

Network-wide Packet Behaviorに基づくパケット分類を用いた ネットワーク故障箇所の推定法

渡部 康平[†] 井上 武^{††} 水谷 后宏^{††} 間野 暢^{††} 明石 修^{††}
中川 健治[†]

[†] 長岡技術科学大学 大学院 工学研究科 〒940-2188 新潟県 長岡市 上富岡町 1603-1

^{††} NTT 未来ねっと研究所 〒239-0847 神奈川県横須賀市光の丘 1-1

E-mail: [†]{k.watabe,nakagawa}@nagaokaut.ac.jp,

^{††}{inoue.takeru,mizutani.kimihiro,mano.toru,akashi.osamu}@lab.ntt.co.jp

あらまし ネットワークの大規模・複雑化に伴い、機器がアラートを発しないにも関わらず QoS の低下が起こるサイレント故障が大きな問題となってきた。サイレント故障の問題では、オペレータの負荷軽減のため、故障の発生検知、及び故障箇所の特定が重要な課題となる。ネットワークの故障検知、及び故障箇所特定を実現する技術としては、ネットワークトモグラフィなどが挙げられるが、既存技術ではリンク故障など単純な故障しか検知することができない。本稿では、ポートの通過パケットをネットワーク内での振る舞い毎に分類してカウントし、パケットロス率を計測することで、高度なネットワーク機能の故障に由来する QoS 低下を検知し、故障箇所の特定を実現する技術を提案する。実ネットワークのコンフィギュレーションデータを利用した検証により、限られたポートでのパケットカウントから、高い精度で故障箇所を特定できることを示す。

キーワード 故障箇所推定, ネットワーク監視, パケット分類, パケットロス率, サイレント故障

An Analysis Method of Network Troubles with Path Identification

Kohei WATABE[†], Takeru INOUE^{††}, Kimihiro MIZUTANI^{††}, Toru MANO^{††}, Osamu AKASHI^{††},
and Kenji NAKAGAWA[†]

[†] Graduate School of Engineering, Nagaoka University of Technology
Kamitomiokamachi 1603-1, Nagaoka, Niigata, 940-2188 Japan

^{††} NTT Network Innovation Laboratories, Hikarigaoka 1-1, Yokosuka, Kanagawa, 239-0847 Japan

E-mail: [†]{k.watabe,nakagawa}@nagaokaut.ac.jp,

^{††}{inoue.takeru,mizutani.kimihiro,mano.toru,akashi.osamu}@lab.ntt.co.jp

Abstract As networks get larger and more complex, network troubleshooting becomes a very important issue for network operators. Though an efficient technique to find a cause of a trouble is useful for network troubleshooting, most of the existing techniques like network tomography can find only link trouble. In this paper, we analyze the accuracy of finding in real network data when we find a cause of a trouble with path identification techniques.

Key words Trouble specification, Network monitoring, Path identification, Packet loss, Network troubleshooting

1. はじめに

近年, IoT や M2M の登場, クラウドやネットワーク仮想化の普及, アプリケーションの多様化により, ネットワークはますます大規模・複雑化している。ネットワークに接続する端末数は飛躍的に増加し, 様々な端末に起因するトラブルに対応するオペレータの負荷は増大している。加えて, ルーターやファ

イアーウォールなど様々なネットワーク機能の仮想化により, ネットワーク構成を柔軟に変更可能になった反面, 頻繁な構成変更により, ネットワークの全容を把握することが困難になってきている。また, 多様化するアプリケーションからの品質要求やセキュリティ要求に応えるため, QoS 制御やアクセス制御など, ネットワークに求められる機能はますます広がっており, その傾向は今後も続くと考えられる。

管理・運用するネットワークが大規模・複雑化するに連れ、ネットワークにおけるサイレント故障が大きな問題となってきた。サイレント故障とは、ネットワーク機器内のバグや設定ミス、内部プロセッサの故障などにより、QoSが低下しているにもかかわらず、装置がアラートを発しない故障である。ネットワーク機能の仮想化に伴い、機能のソフトウェア化が進むことで、設定変更や高度な機能の実装は容易になったが、設定ミスやプログラムのバグは発生しやすくなってきている。例えば、設定ミスにより、リンク帯域が著しく小さな値に設定されてしまった場合、ユーザのQoSは低下するが、システムはアラートを発することはない。長期に渡り複数の管理者によりメンテナンスされることで、このような設定ミスは発生しやすくなる。

サイレント故障の問題では、いかに故障の発生を検知し、故障箇所を特定するかが重要である。サイレント故障は、システムがアラートを発しないため、故障の発生を検知することが困難であり、オペレータは個別に機器を監視するなどして対応しているのが現状である。このような個別の監視では、ネットワークの大規模化に伴い、監視対象箇所が増加し、オペレータの対応が困難になる。また、仮にトラフィック監視やユーザのクレームからQoSの低下を検知できたとしても、QoS低下の原因を特定することは難しい。

ネットワークの故障検知、及び故障箇所特定を実現する技術としては、ネットワークトモグラフィがあるが、ネットワークトモグラフィではフォワーディングに関する故障やリンク故障のみで、アクセス制御やQoS制御などより高度なネットワーク機能の故障を調べることができない。多くのネットワークトモグラフィの研究[1], [2]では、プローブパケットを送信することで計測したエンド・ツー・エンドのQoS情報から、トポロジとフォワーディングの情報を用いてリンク毎のパケットロス率などを推定することで、リンク単位でQoS低下の原因を特定している。しかし、トポロジとフォワーディングに関する情報のみを使ってプローブパケットを構成しても、アクセス制御やQoS制御に関する機能の故障に由来するQoS低下をプローブパケットが経験できるとは限らない。

また、アクセス制御などより高度なネットワーク機能に関する故障を検知し、故障箇所を特定する技術としては、Automatic Test Packet Generation (ATPG) [3]があるが、プローブパケットを送出して計測を行うため、計測のオーバーヘッドがある上、ユーザトラフィックが受ける品質を代表しているとは限らない。ATPGでは、フォワーディングに加えて、アクセス制御などに関する情報を用いてプローブパケットを構成し、プローブパケットの到達性を調べることで、QoS低下を招く原因となっている故障箇所を特定する。前述の帯域設定のミスのように一部のパケットのみがロスするようなケースでは、ユーザのパケットがロス率が增加しても、プローブパケットがロスするとは限らない。

本稿では、Atomic Predicate Verifier (APV) [4]のコンセプトに基づき通過パケットをnetwork-wide packet behavior(ネットワーク全体でのパケットの振る舞い)毎に分類して、パケットロス率を計測することで、サイレント故障の検知、及び故障箇所

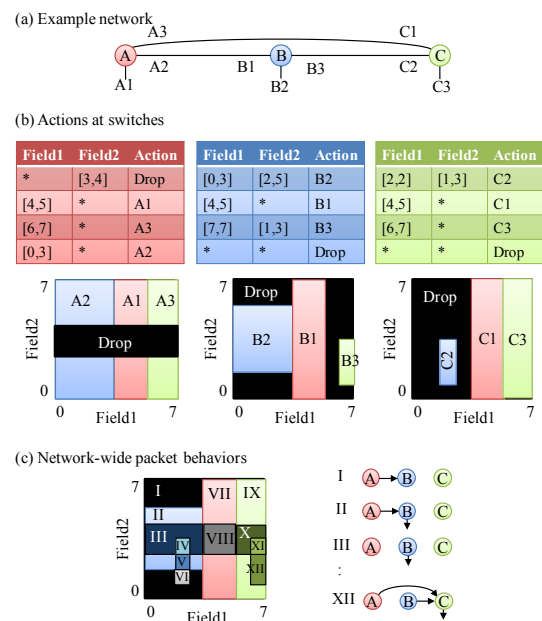


Fig. 1 APVに基づくパケットヘッダの分割とAP番号毎のパケットの振る舞い

所の特定を実現する技術を提案する。我々の手法では、プローブパケットを利用せず、ネットワーク内の各ポートにおいて通過パケットをカウントすることでパケットロス率を計測するため、ユーザトラフィックのQoSを直接監視する。APVによりパケットのヘッダ空間を漏れ無く分類するため、ネットワークのトランジットトラフィックをすべて監視することができ、限られたカウンタ数でも理論的にQoS低下を見逃すことがない。

本稿の構成は以下のとおりである。まず2節において、故障箇所の特定のための重要な要素技術となるパケット分類とパケットカウントの方法について説明する。次に3節で、故障の検知、及び故障箇所の特的手法について述べ、4節で故障箇所の特精度について評価する。最後に、5節で、本稿のまとめと今後の展望について述べる。

2. APによるパケット分類とパケットのカウント

本稿で提案する故障箇所特定技術では、ネットワーク内の各スイッチのポートにおいて、通過するパケットを振る舞い毎に分類してカウントすることで、振る舞い毎のパケットロス率を計測するため、パケット分類の技術が重要となる。本節では、本稿で提案する故障箇所特定の要素技術となるパケット分類の技術について述べる。

Atomic Predicate Verifier (APV) [4]では、各スイッチでのパケットの扱い毎にヘッダ空間を分割した部分空間 (predicate) の情報から、network-wide packet behavior 毎に分割した部分空間 (Atomic Predicate; AP) を得るアルゴリズムを提案している。Fig. 1は、三つのスイッチからなるネットワークにおけるヘッダ空間の分割の例である。ヘッダ空間は、3 bit のフィールド二つからなり、計6 bit としている。それぞれのフィールドの値に依存して各スイッチのアクションは異なり、Fig. 1(b)のようにヘッダ空間が分割されている。predicateはFig. 1(b)

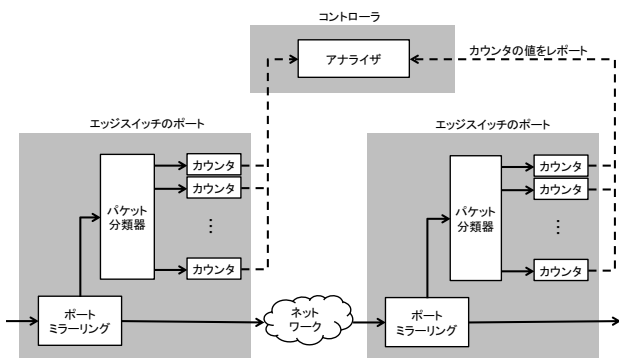


Fig. 2 提案する故障箇所特定法のシステム構成

の表の各行に対応し、パケットのヘッダが含まれる predicate 次第で各スイッチのアクションが決まる。すべてのスイッチでの predicate を考慮に入れると、ネットワーク全体で捉えた場合のパケットの振る舞いは、predicate の組み合わせで Fig. 1 のように 12 種類あり、ヘッダ空間は I~XII の AP に分割される。AP 番号が異なるパケットは、ネットワーク内での振る舞いが異なるため、各スイッチにおいて Forwarding Information Base (FIB) や Access Control List (ACL), Priority Queueing (PQ) などで記述されたアクションが少なくとも一つは異なる。つまり、AP 番号が異なれば、パケットがキューイングされるキューが少なくとも一つは異なり、経験する QoS も異なる。

文献 [5] では、Multi-valued Decision Diagram (MDD) と呼ばれるデータ構造により、AP 番号に基づくパケット分類をメモリ上で高速に実施することを可能にした。文献 [5] の技術を利用することで、10Mpps の速度でパケットを分類できるため、各スイッチのポートを通過する多数のパケットを AP 番号毎にカウントすることができる。

3. AP 毎のパケットカウントに基づく故障箇所特定

本稿で提案する故障箇所特定技術では、ネットワークの各ポートにおいて、通過パケットを AP 番号毎にカウントすることでパケットロス率を計測し、パケットロス率の情報から故障箇所を predicate 単位で特定する。提案する故障箇所特定法のシステム構成の概要を Fig. 2 に示す。ネットワーク内の各ポートを通過するパケットを MDD を利用したパケット分類器で分類し、カウンタが記録したパケット数を定期的にアナライザに集約し、集約した情報を基に故障原因となっている predicate を特定する。パケットのカウントを行うポートは、すべてのエッジスイッチの外部と繋がるポートとネットワーク内部の一部のポートであり、ネットワーク内部でカウントを行うポート数は最小限に留める。本節では、ネットワーク内部でカウントを行うポート数の最小化しつつ、集約したカウンタの情報から故障原因となっている predicate を特定する方法について説明する。

3.1 ネットワークエッジのポートのパケットカウントから故障箇所特定する方法

ネットワーク内で適応され得る predicate の全集合を $R_\Omega = \{r_j | j = 0, 1, \dots\}$ として、AP 番号が i のパケットがネット

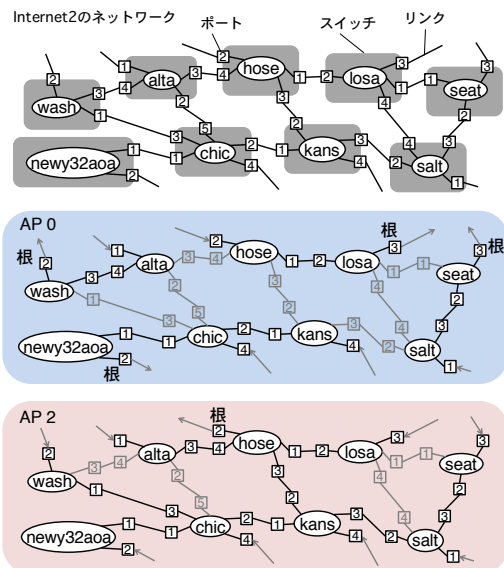


Fig. 3 ネットワークとツリーの例

ワーク内で受ける predicate の集合を behavior $B_i \subset R_\Omega$ と定義すると、 B_i に含まれる FIB の情報からパケットが通過するポートとスイッチのツリー t_i^k ($k = 0, 1, \dots$) を抽出することができる。Fig. 3 は Internet2 におけるネットワークのコンフィギュレーションデータ [6] を基にパケット分類を行い、 B_0 、及び B_2 に含まれる FIB 情報からツリーを抽出した例である。AP 番号 2 の AP に分類されるパケットは、各スイッチの外部と繋がるポートから入り、スイッチ hose の 2 番ポートから外部へ出て行く。AP 番号 0 の例のように、一つの behavior から複数のツリーが抽出される場合もある。

B_i に含まれる predicate が適用される位置を考慮して、各 predicate をツリーの辺に対応付けることで、各 AP 番号のパケットが適用される predicate とその順番がわかる。本稿では、抽出したツリーの中からトランジットトラヒックが流れるツリーのみを抜き出して利用し、トランジットトラヒックが流れるツリーの辺に対応付けられる predicate のみが、故障原因として識別対象となる。predicate をツリーの辺に対応させるときには、パケットカウントと predicate 適用の順番に留意する必要がある。あるポートで適用される predicate がパケットカウントよりも前に適用される場合は、そのポートを表す頂点の葉側に繋がる辺に predicate を対応させ、後に適用される場合は根側の辺に対応させる。以降では、out ポートにおいては predicate の適用後、in ポートにおいては predicate の適用前にパケットカウントが行われるとして検証を行うが、実装に合わせて柔軟に対応することが可能である。

トランジットトラヒックが流れるツリーの葉に位置するネットワークエッジのポートで単位時間あたりに通過したパケットの総数とツリーの根に位置するネットワークエッジのポートにおける通過パケット数をカウントし、比較することで、ツリー毎のロス率 $\{l_{t_i^k}\}$ を計測することができ、品質劣化が発生しているツリーを特定することができる。各ツリーの流入パケット数の総和と流出パケットの数を差を一定時間毎に取り、パケッ

トロス率 $\{l_{t_i}^k\}$ をツリー毎に計測する。パッシブ計測によるポートのパケットカウン트의情報からパケットロス率を推定する技術は、Gu ら [7] により提案されており、本稿でも同様の方法を利用する。計測したパケットロス率に対してしきい値 δ を設けて、 $l_{t_i}^k > \delta$ となるツリーを品質劣化ツリーと定義する。

上記のように特定した品質劣化ツリーのパターンから、品質劣化の原因となっている故障箇所の predicate を絞り込むことができる。predicate r_j ($j = 1, 2, \dots$) が対応付けられる辺を含むツリーの集合を T_{r_j} とし、predicate r_j に関する故障により T_{r_j} のすべてのツリーのパケットロス率が δ 以上となると仮定する。品質劣化ツリーの集合を $T_{\text{loss}} = \{t_i^k \mid l_{t_i}^k > \delta\}$ 、ツリー t_i^k の辺に対応付けられた predicate の集合を $R_{t_i^k}$ とすると、故障原因と予想される predicate は $\bigcap \{R_{t_i^k} \mid t_i^k \in T_{\text{loss}}\} - \bigcup \{R_{t_i^k} \mid t_i^k \in \bar{T}_{\text{loss}}\}$ に絞られる。ただし、この絞り込みにより故障原因となる predicate が一意に特定できるとは限らないことに注意する。この絞り込みのアプローチは、ATPG [3] における故障原因の絞り込みと同様のアプローチである。

実ネットワークにおいては、すべてのトランジットトラヒックに関するツリーについてパケットが十分に流れるとは限らないため、一部のツリーの品質劣化情報は利用できない可能性があり、故障原因を十分に絞り込めない可能性もある。品質劣化を判断するためのトラヒックが十分でないツリーについては、 T_{r_j} から取り除き、パケットロス率は δ 以下であるとして上記の絞り込みを行う。取り除いた結果、 $T_{r_j} = \emptyset$ となる predicate r_j については故障が発生しても、その発生を検知することができないが、影響を受けるトラヒックも少ない。

3.2 ネットワークエッジのポート以外のパケットカウント

ネットワークエッジのポートにおけるパケットカウントにより計測した品質劣化ツリーのパターンでは識別することができない故障箇所の predicate を特定するためには、ネットワークエッジのポート以外でもパケットのカウントを行う必要がある。パケットカウントをツリーの中間ポートでも行うようにすると、カウントを行うポートを挟んでツリーを二つのツリーに分割して、それぞれのツリーでパケットロス率を計測することができる。よって、品質劣化原因がカウントを行うポートを挟んで上流側にあるか、下流側にあるかを区別することができる。例えば、Fig. 3 に示した Internet2 の AP 番号 2 について、kans の 3 番ポートでパケットカウントを行うと、品質劣化の原因がスイッチ seat, salt で適用される predicate にあるか、それ以外のスイッチのポートで適用される predicate にあるかを識別することができる。

必要最小限のポートでのパケットカウントで効率的な故障箇所の特定を実現するためには、パケットカウントを行うポートとパケットカウントを行うことで識別可能になる predicate の関係を明らかにする必要がある。 $R_{T_{\text{loss}}}$ に含まれる predicate の任意のペア (r_x, r_y) について、 T_{loss} に含まれるツリー上で r_x, r_y に対応付けられる辺の中間に位置するポートの集合を $S_{(r_x, r_y)}$ とする。ただし、 r_x, r_y に対応付けられる辺が同じ場合など、中間にポートがない場合は、 $S_{(r_x, r_y)}$ は空集合とする。

ネットワークエッジのポートにおいてパケットカウントを行っ

Table 1 検証に利用したネットワークの統計情報

	Internet2	Stanford	Purdue
スイッチ数	9	16	1,646
エッジスイッチ数	9	12	92
利用ポート数	157	114	3,541
ネットワークエッジポート数	116	39	482

ている状態から、パケットカウントを行うポートを必要最低限の数だけ追加する問題を考えると、集合被覆問題 (Minimum set covering problem) として定式化することができる。ここで、ポート s_l ($l = 1, \dots, n$) が含まれる $S_{(r_x, r_y)}$ の (r_x, r_y) からなる集合 $P_{s_l} = \{(r_x, r_y) \mid s_l \in S_{(r_x, r_y)}\}$ を定義すると、 P_{s_l} はポート s_l でパケットカウントを行った際に、故障原因が識別可能となる predicate ペアの集合である。 $\{P_{s_l}\}$ の和集合を $P_\Omega = \bigcup_{l=1}^n P_{s_l}$ とすると、識別可能な故障原因となる predicate を最大化しつつ、パケットカウンティングを行うポートを最小化する問題は、 $\{P_{s_l}\}$ のうち P_Ω をすべてカバーする最小の部分集合を求める集合被覆問題であることがわかる。集合被覆問題は一般には NP 困難であることが知られているため、規模が大きい場合には貪欲法による解 [8] を利用する必要がある。集合被覆問題は貪欲法により良い解が得られることが知られており、今回の場合、近似比は $H(P_{s_l})$ となる [8]。ここで $H(x)$ は調和関数を表す。

4. 実ネットワークのデータを用いた故障箇所特定精度の評価

提案法による故障箇所の特定精度を調査するため、Internet2, Stanford, Purdue の三つの実ネットワークのコンフィギュレーションデータを用いて検証を行った。検証に利用したネットワークの規模は、Table 1 の通りである。三つのネットワークのデータセットのうち、Internet2 については FIB のみ、Stanford は FIB と ACL, Purdue については ACL のデータが含まれている。本手法は FIB のデータを基にしてツリーを構成するため、Purdue のデータセットにはトポロジ情報を基に最短経路を算出して FIB を追加した。

4.1 ネットワークエッジのポートでのパケットカウントに基づく故障特定精度

まず、Internet2 の FIB, Stanford と Purdue の FIB, Stanford の FIB/ACL, それぞれの情報を基に AP を構成し、ネットワークエッジのポートのみのパケットカウントで故障箇所の特定を行った場合の特定精度を検証した。各ネットワークについてツリーを抽出し、識別対象となる predicate 数を算出した結果を Table 2 に示す。Purdue の FIB/ACL については、AP 数が 70 万を超えるためツリーを構成することができなかった。識別対象外の predicate には、FIB の drop, ACL の deny に関する predicate などが含まれており、トランジットトラヒックのツリーではないため利用されないツリーの大部分は、ツリーの根に繋がる辺に drop や deny に関する predicate が対応付けられているツリーである。故障箇所の特定を行った結果、Internet2 の FIB については、すべてのツリーに関するパ

Table 2 各ネットワークのツリー抽出結果

	Internet2 (FIB)	Stanford (FIB)	Purdue (FIB)	Stanford (FIB/ACL)
AP 数	219	306	706,769	3,069
利用する AP 数	180	218	142	2,179
ツリー数	446	4,425	184,476	156,405
利用するツリー数	305	389	510	4,058
predicate 数	175	146	4,285	437
識別対象 predicate 数	140	53	918	172

ケットロス率の情報が得られれば、ネットワークエッジのポートにおけるパケットカウントにより、故障原因として一意に特定できない predicate は存在しないことが分かった。一方、Stanford の FIB 及び FIB/ACL, Purdue の FIB については、ネットワークエッジのポートのみのパケットカウントでは原因となる predicate を一意に特定できない故障が存在することが確認できた。それぞれの結果について、識別対象の predicate が一様の確率で一つだけ故障するとして、故障原因として絞り込まれる predicate 数の分布を Fig. 4 に示す。Table 1 に示す通り、Internet2 ではすべてのスイッチがエッジスイッチであり、ネットワークエッジのポートが全体に占める割合も他の二つのネットワークに比べて非常に高い。対して Stanford や Purdue は、ネットワークエッジのポート数が全体に占める割合が小さく、ネットワークエッジのポートだけのカウントでは、得られる情報が限定的になってしまうため、故障原因として一意に特定できない predicate が存在してしまっている。

Fig. 4 の結果を見ると、Stanford の FIB/ACL や Purdue の FIB のように、predicate 数、スイッチ数の意味で規模が大きいネットワークほど、ネットワークエッジのポートでのパケットカウントの情報だけでは、故障箇所を特定することが難しいことがわかる。Fig. 4 の横軸 1 に対する縦軸の値が、故障箇所を一意に特定できる確率に相当する。Stanford の FIB では、77% の確率でネットワークの故障箇所を一意に特定できるが、Purdue の FIB では 53%、Stanford の FIB/ACL では 12% の確率でしか故障箇所を一意に特定できない。Purdue の FIB では、最悪の場合、27 個までしか故障原因と疑われる predicate を絞り込むことができない。Stanford の FIB/ACL では、故障原因と疑われる predicate を一意に特定できる確率は 12% に過ぎないが、2 個以下に絞り込める確率は 63% に飛躍的に向上する。これは、ツリー上の同じ辺に対応付けられる FIB と ACL の predicate がそれぞれあり、そのいずれかが故障した場合にももう一方の故障と区別することができないためである。

4.2 一部のツリーの情報が利用できない場合の故障特定精度

さらに、上記で検証した四つのケースについて、ランダムに選択したツリーの品質劣化情報が利用できないと仮定して、ネットワークエッジのポートのみのパケットカウントで故障箇所を特定を行った場合の特定精度を検証した。前述の通り、実ネットワークにおいては、すべてのツリーについてパケットロス率を計測するために十分なパケットが流れるとは限らない。トランジットトラヒックについてのツリーの内、40%のツ

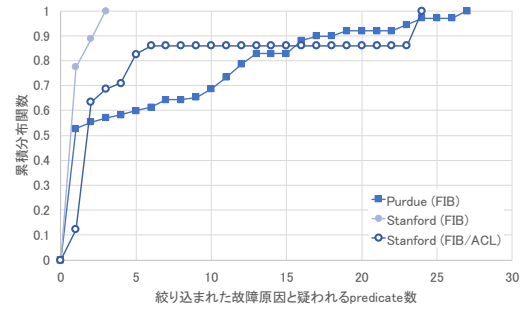


Fig. 4 ネットワークエッジのポートでのパケットカウントのみから故障箇所特定した場合に絞り込まれる故障原因と疑われる predicate 数の分布

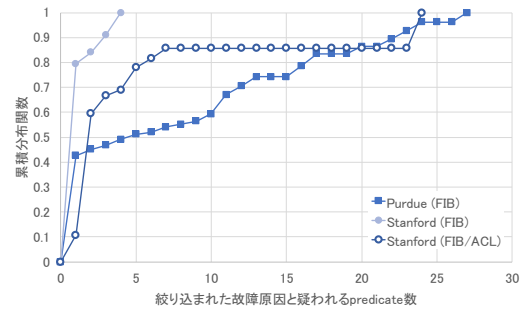


Fig. 5 ネットワークエッジのポートでのパケットカウントのみから故障箇所特定した場合に絞り込まれる故障原因と疑われる predicate 数の分布 (ツリーの 40% のデータが利用不可)

リーの情報が利用できないとした場合に、ネットワークエッジのポートのみのパケットカウントで絞り込まれる故障原因と疑われる predicate 数の分布を導出した。上述の結果と同様に、識別対象の predicate は一様の確率で一つだけ故障するとした。Internet2 の FIB については、40% のツリーの情報が利用できないに関わらず、故障原因として一意に特定できない predicate は存在しないことが分かった。Internet2 の FIB 以外の結果を Fig. 5 に示す。図から、40% のツリーのパケットロス率が利用できないに関わらず、いずれの結果も Fig. 4 と大きな差はない。これは、識別対象の predicate 数に対してツリー数が十分に多く、多くの predicate は複数のツリーの辺に対応付けられているためである。今回利用したデータでは、ツリーの構造は同じでも、異なる AP に属するツリーが多数存在するため、一部のツリーにパケットが流れなくとも、他の AP のツリーでトラヒックが流れていれば、パケットロス率の情報を利用することができる。

4.3 パケットカウントを行うポートの最適化による故障特定精度向上

最後に、ネットワークエッジのポート以外のポートでパケットカウントを行った場合の故障箇所特定精度の向上効果について検証する。Stanford と Purdue の FIB, Stanford の FIB/ACL, それぞれについて、3 節で示した集合被覆問題を解くことで、パケットカウントを行うポートの数を最適化した。なお、パケットロス率に関する情報は、すべてのツリーについて取得可能であるとして検証を行っている。Internet2 の FIB, Stanford と

Table 3 各ネットワークの検証結果

		Internet2 (FIB)	Stanford (FIB)	Purdue (FIB)	Stanford (FIB/ACL)
エッジポートのみでカウント	カウントするポートの割合	116/157	39/114	482/3,541	39/114
	故障原因を一意に特定できる確率	100%	77%	53%	12%
	故障原因を2箇所以下に特定できる確率	100%	89%	55%	63%
エッジポート以外でもカウント	カウントするポートの割合	116/157	45/114	746/3,541	43/114
	故障原因を一意に特定できる確率	100%	100%	100%	15%
	故障原因を2箇所以下に特定できる確率	100%	100%	100%	69%

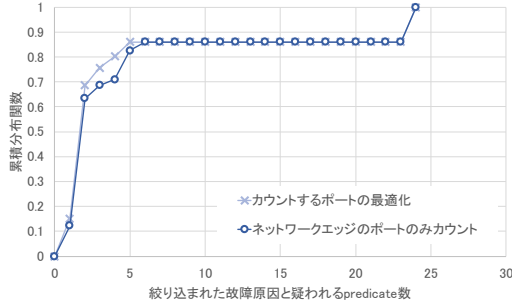


Fig. 6 ネットワークエッジのポート以外のポートでパケットカウントを行った場合に絞り込まれる故障原因と疑われる predicate 数の分布

Purdue の FIB については、総当り探索及び貪欲法、それぞれの方法で最適化を行ったが、同じ解が得られた。一方、Purdue の FIB については、規模が大きいため、総当り探索による解が得られなかったため、貪欲法の解のみを得た。それぞれのネットワークにおけるパケットカウントを行うポートの割合と、故障箇所を一意に特定できる確率をまとめたものを Table 3 に示す。最適化の結果、Stanford と Purdue の FIB については、ネットワークエッジのポート以外にそれぞれ 6 ポート、264 ポートでパケットカウントを行うことで、すべての識別対象 predicate の故障について、一意に故障箇所を特定できることが分かった。一方、Stanford の FIB/ACL の結果では、ネットワークエッジのポート以外に 4 ポートでパケットカウントを行うことで、ネットワークの全ポートでパケットカウントを行うのと同等の故障箇所特定を行うことができることが分かった。Fig. 6 は、ネットワークエッジのポートのみでパケットカウントを行った場合と最適なポートでパケットカウントを行った場合の故障原因と疑われる predicate 数の分布の比較を示す。図から、累積分布関数が上にシフトし、特定の絞り込みがより細かくなっていることが確認できる。カウントを行うポートをわずか 4 ポート増加させただけでも関わらず、故障箇所が一意に特定できる確率は 15%、2 箇所以下に特定できる確率は 69%に改善している。

5. 終わりに

本稿では、ネットワーク中の各スイッチのポートで通過パケットを振る舞い毎にカウントして、パケットロス率を計測することで、ネットワークの故障箇所を検知し、特定する技術を提案した。実ネットワークのコンフィギュレーションデータを利用して、故障箇所の特定精度を検証することで、ネットワー

クエッジのポートでパケットカウントを行うと、FIB に関する故障箇所を一意に特定できることを示した。また、FIB と ACL に関する故障箇所を特定する場合には、69%の確率で故障箇所を 2 箇所以下に絞り込むことができることが分かった。

今後の課題としては、複数箇所の同時故障や AP 数が膨大になる巨大なネットワークへの対応が挙げられる。今回の検証では、故障箇所は高々 1 箇所であると仮定して検証を行ったが、実ネットワークにおいては、必ずしもこの仮定は満たされない。そのため、複数箇所の同時故障を前提に、故障箇所を限定する方法が求められる。また、Purdue の FIB/ACL のデータのように、AP 数が膨大になる大規模なネットワークにおいては、ツリーを合理的な時間で構成することが難しいため、より大規模なネットワークにも対応できるよう、ツリーの構成法を改善する必要がある。

Acknowledgment

本研究の一部は、JSPS 科研費 26880008 の助成を受けたものである。

References

- [1] M. Coates and R. Nowak, "Network Loss Inference Using Unicast End-to-End Measurement," Proceedings of the 13th ITC Specialist Seminar on IP Traffic, Measurement, and Modeling, Monterey, CA, USA, Sept. 2000.
- [2] Y. Gu, G. Jiang, V. Singh, and Y. Zhang, "Optimal Probing for Unicast Network Delay Tomography," Proceedings of the 29th IEEE International Conference on Computer Communication (INFOCOM 2010), pp.1244–1252, IEEE, Shanghai, China, March 2010.
- [3] H. Zeng, P. Kazemian, G. Varghese, and N. McKeown, "Automatic Test Packet Generation," IEEE/ACM Transactions on Networking, vol.22, no.2, pp.554–566, April 2014.
- [4] H. Yang and S.S. Lam, "Real-time Verification of Network Properties Using Atomic Predicates," 2013 21st IEEE International Conference on Network Protocols (ICNP 2013), pp.1–11, Goettingen, Germany, Oct. 2013.
- [5] T. Inoue, T. Mano, K. Mizutani, S.-i. Minato, and O. Akashi, "Rethinking Packet Classification for Global Network View of Software-Defined Networking," Proceedings of 2014 IEEE 22nd International Conference on Network Protocols (ICNP 2014), pp.296–307, Raleigh, NC, USA, Oct. 2014.
- [6] "Internet2 - Forwarding Information Base". <http://vn.grnoc.iu.edu/Internet2/fib/>
- [7] Y. Gu, L. Breslau, N. Duffield, and S. Sen, "On Passive One-way Loss Measurements Using Sampled Flow Statistics," Proceedings of the 28th IEEE International Conference on Computer Communication (INFOCOM 2009), pp.2946–2950, Rio de Janeiro, Brazil, April 2009.
- [8] V.V. Vazirani, Approximation Algorithms, Springer, 2013.