

OpenFlowにおけるリンク毎パケット遅延変動の監視と 劣化リンク特定の効率化

永田 隼也[†] 鶴 正人[†]

[†]九州工業大学情報工学府 〒820-8502 福岡県飯塚市川津 680-4

E-mail: †{syunya@infonet, tsuru@}cse.kyutech.ac.jp

あらまし 近年、クラウドサービスやサーバの仮想化を支える技術として OpenFlow が普及しつつある。それに伴い、OpenFlow を用いた大規模なネットワークの品質維持のために、アクティブ計測による実時間での全リンクの性能計測・状態監視技術が研究されている。先行研究では OpenFlow ネットワークの通信品質の指標として各リンクのパケットロス率に着目していた。本稿では、通信品質の指標として遅延変動に着目し、各リンク・双方向のパケット遅延変動を効率的に監視する手法を提案する。提案手法では、計測端末から計測用パケットを一定時間間隔で全リンクを双方向に通過するようにマルチキャスト転送し、各スイッチでは計測用パケットの到着時間間隔の統計量を取得することで遅延変動を推定する。また、コントローラーがスイッチから観測情報を取得する順序を工夫することにより、遅延変動が大きいリンクを迅速に効率よく特定する手法を提案する。その有効性を大規模トポロジ上のシミュレーションによって示した。また、計測端末の位置による影響も調査した。

キーワード OpenFlow, ネットワーク監視, アクティブ計測, マルチキャスト, パケット遅延変動

OpenFlow-based packet delay variation monitoring per link to efficiently locate deteriorated links

Shunya NAGATA[†] and Masato TSURU[†]

[†] Computer Science and Systems Engineering, Kyushu Institute of Technology Kawazu 680-4, Iizuka-shi,
Fukuoka, 820-8502 Japan

E-mail: †{syunya@infonet, tsuru@}cse.kyutech.ac.jp

Abstract In recent year, OpenFlow is becoming popular as a technology supporting cloud services and server virtualizations. Accordingly, in order to maintain a high quality of large-scale OpenFlow networks, active measurement-based real-time performance and status monitoring schemes have been studied. Our previous study focused on the packet loss rate of each link on an OpenFlow network as an indicator of communication quality. In this report, we focus on the packet delay variation as an indicator of communication quality and propose a method to efficiently monitor the packet delay variation of each link in both direction. In the proposed method, measurement packets are multicast-transferred traversing all links in both directions from a measurement terminal at a constant time interval and each switch acquires statistics of the arrival time interval of the measurement packets, which can be used to estimate the link packet delay variation. We also efficiently identify the links with large delay variations (larger than a threshold) by an appropriate order of retrieving the observed statistics from switches. Its effectiveness has been shown through simulation on a large-scale network topology. We also have investigated the impact of the position of the measurement terminal.

Key words OpenFlow, Network Monitoring, ActiveMeasurement, Multicast, Packet delay variation

1. はじめに

近年、クラウドサービスの普及やサーバの仮想化技術の進展

により、データセンタが大規模・分散化の傾向にある。それに伴い、データセンタ内・データセンタ間のネットワークにおいても大規模・複雑化が進んでいる。実時間での柔軟なサーバ拡

張・移行・縮退に対応するには、ネットワークにも既存のプロトコルの制約を超えた柔軟性・可用性が求められ、それを実現する有効な手段の1つとして OpenFlow 技術が注目されている [1]。一方、大規模な OpenFlow ネットワークにおける最適性・可用性の実現には全ての内部リンクの通信品質を実時間で監視・管理する必要があり、SNMP のような機器の状態情報の定期監視だけでは十分に対応することが出来ず、各リンクに試験パケットを実際に流す能動的な計測の方式が研究されている [2][3]。先行研究 [4][5] では、通信品質の指標としてパケットロス率に着目し、各リンクのパケットロス率を効率的に計測する手法の検討を行った。本稿では、通信品質の指標として遅延変動に着目し、OpenFlow スイッチ間のリンク（非 OpenFlow のスイッチやルータを含む仮想リンク）上のパケット遅延変動の推定を行う。また、統計情報の取得順序を工夫することにより、全てのリンクのパケット遅延変動を確定するまでに必要な統計情報の取得回数を削減する手法を検討する。以降では、まず2章で本計測システムにおける前提条件を説明し、3章で OpenFlow の自由な経路制御を利用した本計測システムの概要を述べる。4章で今回の提案手法を説明し、5章で提案手法の評価を行い、6章で本研究のまとめと今後の課題を示す。

2. 前提条件

本稿は OpenFlow ネットワークを対象とし、各リンクは物理リンクや他ネットワーク上の VPN パスを問わないものとする。ただし、全二重リンクの双方向を個別に監視する必要がある。

2.1 アクティブ計測

ネットワークの通信品質を管理するための計測方法は大きく分けてパッシブ計測とアクティブ計測に分けられる。パッシブ計測は実トラフィックの観測に基づき、アクティブ計測は計測用パケットをネットワーク上に能動的に送信することで品質の推定を行う。本計測は全リンクの走査を目的としているため、本計測システムではアクティブ計測をベースとする。

2.2 先行研究における計測経路の計算

OpenFlow は、ソフトウェアによってパケットの経路制御を柔軟に設定することが可能である。そのため、OFC 側でトポロジの全リンクをカバーする経路を計算し、計測用パケットの経路を自由に設定することが出来る。最も単純な経路はトポロジ上を一筆書きする経路となる。しかし、計測用パケットを一筆書きの経路で転送した場合、パケットは必ずトポロジに存在するリンクの2倍だけホップすることとなる。したがって大規模なネットワークでは、計測時間に大きな影響を及ぼすことになる。また、計測端末に比較的近い箇所で障害リンク（パケットロスや大きな遅延変動が多発するリンク）が生じた場合、計測全体の精度に影響を及ぼす。

以下に先行研究における計測のための経路木生成アルゴリズムについて述べる。

- 最短経路の生成

(1) 計測端末が接続されているスイッチを始点とし、ダイクストラ法を用いて始点から各スイッチへの最短経路を計算する。

- 未使用経路の補完

(1) 始点スイッチから各スイッチに対しての最短経路に含まれていないリンクを検索する。

(2) 検索したリンクの両端のスイッチのうち、始点スイッチからのホップ数が少ないものを選択する。

(3) 選択したスイッチまでの最短経路に、もう一方のスイッチへ延び、そこから折り返して選択したスイッチへ戻る経路を最短経路木に追加する。

(4) 両端のスイッチの始点スイッチからのホップ数が同じである場合、両方を選択し、両スイッチは互いに自身への最短経路に対となっているスイッチへ延びる経路を追加する。

未使用経路の補完までは、両経路木とも同一のアルゴリズムで動作するが、次の折り返し経路の追加からは異なる動作を行う。1つ目の折り返しアルゴリズムを経路生成に利用し完成した経路木を経路木1、2つ目の折り返しアルゴリズムを経路木生成に利用し完成した経路木を経路木2とする。経路木1は経路木2と比較して、計測パスのホップ数が大きいという特徴を持つ。各スイッチのトポロジ上の特徴を判定し、それに応じた折り返し経路を追加する。複数の特徴に該当する場合は連番の若いほうの経路処理を行う。

- 折り返し経路の追加（経路木1）

(1) 未使用経路補完時に選択されたスイッチ：対となるスイッチから延びてきた経路、あるいは折り返してきた経路を1つ選択し、その経路に対して自身の最短経路上の親スイッチへ延びる経路を追加する。

(2) 最短経路上の葉となるスイッチ：自身への最短経路に、親スイッチへ折り返す経路を追加する。

(3) 始点以外で子スイッチを持つスイッチ：子スイッチの中で、葉となるスイッチからのホップ数が最も小さいものを1つ選択し、そのスイッチから延びてくる経路に自身の親スイッチへ延びる経路を追加する。

- 折り返し経路の追加（経路木2）

(1) 始点以外のスイッチ：自身への最短経路に、親スイッチへ折り返す経路を追加する。

上記の3手順を踏むことにより計測経路は重複せず、マルチキャストでパケットが転送されるため全リンクを双方向に走査しつつ、送信される計測用パケットの総数を抑えることが出来る。

実際のトポロジに対し、2種類の経路生成アルゴリズムがどのように適用されるか、例を図1、2に示す。

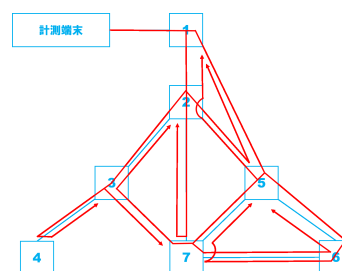


図1 経路木1の応用例

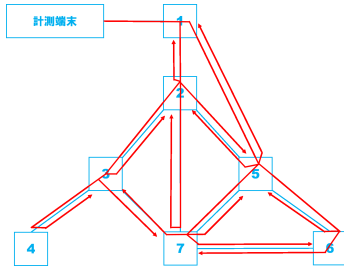


図2 経路木2の応用例

3. 計測システムの概要

本計測システムの流れを図3に示す。本計測システムは大きく分けて3段階のプロセスに分けられる。以後、各段階の動作について詳述していく。

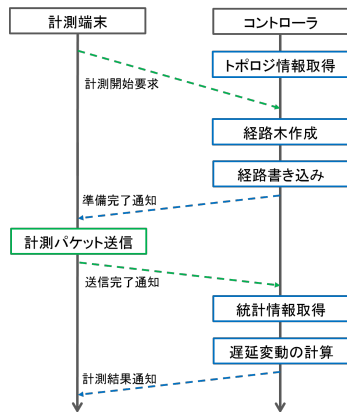


図3 計測環境

- 第一段階

第一段階では、OpenFlow ネットワークのトポロジ把握と、計測用パケットのパスとなる経路木の作成を行う。OFC は、LLDP パケットを OFS 間で交換させ、それによって得られた情報からトポロジの把握を行う。計測端末から「計測準備パケット」が OpenFlow ネットワークに送信され、OFS が受信すると「計測準備パケット」は OFS から OFC へ転送される。その後 OFC は、「計測準備パケット」を受け取った OFS を根とし、全リンク（上り下りを区別）を重複なしに一度で通過する経路木を作成し、この経路木に合わせて各 OFS にフローエントリを適用させる。このフローエントリに入力ポートと出力ポートを結びつけることによって転送を実行する。最後に、OFC から計測端末に対して「計測準備完了パケット」を送信する。

- 第二段階

OpenFlow ネットワークへの計測用パケットの送信と、各 OFS を通過したパケットから得た情報の記録を行う。計測端末は任意の数の計測用パケットを OpenFlow ネットワークの送信する。OFS はフローエントリに従って計測用パケットを転送し、それと同時にフローエントリ毎の統計情報に通過した計測用パケットから得た情報を格納する。経路木の終点に該当するポート（以降、葉ポートとする）で計測用パケットを破棄する。計測端末は任意の数の計測用パケットを送信し終わると、「計測

終了パケット」を OpenFlow ネットワークに送信し、OFS から OFC へ転送される。

- 第三段階

OpenFlow ネットワークにおける全リンク毎のパケット遅延変動を算出する。第二段階において「計測終了パケット」を受け取った OFC は、各 OFS に FlowStatsRequest メッセージを送信する。OFS はこれに対して StatsReply メッセージを OFC に返す。StatsReply メッセージには、FlowStatsRequest メッセージで指定した条件に当てはまるフローエントリの統計情報が含まれている。OFC はこの統計情報を用いてパケット遅延変動を算出する。この結果を OFC は各リンク毎に計測端末に通知することによって、計測端末は計測結果を取得する。

以上が、本計測システムの概要になる。

4. 提案手法

今回の提案手法による改良点は、計測を行う範囲を遅延変動まで拡大した点である。従来手法では、子ノードと親ノードを通過したパケット数を計測し、その差分を求めることでリンク毎のパケットロス率の算出を行っていた。本手法では、通過パケット数だけでなくパケットの到着時間間隔も記録することで、リンク毎のパケットロス率とパケット遅延変動の推定を行う。また、OpenFlow ネットワークにおける遅延変動増大リンク（以降、劣化リンクとする）の特定アルゴリズムの効率化を図る。以降、4.1 節でリンク毎のパケット遅延変動の推定方法、4.2 節で劣化リンクの特定アルゴリズムを説明する。

4.1 遅延変動の推定手法

遅延時間を計測するには、パケットの送信時刻と到着時刻を把握する必要がある。しかし、中継スイッチや受信端末ではパケットの送信時刻が分からないため、遅延時間の計測は困難である。2 点間の遅延時間の計測は困難であるが、遅延変動であればパケットの到着時間間隔から推定することが可能であり、OpenFlow スイッチが単独で実行することが出来る。なお本手法では、OFS がパケットの到着時間間隔の統計情報を取得出来ると仮定する。パケットのシーケンス図を図4に示す。

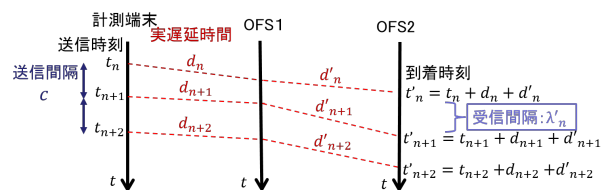


図4 シーケンス図

計測端末から OFS1 へのパケット送信時刻をそれぞれ t_n, t_{n+1}, t_{n+2} とし、OFS2 での到着時刻を t'_n, t'_{n+1}, t'_{n+2} とする。パケットの送信時間間隔を一定値 c とすると (1) 式で表すことが出来る。

$$t_{n+1} = t_n + c \quad (1)$$

また、計測端末-OFS1 間の遅延時間は d_n, d_{n+1}, d_{n+2} , OFS1-OFS2 間の遅延時間は d'_n, d'_{n+1}, d'_{n+2} であるため、OFS2 での到着時刻はそれぞれ (2), (3) 式で表すことができる。

$$t'_n = t_n + d_n \quad (2)$$

$$t'_{n+1} = t_{n+1} + d_{n+1} \quad (3)$$

パケットの到着時間間隔（以降、受信間隔） λ_n は $t'_{n+1} - t'_n$ であるため、(1),(2),(3) 式より、(4) 式に書き換えることができる。

$$\begin{aligned} \lambda'_n &= t'_{n+1} - t'_n = (t_{n+1} + d_{n+1}) - (t_n + d_n) \\ &= c + (d_{n+1} - d_n) \end{aligned} \quad (4)$$

ここで、受信間隔から遅延変動を推定するため、以下の前提条件を定義する。

- d_n, d_{n+1} が互いに同一分布である。
- d'_n, d'_{n+1} が互いに同一分布である。
- d, d' が互いに独立である。

上記の前提条件より、(5) 式は下式に書き換えることができる。

$$\begin{aligned} V[\lambda'_n] &= V[(d_{n+1} - d'_{n+1}) + (d_n - d'_n)] \\ &\cong 2(V[d + d']) \end{aligned} \quad (5)$$

したがって、計測端末から OFS2 までの遅延分散は下式となる。

$$V[d + d'] \cong \frac{V[\lambda'_n]}{2} \quad (6)$$

したがって、パケット送信時間間隔を一定にすることで、到着時間間隔のみで遅延変動の推定が可能となる。ここで、計測端末からある特定のポートまでの遅延分散 $V[d + d']$ をポート遅延分散と定義する。計測端末-OFS2 間のポート遅延分散は、計測端末-OFS1 間の遅延分散 $V[d]$ と OFS1-OFS2 間の遅延分散 $V[d']$ の和であるため、OFS1-OFS2 間の遅延分散は下式で推定することができる。

$$V[d'] = V[d + d'] - V[d] \quad (7)$$

この2つのポートで挟まれた区間の遅延分散を区間遅延分散と定義する。ポート遅延分散と区間遅延分散から、各リンク毎のパケット遅延変動を推定することが可能となる。また、受信時間間隔の分散は下式に書き直すことができる。

$$V[\lambda] = E[\lambda^2] - (E[\lambda])^2 \quad (8)$$

パケットの到着時間間隔の統計量を OFC へ送信することでポート遅延分散を推定することができる。

4.2 劣化リンクの特定アルゴリズム

本劣化リンクの特定手法は、「ポート遅延分散、あるいは区間

遅延分散が OFC で予め設定されている閾値未満であれば、その間のリンクで大きな遅延変動の発生がなかったものとする」という考えのもとに成り立つ。葉ポートで記録したポート遅延分散が閾値未満である場合を図5に、閾値以上である場合を図6に示す。

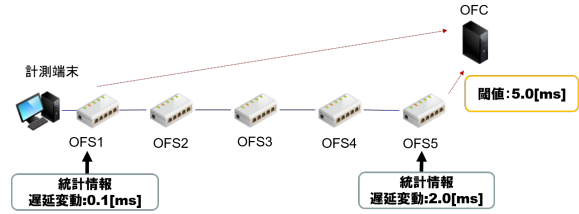


図5 閾値未満の挙動

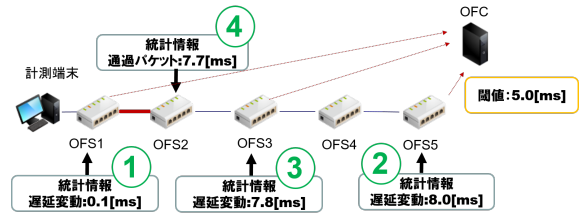


図6 閾値以上の挙動

提案手法では、5つのOFSの内、両端のOFSで記録された遅延変動を確認する。図6のように、末端のOFSの統計情報から算出した遅延変動の値が閾値未満であれば、計測端末-OFS5間に劣化リンクが存在しないとみなす。次に、図7のようにOFS1-OFS2間が劣化リンクであるとする。両端のOFSの遅延変動を確認し、OFCで設定された閾値以上である遅延変動が生じた確認された場合、二部探索で経路の中間にあたるOFSでの遅延変動を確認する。それによって、遅延変動が経路の前半部分であるか、後半部分であるかを判定し、劣化リンクを特定するまで二部探索を繰り返す。

本手法の具体的なアルゴリズムは下記の通りである。

- (1) 経路木における全終点ポート（以下、葉ポート）を保存する。
- (2) 各葉ポートから、経路木における始点ポート（以下、根ポート）までの経路（以下、終端経路）を統計情報取得候補として保存する。
- (3) 根ポートのフローエントリと、各葉ポートのフローエントリの統計情報を OFC へ送信する。
- (4) 各葉ポートで記録された遅延変動が閾値未満であれば、その終端経路では大きい遅延変動が生じなかったと判断し、その終端経路を非統計情報取得候補として保存する。
- (5) 統計情報取得候補の中で、非統計情報取得候補と重複している部分を統計情報取得候補から除外する。
- (6) 残った統計情報取得候補に対して経路毎に統計情報を取得する場合、二部探索を用いる。ここで、統計情報取得候補に共通部分がある場合、共通部分から順に統計情報を取得していく。全てのポートの特定が出来次第、計測終了とする。

5. シミュレーションによる有効性評価

シミュレーショントポロジを用いたシミュレーションの内容について詳述する。

5.1 シミュレーションプログラム

今回使用したシミュレーションプログラムの仕様は以下の通りである。

- 入力：ネットワークトポロジ（スイッチ及び、リンクの情報）、各統計情報（パケットの受信間隔平均、二乗平均、間隔数）

- 出力：計測用パケットの平均ホップ数、取得した統計情報、劣化リンクとその同定にかかった統計情報取得回数

また、シミュレーションでの計測端末、スイッチ、コントローラの使用は以下の通りである。

- スイッチ側：パケット到着時間間隔の取得、統計情報（パケット受信間隔の平均、二乗平均、間隔数）の記録
- 計測端末側：一定間隔での計測用パケット送信
- コントローラ側：閾値の設定、遅延変動の推定、劣化リンクの特定

5.2 シミュレーション内容

シミュレーションには、図7（スイッチ数：43、総リンク数：112）、図8（スイッチ数：19、総リンク数：48）のシミュレーショントポロジを用いた。スイッチ数とリンク数の比較から、平均するとトポロジ2の方がスイッチの次数が多く、スイッチ数、総リンク数ともに少ない。次数が多いため、計測用パケットをマルチキャストする機会が多くなり、平均ホップ数が小さくなる。

本シミュレーションでの評価指標は、全リンクの遅延変動を確定するまでに要した統計情報取得回数と、提案手法で特定した劣化リンクと実際の劣化リンクとの誤差数（劣化リンク判定誤差数）とする。提案手法を経路木1（平均ホップ数大）に適用した場合、提案手法を経路木2（平均ホップ数小）に適用した場合の統計情報取得回数、劣化リンク判定誤差数に対して、以下の条件で比較・評価する。

- 大遅延変動発生リンク数が1~7の場合（ランダムで100パターン）

- 計測端末位置がA（平均ホップ数大）の場合
- 計測端末位置がB（平均ホップ数小）の場合

統計情報取得回数、及び劣化リンク判定誤差数は100パターンの平均を算出する。

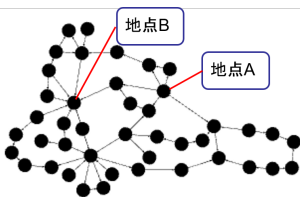


図7 トポロジ1

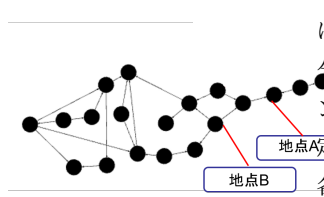


図8 トポロジ2

各経路木、及び計測位置における平均ホップ数を示す。図7

のトポロジ（以下、トポロジ1）における平均ホップ数を表1に、図8のトポロジ（以下、トポロジ2）における平均ホップ数を表2に示す。

表1 トポロジ1における平均ホップ数

	経路木1	経路木2
計測端末位置 A	5.689	4.19
計測端末位置 B	5.036	3.682

表2 トポロジ2における平均ホップ数

	経路木1	経路木2
計測端末位置 A	7.4	5.26
計測端末位置 B	5.545	3.913

5.3 大規模トポロジにおける統計情報取得回数の比較

経路木1、経路木2をそれぞれ適用した場合の提案手法における統計情報取得回数の平均値を、劣化リンクの発生数毎に示す。計測端末位置A（以降、位置A）でのトポロジ1に対して適用した場合を図9に、トポロジ2に対して適用した場合を図10に示す。

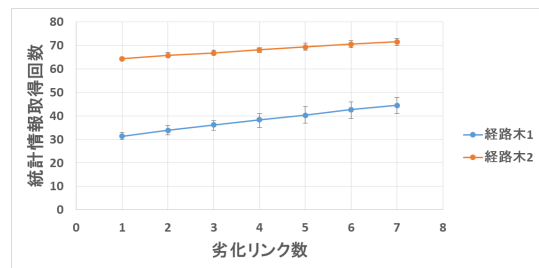


図9 トポロジ1の結果

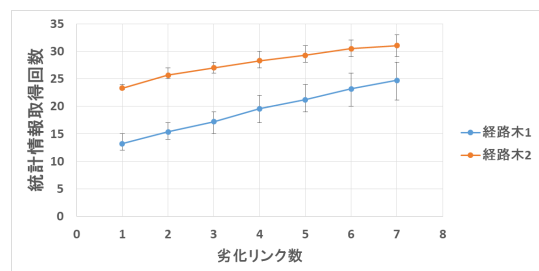


図10 トポロジ2の結果

トポロジ1では全リンク数112に対し、提案手法を経路木1に適用した場合は約3分の1、経路木2を適用した場合は約5分の3のリンクによる取得回数に抑えることが出来た。劣化リンク数の増加に伴い、取得回数がわずかに増加しているが、安定した削減率を実現している。高い削減率の実現に関しては、各終端経路の終点である葉ポートで閾値未満と判断された場合、その終端経路での取得を省略することが出来るためである。また経路木1と経路木2を比較すると、経路木1のほうが大幅な削減率を示したことが確認出来る。これは、経路木1が経路木2と比較して終端経路の平均ホップ数が大きいため、葉ポート

で閾値未満と判断場合に省略することの出来る終端経路が長い
ためであると考えられる。

トポロジ2では、全リンク数48に対し、提案手法を経路木1
を適用した場合は約5分の2、経路木2を適用した場合は約
5分の3のリンクによる取得回数に抑えることが出来た。トポ
ロジ1とトポロジ2を比較して、トポロジ2の削減率の方が高
いのは、前述と同様にトポロジ2の方が終端経路の平均ホッ
プ数が大きいためである。

経路木1を用いて計測端末位置を変更した際の比較として、
トポロジ1の場合を図11に示す。

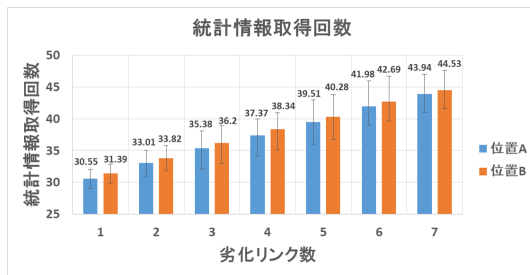


図11 取得回数におけるトポロジ1での計測地点A,Bの比較

計測端末位置をAとBで比較した場合、位置Aは位置Bより
も終端経路の平均ホッパ数が大きいので、わずかであるが取
得回数を抑えることが出来る。

5.4 大規模トポロジにおける劣化リンク判定誤差数の比較

経路木1、経路木2をそれぞれ適用した場合の提案手法にお
ける特定した劣化リンク判定誤差数の平均値を、劣化リンクの
発生数毎に示す。トポロジ1に対して経路木1、経路木2を適
用した場合を図12に、トポロジ1に対して経路木1を適用し、
計測端末位置をA,Bで比較した場合を図13に示す。

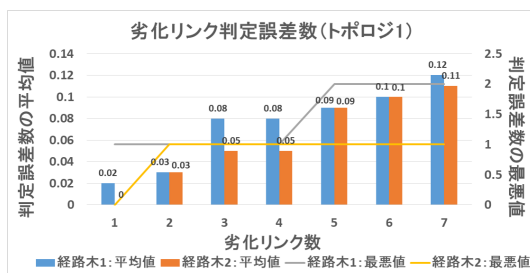


図12 判定リンク誤差におけるトポロジ1での比較

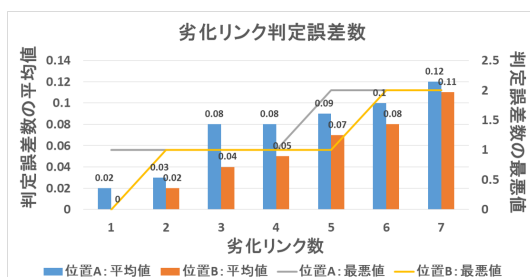


図13 判定リンク誤差におけるトポロジ1での計測端末位置A,Bの
比較

経路木1と経路木2を適用した場合と比較して、経路木1を
適用したの方が経路木2を適用した場合より判定誤差率が高い。
これは平均ホッパ数が大きいため、各リンク毎の推定精度が低
下するためである。また劣化リンク数が増加すると、1つの終
端経路に複数の劣化リンクが生じる場合があるため、精度の低
下は顕著にあらわれる。計測端末位置を変更した場合の結果を
比較すると、位置Bの方が劣化リンク判定誤差率が低く、劣
化リンク数が増加する毎に顕著にあらわれる。これも同様に、
ホッパ数が大きいことで推定精度が低下したためである。

これらの結果から、計測時間や計測精度という観点から見た
場合、計測用パケットの平均ホッパ数は可能な限り低減するこ
とが望まれる。しかし、制御ネットワークの通信負荷やスイッ
チに対するアクセス負荷という観点から見ると、計測用パケッ
トのホッパ数は大きい方が負荷の低減につなげることが出来る。
これらのトレードオフを考慮し、ネットワークの状態によって
2つの経路木と計測端末位置を考慮し、動的に設計することが
望ましいといえる。

6. まとめ

本計測システムはOpenFlowネットワークを考慮し、全リ
ンクに計測用パケットを通過させ、スイッチでは各リンク毎に
計測用パケットの到着時間間隔の統計量を取得することで、遅
延変動を算出する。劣化リンクを特定するとき、全リンクに対
応する統計情報を取得する場合、制御ネットワークやスイッチ
に対するアクセス負荷が増大してしまう。そこで提案手法では
閾値を設定し、計測パスに沿って離れた2つのポート間の遅延
変動が閾値未満であれば、その間に劣化リンクが生じていない
ことを利用し、統計情報の取得順序を工夫することで、全体
の中で少数のリンクのみによる劣化リンクの特定を目指した。提
案手法の大規模トポロジにおける有効性をシミュレーションに
よって検証した結果、取得回数の削減を実現した。今後、統計
情報の取得順序の更なる最適化や、取得回数の削減による計測
精度の劣化を抑える手法について検討する。

謝辞 なお、本研究はJSPS科研費16K00130の助成を受け
たものである。

文 献

- [1] “OpenFlowWhitePaper”, <http://www.openflow.org/documents/openflow-wp-latest.pdf>
- [2] 立花篤男, 渋谷恵美, 長谷川輝之, “OpenFlowネットワークにおける効率的なリンク品質計測 手法の提案”, 信学会ソサイエティ大会講演論文集, no.B-11-12, pp.326, Sep.2013
- [3] 波多野洋一, 山崎康広, 下西英之, “OpenFlowネットワークを用いたネットワーク品質劣化箇所推定における品質計測の効率化”, 電子情報通信学会技術研究報告. NS, ネットワークシステム109(448), pp.25-30, Mar.2010.
- [4] 月岡祐太, 鶴正人, “OpenFlow Networkでの全リンクパケットロス率計測の効率化”, 電子情報通信学会技術研究報告, IN2015-84, pp.77-82, Dec.2015.
- [5] 藤村悠樹, 月岡祐太, 鶴正人, “OpenFlowにおけるパケットロス率計測のための統計情報取得順序の最適化”, 電子情報通信学会技術研究報告, CQ2016-45, pp.73-78, July2016.