで容易に計測が可能である．このとき，フィルタリングを行うことで，特定のプロトコルや IP アドレス，ポート番号などを指定して計測することも可能となる．

一方，ネットワークの内部でトラヒックを計測する場合は，ルータに計測のための機能を持たせることになる．最も簡単な計測手法はポートミラーリングである．計測したトラヒックを出力 (保存) するためのポートをルータに用意し，そこ

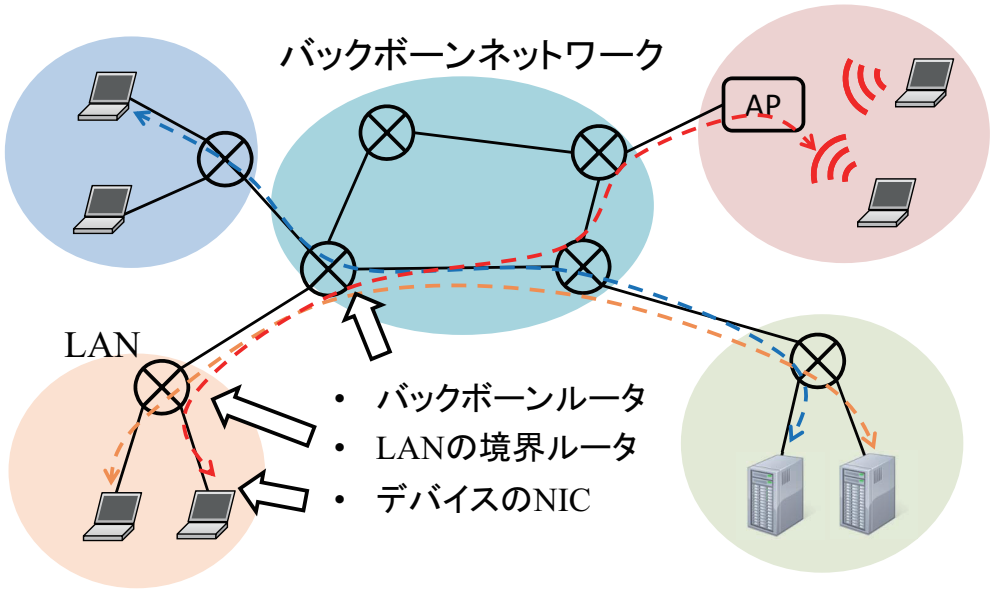

図 1.1: トラヒックの計測地点

に計測用の PC を接続しておく．計測したいトラヒックのパケットがルータを通過するとき，ルータはそのパケットを通常の出力ポートだけでなく，そのコピーを計測用のポートにも出力する．このようにすることで特定のトラヒックのパケットが計測可能となる．なお，プライバシーや使用するメモリ領域の観点から，パケットのヘッダのみや先頭から固定ビット分を記録することが多い．

近年，ネットワークルータにはトラヒック計測のための機能が標準で備えてあるものが多い．Cisco 社の NetFlow [11] や Inmon 社の sFlow [15] などが挙げられる．NetFlow をベースに IETF が標準化を行った規格の IPFIX では，ルータで計測されたトラヒックをどのようにデータの収集地点まで送るかについて定義している．ここで NetFlow について簡単に説明する．まずルータでは転送されるパケットごとのヘッダ情報が読み取られる．このとき，全てのパケットのヘッダを読むのではなく一部のパケットをサンプリングして読み取ることが可能となっている．これはトラヒックを計測する上で，回線速度に対するスケーラビリティを実現させる．すなわち，回線速度が 10 倍や 100 倍になったとしても，サンプリングの頻度を 10 分の 1 や 100 分の 1 にすることで，パケットを処理しきれなくなったり，メモリ領域が不足するということが回避できる．全てのパケット，あるいはサンプリングされたパケットの情報は，フローごとに集約されフローレコードに記録される．フロー集約については次の小節で詳しく述べる．そしてあるタイミングでフローレコードはルータからデータの収集地点へ転送される．そのタイミングとは，

- TCP のコネクションの終了パケットが計測されたとき
- ある期間新しいパケットが到着しなかったとき
- ある期間パケットが途切れなかったとき

- フローレコードを登録するメモリ領域が一杯になったとき

の四つである．収集地点では送られてきたフローレコードを解析して，対象の回線のトラヒックを調べることができる．

本論文では，ネットワーク内部のルータあるいはリンクを流れるトラヒックを計測し，異常トラヒックをオンラインで検出することを考える．このとき，トラヒックのオンライン計測技術として必要になるのは，(i) 流れているパケットをサンプリングできること，(ii) 取得したパケットを任意のフローに集約できること，(iii) 解析地点において一定時間 (任意) ごと，あるいはそれよりも細かい粒度でフローデータが取得可能なこと，の三つである．解析地点は計測地点と離れていてもよいが，その間でデータ損失や大幅な遅延が起こらないことが望ましい．この三つに関連した技術である，パケットサンプリング，フロー集約，オンラインデータ処理について論じる．

1.2.1 パケットサンプリング

ここではパケットサンプリングについて説明する．先に示したように，最近のネットワークルータにはパケットサンプリングの機能を有するものが存在する [11, 15]．また，パケットサンプリングは IETF [14] によって標準化されたフロー計測に関する標準技術 IPFIX でも規定されており，RFC 5474 から RFC 5477 で以下のようなパケットサンプリングを確認できる．

- 規則的サンプリング
  - カウント型サンプリング
  - 時間型サンプリング
- ランダムサンプリング
  - n-out-of-N サンプリング
  - 確率的サンプリング
    - 一様確率サンプリング
    - 非一様確率サンプリング (フロー情報非依存)
    - 非一様確率サンプリング (フロー情報依存)

規則的サンプリングでは，パケットのサンプリングの開始および停止のきっかけが，一定数のパケットの通過 (カウント型) あるいは一定時間の経過 (時間型) によって規定されるため，K パケットごとに一つパケットをサンプリングするといったことが可能となる．一方，ランダムサンプリングは疑似乱数などを用いてどのパケットをサンプリングするか決める．n-out-of-N サンプリングは $[1:N]$ から n 個の数字をランダムに選び，通過順がそれに一致するパケットをサンプリングす

る．確率的サンプリングではパケットごとにある確率 f でサンプリングするかどうか決める．この確率が常に固定されているものは一様確率サンプリングと呼ばれ，状況に応じて変更する場合は非一様確率サンプリングと呼ばれる．非一様確率サンプリングは，確率を変えるときに既に得ているフローの情報を利用するかどうかでさらに区別される．フロー情報依存型の例として，*Sample and Hold* [12] が挙げられる．

本論文では上記のパケットサンプリングのうち，一様確率サンプリングを一貫して用いる．このとき使用する確率をサンプリングレートと呼ぶ．このサンプリングを用いると，トラヒックの到着過程に時間的な相関があるような場合でも各パケットを独立かつ同一にサンプリングすることができ，またサンプリングレートが与えられれば母集団におけるパケット数から標本におけるパケット数を確率的に見積もることができる．本論文でランダムパケットサンプリングと言った場合はこの一様確率サンプリングを指す．

ここで二項分布について説明する．あるフローの母集団におけるパケット数を確率変数 X，ランダムパケットサンプリングののちに抽出されたパケット数を確率変数 Y を用いて表すとき，$X = x$ の条件の下でサンプリングレート f $(0 < f \leq 1)$ でランダムパケットサンプリングを行った場合に $Y = y$ となる確率は，二項分布で以下のように表される．

$$P(Y = y \mid X = x) = \binom{x}{y} f^y (1-f)^{x-y}, \qquad y = 0, 1, \ldots, x \tag{1.1}$$

1.2.2 フロー集約

取得したパケットの情報をそのままパケット単位で保持するのはメモリ容量の面でも，その後のデータ処理の面でも非常に効率が悪い．そこで用いられる技術がフロー集約である．

先にも述べたが，フローは指定する指標あるいはその組み合わせに，同じ値を有するパケット群として定義される．パケット一つ一つのデータには興味がなく，フローとしてのパケット数やバイト数，存続時間などに興味がある場合は，パケット数やバイト数は合計として，存続時間は第一パケットから最終パケットまでの時間差としてそれぞれ集約することができる．五つ組でフローを集約した場合，二つのホストのソケット間の通信をフローとみなすことができ，これは最も細かい粒度でのフローと言える．

一方で DDoS 攻撃や，送信元 IP アドレスを改竄したパケットを用いた SYN フラッド攻撃などは，様々な送信元から特定の宛先にパケットが送られる．攻撃に関わる一連のパケット群を同一のフローとみなすことで，攻撃トラヒックの検出が高パケットレートフローの検出の枠組みで行える．すなわち，宛先 IP アドレス，あるいはそれに宛先ポート番号やプロトコルを加えたものでフローを定義することで，攻撃に関わるパケットを一つのフローとみなすことができる．フローは目的に応じて自由に定義を決められるものとして以降の議論を進める．なお，フロー

集約を実際に計算機で行う際には，フローの識別子を引数としたハッシュ関数を用いることで処理時間が軽減できることに注意する．

1.2.3 スライディングウィンドウ方式

ネットワークを常に監視するには，流れているトラヒックを半永久的に計測，解析し続ける仕組みが必要となる．このとき，計測したデータをいつまでも持ち続けることはメモリ領域の観点から現実的ではない．どこかの時点でそれまでに計測されたデータを解析し，解析が終わったデータは破棄，あるいは二次記憶に出力するなどして次にやってくるデータのためにメモリ領域を開放する必要がある．このとき，解析したい内容によって，どれだけのデータを解析対象として (一時的に) 保持するかが決まってくる．

例えば，各ユーザの月ごとの回線使用量を計算する場合は，一ヶ月間は集計しながらデータを保持し，期間が終了したらまた一から計測し集計していく．一方で，異常な高レートのフローの検出を試みる場合はより細かい尺度で解析する必要がある．これは，月ごとの降水量と，土砂災害の警戒のための時間降水量の計測に類似する．ある時点での時間降水量は 1 分ごとにその時点から 1 時間前までの降水量の合計として計算される．1 分ごとの降水量のデータを 1 時間前までそれぞれ保持しておけば，最新の 1 分間のデータが計測されるたびにそのデータを加え，最も古い 1 分間のデータを差し引くことで容易に計算できることがわかる．

この仕組みはスライディングウィンドウ (SW) 方式と呼ばれる．解析データを保持するスライディングウィンドウ (SW) はベーシックウィンドウ (BW) と呼ばれるウィンドウ (更新単位) に分割されており，その BW の単位で新しいデータを取得し，古い BW と置き換えることで解析対象を細かい粒度で更新していく．なお，SW を BW に分割せずに，解析が終わるとデータを全て置き換えるものをジャンピングウィンドウ (JW) 方式と呼ぶ．月別降水量の例はこちらに相当する．

本論文ではこの SW および BW を時間で規定し，それぞれ T_{SW} および T_{BW} で一定とする．すなわち，まずは T_{SW} 分の解析データを収集し，その後は一定時間 T_{BW} ごとにデータを取得し，その単位で解析データを更新しながら解析していく．各ウィンドウを時間で規定することで，データ処理の間隔を一定にすることができる．また，2 章および 3 章で見るようにパケットレートに興味がある場合には，分母の時間が固定できることで，あらかじめ高パケットレートフロー検出のためのパケット数の閾値を計算することができる．一方で，SW ごとあるいは BW ごとのデータ量 (パケット数，フロー数) は異なるため，データ領域が厳密に制限されるような場合には注意が必要である．

1.3 構成

本論文の次章以降は以下のように構成される．

第 2 章では，高パケットレートフローのオンライン検出におけるパラメータ決定手法について述べる．まずパケットサンプリングによるトラヒックデータの取得とスライディングウィンドウ方式によるデータ更新を採用した高パケットレートフローのオンライン検出手法について述べ，誤検出確率や未検出確率，オンライン検出の実行可能性などを考慮したパラメータ決定手法を提案する．そして実トレースデータを用いた数値実験により評価を行う．本章の内容は，研究業績の雑誌論文 ⟨1⟩，特許 ⟨1⟩，研究会 ⟨1⟩ に関連する．

第 3 章では，一定時間以上高いパケットレートが続くような，持続的高パケットレートフローの検出手法について提案する．本検出手法もパケットサンプリングとスライディングウィンドウ方式を用いている．その後，第 2 章で提案したパラメータ決定手法を踏襲した，本検出手法に対するパラメータ決定手法を提案し，実トレースデータを用いた提案手法の評価を行う．本章の内容は，研究業績の雑誌論文 ⟨2⟩，特許 ⟨2⟩，ならびに研究会 ⟨2⟩ に関連する．

第 4 章では，ポートスキャントラヒックのオンライン検出について述べる．まず TCP ポートスキャンの検出手法について述べたのち，誤検出確率と未検出確率の両方を考慮したパラメータ決定手法を提案する．ここでも実トレースデータを用いた数値実験を行い提案手法の評価を行う．本章の内容は，研究業績の国際会議 ⟨2⟩ に関連する．

最後に第 5 章において本論文の結論を述べる．

第2章

高パケットレートフローの検出におけるパラメータ決定手法

2.1 まえがき

本章では，まずパケットサンプリングによるトラヒックデータの取得とスライディングウィンドウ方式によるデータ更新を採用した高パケットレートフローのオンライン検出手法について述べ，誤検出確率や未検出確率，オンライン検出の実行可能性などを考慮したパラメータ決定手法を提案する．そして実トレースデータを用いた数値実験により評価を行う．

2.2 高パケットレートフローのオンライン検出手法

検出対象のフローをパケットレートが R [packets/s] 以上のフローと定義する．SW 方式を用いて T_{SW} のウィンドウで検出しようとする場合，$x^* = \lceil RT_{\mathrm{SW}} \rceil$ とすると，検出対象となるフローのパケット数は x^* 個以上となる．この検出対象フローをサンプリングレートが f のランダムパケットサンプリングを用いて検出することを考える．任意のフローの母集団におけるパケット数を X，サンプリングされた後のパケット数を Y とすると，式 (1.1) の二項分布を使って x パケットから y パケットだけサンプリングされる確率が計算される．ここで標本における閾値 y^* を設け，y^* 個以上のパケットがサンプリングされたフローを検出することにする．このとき，どのような検出対象のフローでも，サンプルパケット数が y^* 未満となり検出できずに見逃す確率 (未検出確率) を十分小さな誤差 ϵ 以下に抑えるようにパラメータを決定する．

母集団におけるパケット数が x^* のフローを考える．以降このフローを閾値フローと呼ぶ．閾値フローからサンプリングされるパケット数が y^* 未満となる確率，すなわち未検出確率が ϵ 以下に抑えられれば，それ以上のパケット数をもつ検出対象フローの未検出確率も ϵ 以下に抑えることが可能となる．そこで $y^* = y^*(R, T_{\mathrm{SW}}, f, \epsilon)$ を次のように決定する．

$$y^* \quad = \quad \max_{y} \{y;\ \Pr[Y \leq y - 1 \mid X = x^*] \leq \epsilon\}$$

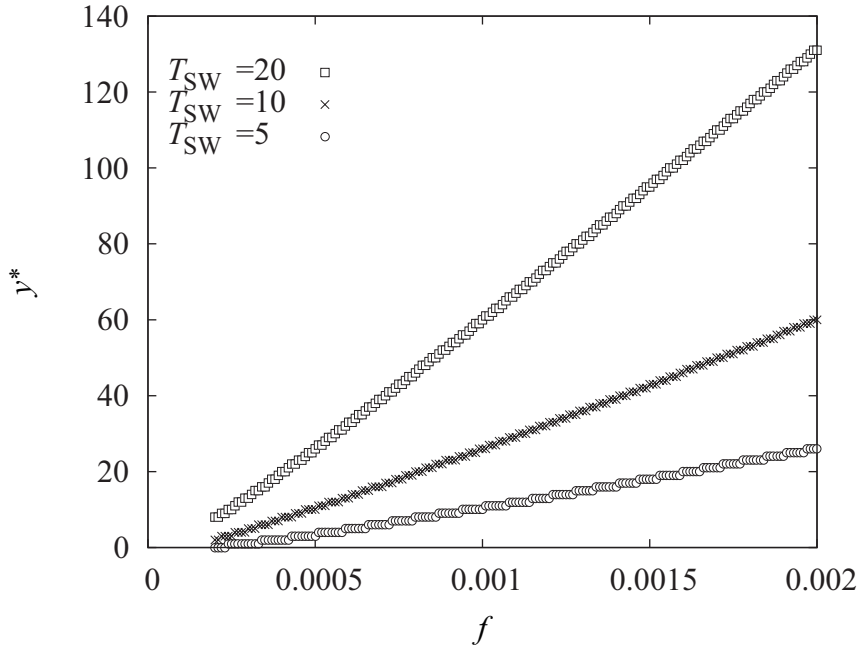

図 2.1: f の関数とみなした y^* $(R = 4,000, \epsilon = 0.01)$

$$= \max_y \left\{ y;\ \sum_{i=0}^{y-1} \binom{x^*}{i} f^i (1-f)^{x^*-i} \leq \epsilon \right\} \tag{2.1}$$

通常，検出対象フローは閾値より多くのパケット数をもつ可能性が高いため，それらのフローを見逃す確率は ϵ を大きく下回ることになる．一方，y^* が決定されると，パケット数が z $(z < x^*)$ のフローを誤検出してしまう確率は，

$$1 - \sum_{i=0}^{\min(y^*-1,z)} \binom{z}{i} f^i (1-f)^{z-i}$$

で与えられる．図 2.1 にサンプルパケット数の閾値 y^* をサンプリングレート f の関数とみなし，$R = 4000$ [packets/s]，$\epsilon = 0.01$ としたときの計算結果を示す．y^* は自然数であるため，T_{SW} を固定して考えると f の階段関数となることがわかる．

SW 方式とランダムパケットサンプリングを組み合わせた上記の検出手法は，予め与えられるパラメータであるパケットレートの閾値 R と未検出確率の許容誤差 ϵ とともに以下の三つの制御パラメータによって特徴付けられる．

- サンプリングレート f
- SW の長さ T_{SW} [s]
- SW 内の BW 数 k

ここで閾値フローのうち，パケットの到着間隔が $1/R$ [s] で一定のフローを考え，この一定レート閾値フローが時刻 t_{R}^* に発生したと仮定し，以下のような最適化問題を解くことで制御パラメータを決定する．

最小化： 低パケットレートフローの誤検出確率
条件： 一定レート閾値フローの未検出確率 $\leq \epsilon$
一定レート閾値フローの検出時刻 $\leq t^*_{\mathrm{R}} + T_{\mathrm{D\_max}}$
オンラインアルゴリズムとして正常に動作

ここで $T_{\mathrm{D\_max}}$ は最大許容検出遅延を表し，予め与えられるパラメータとする．なお，一つ目の未検出確率の条件は，サンプリングレート f と SW の長さ T_{SW} が決定すると式 (2.1) を用いて y^* を決定することで満たされるため，次節ではこの条件を除いた問題を考える．

2.3 パラメータ決定手法

2.3.1 制約条件

前節で示した最適化問題の二つ目と三つ目の制約条件を考える．一定レート閾値フローの検出遅延時間を T_{D} とすると，T_{D} は以下のように上から抑えることができる．

$$T_{\mathrm{D}} \leq T_{\mathrm{SW}} + T_{\mathrm{SW}}/k + \tau \leq T_{\mathrm{D\_max}} \tag{2.2}$$

ここで，τ は SW 内のデータ解析に要する時間の最大値としている．式中の T_{SW}/k は BW の長さを表し，一定レート閾値フローの検出にはデータを集める時間 T_{SW} と集めたデータの処理時間 τ に加えて，最大で 1BW 分の時間がかかることを示している．これは当該フローが多くの場合 BW の途中から始まることに起因する．フローの始まりを含む BW が SW 内でもっとも古い BW となったとき，当該フローの SW におけるパケットレートは R には届いていない．SW 全体でのパケットレートが R に届くのは始まりを含む BW の次の BW がもっとも古い BW となったときである．すなわち，一定レート閾値フローが発生してから SW 全体でとらえるまでに最大で 1BW 分の時間が経過することになる．

一方で，データ処理を含めた検出手法がオンラインアルゴリズムとして正常に動作するための条件は，現在の SW に対する処理が次の BW の取得時刻までに完了することである．すなわち次式で表される．

$$\tau \leq T_{\mathrm{SW}}/k \tag{2.3}$$

ここで，SW の処理時間 τ について見積りを行う．SW が更新されるときには，最も古い BW のデータが削除され，新しい BW のデータが SW に加えられる．その処理時間は BW 内のデータ数に応じて長くなると考え，BW 内のサンプルパケット数の狭義単調増加関数 $G(\cdot)$ とする．母集団において BW 内に N パケットあったとすると，標本における期待値は $N \times f$ パケットである．回線全体のパケットレートの最大値を $C_{\max}$ [packets/s] とすると，BW におけるサンプルパケット数の最大値は $fC_{\max}T_{\mathrm{SW}}/k$ で見積もることができ，τ は以下のようになる．

$$\tau = G\left(\frac{fC_{\max}T_{\mathrm{SW}}}{k}\right) \tag{2.4}$$

なお，$fC_{\max}T_{\mathrm{SW}}/k$ はサンプルフロー数の期待値の上限としてみなすこともできる．

2.3.2 目的関数

次に，最適化問題の目的関数について考える．$P_{\mathrm{WD}}(r)$ をパケットレートが r [packets/s] のフローの誤検出確率とする．

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

ここで，r $(r < R)$ は rT_{SW} が整数になるように選ばれるとする．$rT_{\mathrm{SW}} < y^*$ となるような r に対しては $P_{\mathrm{WD}}(r) = 0$ となるので，$rT_{\mathrm{SW}} \geq y^*$ を仮定する．

目的関数 $P_{\mathrm{WD}}(r)$ を解析的に扱える形で表現することを考える．まず,

$$P_{\mathrm{WD}}(r) = 1 - \sum_{y=0}^{y^*-1} \binom{rT_{\mathrm{SW}}}{y} f^y (1-f)^{rT_{\mathrm{SW}}-y}$$

となることに注意する．ここで二項分布のポアソン近似を適用する (文献 [13] VI.5)．すなわち，十分大きな $x \gg 1$ と十分小さな $f \ll 1$ に対して，二項分布は同じ平均を持つポアソン分布に近似される．

$$\binom{x}{y} f^y (1-f)^{x-y} \approx e^{-fx} \frac{(fx)^y}{y!}$$

このとき誤検出確率は,

$$P_{\mathrm{W}D}(r) \approx 1 - \sum_{y=0}^{y^*-1} e^{-rfT_{\mathrm{SW}}} \frac{(rfT_{\mathrm{SW}})^y}{y!} \tag{2.5}$$

となる．さらに，パケットレートが低いフロー $(r \ll R)$ に対して，ポアソン近似は $P_{\mathrm{W}D}(r)$ の上限を与える．

定理 1. rT_{SW} は整数とする．$\lceil rfT_{\mathrm{SW}} \rceil \leq (y^*-1)/2$ が満たされるとき (すなわち，$y^* \geq 2\lceil rfT_{\mathrm{SW}} \rceil + 1$ となるとき)，以下が成り立つ．

$$P_{\mathrm{W}D}(r) < 1 - \sum_{y=0}^{y^*-1} e^{-rfT_{\mathrm{SW}}} \frac{(rfT_{\mathrm{SW}})^y}{y!} \tag{2.6}$$

定理 1 の証明は付録 A に示す．ここで，rfT_{SW} (> 0) はパケットレートが r のフローから SW にサンプリングされるパケット数の期待値を表すことに注意する．なお，$\lceil rfT_{\mathrm{SW}} \rceil \geq 1$ であるため $y^* \geq 3$ が前提となる．また，y^* は大きいほど式 (2.6) が成り立つ r の範囲が広くなるため好ましい．このことは 2.4.3 でも議論する．

式 (2.5) より，固定された任意の y^* と r に対して，検出対象外のフローの誤検出確率は fT_{SW} に関する狭義単調減少関数となる．このことは，サンプルパケッ

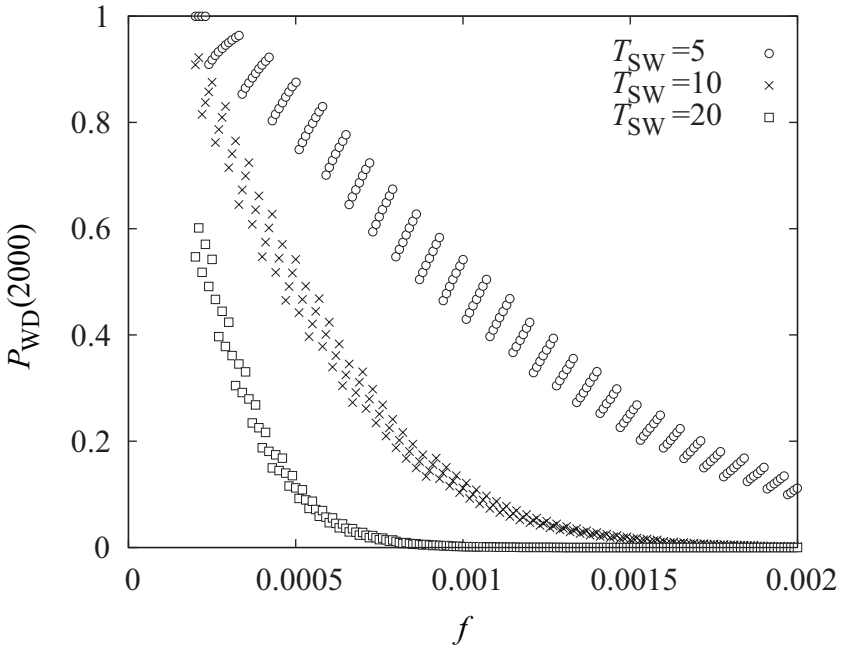

図 2.2: f の関数とみなした誤検出確率 ($R = 4,000$, $r = 2,000$, $\epsilon = 0.01$)

ト数の期待値 $r \times fT_{\mathrm{SW}}$ が増えれば増えるほど，そこから得られる情報も増え，検出の精度も上がるであろうという直感とも一致する．したがって，fT_{SW} を最大化することがおおよそ検出対象外フローの誤検出確率を最小化することになると考えられる．

この考えを検証するための数値計算の結果を示す．図2.2は，$R = 4,000$, $\epsilon = 0.01$ としたときのパケットレート $r = 2000$ [packets/s] のフローに対する誤検出確率を，f の関数とみなして示したものである．図より $P_{\mathrm{WD}}(r)$ は狭義単調減少関数にはなっていないことがわかる．この原因としては，y^* が自然数であることを思い出すと，T_{SW} が固定された状況では f を大きくするほどサンプルパケット数は増加し，y^* が変わらないかぎりは誤検出確率が上がるためである．

図 2.2 と同じ設定で，今度は $r = 2,000$ [packets/s] のフローの誤検出確率を fT_{SW} の関数として描画したものを図 2.3 に示す．この図で誤検出確率は T_{SW} の値に関わらず非常に似通った特徴を示しており，fT_{SW} の狭義単調減少関数で抑えられることがわかる．この現象は r の値を変えても同様に観測された．以上の議論より，サンプリングレートとウィンドウの長さの積 fT_{SW} が検出対象外のフローの誤検出確率を制御する決定的な要素と結論づける．

2.3.3 大域的最適解

これまでの議論から，検出手法のパラメータ決定問題は以下のように記述される．

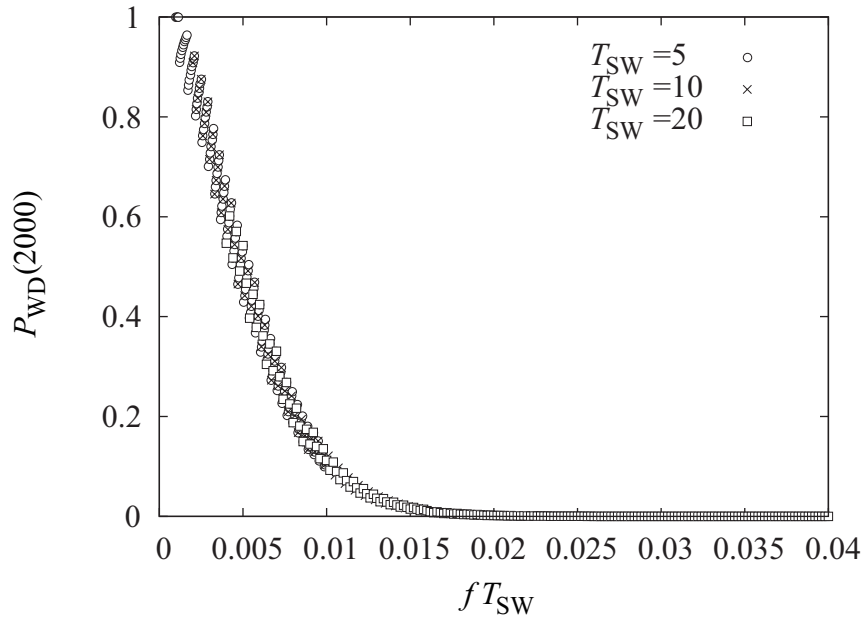

図 2.3: fT_{SW} の関数とみなした誤検出確率 ($R = 4,000$, $r = 2,000$, $\epsilon = 0.01$)

P: 最大化 fT_{SW}

条件 $T_{\mathrm{SW}} > 0$, $f > 0$, k は自然数

$$\frac{k+1}{k}T_{\mathrm{SW}} + G\left(\frac{fC_{\max}T_{\mathrm{SW}}}{k}\right) \leq T_{\mathrm{D\_max}}$$

$$G\left(\frac{fC_{\max}T_{\mathrm{SW}}}{k}\right) \leq \frac{T_{\mathrm{SW}}}{k}$$

ここで，下の二つの制約条件は (2.2)，(2.3)，そして (2.4) から導かれる．

$G(x)$ $(x \geq 0)$ は x の正の狭義単調増加関数とした．問題を簡単にするため微分可能と仮定すると，$G(x)$ の逆関数 $G^{-1}(x)$ が存在し，一次導関数は $G'(x) > 0$ $(x \geq 0)$ となる．

(f, T_{SW}) は最適化問題 P の実行可能解とする．このとき，最後の制約条件より，

$$\frac{T_{\mathrm{SW}}}{k} \geq G\left(\frac{fC_{\max}T_{\mathrm{SW}}}{k}\right) > G(0)$$

となり，最後から二つ目の制約条件から

$$\begin{aligned} T_{\mathrm{D\_max}} &\geq (k+1)G\left(\frac{fC_{\max}T_{\mathrm{SW}}}{k}\right) + G\left(\frac{fC_{\max}T_{\mathrm{SW}}}{k}\right) \\ &> (k+2)G(0) \end{aligned}$$

が導かれる．これは最適化問題 P が実行可能解を持つための必要条件である．$G(0)$ は SW におけるデータ処理のオーバヘッドとして解釈でき，最大許容検出遅延は k が自然数であることを考慮すると次のようにして設定される必要がある．

$$T_{\mathrm{D\_max}} > 3G(0) \tag{2.7}$$

これ以降，$T_{\mathrm{D\_max}} > 3G(0)$ とし，$\mathcal{K}$ を以下のような空でない自然数の集合とする．

$$\mathcal{K} = \left\{1, 2, \ldots, \left\lfloor \frac{T_{\mathrm{D\_max}}}{G(0)} \right\rfloor - 2\right\}$$

このとき，もし $k \in \mathcal{K}$ であれば，最適化問題 P は実行可能解を持つことになる．

定理 2. $k \in \mathcal{K}$ が与えられたとき，最適化問題 P の大域的最適解 (f^*, T^*_{SW}) が唯一存在し，以下で与えられる．

$$\begin{aligned} f^* &= \frac{k+2}{C_{\max} T_{\mathrm{D\_max}}} G^{-1}\left(\frac{T_{\mathrm{D\_max}}}{k+2}\right) \\ T^*_{\mathrm{SW}} &= \frac{k}{k+2} T_{\mathrm{D\_max}} \end{aligned}$$

定理 2 の証明は付録 B に示した．この証明において，最適化問題 P の最後の二つの制約条件は所与 $k \in \mathcal{K}$ に対して有効制約となっており，$f = f^*$ および $T_{\mathrm{SW}} = T^*_{\mathrm{SW}}$ としたときにそれぞれの制約条件が等号で成り立つことを示している．

最適化問題 P の目的関数の最適値 $f^* T^*_{\mathrm{SW}}$ は $kG^{-1}(T_{\mathrm{D\_max}}/(k+2))$ に比例する．したがって，k の最適解 $k = k^*$ は

$$k^* = \arg\max_{k \in \mathcal{K}} kG^{-1}\left(\frac{T_{\mathrm{D\_max}}}{k+2}\right) \tag{2.8}$$

で与えられ，最適化問題 P における f と T_{SW} の最適解 $f = f^*$ と $T_{\mathrm{SW}} = T^*_{\mathrm{SW}}$ はそれぞれ以下で与えられる．

$$f^* = \frac{k^*+2}{C_{\max} T_{\mathrm{D\_max}}} G^{-1}\left(\frac{T_{\mathrm{D\_max}}}{k^*+2}\right) \tag{2.9}$$

$$T^*_{\mathrm{SW}} = \frac{k^*}{k^*+2} T_{\mathrm{D\_max}} \tag{2.10}$$

2.3.2 節で示したように，fT_{SW} の関数である誤検出確率は，y^* が一定の範囲では f に関して増加関数になる．このことを考慮し，三つの制御パラメータ k, T_{SW}，f を以下のように設定する．

1. 式 (2.8) と式 (2.10) を用いて $k = k^*$ と $T_{\mathrm{SW}} = T^*_{\mathrm{SW}}$ を設定
2. 式 (2.9) を用いて $f = f^*$ を設定
3. 式 (2.1) を用いて y^* を設定
4. $f^{**} = \arg\min_f \left\{ \sum_{i=0}^{y^*-1} \binom{\lceil RT^*_{\mathrm{SW}} \rceil}{i} f^i (1-f)^{\lceil RT^*_{\mathrm{SW}} \rceil - i} \leq \epsilon \right\}$ を用いて $f = f^{**}$ を設定

2.4 数値実験

この節ではトレースデータに対する実験結果と，SW 方式に関するいくつかの基本的な性質について議論する．実験には CAIDA トレースデータ [6] を用いた．このトレースデータは 2009 年 5 月 31 日の 6:00 から 6:05 にかけて 10Gbps のバックボーンリンクで計測された．この 300 秒のトレースデータは 152,593,821 パケットからなり，五つ組を用いてフローを定義すると，13,603,014 フローからなる．

表 2.1: 制御パラメータと閾値 ($T_{\mathrm{D\_max}} = 10$)

R	k^*	T^*_{SW}	$\epsilon = 0.01$		$\epsilon = 0.05$		$\epsilon = 0.1$	
			f^{**} $(\times 10^{-4})$	y^*	f^{**} $(\times 10^{-4})$	y^*	f^{**} $(\times 10^{-4})$	y^*
1000	61	9.6825	8.679	3	9.452	5	9.577	6
2000			8.985	9	9.401	12	9.181	13
4000			9.511	24	9.613	28	9.604	30

2.4.1 実験準備

実験を行うにあたり，まず SW 内のデータに対する処理時間を，BW 内にサンプリングされたフロー数 (パケット数で上から抑えられる) に線形に増加する以下のような関数で評価する．

$$G(x) = \Delta_1 x + \Delta_2$$

Δ_1 [s] はサンプリングされた各フローごとにかかる処理時間を表し，Δ_2 [s] はサンプリングされたフロー数とは独立に SW の処理にかかる処理時間を表す．Δ_2 は削除する最も古い BW のメモリ空間を解放したり，検出結果を集約し，収集地点へ転送するのにかかる時間などを含む．ここで，実行可能解を持つための条件である式 (2.7) より $T_{\mathrm{D\_max}} > 3\Delta_2$ が成り立つと仮定する．

このとき，k の最適解 k^* は以下で与えられる．

$$k^* = \begin{cases} \underset{k \in \{k^-, k^+\}}{\arg\max} \left\{ \dfrac{k}{k+2} T_{\mathrm{D\_max}} - k\Delta_2 \right\}, & T_{\mathrm{D\_max}} > 9\Delta_2/2 \\ 1, & 3\Delta_2 < T_{\mathrm{D\_max}} \leq 9\Delta_2/2 \end{cases}$$

ただし，

$$k^- = \left\lfloor \sqrt{\frac{2T_{\mathrm{D\_max}}}{\Delta_2}} - 2 \right\rfloor,\ k^+ = \left\lceil \sqrt{\frac{2T_{\mathrm{D\_max}}}{\Delta_2}} - 2 \right\rceil$$

である．T^*_{SW} は式 (2.10) で与えられ，式 (2.9) で示された f^* は次式のようになる．

$$f^* = \frac{T_{\mathrm{D\_max}} - (k^* + 2)\Delta_2}{C_{\max}\Delta_1 T_{\mathrm{D\_max}}}$$

本節の実験では $\Delta_1 = 5 \times 10^{-4}$ [s]，$\Delta_2 = 5 \times 10^{-3}$ ，そして $C_{\max} = 2 \times 10^6$ [packets/s] とした．また $T_{\mathrm{D\_max}}$ に関しては，$T_{\mathrm{D\_max}} = 10,\ 20$ [s] の二通りを考える．さらにこの二通りに対して，パケットレートの閾値を $R = 1000, 2000, 4000$ [packets/s] の三通り，未検出確率の許容誤差を $\epsilon = 0.01, 0.05, 0.10$ の三通りにそれぞれ設定した．2.3 節の最後に示した手順にしたがって計算した制御パラメータを表 2.1 と表 2.2 に示す．

2.4.2 性能評価指標

提案手法を評価するために，最も情報が多く理想的な状況である $f = 1.0$ (全てのパケットをサンプリング) のときとの比較を行う．ただし，理想的な状況でも k

表 2.2: 制御パラメータと閾値 ($T_{\mathrm{D\_max}} = 20$)

R	k^*	T^*_{SW}	$\epsilon = 0.01$		$\epsilon = 0.05$		$\epsilon = 0.1$	
			f^{**} $(\times 10^{-4})$	y^*	f^{**} $(\times 10^{-4})$	y^*	f^{**} $(\times 10^{-4})$	y^*
1000	87	19.5506	9.605	10	9.312	12	9.696	14
2000			9.737	25	9.522	28	9.513	30
4000			9.720	57	9.653	62	9.657	65

と T_{SW} は提案手法と同じ値を用いた．ここで検出対象フローを検出した時刻を，理想的な状況では t_1，提案手法では t_2 でそれぞれ表す．簡単のため，どちらの状況でも検出した時刻にはそのフローを初めて検出した SW の取得完了時刻を採用した．言い換えるならば，SW の処理時間がどちらも等しく，十分短いという状況を想定している．

検出対象フローを検出結果に応じて次の四つのクラスに分類する．(i) $t_1 > t_2$, (ii) $t_1 = t_2$, (iii) $t_1 < t_2 < \infty$, そして (iv) $t_2 = \infty$ である．ここで，$f = 1.0$ の理想的な状況では誤検出も未検出も起こらないことに注意する．一方で提案手法ではそのどちらも起こりうる．それゆえ，検出対象フローもパケットレートが閾値に達する前に検出されるということが起こる．これがクラス (i) に相当する．検出対象フローの検出という観点から見ると，クラス (i) とクラス (ii) は確実に $T_{\mathrm{D\_max}}$ 以内に検出されたフローということになる．

クラス (iii) は提案手法で検出はできたが，検出遅延は最大許容検出遅延を過ぎてしまっている可能性があるフローである．ここで注意したいが，$t_1 < t_2$ だからと言って，必ずしも検出遅延が最大許容検出遅延を過ぎるとは限らない．例えばパケットレートが $2R$ の一定レートフローが時刻 t_0 に発生したと考える．発生してからおよそ $T_{\mathrm{SW}}/2$ 過ぎると SW 内のパケット数が閾値に到達する．t_1 はこのときの時刻を示すことになる．すなわち，$T_{\mathrm{D\_max}}$ の約半分の時間が基準となる．仮に $t_2 > t_1$ であっても，$t_2 \leq t_0 + T_{\mathrm{D\_max}}$ であれば遅延制約は満たされている．最後のクラス (iv) は未検出，すなわち検出対象フローを検出することができなかったことを意味する．

一方，検出対象外のフローについては，次式で定義される誤検出確率 FPR により評価する．

$$\mathrm{FPR} \quad = \quad \frac{\text{誤検出されたフロー数}}{\text{検出対象外フロー数}} \tag{2.11}$$

また，誤検出された検出対象外フローのパケットレートについても議論する．

2.4.3 実験結果

まず，実験に使用した CAIDA トレースデータに関して基本的な性質を示す．図 2.4 と図 2.5 はそれぞれ $T_{\mathrm{SW}} = 9.6825$，$k = 61$，$f = 1$ (表 2.1 参照) のときと，$T_{\mathrm{SW}} = 19.5506$，$k = 87$，$f = 1$ (表 2.2 参照) のときに，各フローをパケットレー

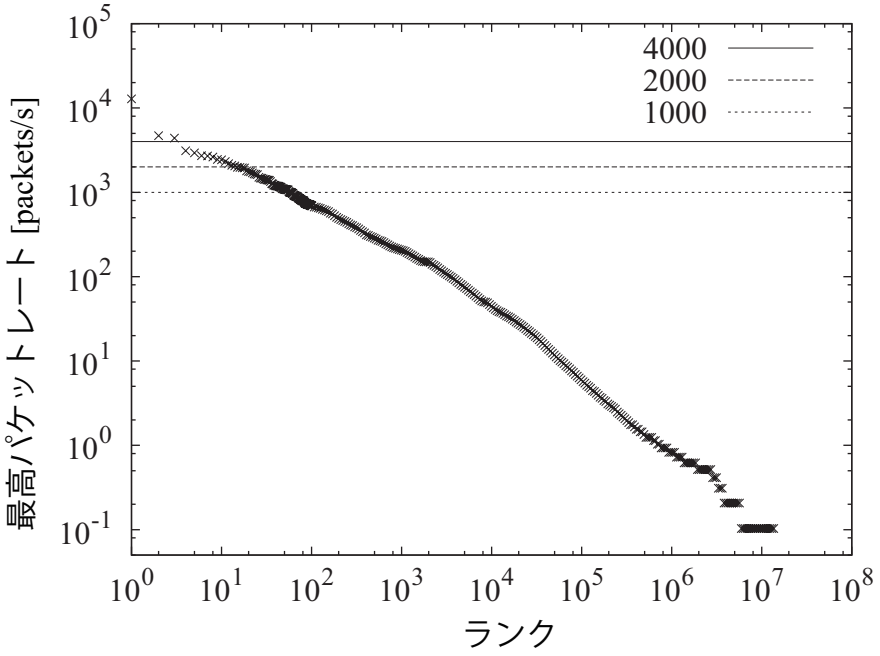

図 2.4: フローの最高パケットレートに基づくランキング ($T_{\mathrm{D\_max}} = 10$)

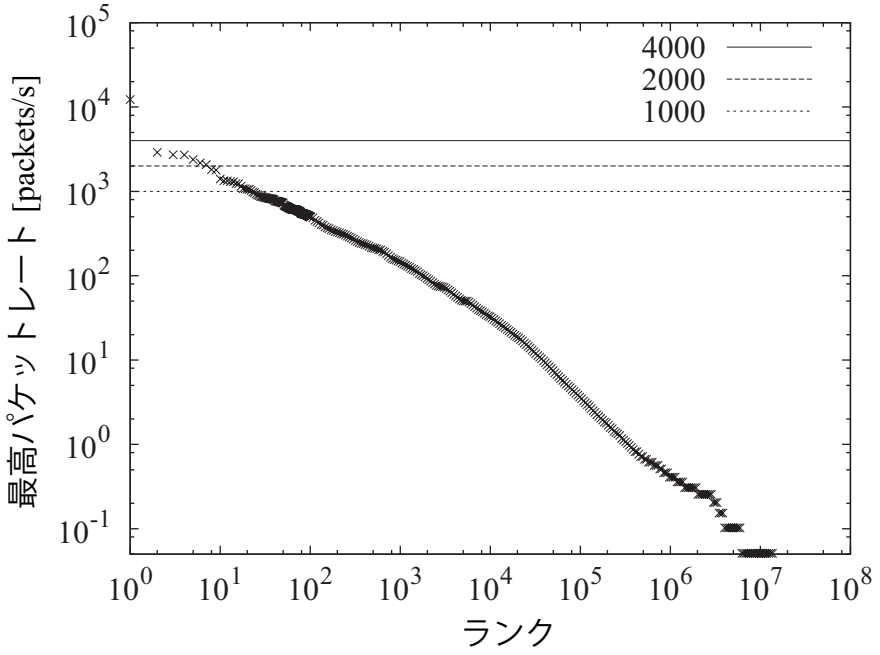

図 2.5: フローの最高パケットレートに基づくランキング ($T_{\mathrm{D\_max}} = 20$)

表 2.3: 検出対象フローの検出数と検出率 ($T_{\mathrm{D\_max}} = 10$)

$\epsilon = 0.01$					
R	N_{T}	(i) $t_1 > t_2$	(ii) $t_1 = t_2$	(iii) $t_1 < t_2 < \infty$	(iv) $t_2 = \infty$
1000	58	57.388 ± 0.047	0.226 ± 0.030	0.317 ± 0.034	0.069 ± 0.016
		0.993 ± 0.001		$5.47 \times 10^{-3} \pm 5.86 \times 10^{-4}$	$1.19 \times 10^{-3} \pm 2.79 \times 10^{-4}$
2000	14	13.813 ± 0.027	0.081 ± 0.017	0.084 ± 0.019	0.022 ± 0.009
		0.992 ± 0.001		$6.00 \times 10^{-3} \pm 1.36 \times 10^{-3}$	$1.57 \times 10^{-3} \pm 6.50 \times 10^{-4}$
4000	3	2.946 ± 0.014	0.027 ± 0.010	0.024 ± 0.009	0.003 ± 0.003
		0.991 ± 0.003		$8.00 \times 10^{-3} \pm 3.16 \times 10^{-3}$	$1.00 \times 10^{-3} \pm 1.13 \times 10^{-3}$
$\epsilon = 0.05$					
R	N_{T}	(i) $t_1 > t_2$	(ii) $t_1 = t_2$	(iii) $t_1 < t_2 < \infty$	(iv) $t_2 = \infty$
1000	58	54.880 ± 0.108	0.869 ± 0.056	1.843 ± 0.083	0.408 ± 0.040
		0.961 ± 0.002		$3.18 \times 10^{-2} \pm 1.44 \times 10^{-3}$	$7.03 \times 10^{-3} \pm 6.86 \times 10^{-4}$
2000	14	13.147 ± 0.055	0.318 ± 0.035	0.442 ± 0.041	0.093 ± 0.019
		0.962 ± 0.003		$3.15 \times 10^{-2} \pm 2.96 \times 10^{-3}$	$6.64 \times 10^{-3} \pm 1.35 \times 10^{-3}$
4000	3	2.777 ± 0.028	0.092 ± 0.019	0.110 ± 0.020	0.021 ± 0.009
		0.956 ± 0.007		$3.67 \times 10^{-2} \pm 6.73 \times 10^{-3}$	$7.00 \times 10^{-3} \pm 2.96 \times 10^{-3}$
$\epsilon = 0.1$					
R	N_{T}	(i) $t_1 > t_2$	(ii) $t_1 = t_2$	(iii) $t_1 < t_2 < \infty$	(iv) $t_2 = \infty$
1000	58	51.849 ± 0.142	1.680 ± 0.078	3.593 ± 0.108	0.878 ± 0.058
		0.923 ± 0.002		$6.19 \times 10^{-2} \pm 1.86 \times 10^{-3}$	$1.51 \times 10^{-2} \pm 1.00 \times 10^{-3}$
2000	14	12.261 ± 0.077	0.558 ± 0.045	0.967 ± 0.060	0.214 ± 0.027
		0.916 ± 0.005		$6.91 \times 10^{-2} \pm 4.30 \times 10^{-3}$	$1.52 \times 10^{-2} \pm 1.92 \times 10^{-3}$
4000	3	2.594 ± 0.036	0.175 ± 0.025	0.196 ± 0.026	0.035 ± 0.011
		0.923 ± 0.009		$6.53 \times 10^{-2} \pm 8.56 \times 10^{-3}$	$1.12 \times 10^{-2} \pm 3.80 \times 10^{-3}$

トの最大値で降順にランク付けし，横軸にランク，縦軸にパケットレートをとったグラフである．なお，どちらのグラフも横軸および縦軸は対数目盛で示している．グラフからパケットレートの分布はべき乗則に従っていることが確認できる．

表 2.3 と表 2.4 は それぞれ $T_{\mathrm{D\_max}} = 10$ と $T_{\mathrm{D\_max}} = 20$ のときのサンプリング実験の結果を示している．ここで N_{T} は検出対象フロー数を示している．それぞれの表では 1000 回の独立なサンプリング実験の平均と 95% 信頼区間が示されている．$T_{\mathrm{D\_max}}$ や ϵ の値に限らず，提案手法はほとんどの検出対象フローを $f = 1$ とした理想的な場合よりも早く検出していることがわかる．この現象は， SW 方式とランダムパケットサンプリングを組み合わせた高パケットレートフローの検出においては一般的な現象である [27]．SW におけるレートを考えると，検出対象フローは最初のパケットを生成してからすぐには R にはならず，次第にそのパケットレートが上がっていく．図 2.6 は典型的な検出対象フローの SW におけるパケットレートの時間変化を表している．時刻 $t_1 = 61T_{\mathrm{BW}}$ $(= T_{\mathrm{SW}})$ において初めてパケットレートが R に達するが，それ以前にパケットレートはかなり R に近い状態になっている．このとき検出確率は $1 - \epsilon$ にかなり近い状態になっており，結果的に提案手法は多くの検出対象フローを t_1 以前に検出することになる．

一方で，クラス (iii) やクラス (iv) の検出対象フロー数はかなり少ない．これはパケットレートが閾値 R ちょうどのフローを $1-\epsilon$ 以上の確率で検出するようにパラメータの決定を行っているためである．トレースデータに含まれるような検出対象フローは多くの場合閾値を上回るため，未検出確率は ϵ を下回ることになる．

表 2.5 と表 2.6 は $T_{\mathrm{D\_max}} = 10$ および $T_{\mathrm{D\_max}} = 20$ のときの，誤検出された検出対象外フローのフロー数 N_{WD} および式 (2.11) で定義される誤検出確率を示している．ただし $\epsilon = 0.01, 0.05, 0.1$ である．表 2.5 より誤検出フロー数はかなりあ

表 2.4: 検出対象フローの検出数と検出率 ($T_{\mathrm{D\_max}} = 20$)

$\epsilon = 0.01$					
R	N_{T}	(i) $t_1 > t_2$	(ii) $t_1 = t_2$	(iii) $t_1 < t_2 < \infty$	(iv) $t_2 = \infty$
1000	22	21.712 ± 0.034	0.140 ± 0.023	0.130 ± 0.023	0.018 ± 0.008
		0.993 ± 0.001		$5.91 \times 10^{-3} \pm 1.03 \times 10^{-3}$	$8.18 \times 10^{-4} \pm 3.75 \times 10^{-4}$
2000	7	6.875 ± 0.021	0.071 ± 0.016	0.044 ± 0.013	0.010 ± 0.006
		0.992 ± 0.002		$6.29 \times 10^{-3} \pm 1.82 \times 10^{-3}$	$1.42 \times 10^{-3} \pm 8.81 \times 10^{-4}$
4000	1	0.984 ± 0.008	0.008 ± 0.006	0.008 ± 0.006	0
		0.992 ± 0.006		$8.00 \times 10^{-3} \pm 5.52 \times 10^{-3}$	0
$\epsilon = 0.05$					
R	N_{T}	(i) $t_1 > t_2$	(ii) $t_1 = t_2$	(iii) $t_1 < t_2 < \infty$	(iv) $t_2 = \infty$
1000	22	20.651 ± 0.069	0.519 ± 0.043	0.741 ± 0.053	0.089 ± 0.018
		0.962 ± 0.003		$3.37 \times 10^{-2} \pm 2.42 \times 10^{-3}$	$4.05 \times 10^{-3} \pm 8.22 \times 10^{-4}$
2000	7	6.508 ± 0.042	0.244 ± 0.030	0.208 ± 0.030	0.040 ± 0.012
		0.965 ± 0.005		$2.97 \times 10^{-2} \pm 4.22 \times 10^{-3}$	$5.71 \times 10^{-3} \pm 1.74 \times 10^{-3}$
4000	1	0.922 ± 0.017	0.041 ± 0.012	0.037 ± 0.012	0
		0.963 ± 0.012		$3.70 \times 10^{-2} \pm 1.17 \times 10^{-2}$	0
$\epsilon = 0.1$					
R	N_{T}	(i) $t_1 > t_2$	(ii) $t_1 = t_2$	(iii) $t_1 < t_2 < \infty$	(iv) $t_2 = \infty$
1000	22	19.382 ± 0.094	0.912 ± 0.056	1.515 ± 0.074	0.186 ± 0.025
		0.923 ± 0.004		$6.89 \times 10^{-2} \pm 3.38 \times 10^{-3}$	$8.45 \times 10^{-3} \pm 1.13 \times 10^{-3}$
2000	7	6.067 ± 0.054	0.375 ± 0.036	0.459 ± 0.039	0.099 ± 0.019
		0.920 ± 0.006		$6.56 \times 10^{-2} \pm 5.53 \times 10^{-3}$	$1.41 \times 10^{-2} \pm 2.73 \times 10^{-3}$
4000	1	0.890 ± 0.019	0.056 ± 0.014	0.054 ± 0.014	0
		0.946 ± 0.014		$5.40 \times 10^{-2} \pm 1.40 \times 10^{-2}$	0

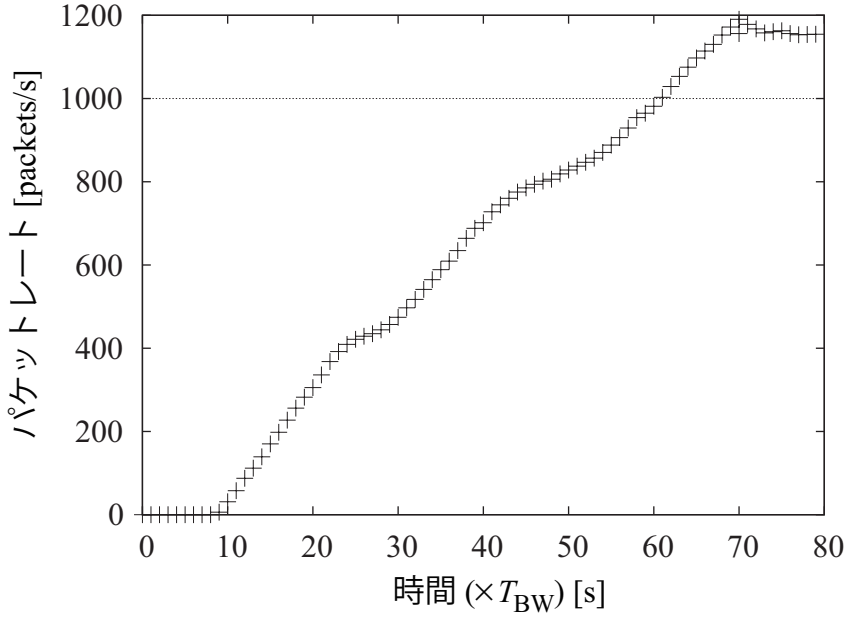

図 2.6: 典型的な検出対象フローのパケットレートの変化 ($k = 61$, $R = 1,000$)

り，ϵ の増加に伴って増えていることがわかる．さらにその傾向は R が小さいときほど顕著であることがわかる．図 2.7 は対象外フローが誤検出されたときの真のパケットレートの累積分布を示している．ただし，$T_{\mathrm{D\_max}} = 10$，$\epsilon = 0.01$ である．図よりかなり低いパケットレートのフローも誤検出されていることがわかる．

検出対象外フローの誤検出は二つの要因によって引き起こされると考えられる．一つはパケットレートの分布である．図 2.4 や図 2.5 で見たように，パケットレートのランキングの分布はべき乗則に従っている．そのため，個々のフローのパケットレートは低くとも，多数のフローが存在することでそのうちのいくつかが結果的に検出されてしまう．文献 [2] でも議論されているように，ランダムパケットサンプリングを用いる限り，これはさけることができない性質である．もう一つの要因はオンラインでの監視を行う際に避けられないものである．パケットレートは低いが長時間存続するようなフローを考えると，ある瞬間の SW における誤検出確率は十分小さいとしても，連続する非常に多くの SW に登場することで結果的にはどこかで誤検出されてしまう．

しかしながらこれらの誤検出は $T_{\mathrm{D\_max}}$ を大きく設定することで軽減することができる．$T_{\mathrm{D\_max}} = 10$ としたときの表 2.5 と，$T_{\mathrm{D\_max}}$ を 2 倍の $T_{\mathrm{D\_max}} = 20$ としたときの表 2.6 を見比べると，$T_{\mathrm{D\_max}}$ を大きくしたときの方が誤検出フロー数 N_{WD} が減っていることがわかる．一般に，$T_{\mathrm{D\_max}}$ を大きくすることは fT_{SW} を大きくすることにつながり，その結果 SW にサンプリングされるパケット数も増えて検出の精度を上げることになる．また，T_{SW} を大きくすることによって長時間存続するフローが出現する SW 数も減るため，先ほどの二つ目の要因に起因する誤検出を軽減することになる．図 2.8 に，$T_{\mathrm{D\_max}} = 20, \epsilon = 0.01$ のときの，検出対象外フローが誤検出されたときの真のパケットレートの累積分布を示す．$T_{\mathrm{D\_max}} = 10$ のときの累積分布である図 2.7 と比べると，低パケットレートフローの誤検出が軽減されているのがわかる．

誤検出フロー数 N_{WD} はサンプルパケット数の閾値 y^* と正の相関を持っていると考えると非常に興味深い．表 2.7 に $T_{\mathrm{D\_max}}, R, \epsilon$ を変えたときの y^* と N_{WD} の関係を示す．今回使用した CAIDA トレースデータだけでなく，別のトレースデータ [29] を使用してもこの関係性は観測された．y^* は $T_{\mathrm{D\_max}}$ の増加関数となることから，$T_{\mathrm{D\_max}}$ を大きくすることで，検出の即応性は犠牲になるが，誤検出を抑えることができると結論づける．

2.4.4 最適パラメータの有効性

最後に提案手法で決定される最適パラメータの有効性について確認する．まず，$T_{\mathrm{D\_max}} = 20$, $\epsilon = 0.05$ とし，制御パラメータは T_{SW} と k を表 2.2 に示した $T_{\mathrm{SW}} = 19.5506$, $k = 87$ で固定する．このとき，f を有効な範囲 $(f \leq f^*)$ で変化させたときの検出精度を確認する．なお，y^* は式 (2.1) に従ってそのつど決定するため，未検出確率は常に $\epsilon = 0.05$ 以下となる．

最適パラメータの有効性を確認するための指標として，検出された全てのフロー

表 2.5: 誤検出されたフロー数と FPR ($T_{\mathrm{D\_max}} = 10$)

$\epsilon = 0.01$		
R	検出対象外フロー数	N_{WD} (FPR)
1000	13602956	2279.53 ± 5.41 $(1.68 \times 10^{-4} \pm 3.98 \times 10^{-7})$
2000	13603000	84.48 ± 1.05 $(6.21 \times 10^{-6} \pm 7.73 \times 10^{-8})$
4000	13603011	7.59 ± 0.39 $(5.58 \times 10^{-7} \pm 2.90 \times 10^{-8})$
$\epsilon = 0.05$		
R	検出対象外フロー数	N_{WD} (FPR)
1000	13602956	526.91 ± 3.35 $(3.87 \times 10^{-5} \pm 2.46 \times 10^{-7})$
2000	13603000	40.50 ± 0.66 $(2.98 \times 10^{-6} \pm 4.88 \times 10^{-8})$
4000	13603011	4.71 ± 0.23 $(3.46 \times 10^{-7} \pm 1.72 \times 10^{-8})$
$\epsilon = 0.1$		
R	検出対象外フロー数	N_{WD} (FPR)
1000	13602956	296.38 ± 2.60 $(2.18 \times 10^{-5} \pm 1.91 \times 10^{-7})$
2000	13603000	29.32 ± 0.59 $(2.16 \times 10^{-6} \pm 4.35 \times 10^{-8})$
4000	13603011	3.30 ± 0.24 $(2.43 \times 10^{-7} \pm 1.78 \times 10^{-8})$

表 2.6: 誤検出されたフロー数と FPR ($T_{\mathrm{D\_max}} = 20$)

$\epsilon = 0.01$		
R	検出対象外フロー数	N_{WD} (FPR)
1000	13602992	144.43 ± 1.51 $(1.06 \times 10^{-5} \pm 1.11 \times 10^{-7})$
2000	13603007	10.86 ± 0.37 $(7.98 \times 10^{-7} \pm 2.69 \times 10^{-8})$
4000	13603013	1.52 ± 0.18 $(1.12 \times 10^{-7} \pm 1.29 \times 10^{-8})$
$\epsilon = 0.05$		
R	検出対象外フロー数	N_{WD} (FPR)
1000	13602992	72.54 ± 1.05 $(5.33 \times 10^{-6} \pm 7.70 \times 10^{-8})$
2000	13603007	6.25 ± 0.30 $(4.59 \times 10^{-7} \pm 2.23 \times 10^{-8})$
4000	13603013	0.67 ± 0.13 $(4.93 \times 10^{-8} \pm 9.40 \times 10^{-9})$
$\epsilon = 0.1$		
R	検出対象外フロー数	N_{WD} (FPR)
1000	13602992	47.35 ± 0.93 $(3.48 \times 10^{-6} \pm 6.84 \times 10^{-8})$
2000	13603007	4.11 ± 0.26 $(3.02 \times 10^{-7} \pm 1.89 \times 10^{-8})$
4000	13603013	0.45 ± 0.12 $(3.31 \times 10^{-8} \pm 8.54 \times 10^{-9})$

表 2.7: サンプルパケット数の閾値 y^* と誤検出フロー数 N_{WD} の関係

R	$T_{\mathrm{D\_max}} = 10$		
	$\epsilon = 0.01$	$\epsilon = 0.05$	$\epsilon = 0.1$
1000	3 / 2279.53	5 / 526.91	6 / 296.38
2000	9 / 84.48	12 / 40.50	13 / 29.32
4000	24 / 7.59	28 / 4.71	30 / 3.30
R	$T_{\mathrm{D\_max}} = 20$		
	$\epsilon = 0.01$	$\epsilon = 0.05$	$\epsilon = 0.1$
1000	10 / 144.43	12 / 72.54	14 / 47.35
2000	25 / 10.86	28 / 6.25	30 / 4.11
4000	57 / 1.52	62 / 0.67	65 / 0.45

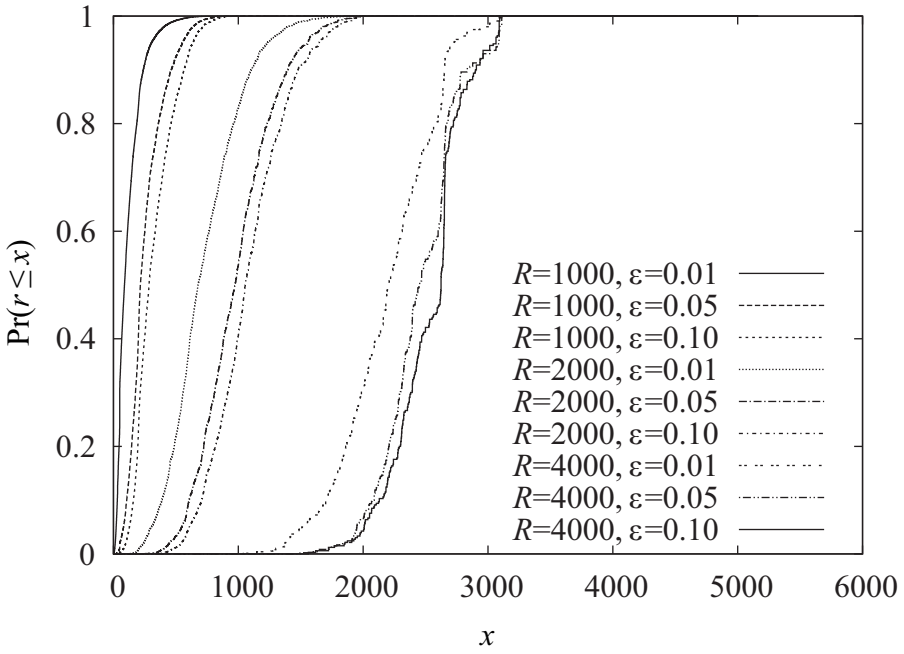

図 2.7: 誤検出されたフローのパケットレートの累積分布 ($T_{\mathrm{D\_max}} = 10$)

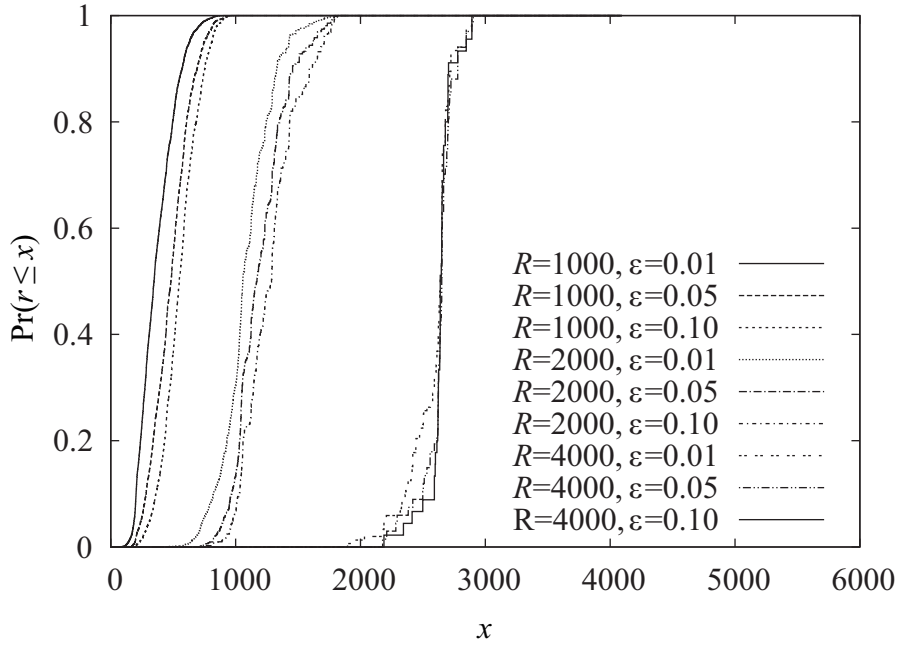

図 2.8: 誤検出されたフローのパケットレートの累積分布 ($T_{\mathrm{D\_max}} = 20$)

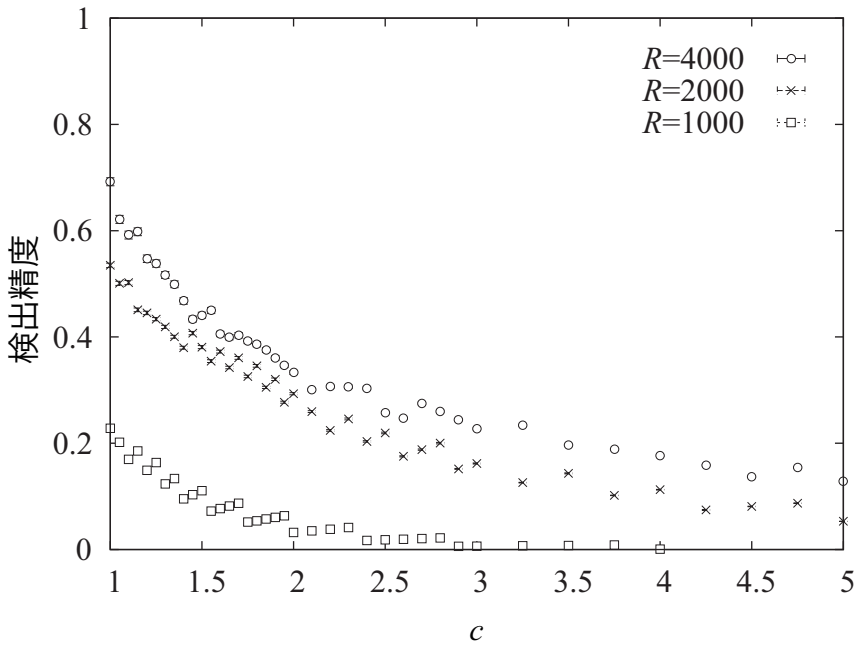

図 2.9: 検出確率の平均と 95%信頼区間 ($f = f^*/c$, $T_{\mathrm{D\_max}} = 20$, $\epsilon = 0.05$)

のうち検出対象フローであった割合を検出精度として定義する．図 2.9 は $f = f^*/c$ として c を介して f を変化させたときの平均の検出精度とその 95% 信頼区間を示している．なお，$c = 1$ のときが最適パラメータである．図より c を大きくすると検出精度は悪化していおり，パラメータを最適化することは非常に重要であると結論づける．

2.5 まとめ

本章では，ランダムパケットサンプリングと SW 方式を組み合わせた高パケットレートフローのオンライン検出手法におけるパラメータ決定手法について提案した．検出手法自体は既存の技術を組み合わせたものであり，その技術が与えられれば誰もが行き着く一般的な手法と言える．しかしながら，そこで使用するパラメータは，各パラメータ単体では変化させたときの影響は簡単に推測がついても，パラメータ全体を同時に決定する際にどのような方針で決定すればよいかは議論されていなかった．本章では，二項分布のポアソン近似を用いて，サンプリングレートと SW の長さの積を最大化することが誤検出確率を最小化する，すなわち標本からの推定精度をもっとも高めることを示した．そして，オンライン検出に関する制約条件の下で上記の積を最大化する問題を解くことでパラメータを決定する手法を提案した．なお，提案したパラメータ決定手法は，パケットサンプリングと時間ベースの計測を組み合わせた手法に対して幅広く適用可能である．

第3章

持続的高パケットレートフロー検出手法とそのパラメータ決定手法

3.1 まえがき

本章では，持続的高パケットレートフローのオンライン検出を考える．検出対象フローは大まかに以下のように特徴づけられる．

- ある固定長の計測期間に渡ってアクティブである (パケットが転送されている)
- その固定長の計測期間に含まれる任意の固定長部分区間におけるパケットレートがあらかじめ与えられる閾値 R を超えている

フローの情報は 2 章と同様に SW 方式とランダムパケットサンプリングを用いて収集する．この枠組みにおいて，検出対象の持続的高パケットレートフローを見逃してしまう未検出確率を保証するサンプルパケット数の閾値の決定方法を提案する．また，検出対象外フローを誤って検出してしまう誤検出確率を最小化するようなパラメータ決定問題を定式化し，それを解くことで適切なパラメータを得られるようにする．そして，トレースデータを用いて提案する検出手法およびパラメータ決定手法の性能評価を行う．

3.2 持続的高パケットレートフローの検出手法

3.2.1 持続的高パケットレートフローとスライディングウィンドウ方式

持続的高パケットレートフローを SW 方式とランダムパケットサンプリングを用いてオンライン検出することを考える．3.1 節で述べたように，持続的高パケットレートフローはそのフローがアクティブである期間，パケットレートを計算する任意の部分区間の時間幅，そして部分区間における最低のパケットレートの三つで特徴づけられる．このような持続的高パケットレートフローをスライディングウィンドウ (SW) 方式と組み合わせて定義する．具体的には，まず 長さが T_{HW}

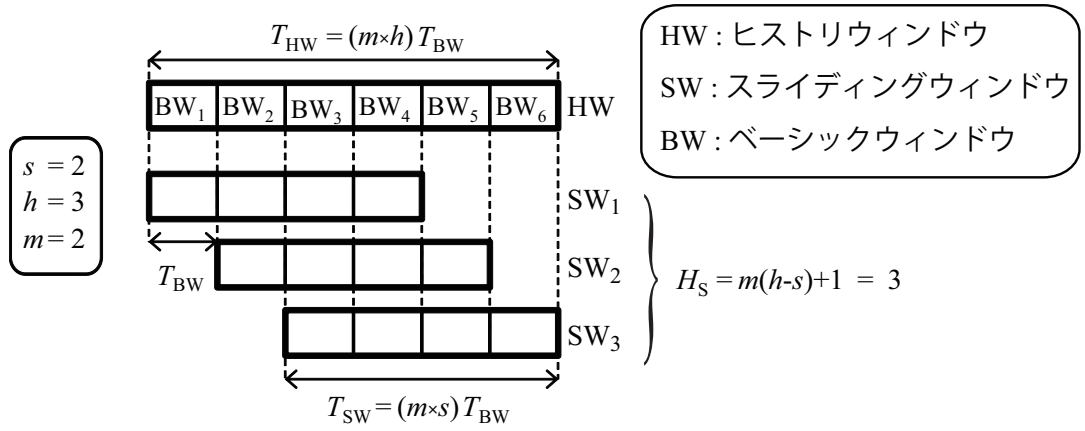

図 3.1: BW, SW, HW の例

のヒストリウィンドウ (HW) を定義する．この長さ T_{HW} は持続的高パケットレートフローのアクティブな時間 (持続時間) に相当する．次に，SW の長さ T_{SW} をパケットレートを計算する部分区間の長さと対応させる．このとき，$T_{\mathrm{SW}}/T_{\mathrm{HW}}$ が自然数 s と h を用いて，次式を満たすことを仮定する．

$$\frac{T_{\mathrm{SW}}}{T_{\mathrm{HW}}} = \frac{s}{h} < 1$$

さらに自然数 m を用いて，SW は $(m \times s)$ 個の BW に分割されるものとする．これ以降は自然数 m を分割因子と呼ぶ．以上より，次式の関係を得る．

$$T_{\mathrm{SW}} = (m \times s)T_{\mathrm{BW}}, \qquad T_{\mathrm{HW}} = (m \times h)T_{\mathrm{BW}}$$

図 3.1 は $(s, h, m) = (2, 3, 2)$ のときの BW，SW，HW の関係を示している．このとき，$T_{\mathrm{SW}} = 4T_{\mathrm{BW}}$ および $T_{\mathrm{HW}} = 6T_{\mathrm{BW}}$ となっていることに注意する．また，HW の中に含まれる SW の数 H_S は次式で与えられる．

$$H_S = m(h - s) + 1 \tag{3.1}$$

通常，3.1 節で述べた特徴を持つ持続的高パケットレートフローは，R 以上のパケットレートを，連続する H_S 個の SW $(T_{\mathrm{SW}} = (m \times s)T_{\mathrm{BW}})$ のいずれにおいても維持することになる．

上記の設定において，サンプリングレートが f $(0 < f \leq 1)$ のパケットサンプリングを用いて持続的高パケットレートフローの検出を試みる．ここで，パケットレートの閾値 R，HW の長さ T_{HW} と SW の長さ T_{SW} (すなわちパラメータ h と s も) はあらかじめ与えられるパラメータとする．一方で，T_{BW} の長さを決める分割因子 m とサンプリングレート f は制御可能なパラメータとする．3.3 節，および 3.4 節において，五つのパラメータ R, T_{HW}, T_{SW}, m, f が与えられた下での持続的高パケットレートフローの検出手法について議論している．また，3.5 節においてパラメータ m と f の決め方について議論する．

Step 1: T_{SW} [s] の間，転送されるパケットを独立かつランダムに確率 f でサンプリングし，パケットデータをフローデータに集約することで新しい BW を作成する．

Step 2: 現在の SW を，最も古い BW の廃棄と新しい BW の追加により更新する．もし計測開始直後であるため，現在の SW に含まれる BW の数が $m \times s$ に満たない場合は新しい BW の追加のみを行う．

Step 3: 更新された SW を検査し，サンプルパケット数が θ^* 以上のフローを特定する．

Step 4: 現在の HW を，最も古い SW の廃棄と新しい SW の追加により更新する．もし計測開始直後であるため，現在の HW に含まれる SW の数が $m \times h$ に満たない場合は新しい SW の追加のみを行う．

Step 5: 更新された HW において，そこに含まれる全ての SW でサンプルパケット数が θ^* 以上となったフローを全て検出し，Step 1 に戻る．

図 3.2: 持続的高パケットレートフローの検出手順

3.3 ランダムパケットサンプリングを用いた検出手法

この節では，五つのパラメータ $R, T_{\mathrm{HW}}, T_{\mathrm{SW}}, m, f$ が与えられた下での持続的高パケットレートフローの検出手法について議論する．Z を単一の SW に含まれる任意のフローのパケット数とする．また，W を Z から SW においてサンプリングされたパケット数とする．このとき，二項分布を用いて以下の確率が得られる．

$$\Pr(W \geq w \mid Z = z) = 1 - \sum_{i=0}^{w-1} \binom{z}{i} f^i (1-f)^{z-i}, \qquad w = 0, 1, \ldots, z$$

サンプリングされたパケット数に基づいて対象フローを検出することを考える．なお，持続的高パケットレートフローは連続する H_S 個の SW のいずれにおいても，少なくとも $\lceil RT_{\mathrm{SW}} \rceil$ 個のパケットを保持する．ここで，$z^* = \lceil RT_{\mathrm{SW}} \rceil$ を定義しておく．未検出確率 $\Pr(W < w \mid Z = z)$ が小さいときは $w - 1 \ll fz$ となり，また，もし $w - 1 \ll fz$ であれば $\Pr(W < w \mid Z = z)$ は z の減少関数となる．それゆえ，あるフローから w 個のパケットがサンプリングされた場合，そのフローが SW の中に z^* 以上のパケットを含んでいたかどうかを，未検出確率 $\Pr(W < w \mid Z \geq z^*)$ の上限 $\Pr(W < w \mid Z = z^*)$ を用いて判断することができる．

そこで，ある閾値 θ^* をあらかじめ設定しておき，連続する H_S 個の SW のいずれにおいても閾値 θ^* 以上のパケットを持続的高パケットレートフローとみなして検出する．検出の手順を図 3.2 にまとめた．閾値 θ^* の設定方法は次節で行う．

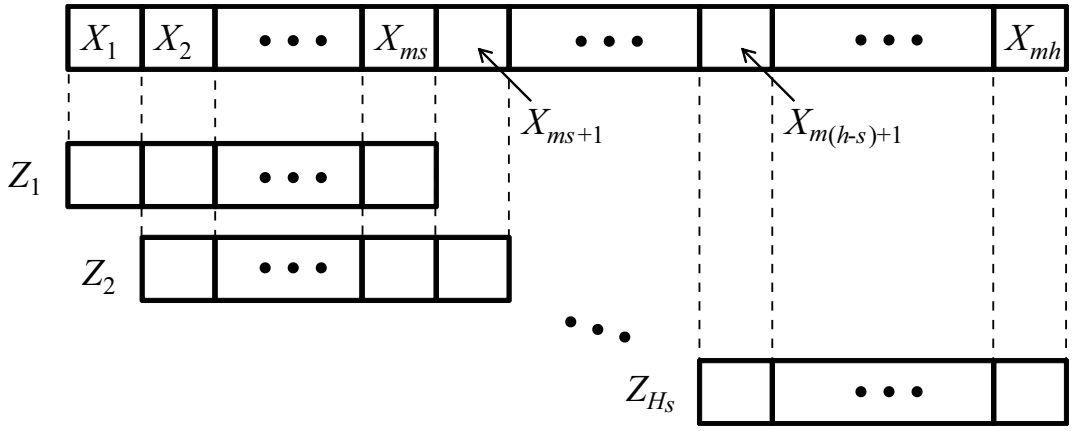

図 3.3: HW, SW, BW のパケット数の関係

3.4　閾値の設定方法

本節では五つのパラメータが与えられたとき，どのようにサンプルパケット数の閾値 θ^* を決定するかを議論する．なお，五つのパラメータは，検出対象のパケットレートの閾値 R [packets/s]，HW の長さ T_{HW} [s]，SW の長さ T_{SW} [s]，T_{BW} [s] の長さを決定する分割因子 m, そしてサンプリングレート f である．θ^* を決定する際の方針として，検出対象である持続的高パケットレートフローが $1-\epsilon$ 以上の確率で検出される (未検出確率が ϵ 以下となる) ようにする．そのための手段として，HW 内の H_S 個全ての SW におけるパケット数がちょうど z^* であるようなフローを考え，そのようなフローの中で最も未検出確率が高くなるフローを特定する．最も検出が困難な検出対象フローを $1-\epsilon$ 以上の確率で検出できるように θ^* を設定すれば，全ての検出対象フローが検出されることに注意する．本節の以降の議論は数学的な要素を多く含むため，興味がなければ θ^* の設定手順を図 3.4 にまとめてあるので読み飛ばして構わない．

任意のフローに対して，X_i $(i = 1, 2, \ldots, mh)$ を HW 内の i 番目の BW におけるパケット数として定義する．同様に，Z_i $(i = 1, 2, \ldots, H_S)$ を HW 内の i 番目の SW におけるパケット数として定義する．このとき，

$$Z_i = X_i + X_{i+1} + \cdots + X_{i+ms-1}, \qquad i = 1, 2, \ldots, H_S$$

となっており，フローは H_S 次元の非負整数ベクトル $\boldsymbol{Z} = (Z_1, Z_2, \ldots, Z_{H_S})$ として特徴付けられる．Z_i と X_i の関係を図 3.3 に示す．

ここで二つのフロー，フロー 1 とフロー 2 について考える．フロー n $(n = 1, 2)$ は H_S 次元の非負整数ベクトル $\boldsymbol{Z}_n$ で表す．もし $\boldsymbol{Z}_1 \leq \boldsymbol{Z}_2$ が成り立つとき，フロー 1 の検出確率はフロー 2 の検出確率以下になることは容易に見て取れる．そこで，持続的高パケットレートフローの中でも最少のパケット数で構成されるフロー，すなわち HW 内の全ての H_S 個の SW においてパケット数が z^* のフローを考える．

$$Z_i = X_i + X_{i+1} + \cdots + X_{i+ms-1} = z^*, \qquad i = 1, 2, \ldots, H_S \tag{3.2}$$

式 (3.2) は X_i $(i = 1, 2, \ldots, mh)$ が周期 ms の周期性を持つことを示している．すなわち，

$$X_i = X_{i+ms}, \qquad i = 1, 2, \ldots, H_S - 1$$

となる．したがって，式 (3.2) 成り立つときは，$\{X_i;\ i = 1, 2, \ldots, ms\}$ が HW 全体に渡るパケット数の分布を決定する．以降，フローはベクトル $\boldsymbol{X} = (X_1, X_2, \ldots, X_{ms})$ によって特徴付けられる．併せて $\boldsymbol{x}$ を最初の ms 個の BW におけるパケット数を表す非負整数 $1 \times ms$ ベクトルとして定義する．

$$\boldsymbol{x} \in \mathcal{X} = \{(x_1\ x_2\ \ldots\ x_{ms});\ x_1 + x_2 + \cdots + x_{ms} = z^*\}$$

任意のフローに対して，Y_i $(i = 1, 2, \ldots, mh)$ を HW 内の i 番目の BW におけるサンプルパケット数として，また W_i $(i = 1, 2, \ldots, H_S)$ を i 番目の SW におけるサンプルパケット数としてそれぞれ定義する．このとき，

$$W_i = Y_i + Y_{i+1} + \cdots + Y_{i+ms-1}, \qquad i = 1, 2, \ldots, H_S \tag{3.3}$$

の関係が成り立っている．閾値 w が与えられると，式 (3.2) を満たす持続的高パケットレートフローの検出確率 $P(w \mid \boldsymbol{x})$ $(\boldsymbol{x} \in \mathcal{X})$ は以下のように定義される．

$$P(w \mid \boldsymbol{x}) = \Pr\left(W_1 \geq w, W_2 \geq w, \ldots, W_{H_S} \geq w \mid \boldsymbol{X} = \boldsymbol{x}\right) \tag{3.4}$$

HW 内の i 番目の SW に注目したとき，その SW において閾値を超えるパケットがサンプリングされる確率は次のようになる．

$$\begin{aligned} p(w \mid z^*) &= \Pr\left(W_i \geq w \mid \boldsymbol{X} = \boldsymbol{x}\right) \\ &= \Pr\left(W_i \geq w \mid X_i + X_{i+1} + \cdots + X_{i+ms-1} = z^*\right) \\ &= 1 - \sum_{j=0}^{w-1} \binom{z^*}{j} f^j (1-f)^{z^*-j} \end{aligned} \tag{3.5}$$

ここで，$P(w \mid \boldsymbol{x}) \neq p^{H_S}(w \mid z^*)$ となることに注意する．これは W_i と W_j $(1 \leq j - i \leq ms - 1)$ は Y_n $(n = j, j+1, \ldots, i+ms-1)$ を共有することで独立とはならないためである．それゆえ，検出確率 $P(w \mid \boldsymbol{x})$ は $\boldsymbol{x} \in \mathcal{X}$ の分布に依存し，式 (3.4) に基づいて $P(w \mid \boldsymbol{x})$ を計算することは困難である．さらに，式 (3.2) を満たす持続的高パケットレートフローの分布の異なり数は ${}_{ms}H_{z^*} = {}_{ms+z^*-1}C_{z^*}$ で与えられ，その分布の異なり方も様々である．

検出確率を最小にするフローを見つけるには上記のような問題があるが，この問題に対して，まず式 (3.2) を満たす持続的高パケットレートフローの検出確率の上界と下界を示す．

$$p^{H_S}(w \mid z^*) \leq P(w \mid \boldsymbol{x}) \leq p^{\lceil H_S/(ms) \rceil}(w \mid z^*), \qquad \boldsymbol{x} \in \mathcal{X} \tag{3.6}$$

式 (3.6) の導出は付録 C で行っている．

なお，ここで示した上界は上限となっており，例えば $\boldsymbol{x} = (0, 0, \ldots, 0, z^*)$ はその上限を与える分布である．一つの BW にパケットを集中させた分布が上限を与えるという事実は，式 (3.2) を満たすフローの中で，パケットを一様に分布させたフローが検出確率が低くなるのではないかということを予期させる．そこで次のように z^* 個のパケットをなるべく一様に配置した分布 $\boldsymbol{x} = \boldsymbol{x}^* = (x_1^*, x_2^*, \ldots, x_{ms}^*)$ を考える．それぞれの要素の値は，$z_1 = z^* \bmod ms$ を用いて以下で与える．

$$x_i^* = \begin{cases} \left\lceil \dfrac{z^*}{ms} \right\rceil, & i = 1, 2, \ldots, z_1 \\ \left\lfloor \dfrac{z^*}{ms} \right\rfloor, & i = z_1 + 1, z_1 + 2, \ldots, ms \end{cases} \tag{3.7}$$

式 (3.7) の理論的根拠は付録 D に示した．付録 D で示しているように，h/s が大きいときは $\boldsymbol{x}^*$ で特徴づけられたフローは最小の検出確率となるフローをかなりの精度で近似する．あとはサンプルパケット数の閾値 θ^* をどのように決めるかを考えればよい．閾値 θ^* は本来次式で定義される．

$$\theta^* = \max\{w;\ P(w \mid \boldsymbol{x}) \geq 1 - \epsilon \ \text{ for all } \ \boldsymbol{x} \in \mathcal{X}\}$$

この閾値 θ^* をこれまでの議論を踏まえて次式で近似する．

$$\theta^* \approx \max\{w;\ P(w \mid \boldsymbol{x}^*) \geq 1 - \epsilon\} \tag{3.8}$$

しかしながら，残念なことに $\boldsymbol{x}^*$ が与えられたとしても，$P(w \mid \boldsymbol{x}^*)$ を数値計算では 解くことは s および h が小さい場合を除いて計算量の面で非常に難しい．そこで，モンテカルロシミュレーション実験によって $P(w \mid \boldsymbol{x}^*)$ の値を求めることにする．$\boldsymbol{x}^* \in \mathcal{X}$ であるため，式 (3.6) より次の関係が成り立つ．

$$p^{H_S}(w \mid z^*) \leq P(w \mid \boldsymbol{x}^*) \leq p^{\lceil H_S/(ms) \rceil}(w \mid z^*)$$

さらに，$p(w \mid z^*)$ と $P(w \mid \boldsymbol{x}^*)$ は w の減少関数であるため，

$$\theta^- \leq \theta^* \leq \theta^+ \tag{3.9}$$

となる．ここで，

$$\theta^- = \max\{w;\ p^{H_S}(w \mid z^*) \geq 1 - \epsilon\}, \qquad \theta^+ = \max\{w;\ p^{\lceil H_S/(ms) \rceil}(w \mid z^*) \geq 1 - \epsilon\}$$

である．この θ^- と θ^+ は簡単に計算できることに注意する．したがって，モンテカルロシミュレーション実験により θ^* を探す際には，探索範囲を式 (3.9) で与えられる範囲に制限することができる．

一般に ϵ は小さく設定するため，検出対象外のフローであっても，各 SW におけるパケット数が z^* に近いようなフローは非常に高い確率で検出してしまう．そこで，検出されたフローを $N+1$ 個のクラスに多段の閾値を用いて分類することを考える．使用する多段の閾値 θ_i $(i = 1, 2, \ldots, N)$ は自然数をとり，$\theta^* < \theta_1 < \theta_2 < \cdots < \theta_N$

を満たすように設定する．ここで $\theta_0 = \theta^*$ としておく．もし全 SW の中での最少のサンプルパケット数 $W_{\min}$ が $\theta_{i-1} \leq W_{\min} < \theta_i$ $(i = 1, 2, \ldots, N)$ を満たすとき，そのフローは i 番目のクラスに分類し，$W_{\min} \geq \theta_N$ となるときは $N+1$ 番目のクラスに分類する．通常，より高いクラスに分類された検出フローはより高いパケットレートを有することを意味する．

あるフローが i 番目 $(i = 1, 2, \ldots, N+1)$ のクラスで検出されたとき，そのフローは誤検出確率の観点から評価をすることができる．検出対象外のフローは少なくとも一つの SW でパケット数が z^* 未満となっている．そのため，

$$\begin{aligned}
\mathrm{FPR} &= \Pr(W_i \geq w \ (i = 1, 2, \ldots, H_S) \mid Z_j < z^* \text{ for some } j \ (j = 1, 2, \ldots, H_S)) \\
&\leq \Pr(W_j \geq w \mid Z_j = z^* - 1) \\
&\qquad \cdot \Pr(W_i \geq w \ (i \neq j, i = 1, 2, \ldots, H_S) \mid W_j \geq w, Z_j = z^* - 1) \\
&\leq \Pr(W_j \geq w \mid Z_j = z^* - 1) \\
&= p(w \mid z^* - 1)
\end{aligned}$$

となる．したがって，誤検出確率の上限 $\overline{\mathrm{FPR}_i}$ $(i = 1, 2, \ldots, N+1)$ が次式で与えられる．

$$\overline{\mathrm{FPR}_i} = p(\theta_{i-1} \mid z^* - 1), \tag{3.10}$$

ここで，$p(.)$ は式 (3.5) で与えられる．なお，$i = 1$ すなわち θ^* で検出されたフローの誤検出確率の上限は $\overline{\mathrm{FPR}_1} \approx 1$ となっていることに注意する．これは，クラス 1 で検出されたフローは検出対象外フローである可能性が十分高いことを示している．サンプルパケット数の閾値 θ^* および多段の閾値 θ_i の設定の仕方を図 3.4 にまとめてある．

3.5 パラメータ決定手法

前節では，パケットレートの閾値 R [packets/s]，HW の長さ T_{HW}，SW の長さ T_{SW}，分割因子 m，サンプリングレート f のパラメータによって特徴づけられる任意のシステムにおいて，サンプルパケット数の閾値 θ^* を調整することによって検出確率 $1 - \epsilon$ 以上 (未検出確率 ϵ 未満) が達成されることを示した．

本節では，誤検出確率に注目する．一般的に，ランダムパケットサンプリングを用いた検出手法はその大きな誤検出確率が宿命となっている．その理由としては，パケットレートが閾値より僅かに低いだけのフローというのはかなり高い確率で検出されてしまうためである．R，T_{HW}，T_{SW} は検出対象のフローを定義する，あらかじめ与えられるパラメータであるため，残りの二つのパラメータ m と f を適切に制御して誤検出確率を下げることを考える．なお，自然数 m に関して，

$$T_{\mathrm{BW}} = \frac{T_{\mathrm{HW}}}{mh} = \frac{T_{\mathrm{SW}}}{ms}$$

が自然数 s と h とともに成立することに注意する．

Step 1: j 番目 $(j = 1, 2, \ldots, mh)$ の BW に x_i^* 個 $i = (j - 1 \bmod ms) + 1$ のパケットを保持しているフローを想定する．ただし，x_i^* $(i = 1, 2, \ldots, ms)$ は式 (3.7) で与えられる．

Step 2: 確率 f でランダムパケットサンプリングのモンテカルロシミュレーション実験を繰り返し行い，以下の式を満たすような θ^* を特定する．

$$\theta^* = \max\{w;\ \Pr(W_1 \geq w, W_2 \geq w, \ldots, W_{H_S} \geq w) \geq 1 - \epsilon\},$$

ただし，W_i $(i = 1, 2, \ldots, H_S)$ は i 番目の SW におけるサンプルパケット数である．

Step 3: 検出されたフローをサンプルパケット数に応じてクラス分けるために多段の閾値 θ_i $(i = 1, 2, \ldots, N)$ を適当な自然数 N に対して用意する．このとき，$\theta^* < \theta_1 < \theta_2 < \cdots < \theta_N$ としておく．また，誤検出確率の上限 $\overline{\mathrm{FPR}}_i$ $(i = 1, 2, \ldots, N + 1)$ を式 (3.10) により計算する．ただし，$\theta_0 = \theta^*$ である．多段の閾値を用いた検出の様子は 3.6 節で確認できる．

図 3.4: サンプルパケット数の閾値 θ^* の設定手順

2 章で議論したパラメータデザインを持続的高パケットレートフローの検出にも適用する．2 章では fT_{SW} を最大化することが誤検出確率を最小にすることを述べた．本章の場合は fT_{SW} もしくは fT_{HW} が最大化の目的関数に相当する．しかしながら本節では T_{SW} も T_{HW} もどちらも所与のパラメータとして扱っている．すなわち，f の最大化をすることが誤検出確率を最小にすることを意味している．2 章と同様に，持続的高パケットレートフローの検出手法がオンラインのアルゴリズムとして機能する，すなわち新しい BW が取得される前に現在の SW および HW の解析を終わらせることを制約条件とする．その他，解析時間の見積りや導入するパラメータ等は 2 章を踏襲する．

また，2 章では最大許容検出遅延に関して制約条件を設けていたが，本章でも同様に制約条件として加えることは可能である．その制約条件は，

$$T_{\mathrm{HW}} + T_{\mathrm{BW}} + G(fC_{\max}T_{\mathrm{BW}}) \leq T_{\mathrm{D\_max}}$$

となる．オンラインのアルゴリズムとして機能することと，検出対象フローが最大許容検出遅延以内に検出されることを制約条件として，誤検出確率を最小にする最適化問題を次のように考える．

$$\begin{aligned}
P^*:\ \text{最大化} \quad & f, \\
\text{条件} \quad & f > 0 \\
& m \text{ は自然数} \\
& G(fC_{\max}T_{\mathrm{BW}}) \leq T_{\mathrm{BW}} && (3.11) \\
& T_{\mathrm{HW}} + T_{\mathrm{BW}} + G(fC_{\max}T_{\mathrm{BW}}) \leq T_{\mathrm{D_max}} && (3.12)
\end{aligned}$$

ただし，$T_{\mathrm{BW}} = T_{\mathrm{HW}}/(mh)$ である．

$\mathcal{M}$ を次のように定義する．

$$\mathcal{M} = \{m_1, m_1 + 1, \ldots, m_2\} \tag{3.13}$$

ε^* は十分小さな正の値とすると，

$$m_1 = \left\lceil \frac{T_{\mathrm{HW}}}{h\{T_{\mathrm{D\_max}} - T_{\mathrm{HW}} - G(0)\}} - \varepsilon^* \right\rceil, \qquad m_2 = \left\lfloor \frac{T_{\mathrm{HW}}}{hG(0)} - \varepsilon^* \right\rfloor$$

となる．付録 E より，もし，

$$G(0) < \min\left(\frac{T_{\mathrm{HW}}}{h}, \frac{T_{\mathrm{D\_max}} - T_{\mathrm{HW}}}{2}\right) \tag{3.14}$$

が成り立つのであれば，固定された $m \in \mathcal{M}$ に対してこの問題は次の条件のときに実行可能となる．その条件とは，式 (3.11) か式 (3.12) の条件のいずれかが有効制約になっており，分割因子の最適解 m^* とサンプリングレートの最適解 f^* が次のように見つかるときである．

$$m^* = \arg\max_{m \in \mathcal{M}} \; mG^{-1}(u(m)), \tag{3.15}$$

$$f^* = \frac{m^* h}{C_{\max} T_{\mathrm{HW}}} \cdot G^{-1}(u(m^*)) \tag{3.16}$$

ここで，$u(m)$ は次式で与えられる．

$$u(m) = \min\left(\frac{T_{\mathrm{HW}}}{hm}, T_{\mathrm{D\_max}} - T_{\mathrm{HW}} - \frac{T_{\mathrm{HW}}}{hm}\right), \qquad m \in \mathcal{M} \tag{3.17}$$

注意 1. 理想としては，持続的高パケットレートフローは *3.1* 節で述べたように定義されるほうがよい．すなわち，長さ T_{SW} の任意の部分区間でパケットレートが R 以上となるフローである．そういったフローを本章では *SW* 方式と組み合わせて近似的に扱っている．このとき，分割因子 m を大きくすることで理想と近似の乖離を小さくできることに注意したい．もちろんこちらを制約条件に加えることもでき，最適解 $(f^*(m^*), m^*)$ を見つける際，$\mathcal{M}$ の代わりに空でない $\mathcal{M}$ の部分集合

$$\mathcal{M}_{\mathrm{sub}} = \{m_{\min}, m_{\min} + 1, \ldots, m_2\},$$

を用意すればよい．なお，$m_{\min} \geq m_1 + 1$ である．同じ理由で，もし式 *(3.15)* で定義される m^* が一意に定まらなければ，その中で最も大きな値を選べばよい．

3.6 数値実験

この節では 2 種類の実トレースデータを使った数値実験の結果を示す．まずトレースデータと評価指標について情報を示し，数値実験の結果を示すとともに議論を行う．

表 3.1: トレースデータの情報

トレースデータ	計測期間 [s]	総パケット数	総フロー数
バックボーン	3,600	410,707,946	23,779,436
バックスキャッタ	3,600	4,843,696	215,433

3.6.1　トレースデータと評価指標

数値実験を行うにあたり，CAIDA [5] が公開している 2 種類の 1 時間のトレースデータを用いる．一つは 2011 年 7 月 21 日にシアトルからシカゴに向けて送られたバックボーンリンクのトラヒック [7] で，もう一つは 2008 年 2 月 20 日に計測されたバックスキャッタのトラヒックを含むトレースデータ [8] である．前者をバックボーントレース，後者をバックスキャッタトレースと呼ぶことにする．バックスキャッタのトラヒックは一般に DoS 攻撃を受けた標的ホストから発信されるトラヒックを指す．SYN Flood 攻撃を例にとると，DoS 攻撃を行うホストは自身の IP アドレスを偽って標的ホストに SYN パケットを送りつける．受けた標的ホストは SYN/ACK パケットを送り返す．このとき相手の IP アドレスは偽られているためまったく別のホスト，あるいは存在しない IP アドレスに向けて送られる．バックスキャッタトラヒックが計測されているということは同じ送信元 IP アドレスから複数の宛先 IP アドレスに向けてパケットが大量に送られることになる．

表 3.1 にトレースデータの情報をまとめた．バックボーントレースでは基本的な 5 つ組でフローを定義し，バックスキャッタトレースでは，バックスキャッタトラヒックをとらえるために送信元 IP アドレスのみでフローを定義する．

提案手法を評価するため，検出成功確率 (True Positive Ratio: TPR) と誤検出確率 (False Positive Ratio: FPR) の二つの指標を考える．それぞれ次のように定義する．まず，$i_{\max}$ を時間長 $T_{\mathrm{M}} = 3,600$ [s] の計測期間にわたる，HW の総数 (HW の更新回数と同じ) として次式のように定義する．

$$i_{\max} = \lfloor T_{\mathrm{M}}/T_{\mathrm{BW}} \rfloor - T_{\mathrm{SW}}/T_{\mathrm{BW}} + 1$$

同様に，$\mathcal{S}(i)$ $(i = 1, 2, \ldots, i_{\max})$ を i 番目の HW においてアクティブであったフローの総数として定義する．このとき TPR および FPR はそれぞれ次のように定義する．

$$\mathrm{TPR} = \frac{\displaystyle\sum_{i=1}^{i_{\max}} \sum_{j \in \mathcal{S}(i)} t(i,j) d(i,j)}{\displaystyle\sum_{i=1}^{i_{\max}} \sum_{j \in \mathcal{S}(i)} t(i,j)}, \qquad \mathrm{FPR} = \frac{\displaystyle\sum_{i=1}^{i_{\max}} \sum_{j \in \mathcal{S}(i)} \{1 - t(i,j)\} d(i,j)}{\displaystyle\sum_{i=1}^{i_{\max}} \sum_{j \in \mathcal{S}(i)} \{1 - t(i,j)\}}$$

ただし，

$$t(i,j) \;=\; \begin{cases} 1, & \text{フロー } j \text{ が } i \text{ 番目の HW において検出対象フローである} \\ 0, & \text{その他} \end{cases}$$

表 3.2: 実験に用いた入力パラメータ

パラメータ	バックボーン	バックスキャッタ
R [packets/s]	500	50
T_{HW} [s]	300	300
T_{SW} [s]	60	60
$T_{\mathrm{D\_max}}$ [s]	310	310
Δ_1 [s]	10^{-3}	10^{-2}
Δ_2 [s]	10^{-1}	1
$C_{\max}$ [packets/s]	10^6	10^4
ϵ	0.05	0.05
(s,h)	(1,5)	(1,5)
z^*	30,000	3,000

表 3.3: 制御パラメータ f^* および m^*

パラメータ	バックボーン	バックスキャッタ
サンプリングレートの最適値 f^*	9.8×10^{-4}	8.0×10^{-3}
分割因子の最適値 m^*	12	12

$$d(i,j) \;=\; \begin{cases} 1, & \text{フロー } j \text{ が } i \text{ 番目の HW で検出された} \\ 0, & \text{その他} \end{cases}$$

である．

3.6.2 実験結果

まず，入力パラメータを設定する．簡単のため，BW におけるデータの処理時間 $G(\cdot)$ は 2 章と同様に，

$$G(x) = \Delta_1 x + \Delta_2$$

として与える．表 3.2 に入力パラメータを示す．$T_{\mathrm{SW}}/T_{\mathrm{HW}} = 1/5$ であるため，

$$s = 1, \qquad h = 5$$

である．また，バックボーントレースでは $z^* = \lceil RT_{\mathrm{SW}} \rceil = 30,000$，バックスキャッタトレースでは $z^* = 3,000$ となる．

続いて，式 (3.15) で定義される分割因子 m の最適解 m^* と，式 (3.16) で定義されるサンプリングレート f の最適解 f^* をそれぞれ得る．このとき求めた最適解は表 3.3 に示す．どちらの場合にも $m^* = 12$ となり，$T_{\mathrm{BW}} = T_{\mathrm{SW}}/(m^* s) = 5$ [s] である．式 (3.1) より，HW 内の 重複を許す SW の数は $H_S = 49$ となる．

次に，検出する際のサンプルパケット数の閾値 θ^* を試行回数 10^6 のモンテカルロシミュレーション実験により求めた．このとき，実験には z^* を一様に分布させ

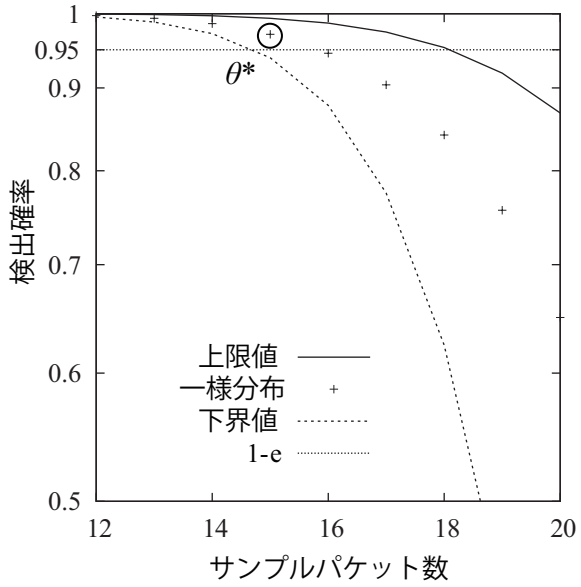

図 3.5: 一様分布フローと検出確率の関係 (バックボーン)

表 3.4: 多段閾値の値

	θ^*	θ_1	θ_2	θ_3	θ_4
バックボーン	15	22	30	37	45
バックスキャッタ	11	16	22	27	33

たフローを用いた．図 3.5 と図 3.6 はバックボーントレースとバックスキャッタトレースのそれぞれについて，横軸の閾値候補を変化させたときの検出確率と上下界値を縦軸にとってプロットしたものである．図より一様分布の検出確率は上下界の間に収まっていることがわかる．結果として，バックボーントレースでは $\theta^* = 15$，バックスキャッタトレースでは $\theta^* = 11$ がそれぞれ求まった．多段閾値 θ_i $(i = 1, 2, 3, 4)$ については

$$\theta_i = \theta^* + \lfloor i\theta^*/2 \rfloor$$

となるように設定した．すなわち，SW におけるサンプルパケット数の最小値 $W_{\min}$ の値に応じて検出されたフローは五つのクラスに分けられる．より正確に言うと，ある i $(i = 1, 2, \ldots, 4)$ について $\theta_{i-1} \leq W_{\min} < \theta_i$ となっていれば，検出されたフローはクラス i に分類する．ただし，$\theta_0 = \theta^*$ として扱い，もし $W_{\min} \geq \theta_4$ となったときはクラス 5 に分類する．それらの結果を表 3.4 および表 3.5 に示す．

表 3.5 より，クラス 1 の誤検出確率の上限 $\overline{\mathrm{FPR}_1}$ はおよそ 1 となっている．これはクラス 1 で検出されたフローは検出対象外のフローである可能性が十分に高いことを示している．これはクラス 2 についても同様に言える．一方で，クラス 5 で検出されたフローは非常に高い確率で検出対象フローであると言える．

図 3.7 および図 3.8 はバックボーントレースとバックスキャッタトレースのそれぞれの検出結果を示している．それぞれのグラフで横軸は時刻を，縦軸は検出さ

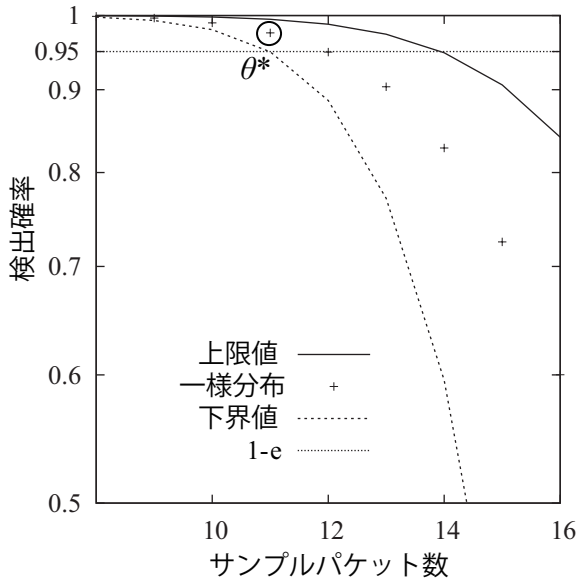

図 3.6: 一様分布フローと検出確率の関係 (バックスキャッタ)

表 3.5: 検出されたフローの分類範囲および誤検出確率の上限

	バックボーン		バックスキャッタ	
	サンプルパケット数	$\overline{\mathrm{FRP}}_i$	サンプルパケット数	$\overline{\mathrm{FRP}}_i$
クラス 1	[15,21]	0.999	[11,15]	0.999
クラス 2	[22,29]	0.933	[16,21]	0.966
クラス 3	[30,36]	0.480	[22,26]	0.686
クラス 4	[37,44]	0.098	[27,32]	0.295
クラス 5	45 or more	0.004	33 or more	0.045

れたフローの ID を示している．‘検出対象’ の印はその番号のフローがその時刻に検出対象，すなわちその時刻の HW において全ての SW で z^* 以上のパケットを有することを示している．図 3.7 では，フロー 1 から 14 までがどこかの時刻で検出対象となっており，その箇所は多くの場合クラス 3 以上で検出されている．一方，フロー 15 から 25 は検出対象外フローが誤検出されたことになる．しかしながら，検出されたときのクラスは多くの場合 1 か 2 である．図 3.8 でも同様の結果になっている．検出されたフローはうまくクラス分けされていると言える．

この持続的高パケットレートフローの検出手法は往々にしてパケットレートが検出対象となる閾値に届く前に検出する傾向がある．この理由としては，パケットレートが高いフローは発生直後からいきなり高いパケットレートで転送を始めるわけではなく，それまでもある程度のパケットレートでパケットを転送しており，そのパケットをつかまえてしまうことで厳密に検出対象となる前に検出してしまうと考えられる．図 3.9 はバックボーントレースのフロー 7 のパケットレートの変化を表している．フロー 7 は期間中 2 箇所でしか検出対象としての条件を満たさないが，それ以外の箇所でもそこそこのパケットを転送しているため，頻繁に

表 3.6: 各指標の 100 回の平均と 95%信頼区間

	バックボーン	バックスキャッタ
TPR	0.99961 ± 0.00509	0.99999 ± 0.00013
FPR	$1.4070 \times 10^{-6} \pm 1.93 \times 10^{-8}$	$9.2950 \times 10^{-6} \pm 7.252 \times 10^{-7}$
誤検出 最少パケット数	10264.7 ± 3175.4	997.47 ± 284.73

検出されている．

最後に，100 回の独立なサンプリング実験により求めた TPR および FPR の値を 95% 信頼区間とともに 表 3.6 に示した．また，誤検出されたフローのうちパケットレートが最も低かったフローの SW におけるパケット数を記録し，100 回の平均と 95% 信頼区間を求めた．TPR からわかることは，検出確率を $1-\epsilon = 0.95$ と設定していてもそれよりもはるかに高い確率で検出していることがわかる．これは，検出対象フローの中でも最も検出しにくいフローを基準にパラメータを決定しているためである．また，FPR も非常に小さな値となっている．さらに，誤検出されたフローのうちパケットレートが最も低いフローでも，パケット数は検出対象の 3 分の 1 程度はあることから，提案手法はよく機能していると結論付ける．

3.7　まとめ

この章では持続的高パケットレートフローを SW 方式とランダムパケットサンプリングを組み合わせてオンラインで検出する手法を提案した．また，サンプルパケット数の閾値 θ^* を未検出確率を考慮して設定する方法，およびサンプリングレート f と ウィンドウの分割因子 m を誤検出確率が最小になるように定める方法を提案した．実トレースデータを使ったサンプリング実験により，提案手法が設計どおりに動作することが確認された．

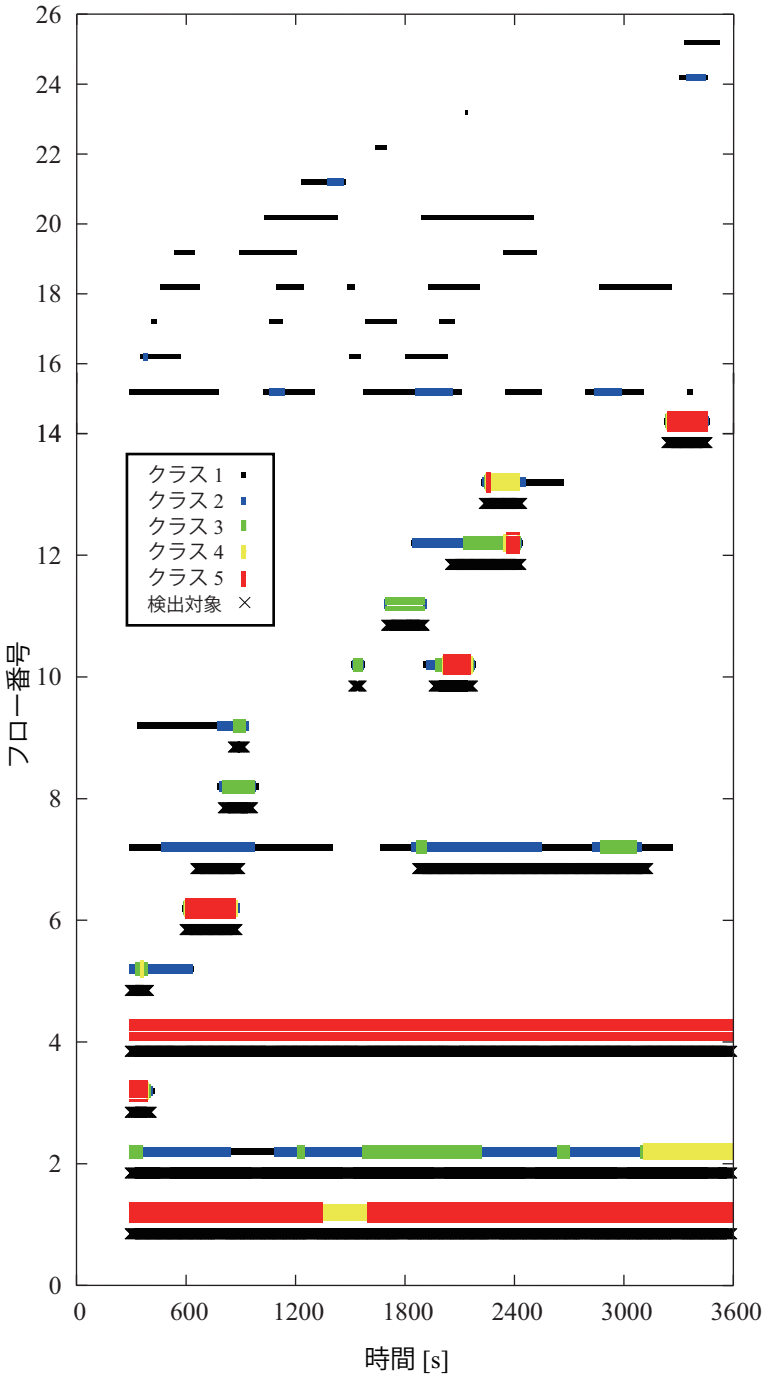

図 3.7: 検出対象と検出結果 (バックボーン).

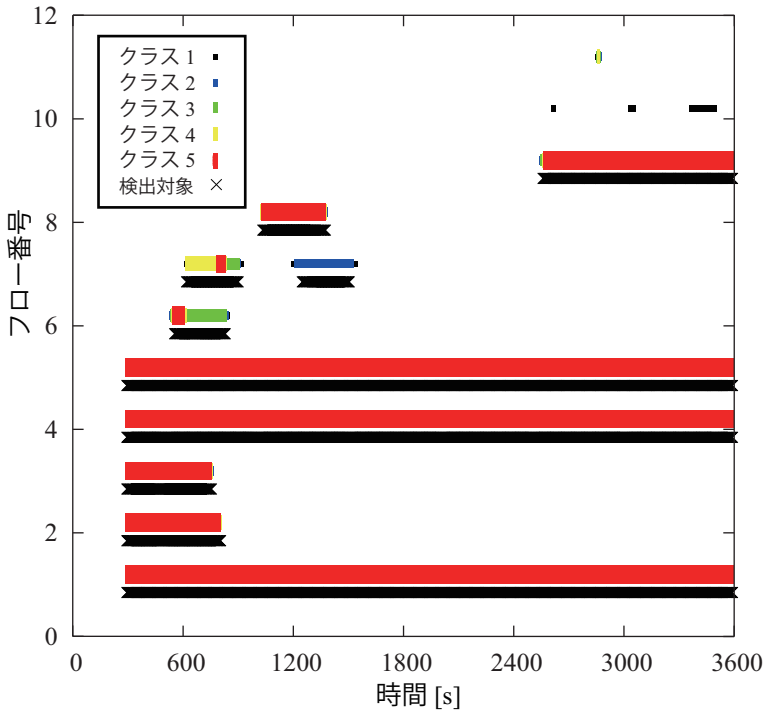

図 3.8: 検出対象と検出結果 (バックスキャッタ).

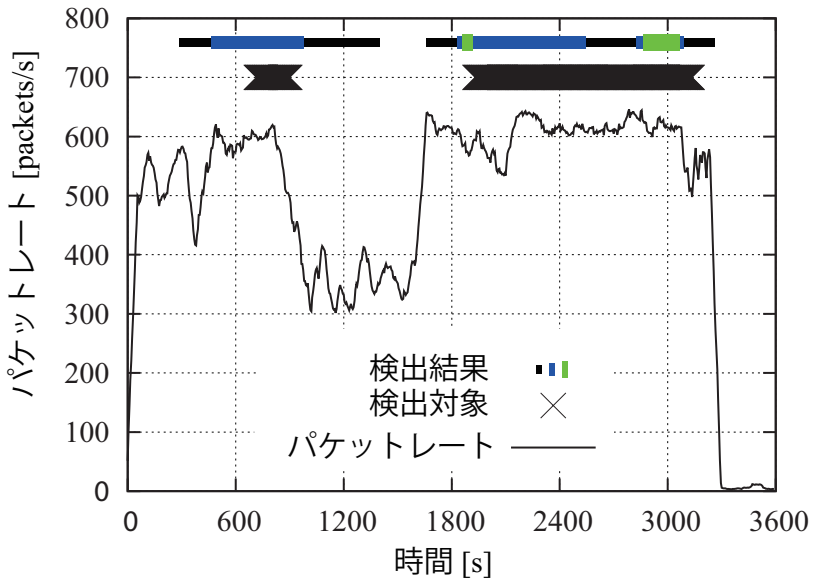

図 3.9: バックボーントレースのフロー 7 のパケットレート

第4章

TCPポートスキャンの検出におけるパラメータ決定手法

4.1 まえがき

本章では，ジャンピングウィンドウ (JW) アルゴリズムとパケットサンプリングを用いた TCP ポートスキャンの検出手法とそのパラメータ決定手法について述べる．ポートスキャンとはネットワーク経由で相手のホストに対して探索パケットを送り，侵入可能なポートがあるかどうか探す行為を指す．ここでは探索パケットに TCP の接続確立要求パケットである SYN パケットを使用する，TCP ポートスキャンの検出を試みる．通常，ポートスキャンは一つのホストから多数の IP アドレスとポート番号の組に対して行われる．すなわち，TCP ポートスキャンが行われると多数の異なる宛先に SYN パケットが送られることになる．しかし SYN パケットを受け取ったホストの多くはそのポート番号を開いておらず，TCP のコネクションは確立されない．本章ではこの特徴に注目した TCP ポートスキャンの検出手法について述べ，その際に使用するパラメータの決定手法について提案する．

4.2 TCPポートスキャン検出手法

本節では，まず TCP ポートスキャンを JW 方式とランダムパケットサンプリングを用いて検出する方法を説明する．次に，検出方法の実行可能性について議論する．その後，ホストごとの SYN パケットのみからなるフロー数を計測する方法について述べる．

ここで，変数 α $(0 < \alpha \leq 1)$ および β $(0 < \beta \leq 1)$ をそれぞれ誤検出確率 FPR の最大許容値および未検出確率 FNR の最大許容値として定義しておく．すなわち，検出対象外のホストを誤検出してしまう確率は α 以下に押さえ，検出対象のホストを検出できずに見逃してしまう確率は β 以下に押さえるようにする．

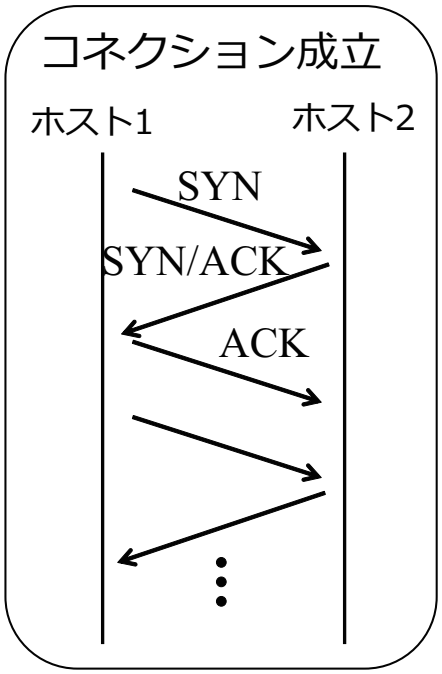

図 4.1: TCP コネクション成立

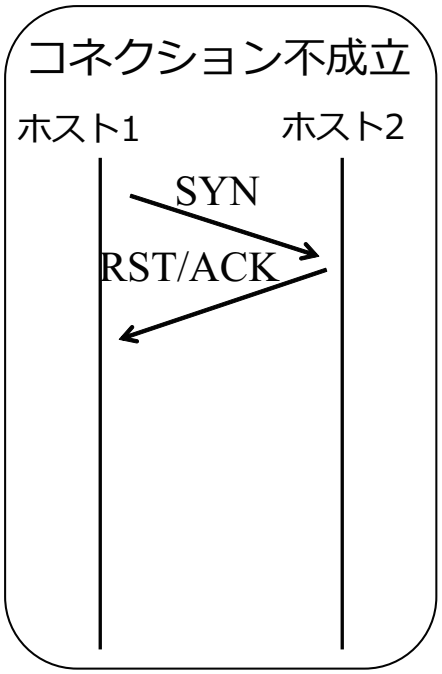

図 4.2: TCP コネクション不成立

4.2.1 TCP ポートスキャンのオンライン検出手法の概要

SYN パケットを送ることで開いているポートを探す TCP ポートスキャンを検出することを考える．図 4.1 に示すように，SYN パケットは TCP コネクションの確立のための 3 ウェイハンドシェイクで最初に送られるパケットとして使用される．SYN パケットを受け取ったホストは送られてきたパケットの宛先ポート番号に指定されたポートを開いていれば SYN/ACK パケットを返送し，さらに ACK パケットを受け取ることでコネクションが確立される．しかし，もし SYN パケットを受け取ったホストにおいて指定された番号のポートが閉じられている場合は，図 4.2 に示すように RST/ACK パケットが返送され，TCP のコネクションは確立されない．通常のホストはわざわざ不必要なポートを開けておくことはないので，TCP ポートスキャンが行われると多数のコネクションの不成立が発生することになる [20]．ポートスキャンの検出にはこのコネクションの不成立数に注目したものが存在する [23].

本章では，ローカルネットワークの管理者の立場で，そのローカルネットワークがインターネットと接続しているゲートウェイやルータにおいて，ローカルエリア内でポートスキャンを行っているホストあるいはポートスキャンを受けているホストを検出することを試みる．具体的には，コネクションの不成立数があらかじめ決定した閾値以上となったホストを検出する．図 4.2 に示すように，SYN パケットと RST/ACK パケットのペアが計測されるとコネクションの不成立が判断されるが，片方向のトラヒックだけを計測しても SYN パケットのみか RST/ACT のみが計測されることになり，コネクションの不成立は判断可能である [18, 31]. 一般的な消費者向けのネットワークでは，インターネットのトラヒックはアップ

ロードとダウンロードで転送量が非対称であることが知られており，片方向のトラヒックだけを計測するのであればローカルネットワークから外部へ向けてのアップロードのトラヒックを計測する方が処理負荷や使用するメモリ領域を軽減できる．そこで外向きのトラヒックのみを計測することとし [32]，さらに処理負荷を軽減させるためにパケットサンプリングを採用する．

本論文では，SYN パケットのみで構成されるフローを SYN-only フローと，RST/ACK パケットのみで構成されるフローを RST-only フローとそれぞれ呼ぶことにする．このとき，ローカルネットワーク内でポートスキャンを実行しているホストがいる場合は，そのホストから通常よりも多くの SYN-only フローが生成されることになり，またあるホストが外部からポートスキャンを受けると，そのホストから通常よりも多くの RST-only フローが生成されることになる．これ以降，詳細な議論はローカルネットワーク内から行われるポートスキャンの検出，すなわち SYN-only フロー数が閾値以上のホストの検出に限定する．なぜならば，RST-only フロー数の議論は，基本的に SYN パケットと RST/ACK パケットを置き換えることで成り立つからである．外部から行われるポートスキャンの検出については，付録 F で述べる．

検出対象について改めて定義する．SYN-only フロー数が適当な長さの時間ウィンドウにおいてあらかじめ定めた閾値 θ 以上となったホストを検出することにする．ここで時間ウィンドウの長さは T_{JW} [s] とし，その更新には JW 方式を用いる．また，トラヒックは全てのパケットを計測するのではなく，ランダムパケットサンプリングで得られたデータから検出を行う．

1 章の図 1.1 にトラヒックの計測地点として 3ヶ所挙げたが，本章で想定する計測地点はそのうちの “ LAN の境界ルータ” に当たる．それ以外の地点で行おうとすると，バックボーンルータでは把握しておかなければならないホスト数が膨大になりすぎ，なおかつ同じホストが生成するフローが常に同じルータを通過するとは限らないため，SYN-only フロー数が閾値以上のホストを検出することには不向きである．また，検出したとしてもそのホストが直接の管理下にないため，直接の管理者と比べるとその後の対応に余計な時間がかかることになる．ホスト自身の NIC で行おうとすることは，そもそも全ホストに計測や検出の仕組みを組み込むこと自体にコストがかかり現実的ではない．

4.2.2 検出手法の実行可能性

検出対象に関して次のように想定する．ポートスキャンを実行しているホストは T_{JW} [s] のウィンドウにおいて，SYN-only フローを S_{B} だけ，正常なフローとは別に生成するものと考える．ここで S_{B} は，

$$S_{\mathrm{B}} \geq N_{\mathrm{S}}, \quad N_{\mathrm{S}} \text{は正の整数} \tag{4.1}$$

を満たしているものとする．つまりポートスキャンのマルウェアは実行ホストにおいてバックグラウンドでスキャンを実行し，少なくとも N_{S} だけの SYN-only フロー

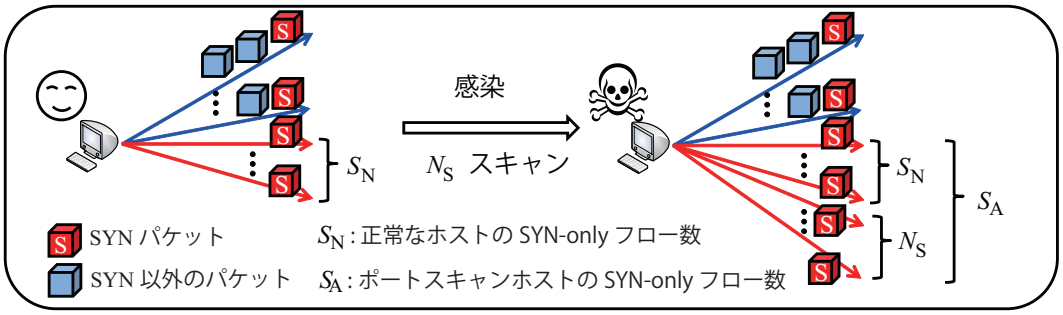

図 4.3: 正常なホストとポートスキャン実行ホストの SYN-only フロー数

を T_{JW} の間に生成するものと考える．ここで，通常のトラヒックにも SYN-only フローは含まれることに注意する．

図 4.3 に示すように，S_N および S_A をそれぞれ正常なホストおよびポートスキャン実行ホストが T_{JW} の間に生成する SYN-only フロー数と定義する．定義より，

$$P[S_A < \theta] = P[S_N + S_B < \theta] \leq P[S_N + N_S < \theta]$$

となる．一般に，誤検出確率 FPR は θ の減少関数であり，未検出確率 FNR は θ の増加関数となる．そこで，$\theta_{FPR} := \theta_{FPR}(\alpha)$ と $\theta_{FNR} := \theta_{FNR}(\beta, N_S)$ を次のように定義する．

$$\begin{aligned} \theta_{FPR} &= \min_{\theta \in \mathbb{N}} \{\theta;\ P[S_N \geq \theta] \leq \alpha\} \\ \theta_{FNR} &= \max_{\theta \in \mathbb{N}} \{\theta;\ P[S_N + N_S < \theta] \leq \beta\} \end{aligned}$$

このとき，検出手法のパラメータ決定が実行可能であるということは，FPR の最大許容値 α と FNR の最大許容値 β が同時に達成されることであるが，これは $\theta_{FPR} \leq \theta \leq \theta_{FNR}$ を満たすような θ が存在することと等価である．すなわち，FPR の最大許容値 α と FNR の最大許容値 β の達成可能性は正常なホストが生成する SYN-only フロー数 S_N の分布に依存する．さらに，$\theta_{FPR} < \theta_{FNR}$ であればサンプリングレート f を 1 より小さくできる可能性がある．

一方で，もし $\theta_{FPR} \geq \theta_{FNR}$ となる場合は，ポートスキャンの実行ホストを α と β の制約を満たしたまま検出することはできない．ここで，θ_{FNR} は N_S の減少関数であり，θ_{FPR} は N_S とは独立であることに注意する．それゆえ，N_S を大きな値に設定すると，FPR と FNR はそれぞれ α と β について同時に達成できる．より具体的には，本章で扱うパラメータ決定において実行可能となるには，N_S は次の不等式を満たす必要がある．

$$N_S \geq \theta_{FPR} - \theta' \tag{4.2}$$

ただし，

$$\theta' = \max_{\theta \in \mathbb{N}} \{\theta;\ P[S_N < \theta] \leq \beta\}$$

である．これらのパラメータの関係を図 4.4 に示す．

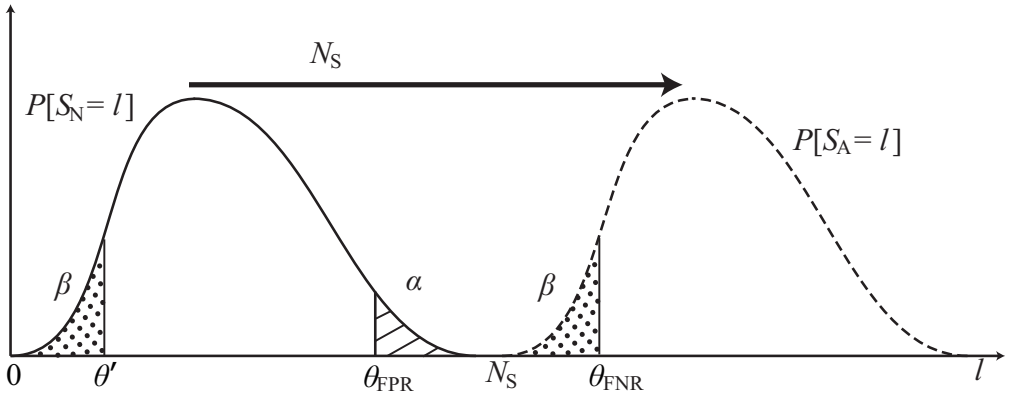

図 4.4: SYN-only フロー数の確率関数

4.2.3 ホストごとの SYN-only フロー数の計測方法

ここでは，ホストごとの SYN-only フロー数を，1.2.2 節で述べた一般的なフロー集約を使用して計測する方法と，ブルームフィルタ [4] を使用して計測する方法を紹介する．本章では TCP のフローにのみ注目しているため，フローは五つ組からプロトコルを除いた四つ組で定義されることに注意する．この四つ組を フロー ID として扱う．

最初に一般的なフロー集約を使用する方法について述べる．パケットの情報をフローに集約する際に，SYN パケットが含まれているかどうかと，SYN パケット以外のパケットが含まれているかどうかの情報を保持するためのフラグをそれぞれ用意し，当該パケットが来たらフラグを立てる．ウィンドウの時間長 T_{JW} が経過した時点で，もし，SYN パケット用のフラグが立ち，SYN パケット以外のパケット用のフラグが立っていなければそのフローは SYN-only フローということになる．全ての SYN-only フローをホストの番号ごとに仕分けることで，ホストごとの SYN-only フロー数が計測できる．

次に，ブルームフィルタを使用した計測方法を述べる．ブルームフィルタは ID を持つ要素の集合をハッシュ関数を用いて集約した形で保持するデータ構造をしており，ある要素がその集合に含まれているかどうかの判定と，その集合に含まれていない要素を新たに登録することが可能である．ブルームフィルタは集約による効率的なメモリ領域利用の反面，登録されていない要素を登録されていると判定してしまう偽陽性の可能性が存在する．しかしながら，十分な大きさのメモリ領域と適当なハッシュ関数を用意すると，偽陽性の可能性は十分小さくすることが可能である．本論文ではブルームフィルタによる偽陽性は起こらないものとして考え，SYN-only フロー数の計測に利用する．以降はブルームフィルタを単にフィルタと呼ぶ．

まず，二つのフィルタ F_{SYN} と F_{Others} を用意する．F_{SYN} は SYN パケット用のフィルタであり，F_{Others} はそれ以外のパケット用のフィルタである．それぞれのフィルタは，ある ID について問い合わせると，その ID がすでに登録されているかどうかを判定できるものとする．また，ホストごとの SYN-only フロー数を

Step 1: フィルタ F_{SYN} と F_{Others}，カウンタ C_i $(i = 1, 2, \ldots, N_{\mathrm{LAN}})$ を初期化し，新しい JW での計測を開始する．

Step 2: パケットが取得されるのを待つ．この間に JW の終了時点に達したら C_i $(i = 1, 2, \ldots, N_{\mathrm{LAN}})$ の結果を出力後に Step 1 へ戻る．取得したパケットについて，ホスト番号 i，フロー ID j，SYN パケットかどうか，の 3 点を調べ，SYN パケットであれば Step 3 へ，SYN パケット以外であれば Step 4 へそれぞれ進む．

Step 3: F_{SYN} へフロー ID j が登録されているかどうかの問い合わせを行い，登録されていれば何もしない．未登録であれば新たに登録し，ホスト i のカウンタ C_i を 1 だけ増やす．Step 2 へ戻る．

Step 4: F_{Others} へフロー ID j が登録されているかどうかの問い合わせを行い，登録されていれば何もしない．未登録であれば j を F_{Others} に新たに登録し，さらに j が F_{SYN} に登録されているかどうか問い合わせ，登録されていれば C_i を 1 だけ減らす．Step 2 へ戻る．

図 4.5: ホストごとの SYN-only フロー数計測手順

数えるためのカウンタ C_i $(i = 1, 2, \ldots, N_{\mathrm{LAN}})$ を用意する．ただし，N_{LAN} は監視しているローカルエリアネットワーク内のホスト数とする．JW が更新されると，フィルタ F_{SYN} と F_{Others}，カウンタ C_i を初期化する．そこからウィンドウの時間長である T_{JW} の間，以下の手順に従って処理を行う．

もし SYN パケットがサンプリングされたら F_{SYN} に同じ ID が登録されていないかどうか確認する．もし登録されていなければそのパケットの ID を新たに登録し，送信元ホストのカウンタに 1 を加算する．一方，もし SYN パケット以外のパケットがサンプリングされたら，そのパケットの ID が F_{Others} に登録されていないか確認する．登録されていなければ新たにその ID を登録し，今度は F_{SYN} に ID が登録されていないかどうか確認する．もし F_{SYN} に ID が登録されていれば，そのパケットの送信元ホストのカウンタを 1 だけ減ずる．ウィンドウの終了時点で，カウンタの値が閾値以上となったホストを検出する．その後は JW を更新し，上記の手順を繰り返す．以上の手順を図 4.5 にまとめる．

4.3 パラメータ決定手法

4.3.1 ランダムパケットサンプリングにおける検出確率

ローカルネットワークから外部へ出て行くフロー情報の収集にはランダムパケットサンプリングを用いる．そしてサンプリングされた中で，SYN パケットのみからなるフローを sampled-SYN-only フローと呼ぶことにする．

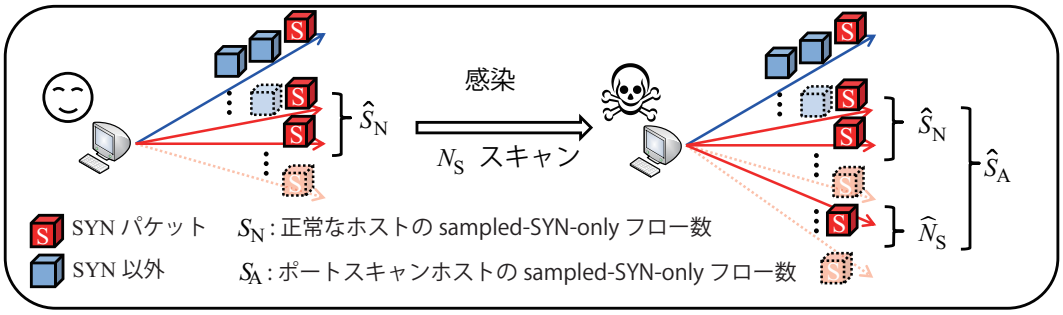

図 4.6: 正常なホストとポートスキャン実行ホストの sampled-SYN-only フロー数

T_{JW} の間，外部へ出ていくパケットを確率 f $(0 < f \leq 1)$ でランダムにサンプリングし，JW の終了時点で $\hat{\theta}$ 以上の sampled-SYN-only フローを生成したホストを検出することにする．ここで，次のことに注意する．

1. SYN-only フローは確率 $1 - f$ でサンプリングされない
2. l $(l \geq 2)$ パケットからなる正常な TCP フローは確率 $f(1-f)^{l-1}$ で sampled-SYN-only フローになる

1. は未検出につながり，2. は誤検出の原因となる．それゆえ，あるホストによって生成される sampled-SYN-only フローの数は，一つのホストが生成するフロー数の分布だけでなく，フロー長の分布にも依存する．ただし，フロー長とはそのフローに含まれるパケット数を表す．

正常なホストから生成される sampled-SYN-only フロー数の分布を得るために，以下のようにしてランダムパケットサンプリングを用いた検出手法をモデル化する．D をランダムに選ばれたホストが T_{JW} の間に生成する SYN パケットを含むフロー数として，また，L をランダムに選ばれたフローのフロー長としてそれぞれ定義する．また，図 4.6 に示すように，$\hat{S}_N$ は正常なホストから生成される sampled-SYN-only フロー数として定義する．

簡単のため，正常なホストから生成されるフロー数 D が与えられた下では，それらのフロー長は独立かつ同一の分布に従うと想定する．すると，

$$\Pr[\hat{S}_N = m] = \sum_{n=m}^{\infty} \binom{n}{m} q_n^m (1-q_n)^{n-m} \Pr[D = n] \tag{4.3}$$

を得る．ただし，q_n $(n = 1, 2, \ldots)$ は，あるサンプリングされたフローが，そのフローを生成したホストが T_{JW} の間にフローを n だけ生成していたという条件の下で，sampled-SYN-only フローである確率を表している．

$$q_n = \sum_{i=1}^{\infty} f(1-f)^{i-1} \Pr[L = i \mid D = n] \tag{4.4}$$

次に，ポートスキャンを実行しているホストから生成される sampled-SYN-only フロー数 $\hat{S}_A$ の分布を考える．4.2.2 小節と同様に，ポートスキャン実行ホストは

S_{B} だけ SYN-only フローを正常なトラヒックとは別に T_{JW} の間に生成すると想定する．ただし，式 (4.1) は成立しているものとする．$\hat{S}_{\mathrm{B}}$ を S_{B} の SYN-only フローからサンプリングされた sampled-SYN-only フローの数とする．同様に $\hat{N}_{\mathrm{S}}$ を N_{S} の SYN-only フローからサンプリングされた sampled-SYN-only フローの数とする．このとき，$l = 1, 2, \ldots$ に対して，二項分布より

$$\Pr[\hat{S}_{\mathrm{B}} < l] \leq \Pr[\hat{N}_{\mathrm{S}} < l] = \sum_{i=0}^{l-1} \binom{N_{\mathrm{S}}}{i} f^i (1-f)^{N_{\mathrm{S}}-i}$$

となる．そして，ポートスキャンを実行しているホストからサンプリングされる sampled-SYN-only フロー数が l 未満となる確率 $\Pr[\hat{S}_{\mathrm{A}} < l]$ $(= \Pr[\hat{S}_{\mathrm{N}} + \hat{S}_{\mathrm{B}} < l])$ は次式で上から押さえられる．

$$\Pr[\hat{S}_{\mathrm{A}} < l] \leq \Pr[\hat{S}_{\mathrm{N}} + \hat{N}_{\mathrm{S}} < l] = \sum_{i=0}^{l-1} \Pr[\hat{S}_{\mathrm{N}} = i] \Pr[\hat{N}_{\mathrm{S}} < l - i] \tag{4.5}$$

4.3.2 パラメータ決定

ここでは，パラメータ決定の問題が少なくとも $f = 1$ で解を持つ，すなわち所与 T_{JW}，N_{S}，α，β に対して式 (4.2) が成り立つことを前提とする．この条件の下，サンプリングレート f と sampled-SYN-only フロー数の閾値 $\hat{\theta}$ を決定することが目的となる．このとき，サンプリングレートは小さいほど，使用するメモリ領域や処理のオーバヘッドを軽減できるため望ましい．そこで，パラメータ決定の問題をサンプリングレート f の最小化を目的とし，FPR の α と FNR の β を制約条件とした最適化問題として定式化する．

まず，FPR である $\Pr[\hat{S}_{\mathrm{N}} \geq \hat{\theta}]$ と FNR の上界である $\Pr[\hat{S}_{\mathrm{N}} + \hat{N}_{\mathrm{S}} < \hat{\theta}]$ はどちらも閾値 $\hat{\theta}$ とサンプリングレート f の関数であることに注意する．FPR $\leq \alpha$ を満たす閾値の下界値 $\hat{\theta}_{\mathrm{FPR}}$ と FNR $\leq \beta$ を満たす閾値の上界値 $\hat{\theta}_{\mathrm{FNR}}$ はそれぞれ，

$$\hat{\theta}_{\mathrm{FPR}}(\alpha, f) = \min_{\hat{\theta} \in \mathbb{N}} \left\{ \hat{\theta};\ \Pr[\hat{S}_{\mathrm{N}} \geq \hat{\theta}] \leq \alpha \right\} \tag{4.6}$$

$$\hat{\theta}_{\mathrm{FNR}}(\beta, f) = \max_{\hat{\theta} \in \mathbb{N}} \left\{ \hat{\theta};\ \Pr[\hat{S}_{\mathrm{N}} + \hat{N}_{\mathrm{S}} < \hat{\theta}] \leq \beta \right\} \tag{4.7}$$

で与えられる．ここで，$\Pr[\hat{S}_{\mathrm{A}} < \hat{\theta}] \leq \Pr[\hat{S}_{\mathrm{N}} + \hat{N}_{\mathrm{S}} < \hat{\theta}]$ に注意する．以上より，パラメータ決定の問題は次のように定式化される．

$$\begin{aligned} \text{最小化} \quad & f, \\ \text{条件} \quad & \epsilon \leq f \leq 1 \\ & \hat{\theta}_{\mathrm{FPR}}(\alpha, x) \leq \hat{\theta}_{\mathrm{FNR}}(\beta, x) \ \text{ for all } x \in [f, 1] \end{aligned} \tag{4.8}$$

ただし，ϵ は十分小さな正の実数であり，これは大域的最小解の存在を保証するために導入されている．

注 1. $x \in [\epsilon, 1]$ に対して $\hat{\theta}_{\mathrm{FPR}}(\alpha, x) = \hat{\theta}_{\mathrm{FNR}}(\beta, x)$ となる解は複数存在する可能性があり，そのときは式 (4.8) より最も小さいものを選ぶ．
注 2. 一旦，最小のサンプリングレート $\hat{f}_{\mathrm{opt}}$ が決定されると，sampled-SYN-only フローの閾値の最適値 $\hat{\theta}_{\mathrm{opt}}$ は $\hat{\theta}_{\mathrm{opt}} = \hat{\theta}_{\mathrm{FNR}}(\beta, \hat{f}_{\mathrm{opt}})$ として設定できる．

4.4 数値実験

4.4.1 実験準備

提案するパラメータ決定手法の効果を確認するため，[1] 内の実トレースデータ *Trace 3* を使用する．このトレースデータは 2003 年にオランダのある大学と学術ネットワークとを接続する地点において計測された 900 秒間の外向きのトラヒックデータで，学内のホスト数は 1,405，転送されたパケット数は 927,006 パケットである．

提案手法では，正常なトラヒックの情報が必要となる．そこで，このトレースデータにはポートスキャンのトラヒックは含まれていないものと想定し，トレースを前半 600 秒と後半 300 秒に分けてそれぞれ学習データとテストデータとした．学習データを用いて，D と L の結合分布を作り，そこからフロー数分布 $\Pr[D = m]$ と条件付きフロー長分布 $\Pr[L = l \mid D = m]$ を得た．なお，T_{JW} [s] は $T_{\mathrm{JW}} = 5, 10, 20, 30, 60, 150$ と設定した．

一方，後半のテストデータはサンプリング実験に使用した．実験では FPR の最大許容値 α と FNR の最大許容値 β をそれぞれ $\alpha = 10^{-5}$，$\beta = 10^{-2}$ と設定した．ポートスキャン実行ホストの検出においては，FPR は十分に小さくなることが望ましい．なぜならば，一度疑わしいと判断されたホストは，管理者等によってネットワークからの切り離しが行われたり [28]，より精度の高い判断を求められたりするため，そこに費やされる手間やコストを考えて $\alpha < \beta$ とした．

上記の条件の下で，テストデータ使用時の FPR と FNR について調べていく．FPR を計算するときは，テストデータそのものに対してサンプリング実験を行う．一方， FNR を計算するときは，テストデータを加工したのちにサンプリング実験を行う．ここで行う加工は，各 JW において一つ以上のパケットを生成した全てのホストに対して，SYN-only フローを N_{S} だけ追加で発生させ，擬似的にポートスキャンを行わせるというものである．

まず最初に，学習データを使用して N_{S} が取り得る最小値，すなわち式 (4.2) の $\hat{\theta}_{\mathrm{FPR}} - \theta'$ を $T_{\mathrm{JW}} = 5, 10, 20, 30, 60, 150$ のそれぞれについて $\alpha = 10^{-5}$ と $\beta = 10^{-2}$ に対して求めた．その結果を表 4.1 に示す．計測時間が長くなるとその分各ホストが生成する SYN-only フロー数も増え，そういったホストを誤検出しないようにするため N_{S} が取り得る最小値は JW の長さ T_{JW} に応じて大きくなっていると考えられる．

次に，N_{S} を表 4.1 の最小値に設定し，サンプリングレートを $f = 1$ としてテストデータに対して数値実験を行う．なお，正常なホストの中で SYN-only フローをまっ

表 4.1: N_{S} が取り得るの最小値と $f = 1$ のときの FPR と FNR

T_{JW}[s]	N_{S} の最小値	総ホスト数	FPR	FNR
5	22	5,012	7.981×10^{-4} (4)	0
10	41	3,346	2.989×10^{-4} (1)	0
20	57	2,270	4.405×10^{-4} (1)	0
30	68	1,845	5.420×10^{-4} (1)	0
60	112	1,251	0 (0)	0
150	132	709	2.821×10^{-3} (2)	0

たく生成しないホストの割合が β よりも大きかった，すなわち $\Pr[S_{\mathrm{N}} = 0] > 10^{-2}$ であったため，$\theta' = 0$ である．すなわち，閾値は $\hat{\theta}_{\mathrm{opt}} = N_{\mathrm{S}}$ となる．

このときの結果を表 4.1 に示す．表では総ホスト数，FPR，FNR を列挙している．なお，FPR 列の括弧内の数字は誤検出されたホスト数を表している．各ホストは JW が更新されると同じホスト番号でも別のホストとみなすため，T_{JW} が大きくなるほど総ホスト数は減少している．FPR は $T_{\mathrm{JW}} = 60$ を除く全てにおいて α よりも大きな値となっている．学習データを基に推定した分布とテストデータの分布が一致する理想的な状況であれば，FPR が $\alpha = 10^{-5}$ と等しくなるため，推定がうまくできていないことを意味する．図 4.7 に学習データを基に作成した FPR の分布と，テストデータをそのまま使用した FPR の分布を示す．FPR $= \alpha$ 付近での分布がかけ離れていることがわかる．

一方，FNR に関しては全ての場合において 0 となった．これは，学習データにおいて $\Pr[S_{\mathrm{N}} = 0] > 10^{-2}$ であり，$\hat{\theta}_{\mathrm{opt}} = N_{\mathrm{S}}$ となったことから，$f = 1$ の場合には全てのホストが検出される状況になったためである．学習データを基に作成した FNR の分布と，テストデータをそのまま使用した FNR の分布を図 4.8 に示す．両者が重なっていることがわかる．

4.4.2 実験結果

$T_{\mathrm{JW}} = 5, 10, 20, 30, 60, 150$ のそれぞれに対して，N_{S} の値として，500，1000，1500，の 3 種類を設定して実験を行う．この数字はポートスキャンを行えるフリーソフト Nmap [22] をデフォルトのまま使用したときに，1000 の宛先に SYN パケットを数秒から十数秒かけて送るのを参考にした．いずれの N_{S} についても，表 4.1 で示した最小値よりも大きいため，サンプリングレートを 1 未満に設定できる可能性がある．

図 4.9 は式 (4.6) で定義される $\hat{\theta}_{\mathrm{FPR}}$ と式 (4.7) で定義される $\hat{\theta}_{\mathrm{FNR}}$ をサンプリングレート f の関数として示したものである．$\hat{\theta}_{\mathrm{FNR}}$ は f，T_{JW}，および N_{S} の関数だが，T_{JW} については感度がほとんどない．これは $\hat{N}_{\mathrm{S}}$ が $\hat{S}_{\mathrm{N}}$ と比較して非常に大きいためである．$\hat{\theta}_{\mathrm{FNR}}$ は N_{S} に関しては増加関数であり，f に関しては減少関数となる．一方で，$\hat{\theta}_{\mathrm{FPR}}$ は f と T_{JW} の関数であり，N_{S} とは独立である．グラフから

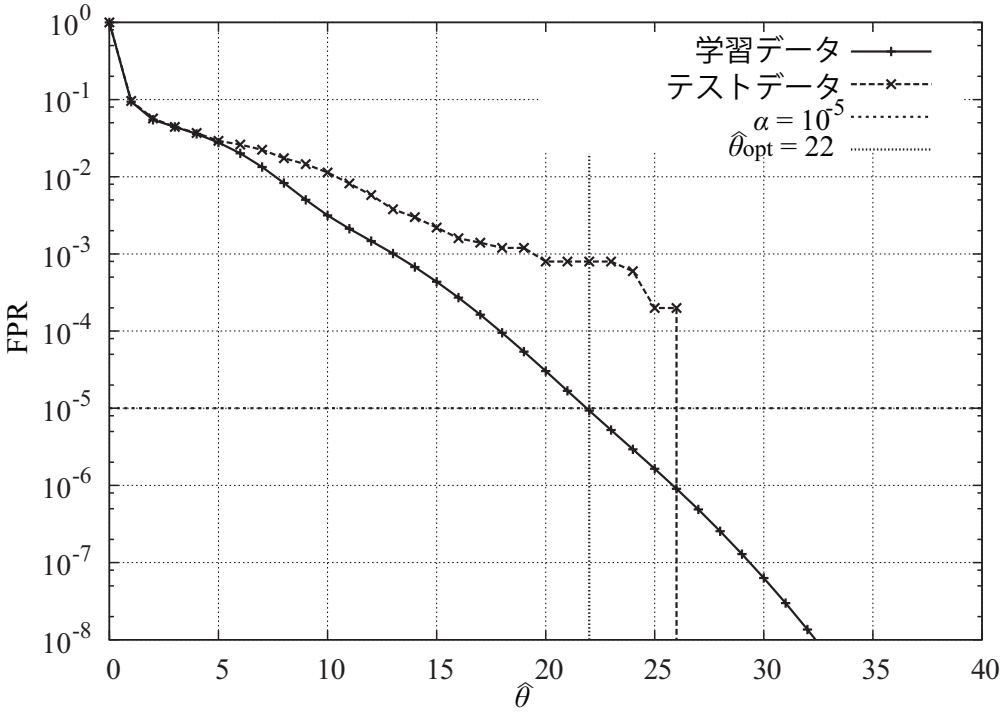

図 4.7: $f = 1$ としたときの FPR と閾値の関係

もわかるが $\hat{\theta}_{\mathrm{FPR}}$ は非常に興味深い形をしている．一部のサンプリングレートで誤検出率が上がるのは，フロー長分布が大きく関係していると推測できる．例えば，フロー長 l $(l \geq 2)$ のフローが sampled-SYN-only フローになる確率は $f(1-f)^{l-1}$ であることを述べた．この式を f について微分すると $f = 1/l$ で極大値を取ることがわかる．すなわち，フロー長 l のフローはサンプリングレート $f = 1/l$ でもっとも sampled-SYN-only フローになりやすいということになる．図 4.9 を見るとサンプリングレートが 0.5 から 0.1 のあたりで $\hat{\theta}_{\mathrm{FPR}}$ のグラフは閾値の値が大きくなっている．これはフロー長が 2 から 10 のフローが sampled-SYN-only フローになりやすい範囲であり，そういったフローを多く生成する正常なホストを誤検出してしまわないようにするために閾値が上がると考えられる．実際に，学習データには SYN パケットを含めたフロー長が 6 のフローが最も多く含まれており，全体の 3 分の 1 近くを占めていた．次いで，フロー長が 5 のフローが全体の 10 分の 1 強を占めていた．

$\hat{\theta}_{\mathrm{FPR}}$ と $\hat{\theta}_{\mathrm{FNR}}$ の線が交わっている点が最適なパラメータセット $\hat{\theta}_{\mathrm{opt}}$ と $\hat{f}_{\mathrm{opt}}$ を与えている．ただし，$\hat{\theta}_{\mathrm{FPR}}$ と $\hat{\theta}_{\mathrm{FNR}}$ は自然数を取るため，そのグラフは階段関数となることに注意する．$\hat{\theta}_{\mathrm{FPR}}(T_{\mathrm{JW}} = 5)$ と $\hat{\theta}_{\mathrm{FNR}}(N_{\mathrm{S}} = 1000)$ について交点付近を拡大したものを図 4.10 に示す．$\hat{\theta}_{\mathrm{FPR}}$ と $\hat{\theta}_{\mathrm{FNR}}$ はサンプリングレートを下げていくとあるところで 1 だけ下がる．その下がる直前の点において，それぞれ FPR$= \alpha$ あるいは FNR$= \beta$ となる．図 4.10 においては，サンプリングレートが 0.02 付近で式 (4.8) を満たす最適パラメータセットを与える．このとき，$\hat{\theta}_{\mathrm{FNR}}$ は変化する直前であり，FNR $= \beta$ を満たす．一方で FPR は α 未満となる．

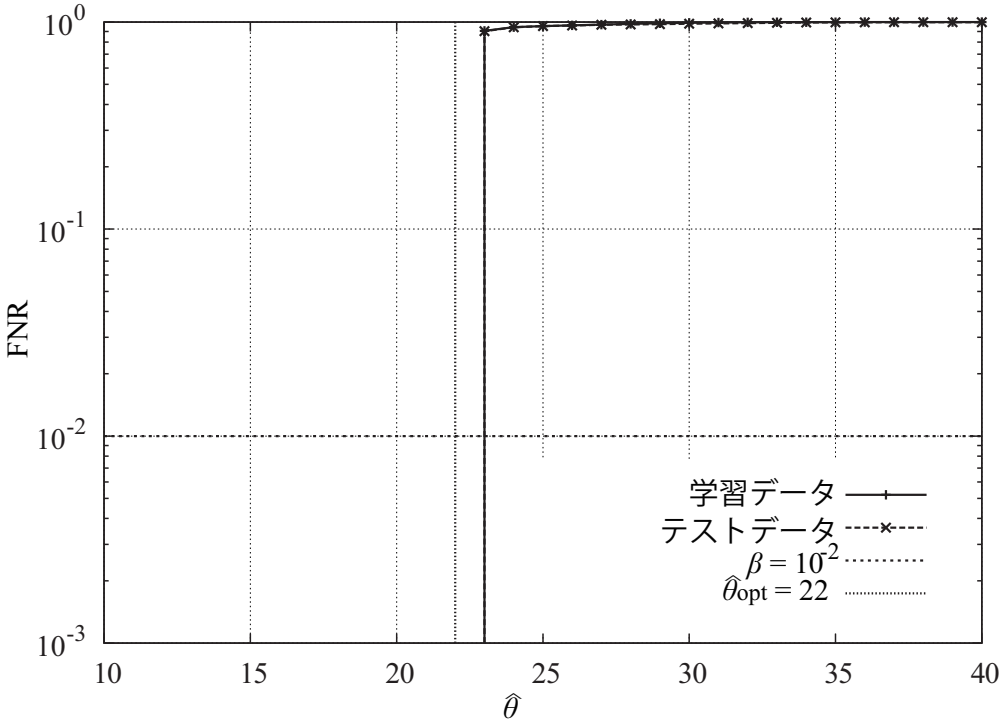

図 4.8: $f = 1$ としたときの FNR と閾値の関係

最適なパラメータセット $\hat{\theta}_{\mathrm{opt}}$ と $\hat{f}_{\mathrm{opt}}$ を表 4.2 に示す．N_{S} を固定すると，最適なサンプリングレート $\hat{f}_{\mathrm{opt}}$ は T_{JW} に応じて増加する．ポートスキャンのフローレートを考えたとき，その値は $N_{\mathrm{S}}/T_{\mathrm{JW}}$ となり，T_{JW} が大きいとそれだけ検出対象のパケットレートは下がることになる．すなわち，パケットレートが低いものを検出しようとすると，それだけ検出が難しくなることを意味している．

$N_{\mathrm{S}}/T_{\mathrm{JW}}$ を固定した場合 (例えば $N_{\mathrm{S}}/T_{\mathrm{JW}} = 500/10, 1000/20, 1500/30$) は，最適なサンプリングレートは T_{JW} を大きくするほど小さくなる．この理由を次のように考える．FPR および $\hat{\theta}_{\mathrm{FPR}}$ は一部の SYN-only flow やサンプリング後に sampled-SYN-only flow になりやすいフローを多く送信するホストの影響を強く受ける．T_{JW} が小さいとこのホストがアクティブな期間を JW 全体で捕まえやすくなる．一方 T_{JW} を大きくすると，非アクティブな期間を含め，ならされた形で SYN-only フロー群を捕らえることになる．また，ある正常な SYN パケットを含むフローに注目したとき，T_{JW} が小さいと，SYN パケットを含む部分と含まない一つないし複数の部分に分断される可能性が高まる．この分断が起こることによって，この正常フローがサンプリング後に SYN-only フローとなる可能性が高まることになる．こういった要因が，$N_{\mathrm{S}}/T_{\mathrm{JW}}$ が一定のときに T_{JW} が小さいときのほう高いサンプリングレートを必要とさせていると考える．表 4.1 において，N_{S} の最小値は T_{JW} に対して線形ではないことからも推察できる．

最後に，最適パラメータがうまく機能することを確認するため，1000 回の独立なサンプリング実験を行った．表 4.3 は FPR および FNR を平均の誤検出回数および未検出回数と併せて示している．ただし，$N_{\mathrm{S}} = 1000$ であり，信頼区間は

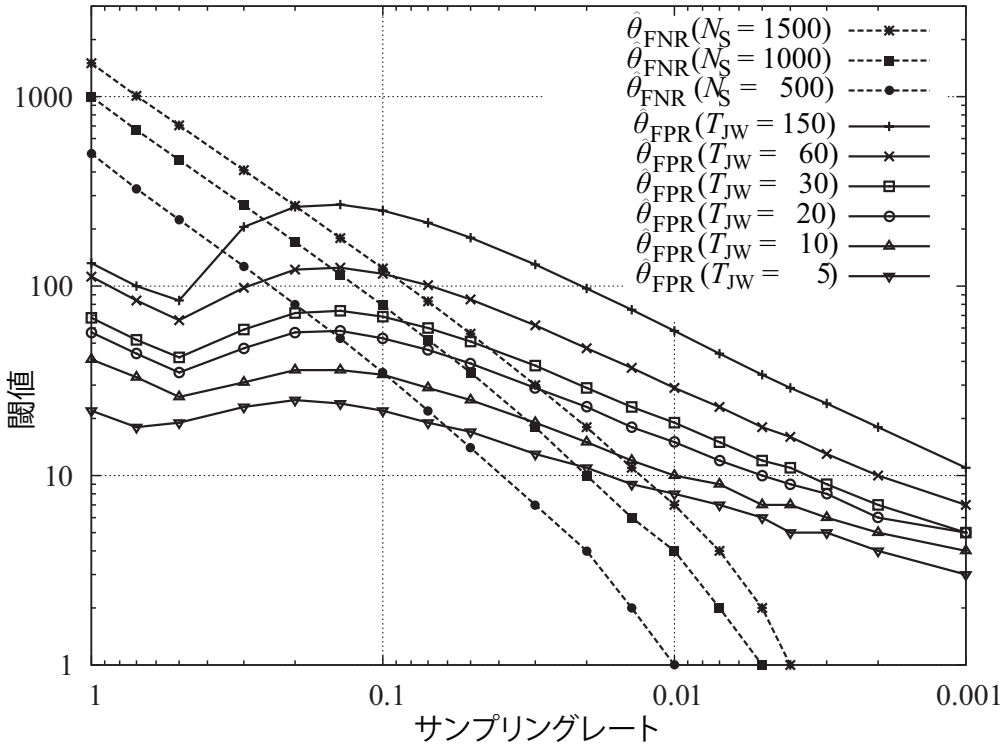

図 4.9: 閾値 $\hat{\theta}_{\mathrm{FPR}}$ および $\hat{\theta}_{\mathrm{FNR}}$ とサンプリングレートの関係

95% とした．表から，FPR と FNR のどちらも制約条件 FPR $\leq \alpha = 10^{-5}$ と FNR $\leq \beta = 10^{-2}$ を満たしていることがわかる．

図 4.11 および 図 4.12 に，$T_{\mathrm{JW}} = 5$，$N_S = 1000$ のときの，学習データを基にした場合とテストデータをそのまま使用した場合の FPR および FNR のグラフを示す．ただし，サンプリングレートと閾値は表 4.2 の提案手法で決定したパラメータを使用した．

FPR について，図 4.7 と図 4.11 を比較すると，サンプリングレートが 1 未満の最適パラメータを使用したときの方がうまく近似できている．この理由として，FPR は正常なホストの中で SYN-only フロー数が上位のホストの影響を強く受ける．サンプリングレートが 1 のとき，$\hat{\theta}_{\mathrm{FPR}}$ はこの上位のホストの分布に直接影響される．一方でサンプリングレートを下げることで，上位のホストからの sampled-SYN-only フロー数はばらけ，ある程度の違いは吸収される形で分布を形成する．そのため，学習データとテストデータのそもそもの分布に違いがあったとしても，サンプリングによってその差を吸収できると考えられる．

FNR については図 4.12 を見るとわかるように，学習データによる分布とテストデータによる分布が重なっている．これは，学習データおよびテストデータのどちらも大半のホストが SYN-only フローを生成していないため，仮想的に発生させる N_{S} が分布を支配しているためである．

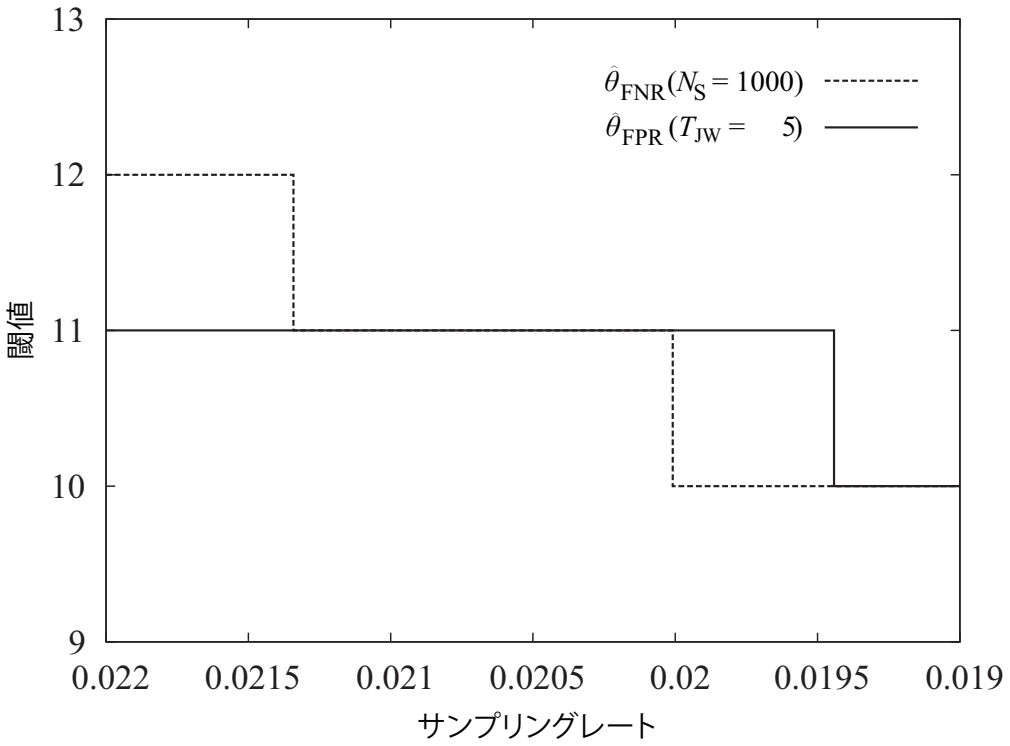

図 4.10: 交点付近における閾値 $\hat{\theta}_{\mathrm{FPR}}(T_{\mathrm{JW}} = 5)$ および $\hat{\theta}_{\mathrm{FNR}}(N_{\mathrm{S}} = 1000)$ とサンプリングレートの関係

4.5 まとめ

本章では TCP ポートスキャンおよびその実行ホストを JW 方式とランダムパケットサンプリングを用いてオンラインで検出する手法について議論した．そして，その検出手法において FPR と FNR を十分小さく押さえた上で，サンプリングレートを最小化するようなパラメータ決定手法を提案した．実トレースデータを用いたサンプリング実験の結果から，提案手法がうまく機能していることが確認された．

表 4.2: パケットレートの最適解 $\hat{f}_{\mathrm{opt}}$ と閾値の最適解 $\hat{\theta}_{\mathrm{opt}}$

W	$N_{\mathrm{S}} = 500$		$N_{\mathrm{S}} = 1,000$		$N_{\mathrm{S}} = 1,500$	
[sec]	$\hat{f}_{\mathrm{opt}}$	$\hat{\theta}_{\mathrm{opt}}$	$\hat{f}_{\mathrm{opt}}$	$\hat{\theta}_{\mathrm{opt}}$	$\hat{f}_{\mathrm{opt}}$	$\hat{\theta}_{\mathrm{opt}}$
5	0.0577697	18	0.0200088	11	0.0106205	8
10	0.0939238	33	0.0303371	19	0.0151368	13
20	0.1509590	58	0.0590948	43	0.0276648	28
30	0.1850930	73	0.0842024	65	0.0418499	46
60	0.2617810	109	0.1512100	126	0.0918684	113
150	0.3667110	159	0.2613340	230	0.1978690	262

表 4.3: 誤検出確率と未検出確率

W [s]	誤検出ホスト数	FPR ($\times 10^{-6}$)	未検出のホスト数	FNR ($\times 10^{-3}$)
5	0.009 ± 0.006	1.796 ± 1.168	50.062 ± 0.450	9.988 ± 0.090
10	0.007 ± 0.005	2.092 ± 1.545	33.378 ± 0.344	9.975 ± 0.103
20	0.004 ± 0.004	1.762 ± 1.724	22.496 ± 0.291	9.910 ± 0.128
30	0.004 ± 0.004	2.168 ± 2.121	18.107 ± 0.252	9.814 ± 0.136
60	0.002 ± 0.003	1.599 ± 2.215	12.245 ± 0.220	9.788 ± 0.176
150	0.001 ± 0.002	1.410 ± 2.764	6.960 ± 0.156	9.817 ± 0.221

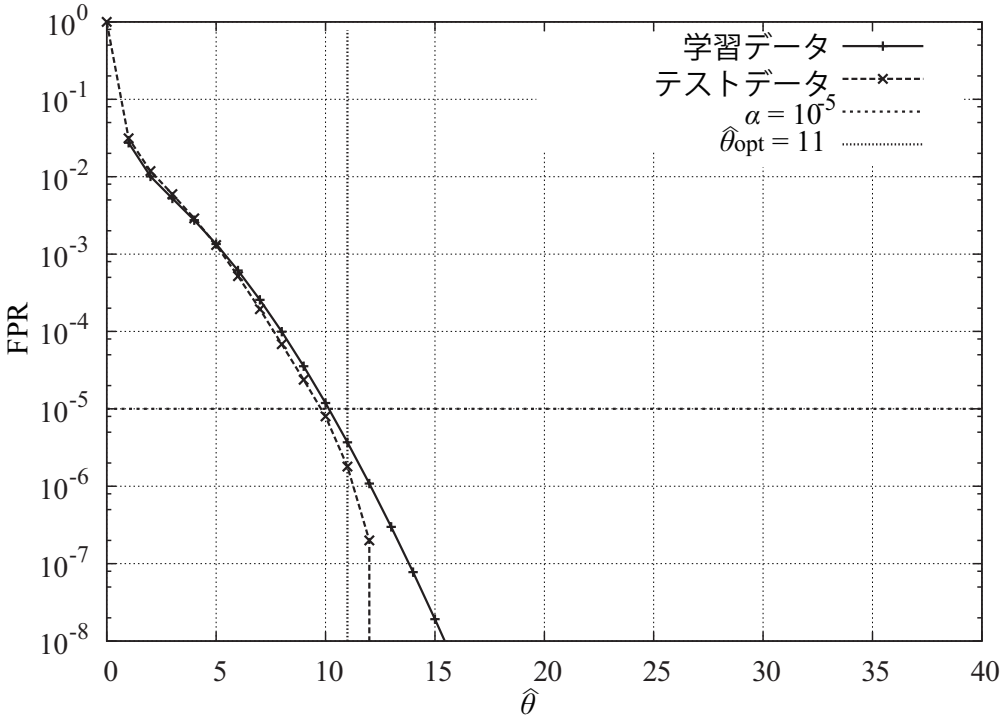

図 4.11: $f = \hat{f}_{\mathrm{opt}}$ としたときの FPR と閾値の関係

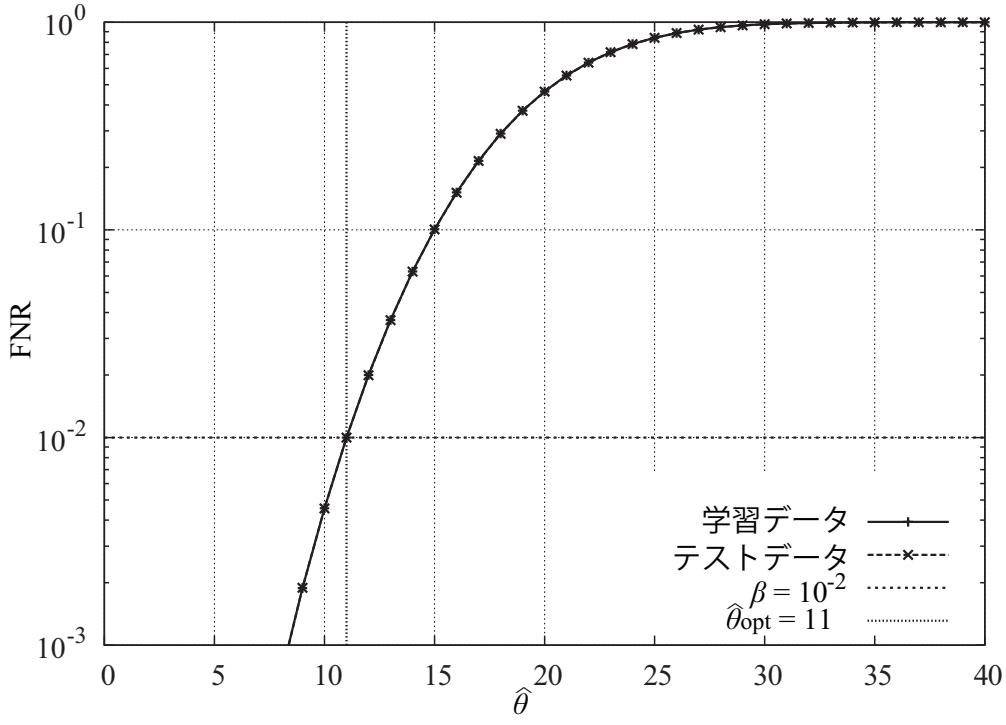

図 4.12: $f = \hat{f}_{\mathrm{opt}}$ としたときの FNR と閾値の関係

第5章

結論

本論文では，異常トラヒックのオンライン検出について論じてきた．オンライン検出のための計測技術については1章で述べた．検出対象となる異常トラヒックには，DoS攻撃などのネットワーク攻撃や回線の帯域を浪費するようなフローの特徴を捉えている高パケットレートフローおよび持続的高パケットレートフロー，そしてワームやボットの感染拡大の手段となるポートスキャンを想定し，2章から4章でそれぞれ扱った．これまでに行った議論や得られた結果についてまとめる．

1章では，異常トラヒックのオンライン検出に関する背景とトラヒック計測技術について述べた．オンラインで検出するための工夫としては，使用するメモリ領域や処理時間を軽減させるパケットサンプリングやフロー集約，オンラインデータ処理のためのスライディングウィンドウ (SW) 方式を，共通する技術として使用した．

2章では，ランダムパケットサンプリングと SW 方式を組み合わせた高パケットレートフローのオンライン検出手法におけるパラメータ決定手法について提案した．検出手法自体は既存の技術を組み合わせたものであり，その技術が与えられれば誰もがたどり着くであろう一般的な手法と言える．しかしながら，そこで使用するパラメータは，各パラメータ単体では変化させたときの影響は推測できても，パラメータ全体を同時に決定する際にどのような方針で決定すればよいかは議論されていなかった．そこで，まず二項分布のポアソン近似を用いて，サンプリングレートと SW の長さの積を最大化することが誤検出確率を最小化する，すなわち標本からの推定精度をもっとも高めることを示した．そして，オンライン検出に関する制約条件の下で上記の積を最大化する問題を解くことでパラメータを決定する手法を提案した．なお，提案したパラメータ決定手法は，パケットサンプリングと時間ベースの計測を組み合わせた手法に対して幅広く適用可能である．

3章では，一定期間高いパケットレートを維持するような，持続的高パケットレートフローのオンライン検出手法と，そのパラメータ決定手法を提案した．SW 方式を用いた検出手法では，一定期間に含まれる全ての SW においてパケットレートが閾値以上のフローを検出対象とした．このとき，対象フローの検出確率は独立でない確率変数の同時確率となるため，数値計算で直接求めることが非常に困難である．また，検出対象フローの中でも BW ごとのパケット数の分布によって

検出確率が変わるため，全ての検出対象フローの検出確率を保証するには対象フローのうちで最も検出確率が低いフロー (閾値フロー) のパケット数の分布を特定する必要がある．残念ながら閾値フローには共通する法則はなかったが，検出対象の条件を満たす最低限のパケットを各 BW に一様に配置した分布のフローが閾値フローと一致，あるいはかなり近い検出確率を示すことを見い出した．また，その一様分布のフローの検出確率はモンテカルロシミュレーション実験により，検出割合として求めることができる．これにより決定したパラメータを使用して実験を行ったところ，非常に良好に動作することが確認できた．

4 章では，SYN パケットを使って宛先のポートが開いているかどうかを探査する TCP ポートスキャンをオンラインで検出する方法について述べ，そのパラメータ決定手法を提案した．TCP ポートスキャンを行っているホストがローカルネットワークにいるとき，外向きのトラヒックを計測すると SYN パケットのみのフローがそのホストから多数送り出される．そこで，その数が閾値を超えたホストを，パケットサンプリングで取得したデータにおける SYN パケットのみのフロー数を，設定した閾値と比較して検出する方法を用いた．正常なホストのフロー数分布とフロー長分布を利用することで誤検出確率 FPR と未検出確率 FNR の両方を保証した上で，サンプリングレートを最小にするようなパラメータ決定を行った．実トレースデータを使ったシミュレーション実験の結果，良好に動作することが確認された．

各章の結果より，異常トラヒックのオンライン検出におけるパラメータの決定について次のようにまとめることができる．まずパケットサンプリングを行うことによって，検出対象の未検出と検出対象外の誤検出が発生するが，これらはサンプリングレートが高いほど良くなる．また，サンプルパケット数やサンプルフロー数の閾値は小さくするほど誤検出が多くなり，大きくするほど未検出が多くなる．一方で，オンラインでの安定動作や使用できるメモリ領域に制限がある場合などはサンプリングレートは下げた方が良い．ウィンドウアルゴリズムを設計する際には，解析対象のデータを集める時間長と更新する時間間隔が決定すべきパラメータとなる．データを集める時間長は，検出対象がアクティブである間は大きくするほどデータがより多く集まるので検出精度は上がる．一方で，データ収集の時間長が，検出対象がアクティブである時間を超えてしまうと，それ以上検出対象の情報が増えることはなく，また，大きくするほどに検出対象が集めたデータ内で相対的に小さくなっていってしまう．ウィンドウの更新間隔は短いほど検査の粒度が上がり，即応性も高くなるが，その一方で，処理時間に関して制約が厳しくなる．これらのパラメータは単体では上記で述べたような点に注意すればよいが，それらを同時に決定する際には，本論文で述べてきたように，目的に応じていずれかの指標を目的関数とし，それ以外を制約条件とした最適化問題を解く手法が有効である．

インターネットは転送されるデータ量も接続ホスト数も拡大しつづけており，また，新しいサービスやアプリケーションが流行る度に転送されるデータの様相も変化している．このようなインターネットを流れるトラヒックの変化は今後も続

いていくと考えられ，その中で異常トラヒックと呼ばれるトラヒックも様々な種類をもって発生し，猛威を振るうと考えられる．その異常トラヒックが，DoS 攻撃のように，被害を被るホスト側の自衛だけでは対処が困難なものであれば，やはりネットワークの内部において管理者の立場から検出できることが望ましい．また，一つひとつのフローは目立たないが，ネットワーク全体を見渡すと実は蔓延しているようなものを見つけようとする場合には，本論文で扱ったような単一の回線もしくはルータでの検出を拡張し，例えば AS 全体としての異常トラヒック検出の仕組みであったり，あるいは複数 AS 間での協調的な測定や検出の仕組みが今後は必要になると考えられる．

付 録A

定理 1 の証明

まず，式 (2.6) を次のように変形する．

$$\sum_{y=y^*}^{rT_{\rm SW}} \binom{rT_{\rm SW}}{y} f^y (1-f)^{rT_{\rm SW}-y} \;<\; \sum_{y=y^*}^{\infty} e^{-rfT_{\rm SW}} \frac{(rfT_{\rm SW})^y}{y!} \tag{A.1}$$

この式は $rT_{\rm SW} < y^*$ のとき成り立つため，これ以降は $rT_{\rm SW} \geq y^*$ と仮定する．式 (A.1) の十分条件は $y = y^*, y^*+1, \ldots, rT_{\rm SW}$ に対して次式で与えられる．

$$\binom{rT_{\rm SW}}{y} f^y (1-f)^{rT_{\rm SW}-y} \leq e^{-rfT_{\rm SW}} \frac{(rfT_{\rm SW})^y}{y!}$$

議論を簡単にするため，$x(r)$ と $\lambda(r)$ を次のように定義する．

$$x(r) = rT_{\rm SW}, \qquad \lambda(r) = frT_{\rm SW}$$

このとき，

$$\begin{aligned}
\binom{x(r)}{y} f^y (1-f)^{x(r)-y} &= \frac{x(r)!}{y!(x(r)-y)!} f^y (1-f)^{x(r)-y} \\
&= \frac{x(r)!}{y!(x(r)-y)!} \left(\frac{\lambda(r)}{x(r)}\right)^y \left(1-\frac{\lambda(r)}{x(r)}\right)^{x(r)-y} \\
&= \left(1-\frac{\lambda(r)}{x(r)}\right)^{x(r)} \cdot \frac{\lambda(r)^y}{y!} \cdot \frac{x(r)!}{(x(r)-y)!(x(r)-\lambda(r))^y} \\
&\leq e^{-\lambda(r)} \cdot \frac{\lambda(r)^y}{y!} \cdot \frac{x(r)!}{(x(r)-y)!(x(r)-\lambda(r))^y}
\end{aligned} \tag{A.2}$$

となる．なお，最後の不等式は全ての $x \geq 0$ に対して，$1-x \leq \exp(-x)$ が成り立つことを用いている．

ここで，次式に注目する．

$$\begin{aligned}
\frac{x(r)!}{(x(r)-y)!(x(r)-\lambda(r))^y} &= \left(\prod_{k=0}^{\lceil\lambda(r)\rceil-1} \frac{x(r)-k}{x(r)-\lambda(r)}\right) \cdot \frac{x(r)-\lceil\lambda(r)\rceil}{x(r)-\lambda(r)} \\
&\quad \cdot \left(\prod_{k=\lceil\lambda(r)\rceil+1}^{y-1} \frac{x(r)-k}{x(r)-\lambda(r)}\right)
\end{aligned} \tag{A.3}$$

式 (A.3) の右辺の最初の要素の分数は 1 より大きく，中央の分数は 1 以下，最後の要素の分数は 1 未満となっている．仮定より，$y^* \geq 2\lceil\lambda(r)\rceil + 1$ であるため，式 (A.3) は全ての $y = y^*, y^* + 1, \ldots, rT_{\mathrm{SW}}$ に対して以下のように書き換えられる．

$$\begin{aligned}\frac{x(r)!}{(x(r)-y)!(x(r)-\lambda(r))^y} &= \left(\prod_{k=1}^{\lceil\lambda(r)\rceil} \frac{x(r)-\lceil\lambda(r)\rceil+k}{x(r)-\lambda(r)} \cdot \frac{x(r)-\lceil\lambda(r)\rceil-k}{x(r)-\lambda(r)}\right) \\ &\quad \cdot \frac{x(r)-\lceil\lambda(r)\rceil}{x(r)-\lambda(r)} \cdot \left(\prod_{k=2\lceil\lambda(r)\rceil+1}^{y-1} \frac{x(r)-k}{x(r)-\lambda(r)}\right)\end{aligned}$$

ここで，$0 < a \leq b \leq x$ を満たすような x, a, b に対して，

$$\frac{x+a}{x} \cdot \frac{x-b}{x} < 1$$

となるため，

$$\frac{x(r)-\lceil\lambda(r)\rceil+k}{x(r)-\lambda(r)} \cdot \frac{x(r)-\lceil\lambda(r)\rceil-k}{x(r)-\lambda(r)} < 1$$

となり，

$$\frac{x(r)!}{(x(r)-y)!(x(r)-\lambda(r))^y} < 1 \tag{A.4}$$

を得る．式 (A.1)，式 (A.2)，そして式 (A.4) より，定理 1 が導かれる． □

付 録B

定理 2の証明

まず，固定された $k \in \mathcal{K}$ に対して，元の問題を以下のような最小化問題に書き換える．

$$
\begin{aligned}
P: \quad \text{最小化} \quad & -fT_{\mathrm{SW}} \\
\text{条件} \quad & T_{\mathrm{SW}} > 0,\ f > 0 \\
& \frac{k+1}{k} T_{\mathrm{SW}} + G\left(\frac{fC_{\max}T_{\mathrm{SW}}}{k}\right) - T_{\mathrm{D\_max}} \leq 0 \\
& G\left(\frac{fC_{\max}T_{\mathrm{SW}}}{k}\right) - \frac{T_{\mathrm{SW}}}{k} \leq 0
\end{aligned}
$$

次に，問題 P の緩和問題 P' を以下で与える．

$$
\begin{aligned}
P': \quad \text{最小化} \quad & -fT_{\mathrm{SW}} \\
\text{条件} \quad & \frac{k+1}{k} T_{\mathrm{SW}} + G\left(\frac{fC_{\max}T_{\mathrm{SW}}}{k}\right) - T_{\mathrm{D\_max}} \leq 0 \\
& G\left(\frac{fC_{\max}T_{\mathrm{SW}}}{k}\right) - \frac{T_{\mathrm{SW}}}{k} \leq 0
\end{aligned}
$$

最初に，緩和問題 P' の大域的最適解が存在することを示す．制約条件が等号で成り立つと仮定すると，

$$
\begin{aligned}
T_{\mathrm{SW}} &= \frac{k}{k+2} T_{\mathrm{D\_max}} > 0 && \text{(B.1)} \\
f &= \frac{k}{C_{\max}T_{\mathrm{SW}}} G^{-1}\left(\frac{T_{\mathrm{SW}}}{k}\right) \\
&= \frac{k+2}{C_{\max}T_{\mathrm{D\_max}}} G^{-1}\left(\frac{T_{\mathrm{D\_max}}}{k+2}\right) > 0 && \text{(B.2)}
\end{aligned}
$$

を得る．ここで，式 (B.2) の不等式は $G(0) < T_{\mathrm{D\_max}}/(k+2)$ の仮定による．したがって，緩和問題 P' は実行可能解 $-fT_{\mathrm{SW}} < 0$ を持つ．よって，以下のような問題を考える．

$$
\begin{aligned}
P'': \quad \text{最小化} \quad & -fT_{\mathrm{SW}}
\end{aligned}
$$

$$\text{条件} \quad \frac{k+1}{k}T_{\mathrm{SW}} + G\left(\frac{fC_{\max}T_{\mathrm{SW}}}{k}\right) - T_{\mathrm{D\_max}} \leq 0 \tag{B.3}$$

$$G\left(\frac{fC_{\max}T_{\mathrm{SW}}}{k}\right) - \frac{T_{\mathrm{SW}}}{k} \leq 0 \tag{B.4}$$

$$-fT_{\mathrm{SW}} \leq -\zeta \tag{B.5}$$

ここで，$\zeta > 0$ は十分小さな正の定数とする．$G(x)$ は x の狭義増加関数であるため，式 (B.4) および式 (B.5) より，

$$G\left(\frac{\zeta C_{\max}}{k}\right) \leq G\left(\frac{fC_{\max}T_{\mathrm{SW}}}{k}\right) \leq \frac{T_{\mathrm{SW}}}{k}$$

となる．また，式 (B.3) と式 (B.5) より，次式を得る．

$$\begin{aligned}\frac{k+1}{k}T_{\mathrm{SW}} + G\left(\frac{\zeta C_{\max}}{k}\right) \\ \leq \frac{k+1}{k}T_{\mathrm{SW}} + G\left(\frac{fC_{\max}T_{\mathrm{SW}}}{k}\right) \leq T_{\mathrm{D\_max}}\end{aligned}$$

したがって，

$$\begin{aligned}kG\left(\frac{\zeta C_{\max}}{k}\right) &\leq T_{\mathrm{SW}} \\ &\leq \frac{k}{k+1}\left(T_{\mathrm{D\_max}} - G\left(\frac{\zeta C_{\max}}{k}\right)\right)\end{aligned} \tag{B.6}$$

となる．

一方で，式 (B.3) より

$$\begin{aligned}(k+1)\zeta + G(\zeta f C_{\max}) &\leq \frac{k+1}{k}T_{\mathrm{SW}} + G\left(\frac{fC_{\max}T_{\mathrm{SW}}}{k}\right) \\ &\leq T_{\mathrm{D\_max}}\end{aligned}$$

となり，ゆえに

$$G(\zeta f C_{\max}) \leq T_{\mathrm{D\_max}} - (k+1)\zeta \tag{B.7}$$

となる．また，式 (B.5) より，

$$f \geq \frac{\zeta}{T_{\mathrm{SW}}} \tag{B.8}$$

となる．したがって，式 (B.6)，(B.7)，(B.8) とあわせて考えると，次式を得る．

$$\begin{aligned}\frac{(k+1)\zeta}{k\left(T_{\mathrm{D\_max}} - G\left(\frac{\zeta C_{\max}}{k}\right)\right)} \leq \frac{\zeta}{T_{\mathrm{SW}}} \\ \leq f \leq \frac{G^{-1}(T_{\mathrm{D\_max}} - (k+1)\zeta)}{\zeta C_{\max}}\end{aligned} \tag{B.9}$$

式 (B.6) と 式 (B.9) は (f, T_{SW}) の実行可能領域が有界閉集合であることを示している．任意の連続関数は有界かつ閉じた領域内で大域的最小解を持つため，問題 P'' は大域的最小解 (f^*, T^*_{SW}) を持つ．ここで，(f^*, T^*_{SW}) は緩和問題 P' においても大域的最小解になることを示す．もし，(f^*, T^*_{SW}) が 緩和問題 P' の大域的最小解になっていなければ，$-\overline{f} \cdot \overline{T}_{\mathrm{SW}} < -f^* T^*_{\mathrm{SW}} < -\zeta$ を満たす P' の実行可能解 $(\overline{f}, \overline{T}_{\mathrm{SW}})$ が存在する．しかし $(\overline{f}, \overline{T}_{\mathrm{SW}})$ は P'' でも実行可能解であるため，P'' における (f^*, T^*_{SW}) の大域的最適性に反する．

このように，緩和問題 P' には大域的最小解 (f^*, T^*_{SW}) が存在し，

$$-f^* T^*_{\mathrm{SW}} < 0$$

を満たす．したがって，$\overline{f} \cdot \overline{T}_{\mathrm{SW}} > 0$ を満たす局所最小解 $(\overline{f}, \overline{T}_{\mathrm{SW}})$ が存在する．ここで，ラグランジュ関数 $L(f, T_{\mathrm{SW}}, \lambda_1, \lambda_2)$ を緩和問題 P' に導入する．

$$\begin{aligned} L(f, T_{\mathrm{SW}}, \lambda_1, \lambda_2) &= -f T_{\mathrm{SW}} + \lambda_1 \left[G\left(\frac{f C_{\max} T_{\mathrm{SW}}}{k} \right) - \frac{T_{\mathrm{SW}}}{k} \right] \\ &\quad + \lambda_2 \left[\frac{k+1}{k} T_{\mathrm{SW}} + G\left(\frac{f C_{\max} T_{\mathrm{SW}}}{k} \right) - T_{\mathrm{D\_max}} \right] \end{aligned}$$

ここで，λ_1 と λ_2 はラグランジュの未定乗数である．$\overline{f} \cdot \overline{T}_{\mathrm{SW}} > 0$ を満たす局所最小解 $(\overline{f}, \overline{T}_{\mathrm{SW}})$ に対する KKT 条件より，

$$\left. \frac{\partial L}{\partial f} \right|_{(f, T_{\mathrm{SW}}) = (\overline{f}, \overline{T}_{\mathrm{SW}})} = -\overline{T}_{\mathrm{SW}} + (\lambda_1 + \lambda_2) \frac{C_{\max} \overline{T}_{\mathrm{SW}}}{k} G'\left(\frac{\overline{f} C_{\max} \overline{T}_{\mathrm{SW}}}{k} \right) = 0 \tag{B.10}$$

$$\begin{aligned} \left. \frac{\partial L}{\partial T_{\mathrm{SW}}} \right|_{(f, T_{\mathrm{SW}}) = (\overline{f}, \overline{T}_{\mathrm{SW}})} &= -\overline{f} + \frac{k+1}{k} \lambda_2 - \frac{\lambda_1}{k} \\ &\quad + (\lambda_1 + \lambda_2) \frac{\overline{f} C_{\max}}{k} G'\left(\frac{\overline{f} C_{\max} \overline{T}_{\mathrm{SW}}}{k} \right) = 0 \end{aligned} \tag{B.11}$$

となり，式 (B.10) より

$$(\lambda_1 + \lambda_2) \frac{C_{\max}}{k} G'\left(\frac{\overline{f} C_{\max} \overline{T}_{\mathrm{SW}}}{k} \right) = 1 \tag{B.12}$$

となる．この式および式 (B.11) より次式を得る．

$$(k+1)\lambda_1 = \lambda_2 \tag{B.13}$$

$x \geq 0$ を満たす全ての x に対して $G'(x) > 0$ となるため，式 (B.12) は $\lambda_1 + \lambda_2 > 0$ を意味する．したがって，式 (B.13) と併せて考えると，$\lambda_1 > 0$ かつ $\lambda_2 > 0$ となる．ゆえに，KKT の相補性条件より，緩和問題 P' における二つの制約条件はど

ちらも有効制約となっている．結果として，式 (B.1) および式 (B.2) より，緩和問題 P' の局所最小解 $(\overline{f}, \overline{T}_{\mathrm{SW}})$ は次式により一意に求まる．

$$\begin{aligned} \overline{f} &= \frac{k+2}{C_{\max} T_{\mathrm{D\_max}}} G^{-1}\left(\frac{T_{\mathrm{D\_max}}}{k+2}\right) > 0 \\ \overline{T}_{\mathrm{SW}} &= \frac{k}{k+2} T_{\mathrm{D\_max}} > 0 \end{aligned}$$

これらは緩和問題 P' において大域的最小解 (f^*, T^*_{SW}) を与える．さらに，$\overline{f} > 0$ かつ $\overline{T}_{\mathrm{SW}} > 0$ であるため，元の問題 P においても大域的最小解となる． □

付 録C
検出確率の上界と下界

$\boldsymbol{X}_{\mathrm{MTF}} = \boldsymbol{x}$ で特徴づけられる検出対象フローをここではMTF (Minimum Target Flow) と呼ぶ．まず，MTF の検出確率 $P(w \mid \boldsymbol{x})$ の上界を求める．所与 s と h に対し，次のように n を定義する．

$$n = \left\lceil \frac{h}{s} \right\rceil$$

もし $n \geq 2$ であれば，W_{js+1} $(j = 0, 1, \ldots, n-1)$ は定義より互いに独立である (式 (3.3))．このとき Γ_{I} と Γ_{D} を次のように定義する．

$$\Gamma_{\mathrm{I}} = \{1, s+1, 2s+1, \ldots, (n-1)s+1\}, \qquad \Gamma_{\mathrm{D}} = \{1, 2, \ldots, h\} \setminus \Gamma_{\mathrm{I}}$$

すると，

$$\begin{aligned} P(w \mid \boldsymbol{x}) &= \prod_{j \in \Gamma_{\mathrm{I}}} \Pr\left[W_j \geq w \mid \boldsymbol{X}_{\mathrm{MTF}} = \boldsymbol{x}\right] \\ &\qquad \cdot \Pr\left[W_i \geq w \ (i \in \Gamma_{\mathrm{D}}) \mid W_j \geq w \ (j \in \Gamma_{\mathrm{I}}), \boldsymbol{X}_{\mathrm{MTF}} = \boldsymbol{x}\right] \\ &= p^n(w \mid z^*) \cdot \Pr\left[W_i \geq w \ (i \in \Gamma_{\mathrm{D}}) \mid W_j \geq w \ (j \in \Gamma_{\mathrm{I}}), \boldsymbol{X}_{\mathrm{MTF}} = \boldsymbol{x}\right] \\ &\leq p^n(w \mid z^*) \end{aligned}$$

となり，ここから式 (3.6) の上界は導かれる．ここで，$\boldsymbol{x} = (0, 0, \ldots, 0, z^*)$ とすると，

$$\Pr\left[W_i \geq w \ (i \in \Gamma_{\mathrm{D}}) \mid W_j \geq w \ (j \in \Gamma_{\mathrm{I}}), \boldsymbol{X}_{\mathrm{MTF}} = (0, 0, \ldots, 0, z^*)\right] = 1,$$

となるため，

$$P(w \mid (0, 0, \ldots, 0, z^*)) = p^n(w \mid z^*) \tag{C.1}$$

が得られる．したがって，式 (C.1) で表される MTF の上界は，厳密には MTF の上限となっている．

次に MTF の検出確率 $P(w \mid \boldsymbol{x})$ の下界を考える．$\boldsymbol{Y}$ は $(Y_1, Y_2, \ldots, Y_{h+s-1})$ を表すものとする．$\boldsymbol{X}_{\mathrm{MTF}} = \boldsymbol{x}$ が与えられたとき，Y_i $(i = 1, 2, \ldots, h+s-1)$ は互いに独立である．文献 [21] の Theorem 3.10.5 (v) に従うと，確率変数 $\boldsymbol{Y}$ は “正

の関連を有する (positively associated)" ことになる．$\boldsymbol{Y}$ が正の関連を有するとは，任意の増加関数 $f(\boldsymbol{Y})$ と $g(\boldsymbol{Y})$ に対して，

$$\mathrm{Cov}[f(\boldsymbol{Y}), g(\boldsymbol{Y})] \geq 0$$

が成り立つ，あるいは結果として

$$E[f(\boldsymbol{Y})g(\boldsymbol{Y})] \geq E[f(\boldsymbol{Y})]E[g(\boldsymbol{Y})]$$

が成り立つ場合を意味する．

非負の増加関数の積はそれもまた増加関数になるため，帰納法により任意の非負の増加関数 $f_i(\boldsymbol{Y})$ $(i = 1, 2, \ldots, j)$ に対して次式が成り立つ．

$$E\left[\prod_{i=1}^{j} f_i(\boldsymbol{Y})\right] \geq \prod_{i=1}^{j} E\left[f_i(\boldsymbol{Y})\right] \tag{C.2}$$

ここで，$I_i(w, \boldsymbol{Y} \mid \boldsymbol{x})$ $(i = 1, 2, \ldots, h)$ を次のように定義する．

$$I_i(w, \boldsymbol{Y} \mid \boldsymbol{x}) = \begin{cases} 1, & Y_i + Y_{i+1} + \cdots + Y_{i+s-1} \geq w \text{ for a given } \boldsymbol{X}_{\mathrm{MTF}} = \boldsymbol{x}, \\ 0, & \text{otherwise} \end{cases}$$

$I_i(w, \boldsymbol{Y} \mid \boldsymbol{x})$ $(i = 1, 2, \ldots, h)$ は $\boldsymbol{Y}$ の非負増加関数であることは明白である．すなわち，

$$Y_j \geq Y'_j, \qquad j = i, i+1, \ldots, i+s-1$$

であるような $\boldsymbol{Y}$ と $\boldsymbol{Y}'$ に対して，

$$I_i(w, \boldsymbol{Y} \mid \boldsymbol{x}) \geq I_i(w, \boldsymbol{Y}' \mid \boldsymbol{x})$$

が成り立つ．

そこで式 C.2 を $I_i(w, \boldsymbol{Y} \mid \boldsymbol{x})$ に適用すると，

$$\begin{aligned} P(w \mid \boldsymbol{x}) &= \Pr\left[W_1 \geq w, W_2 \geq w, \ldots, W_{h+s-1} \geq w \mid \boldsymbol{X}_{\mathrm{MTF}} = \boldsymbol{x}\right] \\ &= E\left[\prod_{i=1}^{h+s-1} I_i(w, \boldsymbol{Y} \mid \boldsymbol{x})\right] \\ &\geq \prod_{i=1}^{h+s-1} E\left[I_i(w, \boldsymbol{Y} \mid \boldsymbol{x})\right] \\ &= \prod_{i=1}^{h+s-1} \Pr\left[W_i \geq w \mid \boldsymbol{X}_{\mathrm{MTF}} = \boldsymbol{x}\right] = p^{h+s-1}(w \mid z^*) \end{aligned}$$

となる．以上より式 (3.6) の下界が得られる．

付 録D

近似的閾値フロー特定のためのヒューリスティック法

閾値フローは最少のパケット数で構成される検出対象フローのうち，検出確率が最も小さいフローとして定義される．一般的な議論をすると，式 (3.4) 中の検出確率 $P(w \mid \boldsymbol{x})$ は五つのパラメータ s, h, f, z^*, そして w に依存するため，それらのパラメータが与えられた下で閾値フローを解析的に求めることはほぼ不可能である．そこで，検出確率が最小のときと十分近くなるような近似的な閾値フローを考える．

まず，T_{HW} が T_{SW} の倍数になっているような場合を考える．この場合，次のような $1 \times s$ ベクトルで特徴付けられる MTF が検出確率の上限を所与のパラメータに関係なく与える．そのベクトルは，$(z^*, 0, 0, \ldots, 0)$, $(0, z^*, 0, 0, \ldots, 0)$, $\ldots$, $(0, 0, \ldots, 0, z^*)$ である．このことは z^* のパケットを SW 内の s 個の各 BW に一様に分布させると検出確率が最小となるのではないかという考えをもたらす．検出確率の数値計算が比較的容易な s と h が小さなときに，様々なパラメータの組み合わせで確認した結果，z^* が s で割りきれるときは一様に分布させたときが検出確率最小となった．その一例を表 D.1 に示す．このときのパラメータは $T_{\mathrm{HW}} = 3T_{\mathrm{SW}}$, $T_{\mathrm{SW}} = 3T_{\mathrm{BW}}$, $z^* = 9$ である．x_i $(i = 1, 2, 3)$ は X_i $(= X_{i+3} = X_{i+6})$ に割り付けたパケット数である．

次に，T_{HW} は T_{SW} の倍数だが，パケット数 z^* が ms で割りきれず ($r = z^* \bmod ms$)，SW 内の 各 BW に同数を割り付けられない場合を考える．様々なパラメータで数値計算を行ったところ，確認した全て結果でおおよそ一様分布と言える分布が閾値フローとなっていた．言い換えると，閾値フローの分布において，BW 間のパケット数の差は高々 1 であった．表 D.2 は z^* 以外のパラメータが表 D.1 と同じ条件として，検出確率を数値計算によって求めた結果である．$r = 1$ $(z^* = 10)$ のときは余った 1 パケットを SW 内の 2 番目の BW に割り付けており，$r = 2$ $(z^* = 11)$ のときは余った 2 パケットを一つは 2 番目に割り付け，もう一つは 1 番目もしくは 3 番目に割り付けると閾値フローとなった．1 パケットの差は z^* が大きくなると相対的には小さくなっていくので，余ったパケットをどこに配置するかは検出確率にほとんど影響を与えず，無視できると考える．この考えは後ほど現実的な規模の実験を行って確認する．

表 D.1: 検出確率 ($s = 1$, $h = 3$, $m = 3$, $f = 0.5$, $z^* = 9$, $\theta^* = 2$)

x_1	x_2	x_3	検出確率
9	0	0	0.94254321
0	9	0	0.94254321
0	0	9	0.94254321
8	1	0	0.92776197
8	0	1	0.92776197
1	8	0	0.92776197
1	0	8	0.92776197
0	8	1	0.92776197
0	1	8	0.92776197
7	2	0	0.92090750
⋮	⋮	⋮	⋮

x_1	x_2	x_3	検出確率
5	2	2	0.90216273
2	2	5	0.90216273
2	5	2	0.90181115
4	2	3	0.89976624
3	2	4	0.89976624
4	3	2	0.89969143
2	3	4	0.89969143
3	4	2	0.89956427
2	4	3	0.89956427
3	3	3	0.89830086

表 D.2: 閾値フローの検出確率 ($s = 1$, $h = 3$, $m = 3$, $f = 0.5$, $\theta^* = 2$)

z^*	x_1	x_2	x_3	検出確率
10	3	4	3	0.94079264
11	3	4	4	0.96617057
	4	4	3	

最後に，T_{HW} が T_{SW} の倍数ではない場合について考える．まず次のような c を定義する．

$$c = \{m(h-s) - 2 \bmod ms\} + 1$$

このとき，パケット数が x_i $(i = 1, 2, \ldots, c)$ である BW は HW 内に $\lceil (m(h-s)+1)/ms \rceil$ 回出現する．一方，パケット数が x_i $(i = c+1, c+2, \ldots, ms)$ である BW は HW 内に $\lfloor (m(h-s)+1)/ms \rfloor$ 回出現する．数値計算の結果より，閾値フローにおいて x_i $(i = 1, 2, \ldots, c)$ が x_i $(i = c+1, c+2, \ldots, ms)$ よりも大きな値を持ちやすいという傾向が，とりわけ $(m(h-s)+1)/ms$ の値が小さいときに見られた．これは BW ごとの出現回数の違いによってもたらされていると考えられる．表 D.3 は $s = 7$，$m = 1$ $(T_{\mathrm{SW}} = 7T_{\mathrm{BW}})$，$z^* = 7$，$\theta^* = 2$，$f = 0.5$ とし，h を $h = 10, 17, 24$ と変化させたときの閾値フローの分布と検出確率を示している．$h = 10$ のときは x_1，x_2，x_3 にのみパケットを配置しているのに対し，$h = 24$ では一様分布になっている．

たとえ一様分布が閾値フローの分布と異なる場合でも，その検出確率は非常に近い値を取ると推測する (とりわけ $m(h-s)+1$ の値が大きいときには推測する)．実用的な状況を考えると，$T_{\mathrm{HW}}/T_{\mathrm{SW}} = h/s$ はそこそこ大きな値になると考えられ，その場合には一様分布を用いて所与 ϵ に対して θ^* を求めればよいと結論づける．

先ほどの推論を確認するため，ランダムに生成した 10^5 パターンの MTF に対してそれぞれ 10^6 回の独立なサンプリング実験を行った．このとき，一様分布と

表 D.3: 閾値フローの検出確率 ($s = 7$, $m = 1$, $f = 0.5$, $z^* = 7$, $\theta^* = 2$)

h	x_1	x_2	x_3	x_4	x_5	x_6	x_7	検出確率
10	2	3	2	0	0	0	0	0.82739258
17	2	2	2	1	0	0	0	0.72657776
	2	2	2	0	1	0	0	
	2	2	2	0	0	1	0	
	2	2	2	0	0	0	1	
24	1	1	1	1	1	1	1	0.62347192

上限を与える $\boldsymbol{x} = (0, 0, \ldots, 0, z^*)$ の分布についても 10^6 回のサンプリング実験を行い，検出された割合を比較する．なお，実験に使用したパラメータは表 3.2 と同じとした．

図 D.1 に結果を示す．各パネルにおいて，縦軸は検出確率を，横軸は検出確率の低い順につけたランクをそれぞれ示している．左側の 3 枚のパネルは T_{HW} が T_{SW} の倍数になっている場合の，右側の 3 毎のパネルは T_{HW} が T_{SW} の倍数ではないときの結果をそれぞれ示している．一様分布は左側の倍数になっているときにはかなり上位に位置している．一方，右側の倍数になっていない方でも，検出確率自体はトップの分布と比較してもほとんど差がなく十分小さな値となっている．それゆえ，一様分布を用いてサンプルパケット数の閾値 θ^* を決定することを提案する．

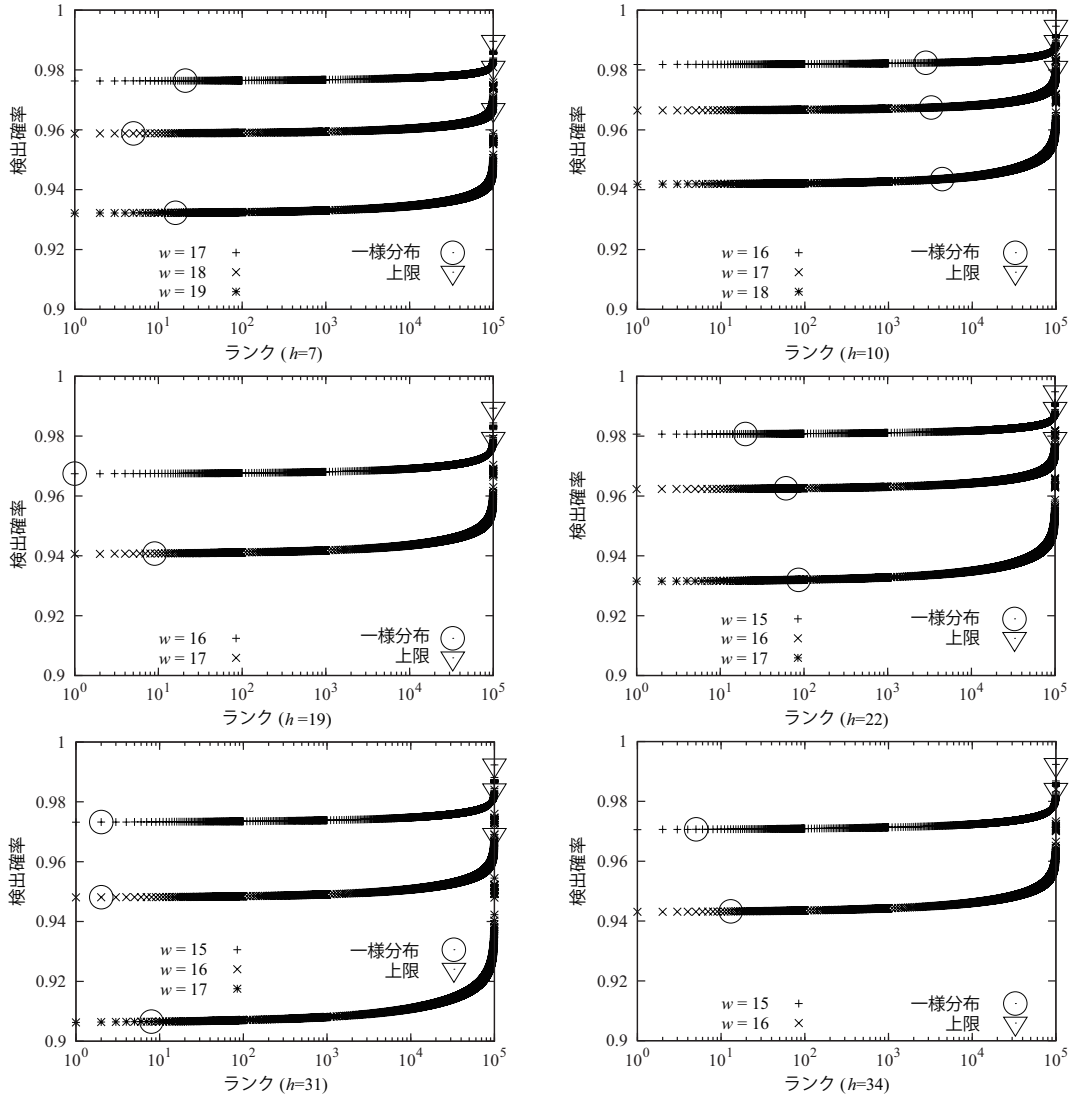

図 D.1: ランダムに生成した MTF の検出確率 ($s = 2,\ m = 3,\ z^* = 30,000,$ $f = 9.8 \times 10^{-4}$, 試行回数 $= 10^6$, 生成パターン数 $= 10^5$, $h = 4, 5, 6, 7, 8, 9$)

付 録E

問題 P^* の最適解

ここでは，問題 P^* の最適解を導く．(f, m) は問題 P^* の実行可能解とする．式 (3.11) の制約条件より，

$$G(0) < G(fC_{\max}T_{\mathrm{BW}}) \leq \frac{T_{\mathrm{HW}}}{mh} \leq \frac{T_{\mathrm{HW}}}{h} \tag{E.1}$$

となる．このとき，$G(0)$ は SW と HW のデータを処理するさいのオーバヘッドとして解釈できる．式 (3.12) の制約条件と式 E.1 の二つ目の不等式より，

$$T_{\mathrm{D\_max}} \;>\; T_{\mathrm{HW}} + \frac{T_{\mathrm{HW}}}{mh} + G(0) > T_{\mathrm{HW}} + 2G(0) \tag{E.2}$$

となる．ここで式 (E.1) と式 (E.2) は $G(0)$ に関する二つの制約条件を提供している．以降では式 3.14 が成り立つものと想定する．

式 (E.1) の二つ目の不等式と式 (E.2) の一つ目の不等式より，

$$m < \frac{T_{\mathrm{HW}}}{hG(0)}, \qquad m > \frac{T_{\mathrm{HW}}}{h\{T_{\mathrm{D\_max}} - T_{\mathrm{HW}} - G(0)\}}$$

がそれぞれ導かれる．式 (3.13) にあるように，空でない自然数の集合 $\mathcal{M}$ を定義する．式 (3.14) の想定の下では，$\mathcal{M} \neq \emptyset$ であり，固定された m に対して問題 P^* が実行可能となることは $m \in \mathcal{M}$ となることの必要十分条件である．

$G(x)$ $(x \geq 0)$ は x に関する正の狭義増加関数であるため，$G(\cdot)$ の逆関数 $G^{-1}(\cdot)$ が存在する．ゆえに，固定された $m \in \mathcal{M}$ に対して，式 (3.11) は

$$f \leq \frac{mh}{C_{\max}T_{\mathrm{HW}}} \cdot G^{-1}\left(\frac{T_{\mathrm{HW}}}{mh}\right)$$

となることを示す．一方で，式 (3.12) より

$$f \leq \frac{km}{C_{\max}T_{\mathrm{HW}}} \cdot G^{-1}\left(T_{\mathrm{D\_max}} - T_{\mathrm{HW}} - \frac{T_{\mathrm{HW}}}{mh}\right)$$

となる．したがって f は次式で表される $f = f^*(m)$ のときに最大となる．

$$f^*(m) = \frac{mh}{C_{\max}T_{\mathrm{HW}}} \cdot G^{-1}(u(m)), \qquad m \in \mathcal{M}$$

$u(m)$ は式 (3.17) で与えられる．

f が決定されると続いて式 (3.15) で定義される m^* を決定する．すなわち，全ての $m \in \mathcal{M}$ に対して $f^*(m^*) \geq f^*(m)$ となる m^* を見つける．まとめると，問題 P^* が実行可能解を持つには，式 (3.14) の想定が成り立つことが必要十分条件であり，問題 P^* の最適解は $(f^*(m^*), m^*)$ で与えられる．

付 録F

外部からの TCP ポートスキャン検出手順

第 4 章ではローカルネットワークのホストから外部に対して行われるポートスキャンの検出について議論している．ここでは，その逆向きで，外部からローカルネットワーク内のホストに対して行われるポートスキャンの検出を考える．

特定の IP アドレスに対して，宛先ポート番号を変えながら応答のあるポートを探す，垂直型ポートスキャンが行われると，その IP アドレスのホストにフロー ID の異なる多数の SYN パケットが到着する．通常，ほとんどのポートは閉じられているため，ホストは到着した SYN パケットのほとんどに対して RST/ACK パケットで返信し，それ以外のパケットは送らず，またコネクションも確立されない．したがって，ローカルネットワークから外部のネットワークへ転送されるトラヒックを計測すると，垂直型ポートスキャンを受けているホストからの，大量の RST/ACK パケットのみのフローが含まれることになる．すなわち，ローカルネットワーク内から外部へのポートスキャンを検出するときと同様に，ホストごとの RST/ACK パケットのみのフロー数を数えることで外部からの垂直型 TCP ポートスキャンを検出できると考えられる．RST/ACK パケットのみのフローを RST-only フローと呼ぶことにし，ブルームフィルタを用いたホストごとの RST-only フロー数の計測手順について図 F.1 に示す．ただし，F_{RST} と F_{Others} は RST/ACK パケット用とそれ以外のパケット用のブルームフィルタであり，C'_i $(i = 1, 2, \ldots, N_{\mathrm{LAN}})$ はローカルホストごとの RST-only フロー数のカウンタである．

各 JW の終了時点で出力される C'_i $(i = 1, 2, \ldots, N_{\mathrm{LAN}})$ の値とあらかじめ決定された閾値を比較することで，RST-only フロー数の多いホスト i をポートスキャンを受けているホストの候補として検出することが可能となる．なお，サンプリングレート $f < 1$ のパケットサンプリングを行う場合の閾値は，4.3 節の sampled-SYN-only フロー数の閾値を決定する手法における SYN パケットを RST/ACK パケットに置き換えることで同様に決定することができる．

Step 1: フィルタ F_{RST} と F_{Others}，カウンタ C'_i $(i = 1, 2, \ldots, N_{\mathrm{LAN}})$ を初期化し，新しい JW での計測を開始する．

Step 2: パケットが取得されるのを待つ．この間に JW の終了時点に達したら C'_i $(i = 1, 2, \ldots, N_{\mathrm{LAN}})$ の結果を出力後に Step 1 へ戻る．取得したパケットについて，ホスト番号 i，フロー ID j，RST パケットかどうか，の 3 点を調べ，RST パケットであれば Step 3 へ，RST パケット以外であれば Step 4 へそれぞれ進む．

Step 3: F_{RST} へフロー ID j が登録されているかどうかの問い合わせを行い，登録されていれば何もしない．未登録であれば j を F_{RST} に新たに登録し，さらに j が F_{Others} に登録されているかどうか問い合わせ，登録されていなければホスト i のカウンタ C'_i を 1 だけ増やす．Step 2 へ戻る．

Step 4: F_{Others} へフロー ID j が登録されているかどうかの問い合わせを行い，登録されていれば何もしない．未登録であれば j を F_{Others} に新たに登録する．Step 2 へ戻る．

図 F.1: ホストごとの RST-only フロー数計測手順

参考文献

[1] R. R. R. Barbosa, R. Sadre, A. Pras, R. van de Meent, "Simpleweb/University of Twente traffic traces data repository," *Technical Report TR-CTIT-10-19*, Centre for Telematics and Information Technology, University of Twente, 2010, Available: `http://doc.utwente.nl/71273`.

[2] C. Barakat, G. Iannaccone, and C. Diot, "Ranking flows from sampled traffic," *Proceedings of ACM CoNEXT '05*, pp. 188–199, 2005.

[3] D. K. Bhattacharyya and J. K. Kalita, *Network Anomaly Detection: A Machine Learning Perspective*, CRC Press, 2013.

[4] B. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Communications of the ACM*, vol. 13, no. 7, pp. 422–426, 1970.

[5] CAIDA: The Cooperative Association for Internet Data Analysis, 2011. Available: `http://www.caida.org/home/`.

[6] The CAIDA Anonymized 2009 Internet Traces - <equinix-sanjose, 20090331>. C. Walsworth, E. Aben, KC Claffy, D. Andersen, 2010. Available: <http://www.caida.org/data/passive/passive_2009_dataset.xml>.

[7] The CAIDA Anonymized 2011 Internet Traces - <equinix-chicago, 20110721>. Available: `http://www.caida.org/data/passive/passive_2011_dataset.xml`.

[8] The CAIDA Backscatter-2008 Dataset - <20080220>. Available: `http://www.caida.org/data/passive/backscatter_2008_dataset.xml`.

[9] CERT advisory CA-2001-23 continued threat of the "Code Red" worm, `http://www.cert.org/advisories/CA-2001-23.html`

[10] M. Crovella and B. Krishnamurthy, *Internet Measurement*, John Wiley & Sons, 2006.

[11] Cisco NetFlow. Available: `http://www.cisco.com/en/US/products/ps6601/products_ios_protocol_group_home.html`.

[12] C. Estan and G. Varghese, "New directions in traffic measurement and accounting: Focusing on the elephants, ignoring the mice," *ACM Transactions on Computer Systems* vol. 21, no. 3, pp. 270–313, 2003.

[13] W. Feller, *An Introduction to Probability Theory and Its Applications*, 3rd Edition, Vol. 1, John Wiley & Sons, 1968.

[14] The Internet Engineering Task Force. Available: `http://www.ietf.org`.

[15] InMon sFlow Probe. Available: `http://www.sflow.org`.

[16] J. Kurose and K. Ross, *Computer Networking: A Top-Down Approach*, sixth edition, Pearson Education, 2012.

[17] C. B. Lee, C. Roedel, and E. Silenok, "Detection and characterization of port scan attacks," *Technical Repport*, UC San Diago, 2003, `http://csweb.ucsd.edu/users/clbailey/PortScans.pdf`.

[18] J. Mai, A. Sridharan, C. Chuah, H. Zang, and T. Ye, "Impact of packet sampling on portscan detection," *IEEE Journal of Selected Areas in Communications*, vol. 24, no. 12, pp. 2285–2298, 2006.

[19] J. Mirkovic and P. Reiher, "A taxonomy of DDoS attack and DDoS defense mechanisms," *ACM SIGCOMM CCR*, vol. 34, pp.39–53, 2004.

[20] H. Moon, s. Yi, and K. Cho, "A modified multi-resolution approach for port scan detection," *Proceedings of IEEE GLOBECOM '10*, 2010.

[21] A. Müller and D. Stoyan, *Comparison Methods for Stochastic Models and Risks*, John Wiley & Sons, 2002.

[22] Nmap. Available: `http://nmap.org/`.

[23] V. Paxson, "Bro: A system for detection network intruders in real-time," *Computer Networks*, vol. 31, pp. 2435–2463 ,1999.

[24] D. Piscitello, "Conficker summary and review," May, 2010, `http://www.icann.org/en/about/staff/security/conficker-summary-review-07may10-en.pdf`

[25] V. A. Siris and F. Papagalou, "Application of anomaly detection algorithms for detecting SYN flooding attacks," *Computer Communications*, vol. 29, pp. 1433–1442, 2006.

[26] tcpdump. Available: <http://www.tcpdump.org>.

[27] S. Venkataraman, D. Song, P. B. Gibbons, A. Blum, "New streaming algorithms for fast detection of superspreaders," *Proceedings of Network and Distributed System Security Symposium (NDSS)*, 2005.

[28] N. Weaver, S. Staniford, V. Paxson, "Very fast containment of scanning worms," *Proceedings of the 13th USENIX Security Symposium*, 2004.

[29] WIDE: the MAWI Working Group, 2010. Available: <http://www.wide.ad.jp/project/wg/mawi.html>.

[30] Wireshark. Available: <http://www.wireshark.org>.

[31] H. Zhang, X. Zhu, and W. Guo, "TCP portscan detection based on single packet flows and entropy," Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human, pp.1056–1060, 2009.

[32] C. C. Zou, W. Gong, D. Towsley, and L. Gao, "The monitoring and early detection of Internet worms," *IEEE/ACM Transactions on Networking*, vol. 13, no. 5, pp. 961–947, 2005.

研究業績

雑誌論文

⟨1⟩ T. Kudo and T. Takine,
"Design of a sliding window scheme for detecting high packet-rate flows via random packet sampling," *Computer Networks*, vol. 55, no. 6, pp. 1351–1363, April, 2011.

⟨2⟩ T. Kudo and T. Takine,
"On-line detection of persistently high packet-rate flows via a sliding window scheme with random packet sampling," *International Journal of Network Management*, vol. 24, no. 1, pp.28–47, January/February 2014.

国際会議

⟨1⟩ T. Kudo, T. Morita, T. Matsuda, and T. Takine,
"PCA-based robust anomaly detection using periodic traffic behavior," *Proceedings of the 1st IEEE Workshop on Traffic Identification and Classification for Advanced Network Services and Scenarios (TRICANS)*, pp. 1350–1354, Budapest, Hungary, June 9-13, 2013.

⟨2⟩ T. Kudo, H. Hamagaki, and T. Takine,
"Design of a portscan detection scheme with random packet sampling," *Proceedings of the 6th International Conference on Security Technology (SecTech 2013), Advanced Science and Technology Letters (ASTL)*, vol. 29, pp. 1–6, Jeju Island, Korea, November 21-23, 2013.

受賞

⟨1⟩ Best Paper Award of the 6th International Conference of Security Technnology (SecTech), November 21-23, 2013.

特許

⟨1⟩ 出願人：日本電信電話株式会社，国立大学法人大阪大学
発明者：上山憲昭, 川原亮一, 原田薫明, 滝根哲哉, 工藤隆則
発明名称：“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”
出願番号：特許出願 2007-326789
公開番号：特許公開 2009-152712

⟨2⟩ 出願人：日本電信電話株式会社，国立大学法人大阪大学
発明者：上山憲昭, 川原亮一, 森達哉, 滝根哲哉, 工藤隆則
発明名称：“高パケットレートフロー検出装置
及び高パケットレートフロー検出手法”
出願番号：特許出願 2010-160924
公開番号：特許公開 2012-23629

研究会 (査読なし)

⟨1⟩ 工藤隆則, 滝根哲哉,
“高パケットレートフローのオンライン検出手法,”
信学技報, vol. 107, no. 525, IN2007–165, pp. 37–42, 2008 年 3 月.

⟨2⟩ 工藤隆則, 滝根哲哉,
“持続的高パケットレートフローのオンライン検出手法,”
信学技報, vol. 110, no. 449, IN2010–179, pp. 223–228, 2011 年 3 月.

⟨3⟩ 森田達也, 工藤隆則, 松田崇弘, 滝根哲哉,
“トラヒックの周期性を利用した PCA による異常トラヒック検出,”
電子情報通信学会総合大会, B-6-79, 2013 年 3 月.

⟨4⟩ 松田崇弘, 森田達也, 工藤隆則, 滝根哲哉,
“異常トラヒック検出のためのロバスト主成分分析手法,”
信学技報, vol. 113, no. 292, NS2013–128, pp. 71–76, 2013 年 11 月.