東京大学
情報理工学系研究科 電子情報学専攻
修士論文

Fast Reroute の代替パスに関する 遅延時間測定と低遅延パスの検出フレームワーク

A Framework for Latency Measurement and Low-Latency Path Detection in Backup Paths of Fast Reroute

48-236406
伊藤 吉彦
Yoshihiko Ito

指導教員　落合秀也 准教授

2025 年 1 月

概要

ビデオ会議やオンラインゲーム，ライブストリーミングなどのリアルタイム通信サービスは，現代の生活において不可欠な存在となっており，高品質かつ耐障害性に優れたネットワークインフラの需要が高まっている．Fast Reroute (FaRe) は，バックアップパス (BPath) を事前計算することによってネットワークの耐障害性を向上させるが，遅延時間の観点でこ BPath の性能を保証することが課題となっている．

本研究では，FaRe が提供する BPath の遅延時間を測定し，より低遅延な代替パスを検出するためのフレームワークを提案する．具体的には，PING CLI を用いた Path Latency Analyzer using Ping (PLA-P) と，Two-Way Active Measurement Protocol (TWAMP) を用いた Path Latency Analyzer using TWAMP (PLA-T) の二つの手法を設計・実装した．PLA-P は特定のパスに沿った往復遅延時間を測定し，PLA-T はリンクレベルの遅延メトリクスを評価する．

シミュレーション環境での実験により，PLA-P は遅延変化へ適応し，低遅延な代替パスを効果的に検出できることが確認された．しかしながら，PLA-P は測定精度の点で課題が残る結果となった．一方，PLA-T は，PLA-P と同様に低遅延代替パスを検出でき，小規模なトポロジーであれば，数 ms の誤差で BPath の測定をすることができた．本研究で行った実験では，トポロジーサイズが大きくなると，測定精度が落ちた．別の状況下での追加実験を行い，より詳細な評価を行う予定である．

本研究で提案したフレームワークは，FaRe 環境における Quality of Service の向上に寄与し，高い信頼性の低遅延パスの検出と活用を可能にする．

Abstract

Real-time communication services such as video conferencing, online gaming, and live streaming have become integral to modern life, driving demands for high-quality and resilient network infrastructures. While Fast Reroute (FaRe) enhances network fault tolerance by precomputing backup paths (BPaths), it struggles to ensure the performance of these paths, particularly in terms of latency.

This research introduces a novel framework for measuring latency in FaRe BPaths and identifying alternative low-latency paths. Two methods are proposed and implemented: the Path Latency Analyzer using PING CLI (PLA-P) and the Path Latency Analyzer using TWAMP (PLA-T). PLA-P utilizes PING CLI to measure round-trip times along specific paths, while PLA-T employs TWAMP to assess link-level delay metrics.

Experimental evaluations conducted on a simulated network environment demonstrated that PLA-P can dynamically adapt to latency changes, effectively identifying low-latency alternative paths, but its measurement accuracy still faces challenges. PLA-T also detected lower latency alternative paths. In small network topology, PLA-T measured a latency with a backup path with only a few milliseconds of error. However, as the topology size increased, the measurement accuracy of PLA-T declined. We plan to conduct additional experiments under different conditions to evaluate in more details.

The proposed framework offers a significant contribution to enhancing the Quality of Service in FaRe environments by enabling reliable detection and utilization of low-latency paths.

目次

図目次

表目次

第 1 章

序論

1.1 背景

近年，ビデオ通話アプリケーションやオンラインゲーム，ライブストリーミングなどのリアルタイム通信を必要とするインターネットサービスは，人々の生活やビジネスに深く根付いてきている．Feldmann ら [1] による報告では，コロナ禍を経てビデオ会議やオンライン教育のトラフィック量が 2 倍以上に増加したとされている．さらに，インターネットは人々の生活に大きな影響を与え，社会インフラとして人々の生活に浸透してきている．総務省の調査によると，2022 年における国内のインターネットの利用率は 84.9 % に達した [2]．また，国外においても，2018 年には世界人口の過半数を占める 39 億人がインターネットを利用し [3]，国内外共にインターネットの普及が急速に進んでいることがわかる．これらの数値はインターネットそしてリアルタイム通信サービスの普及が進み，ネットワークサービスへの期待が増加していることを示している．

2017 年に Khan ら [4] が行った調査によれば，ユーザの期待値と実際の体験にはギャップがあり，インターネットサービスの重要性は高いものの，満足度は低かった．ユーザがサービスから受ける体験 (Quality of Experience, QoE) は，ネットワークやそのサービスの技術的なパフォーマンス (Quality of Service, QoS) を，ユーザの視点から知覚的に評価したものである [5, 6]．そのため，ユーザが満足するネットワークサービスの実現には，高い品質を備えたネットワークが求められる．

QoS を定量的に表現する際，帯域幅，遅延，パケット損失率などのメトリクスが代表的に使用される [7]．トラフィックが広帯域、低遅延、低損失といった QoS 要件を満たすために，上述した QoS メトリクスを利用した経路選択 (QoS ルーティング) [8] が想定される．通常の Interior Gateway Protocol (IGP) では，ホップ数やリンクごとに割り当てられたコストの総和が最小となる経路を選択する．QoS ルーティングでは，遅延 30 ms かつ損失率 99.99 % のような，複数の QoS 要件を最も満たす経路が選択される．単一の要件であれば，通常の IGP と同様に，計測された QoS メトリクスの総和が最小となる経路が選択されるが，複数の要件が合わさった経路の選択は，分散的な経路計算では実現が困難である [9]．そのため，外部コントローラである Path Computation Element (PCE) [10] に，経路計算機能を分離することで，QoS

要件を満たす経路を提供する．

さらに，ネットワークは現代社会の重要なインフラであるため，高い品質に加えて，災害や攻撃などの障害時にもサービスを継続的に提供できる耐障害性が求められる [11]．ネットワークの耐障害性を確保するために，ネットワークの状態の監視，潜在的な障害の検出，リソースのバックアップ，負荷分散などの手法が考えられる [12]．

ネットワークの耐障害性向上のための代表的な技術に，複数のアクティブな経路を用意し，パケットを複製してそれぞれの経路に同時に転送するという手法が挙げられる [13]．用意された経路の一つに障害が発生したとしても残りの経路が有効である限り，理論上サービスダウンタイムを 0 に抑えることができる．しかしながら，パケット処理のために特別な処理が受信側に必要であることや，複数の経路に同時にトラフィックを転送するために豊富なネットワークリソースが必要であることなど，導入コストが高い．

一方，アクティブな経路とは別に，非アクティブな経路を用意し，通常時は主要パス (PPath) を使い，障害発生時にはバックアップパス (BPath) に切り替える手法がある．経路の切り替え位置に応じて Global Repair と Local Repair の二つに大別される [14]．Global Repair は PPath の障害発生箇所に依らずに，先頭ノードでパスを切り替え，ネットワーク全体を End-to-End (E2E) で保護する．Local Repair は障害発生箇所の隣接ノードでパスの切り替えをし, 障害箇所を迂回する．Multiprotocol Label Switching (MPLS) [15] 網や Segment Routing (SR) [16] 網の入口ノードに設定された Secondary Path は Global Repair を実現する BPath である．一方，Local Repair を代表する技術に Fast Reroute (FaRe) がある．FaRe では BPath を事前に計算し，障害を検出した隣接ノードにおいてパスを切り替える [17–19]．IGP におけるパスの再計算には 200 ms 程度必要だが，FaRe では 50 ms 程度で障害回復が可能である (付録 A 参照)．

Global Repair の場合，網の入口ノードのルーティングテーブル (Routing Information Base, RIB) に BPath の経路情報が格納されるため，PCE を利用した QoS 制御が容易である．一方，FaRe では各ノードがそれぞれ BPath を RIB に保管するため，BPath のポリシー遵守や QoS の保証が困難である [12, 20]．

1.2　本研究の貢献

FaRe はネットワークの耐障害性向上を実現する技術であるが，各ノードで BPath の経路計算がなされるため，BPath に切り替え後の性能を保証することが困難である．本研究では，QoS メトリクスの遅延時間に着目し，FaRe が提供する BPath の遅延時間を測定するためのフレームワークを提案した．

通常，パスの性能を評価する際，実際のトラフィックからデータを取得するパッシブ計測の方が正確であるとされている [21] が，FaRe の BPath は通常時にトラフィックが流れないため，測定用パケットを用いるアクティブ計測による計測を行った．

本研究では，PING Command Line Interface (PING CLI) と Two-Way Active Measurement Protocol (TWAMP) を用いた測定フレームワークをそれぞれ提案および開発し，その性能評価を行った．本研究の主な貢献は以下の二つである．

- PING CLI および TWAMP を用いて，FaRe が提供する BPath を測定するフレームワークを設計・実装し，それらの測定性能を評価した．PING CLI を使った測定は，遅延の変化に対する応答性で，TWAMP を使った測定では，測定精度に関して，優れた成績を収めた．
- 上述したフレームワークを使用することで，通常の FaRe が提供する BPath よりも低遅延の BPath を検出することができた．

1.3 本論文の構成

本論文は以下の通り構成される．第 2 章では本研究に関わる技術について解説したのち，ネットワークのパス品質向上に関する研究を紹介し，本研究の立ち位置を明らかにする．そして，第 3 章で想定環境を整理した後，第 4 章で提案するフレームワークを，第 5 章で詳細な実装を説明する．第 6 章では提案フレームワークを評価し，第 7 章にて議論する．最後に，第 8 章で本研究の結論と今後の課題を述べる．

第 2 章

関連技術・関連研究

本章では，初めに Traffic Engineering，QoS 制御について説明し，ネットワークの品質制御に関する技術について説明する．次に，耐障害性を高めるための FaRe 技術について説明する．最後に，ネットワーク品質向上を目的とした取り組みとその課題を整理し、本研究の提案手法がこれらの先行研究をいかに補完し発展させるかを説明する．

2.1 Traffic Engineering (TE)

2.1.1 TE とは

現在，社会インフラとなったインターネットは，Internet Protocol (IP) により接続されている．IP はベストエフォートの原則に基づいて，パケット交換を行うプロトコルである [22]．動画や音楽の配信サービスなど，リアルタイムの通信を必要とする場合，従来の IP ネットワークではユーザに十分な品質のサービスを提供することができない [23]．この問題を解決するために Traffic Engineering (TE) のプロセスが用いられる． TE とは明示的に経路を指定し，トラフィックを制御することで，QoS 制御やトラフィックのロードバランシングを行うものである． TE は，MPLS や SR を用いることで実現できる [24, 25].

2.1.2 SR-TE

SR とは，ソースルーティングの実現方法の一つで，パケットが通過するノードを明示的に指定する経路制御方式である．SR では，データプレーンに MPLS と Internet Protocol Version 6 (IPv6) を指定することができる．ラベルあるいは IPv6 アドレスを用いて転送先の中継ノードを表現し (Segment Identifier, SID と呼ばれる)，パケットに SID を埋め込む．中継ノードは埋め込まれた SID を参照し，次の転送先を決定する．SID は，Open Shortest Path First (OSPF) や Intermediate System to Intermediate System (IS-IS)，Border Gateway Protocol – Link State (BGP-LS) を用いて広報され，ネットワーク全体で一貫性のある経路制御が可能になる．

SR を用いた TE には SR Policy (SRP) [26] が用いられる．SRP は経路に付加された属性であ

る Color，宛先である Endpoint，通過する経路を示す Segment List (SID をリスト化したもの，SL) で構成される．SRP を用いることで，プレフィックス単位，ルーティングテーブル単位での TE が可能になる．また，SRP 自体にも SID を割り当てることができ，Binding-SID (BSID) と称される．SR-TE では，SRP が指定する候補パスに従って TE される．候補パスは，Explicit Path, Dynamic Path，あるいはその複合を指定でき，Explicit Path は SL で明示的に指定されたパスである．一方，Dynamic Path は，IGP コストあるいは TE メトリクスを使って動的に計算されるパスである．

2.2 QoS 制御

2.2.1 本研究における QoS の立ち位置

QoS とは，ネットワークの通信品質を指す語句である．Stankiewicz ら [27] は，Intrinsic QoS，Perceived QoS，Assessed QoS からなる一般的な QoS モデルを提唱した．Intrinsic QoS とは，ネットワーク自体の性能に焦点を当てた概念で，計測された QoS メトリクスを用いて定量的に表現される．代表的な QoS メトリクスは，IP パケットの転送遅延，遅延変動 (ジッター)，パケット損失率，帯域幅などである．Perceived QoS は，エンドユーザがネットワークサービスを使用した際に直接感じる品質を指す語句で，Intrinsic QoS とは異なり，主観的に評価される．Assessed QoS は，一般的には QoE とも呼称され，Perceived QoS を客観的に評価したものである．サービス価格やカスタマーサポートの質などを踏まえ，サービスを受け入れ可能かを総合的に判断する．

以上のように，QoS は，どの視点から品質を評価するかによって，三つに分類されるが，本研究では Intrinsic QoS を QoS として使用する．また，測定する QoS メトリクスを遅延，ジッター，パケット損失率に限定し，以後これら三つを遅延メトリクス群と称する．

2.2.2 遅延メトリクス群の測定に使用される技術

遅延メトリクス群の測定に使用される技術のうち，代表的な二つの技術である PING CLI と TWAMP [28] について紹介する．

1. **PING CLI**
   PING CLI は Internet Control Message Protocol (ICMP), Transfer Control Protocol (TCP), User Datagram Protocol (UDP) のいずれかのパケットを用いて対象のネットワーク機器との疎通性を確認する CLI である [29]．ICMP パケットを用いる場合，PING CLI では指定された宛先に ICMP Echo Request を送信し，ICMP Echo Reply が返って来れば，IP ネットワークとして疎通性があると判断される．PING CLI を使うことで，疎通性の他に，通信の往復時間 (Round Trip Time, RTT)，パケット損失率，パケットが到達可能な最大ホップ数 (Time to Live, TTL) の三つの情報を取得可能である．RTT は遅延メトリクスとして用いられ，複数の遅延メトリクスからジッターを求めることが可能である．

表 2.1: TWAMP の制御パラメータ

パラメータ名	説明
probe-interval (ProInt) [s]	テストパケットの送信間隔
probe-count (ProCon) [-]	プローブ値の算出に使用するテストパケットの数
advertisement-interval (AdInt) [s]	メトリクスの広報周期
advertisement-threshold (AdsThr) [%]	メトリクスの変化量の閾値

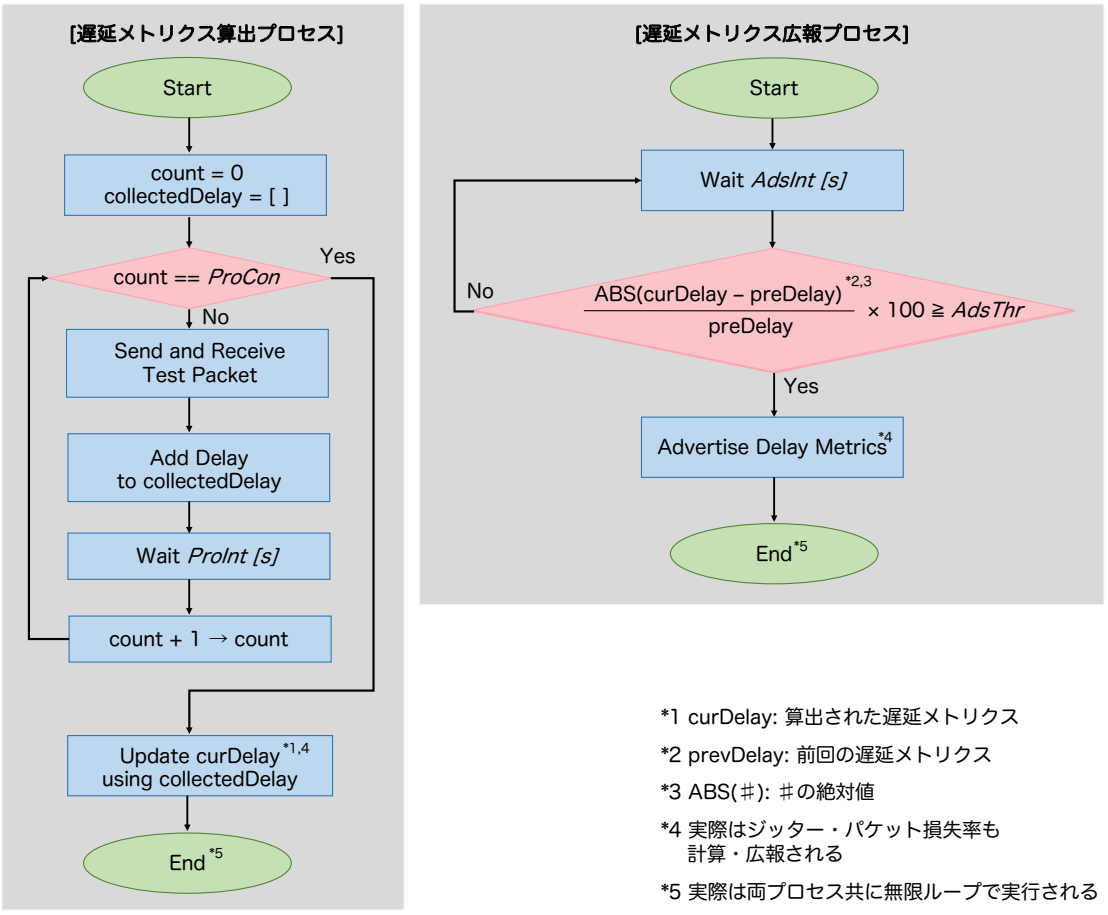


図 2.1: TWAMP における遅延メトリクスの測定・広報のフローチャート

2. **TWAMP**

TWAMP はネットワーク性能を正確に測定するために設計されたプロトコルである．ネットワークの遅延やパケット損失率を測定するための標準化されたプロトコルとして広く使用されている．TWAMP は，測定のためのテストパケットの送受信を実行する Test Session と，そのセットアップ，管理，終了を行う Control Session の二つのセッションから成る．また，TWAMP から Control Session を除外し，Test Session のみで測定を行う TWAMP Light も存在する．TWAMP Light は TWAMP に比べ，Control Session によるオーバヘッドが発生しないため，構成が簡単で軽量であることが知られる．以後，特別な明記がない場合，

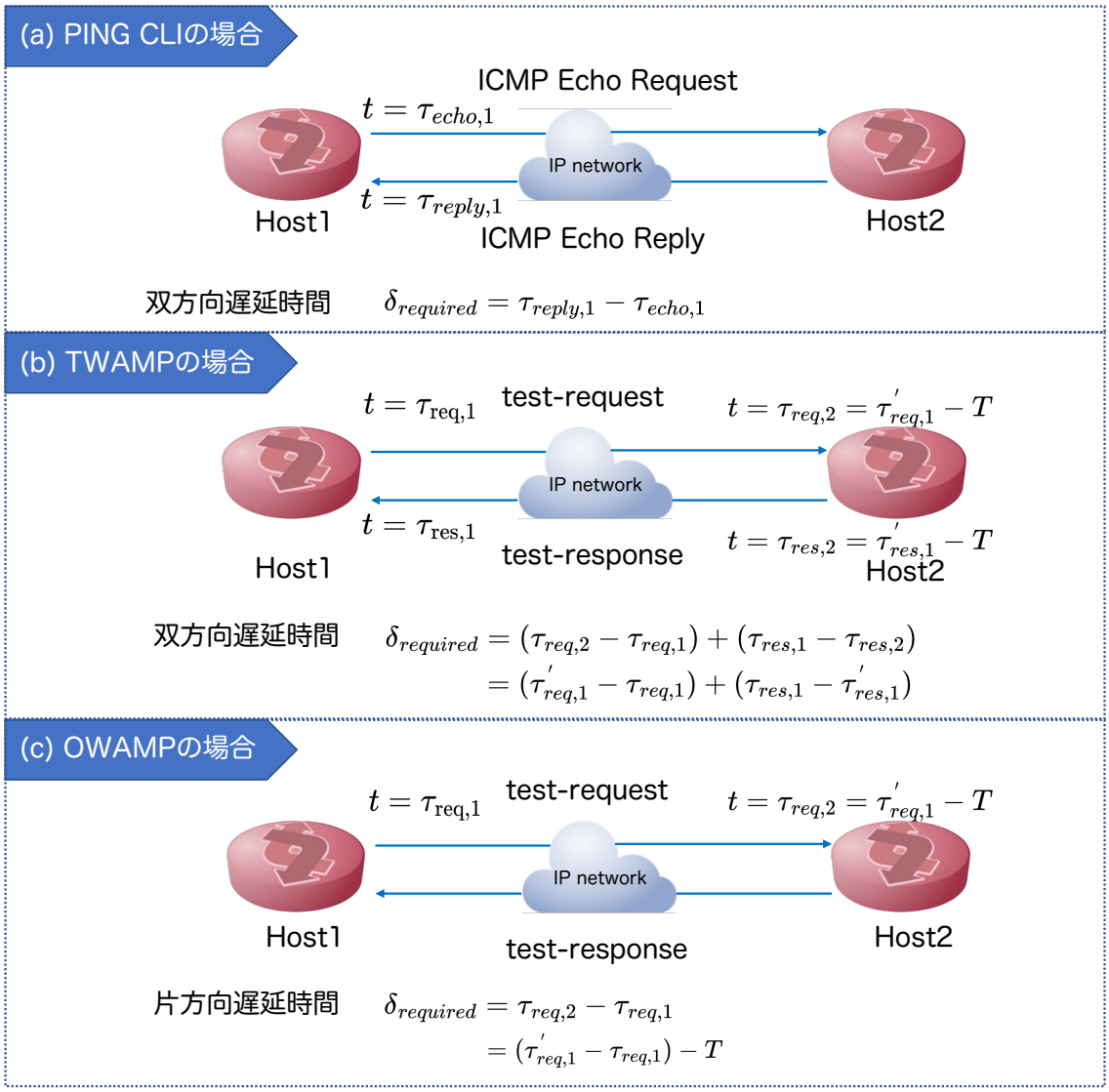


図 2.2: PING CLI，TWAMP，OWAMP の遅延の測定方法の比較．$\tau_{X,Y}$ は HostY におけるタイムスタンプ，Host2 の時刻は Host1 より T だけ進んでいる．

TWAMP Light を TWAMP と称する．

Juniper 社が開発している Junos OS の実装 [30] を元に，TWAMP の測定精度に影響するパラメータを表 2.1 に整理し，TWAMP の測定・広報フローを図 2.1 に示した．TWAMP ではテストパケットを使って，遅延時間を測定し，複数の測定値を使って遅延メトリクスとジッターメトリクスを算出する．ProCon は算出に必要な測定値の数を表し，測定値の平均値が遅延メトリクスとなる．また，測定された遅延メトリクスを，後述する IGP を用いて広報する際，AdsInt と AdsThr を設定する．AdsInt (s) おきに遅延メトリクス群を更新するかを確認し，前回測定された遅延メトリクスと比較して AdsThr (%) 以上変化していれば，ネットワーク全体に広報する．AdsInt を短く，AdsThr を小さく設定すれば遅延メトリクスは頻繁に更新されることになる．

Junos では広報をするタイミングに応じて，Periodic Advertisement (PAd) と Accelerated Advertisement (AAd) を設けている．PAd は前述したように AdsInt おきに遅延メトリクスの広報を行うか判定する手法である．一方，AAd では遅延メトリクスの値が更新されるたびに広報の判定を行う．例として，ProInt，ProCon，AdsInt，AdsThr をそれぞれ 5 s，10 個，120 s，10 % と設定した場合，PAd は 120 s おきに遅延メトリクスが 10 % 以上変化していれば広報する．AAd では，遅延メトリクスの値が更新される，50 s (= 5 $s \times 10$) おきに広報するかの判定が行われる (正確には，50 s + 遅延の測定値 である)．本研究では，AAd の場合における，遅延メトリクスの更新間隔も AdsInt と見なし，表 2.1 に示される 4 つのパラメータが TWAMP の測定精度に影響すると考える．

PING CLI と TWAMP は双方向の遅延を測定する技術であり，片方向の遅延は RTT を半分にすることで計算されるが，通常のネットワークでは送信経路と受信経路が対称でないことがほとんどである [31]．そのような場合，双方向の遅延測定から片方向遅延を正確に算出することは不可能である．そのため，片方向の遅延を正確に測定するために，A One-way Active Measurement Protocol (OWAMP) が提案された [32, 33]．しかしながら，OWAMP を用いた正確な測定にはエンドツーエンドで時刻が同期されている必要があり，時刻同期の精度が測定結果の正確性に大きく影響する．

図 2.2 には双方向測定と片方向測定の遅延時間の測定の仕方について示されている．Host1 と Host2 の時刻同期がされていない場合であっても，PING CLI や TWAMP のような双方向遅延測定では，時刻のずれ T が打ち消され，正確にパケット転送に必要な時間を計測可能である．一方，OWAMP のような片方向遅延測定の場合，両ホストの時刻のずれ T が計測された遅延時間に含まれたままである．

2.2.3　メトリクス広報に使用される技術

ネットワーク全体の QoS を評価するために，2.2.2 で述べた技術を用いて測定した遅延メトリクス群をネットワーク全体に広報する必要がある．本節では IGP の TE 拡張と BGP-LS の二つのプロトコルを用いてネットワーク全体に遅延メトリクス群を広報する仕組みを説明する．

1. **IGP TE 拡張**

   IS-IS や OSPF のようなリンクステート型の経路制御プロトコルでは，隣接ノード，接続リンク，ネットワークの状態情報を情報ユニットにまとめて，広報する．この情報ユニットは，OSPF では Link-State Advertisement (LSA)，IS-IS では Link-State Packet (LSP) と称される．IGP の TE 拡張とは，情報ユニットを TE 向けに拡張し [34, 35]，遅延や実行帯域といったメトリクスを広報するように改良したものである．拡張された IGP では，片方向リンク遅延の統計値 (平均値，最小値，最大値，ばらつき)，片方向リンク損失，片方向使用可能帯域幅などを扱える．広報された TE 情報は，Traffic Engineering Database (TED) に保存される．SR を用いた QoS 制御では，TED に格納された TE 情報を用いて，制約付きパス

を選択する．

2. **BGP-LS**

   BGP-LS [36] は，リンクステート型経路制御プロトコル (IS-IS や OSPF など) のリンクステート情報を広報するための Border Gateway Protocol (BGP) の拡張機能で，主に TE や Software Defined Network (SDN) 環境で使用される [37, 38]．MP_REACH_NLRI と呼ばれるパス属性の領域内に，新たに BGP-LS NLRI 属性の領域を作成し，リンクステート情報を構造化して格納する．また，BGP-LS NLRI には SR に関する情報も格納されており，ノードに一意に割り当てられた SID (Node SID)，グローバルに使用可能な SID の範囲を定義したもの (Segment Routing Global Block，SRGB)，リンクに割り当てられた SID (Adjacency SID, Adj SID) などが BGP-LS で広報される．

2.3 Fast Reroute (FaRe)

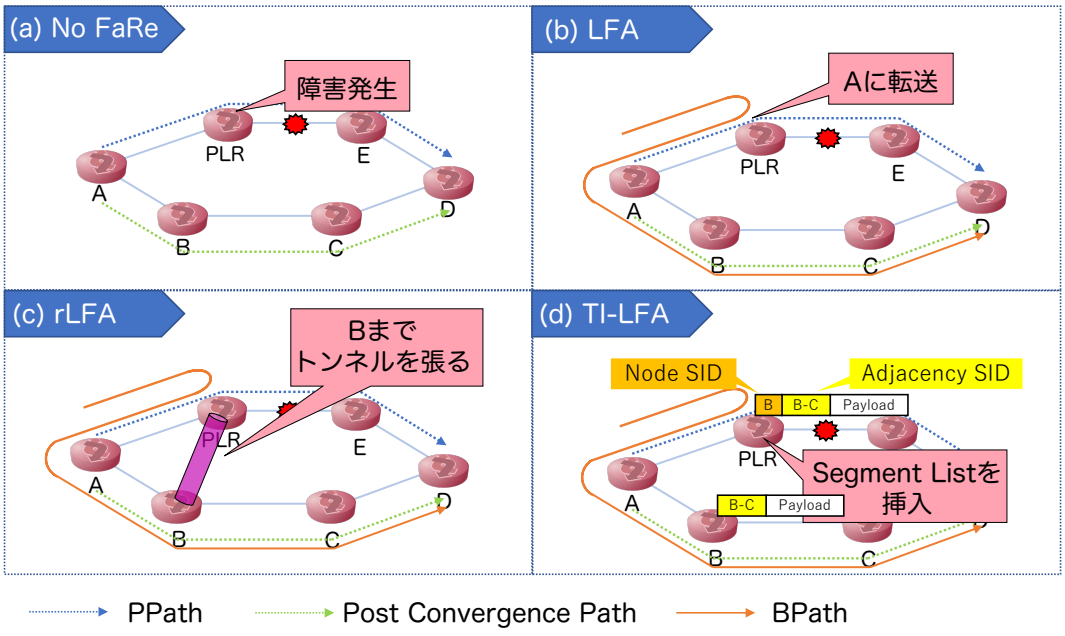


図 2.3: LFA とその発展系が提供する BPath の比較

FaRe は Local Repair の一種で，BPath を事前に計算し，障害発生時にネットワーク全体のパス計算の収束を待たずに，障害を検出したノード (Point of Local Repair, PLR) でパスを切り替える技術である．本節では FaRe の代表的な手法や FaRe の実現に必要な技術について整理する．

FaRe は，ネットワーク全体でパスが収束する前に，経路を切り替える技術である．障害を検出したノードは BPath の経路を使用し，検出していないノードは PPath の経路を使用するため，両者の間で経路の競合が起き，ループが発生する場合がある．ループが発生しない BPath を計算する手法を Loop Free Alternate (LFA) と呼び，リンクステート型プロト

コル上で動作可能である [39]．LFA は障害シナリオのカバー範囲を広げるために，Remote LFA (rLFA) [40, 41]，Topology-Independent LFA (TI-LFA) [42] と拡張されてきた．図 2.3 は LFA，rLFA，および TI-LFA が BPath を提供する仕組みを示す．LFA は最もシンプルな方法であり，rLFA は遠隔ノードへのトンネリングを加え，TI-LFA は SR 技術を活用して障害シナリオに対応する．rLFA では，図 2.3 (c) のように，PLR から遠隔ノードまで，トンネリングプロトコルを用いた経路制御することで，LFA より対応可能な障害シナリオの範囲を広げた．さらに，TI-LFA では，図 2.3 (d) のように，ループが発生しないノードまで SR を用いた経路制御をすることで，トポロジーに依存せず，中継ノードの単一障害シナリオに対して完全に対応可能となった．計算アルゴリズムの詳細は，付録 B に記載する．LFA，rLFA，TI-LFA の三つの計算手法は，実世界のネットワーク機器にデプロイされており，利用可能である [43]．

上述した技術は，個々の運用者が管理しているネットワーク内に制限した FaRe である．一方，インターネットのように異なる運用者が管理する，個々のネットワークが相互に接続するネットワークを対象にした FaRe も存在する．このようなネットワークでは，経路制御のために BGP を使って経路情報の更新をする．遠隔障害により遅延を引き起こす問題に BGP が対応するために，SWIFT と称される FaRe のフレームワークが開発された [44]．SWIFT では，BGP の Update Message の小さなバーストから障害範囲を予測することで，BGP の遅延を解消する．また，独自のエンコーディングスキームを導入し，フォワーディングテーブルのうち，障害の影響を受けたエントリを迅速に更新可能にした．

本研究では，デプロイの容易性と，障害シナリオのカバー範囲を考慮して，FaRe の技術を TI-LFA に限定して議論を進める．

2.3.1　障害検出技術

障害回復に必要な時間は，障害を検出するまでの時間と BPath への経路の切り替えに必要な時間に依存するため，障害の検出速度は重要な要素である．一般的なネットワーク障害の検出手法は次の 2 つである．

1. **ハードウェアベースの障害検出**
   物理層やリンク層を直接監視して障害を検出する手法である．例えば，ネットワークデバイスの物理的なリンク状態を監視し，リンクダウン信号を検出した際，即座に障害と判定する．この手法ではループや輻輳などの上位層の障害を検出することはできない．
2. **Bidirectional Forwarding Detection (BFD)**
   BFD とは，高速な障害検出を実現するために設計されたプロトコルで，隣接ノード間で超短間隔で Hello パケットを送受信することで，リンク状態を監視する手法である [45]．正常時はパケットを定期的に交換できている状態であり，一定時間応答がない場合，障害を検出する．

2.4　パスのネットワーク品質向上に関する研究

表 2.2: パス品質向上に関する先行研究と本研究の比較

論文	SDN	耐障害性実現手法	評価対象	使用技術
[46]	×	トラフィック分散	遅延， パケットロス率	eBPF
[47]	○	トラフィック分散	帯域幅	SRv6, PCEP* [10]
[48]	○	トラフィック分散	帯域幅，遅延	SR，OpenFlow
[49]	○	BPath	遅延， パケットロス率	SRv6, TWAMP, BGP-LS
[50]	○	BPath (FaRe)	帯域幅	SR, PCEP
[51]	○	BPath (FaRe)	N/A	N/A
[12]	×	BPath (FaRe)	帯域幅， パケットロス率	N/A
本研究	○	BPath (FaRe)	遅延，ジッター， パケットロス率	SR, PCEP, PING CLI, TWAMP, BGP-LS etc.

本節では，表 2.2 に整理した，使用されるパスの品質向上を目的とした研究と比較し，本研究の立ち位置を明らかにする．

初めに，トラフィックを制御することで，耐障害性と QoS の向上を図った研究について説明する．Sepher ら [46] は，SRv6 を利用した輻輳制御メカニズムである Sepitto v2 を提案した．中間ノードでは Extended Berkeley Packet Filter (eBPF) の機能を使って収集したトラフィック統計を，データプレーンを介してエッジノードに提供し，エッジノードでは，収集した統計データを使って，輻輳リンクを迂回するルートを動的に選択した．Sepitto v2 は，低遅延，低損失といったポリシーに基づき，トラフィックを分散することで，全体的なネットワーク効率を向上させた．Eryc ら [47] は，戦術ネットワークにおけるネットワークの輻輳を緩和し，トラフィックの損失を削減するために，SRv6 と SDN を活用した TactSR アーキテクチャを提案した．戦術ネットワークでは，データの信頼性が極めて重要であるため，ネットワークの可用性と耐障害性を向上させることは極めて重要であった．Path Computing Client (PCC) と呼ばれるノードは，パスの計算を外部に依頼するために，トポロジーやフロー情報を PCE に送信する．PCE において，ネットワーク全体のトポロジーを管理し，輻輳のあるリンクを回避するように SRP を生成することで，トラフィックを分散し，冗長リンクを効果的に活用した．Ohmmar ら [48] は，SDN と SR を活用した QoS に対応したルーティング方法を提案した．SDN コントローラは，ネットワークのリンク情報を収集し，帯域幅や遅延に基づく最適ルートを計算した．また，フローごとに優先度を設定し，重要なフローに優先的にリソース割

り当てを行った．QoS 対応フローは，高いスループットと安定した遅延を維持した．しかし，パケット損失や障害耐性は評価されていなかった．

次に，耐障害性のために BPath を用い，BPath を含めて QoS を向上させた研究について説明する．Zhenlin ら [49] は，SRv6 と SDN を活用した低遅延・高信頼性スライス構築手法を提案した．TWAMP と BGP-LS を使って，ネットワーク全体の QoS メトリクスを収集し，SDN コントローラに転送する．SDN コントローラは，低遅延，低損失の PPath と BPath を計算し，BGP SRv6 Policy によって，計算された経路を定義することで，QoS を考慮したスライシングを実現した．考慮されている BPath は Global Repair のものであった．一方，後述する 3 つの研究は Local Repair の BPath の QoS を評価・向上させることを目的とした．Vítor ら [50] は，最大三つのセグメントで経路を構成することで，ネットワークリソースの最適利用を可能とする SALP-SR を提案した．SALP-SR は，リンク障害後のネットワークパフォーマンスと輻輳管理を向上させることができ，従来の TI-LFA における，障害発生後のネットワーク輻輳に関して考慮されていないという課題を克服した．Liesbeth ら [51] は，TI-LFA が提供する BPath がネットワーク全体に与える付加や遅延の影響を分析した．障害後にトラフィックが正常に復旧する割合，PPath に対する BPath のホップ数の増加率，BPath において同じノードを複数回経由する割合，の三つの観点から評価を行った．Oleksandr ら [12] は，SD-WAN 環境でデータトラフィック伝送の際に，帯域幅とパケット損失率を保護する FaRe QoS スキームを設計した．帯域幅とパケット損失の二つの指標を保護する新しい FaRe のモデルを提案した．各ルーターのバッファ制限を考慮し，フロー条件を満たしながらトラフィックを分散させることで，帯域幅の過負荷を防止した．

第 1.1 節で述べたように，Local Repair の BPath は QoS を保証できないという課題がある．既存研究では，帯域幅を指標とした QoS 保護や，ホップ数の観点から BPath の評価をすることで，上記の課題を解決しようとした．本研究では，FaRe の BPath の遅延時間を測定するフレームワークを実装し，障害後に使用されるパスの遅延時間を提供することを目的とした．

第 3 章

機能要件

本章では，本研究がターゲットとするネットワーク環境を説明する．さらに，提案する測定フレームワークが満たすべき要件について議論する．

3.1 想定環境

1. **対象とする FaRe の計算方法**
   第 2.3 節で述べた手法に加え，様々な目的や環境に適した BPath を算出する手法が存在する中で，本研究では，多くのネットワーク機器ベンダーに実装されていること，および障害シナリオのカバー範囲が広いことを理由に，TI-LFA を採用する．
2. **想定する障害シナリオ**
   TI-LFA がカバー可能な障害シナリオは，単一の中継ノードあるいはリンクに障害が起きた場合である．そのため，本研究では複数箇所の障害に対する高速回復 [52] や，出口ノードの保護 [53] は対象外とする．
3. **想定するネットワーク環境**
   LFA 系列の FaRe は IGP 上で動作する．また，TI-LFA は SR の技術を用いた FaRe である．これら 2 つの理由から本研究では，IGP が動作する SR-MPLS 網を想定する．SRP で指定される候補パスは IGP コストあるいは遅延メトリクスを用いた Dynamic Path のみを想定し，Explicit Path は考慮しないものとする..
   さらに，IGP 上で動作する FaRe を用いる点や，出口ノードの障害シナリオを考慮しない点から，外部のネットワークと接続しない，IntraNet を想定する．

3.2 機能要件

本節では，本研究で提案する測定フレームワークに対する要件を整理する．

1. **アクティブ計測**
   測定対象である FaRe の BPath は，通常時にはトラフィックが流れないため，パッシブ計

測は不可能である．そのため，BPath に測定用のトラフィックを流し，アクティブ計測を行う．アクティブ計測は，実際のトラフィックとは性質が異なるため，その評価に注意を要する．

2. **品質変化に対する追従性**
   ネットワーク品質の評価の際，測定された QoS メトリクスは最新のものである必要がある．測定フレームワークはネットワーク品質の変化を迅速に検出する必要がある．要求される変化検出の所要時間は，ネットワークサービスの要件によって異なる．
3. **デプロイの容易性**
   FaRe において BPath は各ノードで個別に計算されたものであり，BPath の情報を共有するプロトコルは存在しない．しかしながら，BPath の評価に追加のプロトコルが必要となると，ネットワークの運用者に負担を強いることにつながる．そのため，測定フレームワークは既存のプロトコルのみを用いて動作するべきである．
4. **各機能のモジュール性**
   詳細は第 4 章で説明するが，本研究で提案する測定フレームワークは，測定機能とパス計算機能を有する．これらの機能は互いに疎結合であるべきである．

3.3　測定フレームワークの用途

本節では，測定フレームワークの用途について述べる．本研究では，以下の二つの使用方法を想定する．

1. **BPath の遅延時間の測定**
   第 1.1 節で述べたように，FaRe における BPath の性能評価は困難である．そのため，本研究の目的である，BPath の遅延時間の測定に使用することを想定する．
2. **TI-LFA が提供する BPath よりも低遅延なパスの検出**
   遅延測定の結果，TI-LFA によって IGP コストベースで計算される BPath よりも低遅延の BPath を検出することができる．本研究の範囲には，FaRe の BPath 制御は含まれないが，より低遅延の BPath 提供をサポートすることを目的とする．

第 4 章

提案手法

本章では，FaRe の代替パスの遅延時間を測定するフレームワークについて説明する．フレームワークは，ネットワークから遅延メトリクスを収集する機能とパスを計算する機能を有する．本研究では，トポロジー情報をもとにパスを計算した後，PING CLI で遅延メトリクスを測定する PLA-P と，各リンクの遅延メトリクスを測定し，収集したメトリクスを元にパスの遅延時間を計算する PLA-T の二つの手法を提案した．

表 4.1: 測定フレームワークで使用するプロトコル

フレームワーク	測定	メトリクス収集	その他
PLA-P	PING CLI	gRPC	PCE
PLA-T	TWAMP	IGP，BGP-LS	gRPC

4.1 Path Latency Analyzer using PING CLI (PLA-P)

4.1.1 アーキテクチャの概要

本フレームワークは，PING CLI を用いた遅延測定フレームワークであり，表 4.1 に示したプロトコルを用いる．PLA-P は，図 4.1 に示した構成を持つ．トポロジー情報に基づいて測定対象となる BPath を探索し，発見された BPath を候補パスとする SRP を作成する．作成された SRP は，PCEP プロトコルを用いてネットワーク上の PE ノードに適用される．適用された SRP が有効であることが確認した後，PING CLI を使って，遅延メトリクスを収集する．PLA-P とネットワーク上のノード間の通信のうち，PCEP を除いたものは全て gRPC 通信を用いる．

4.1.2 構成要素

PLA-P は Path Explorer (PaE) と Metrics Collector using PING CLI (MC-P) の二つの要素で構成される．

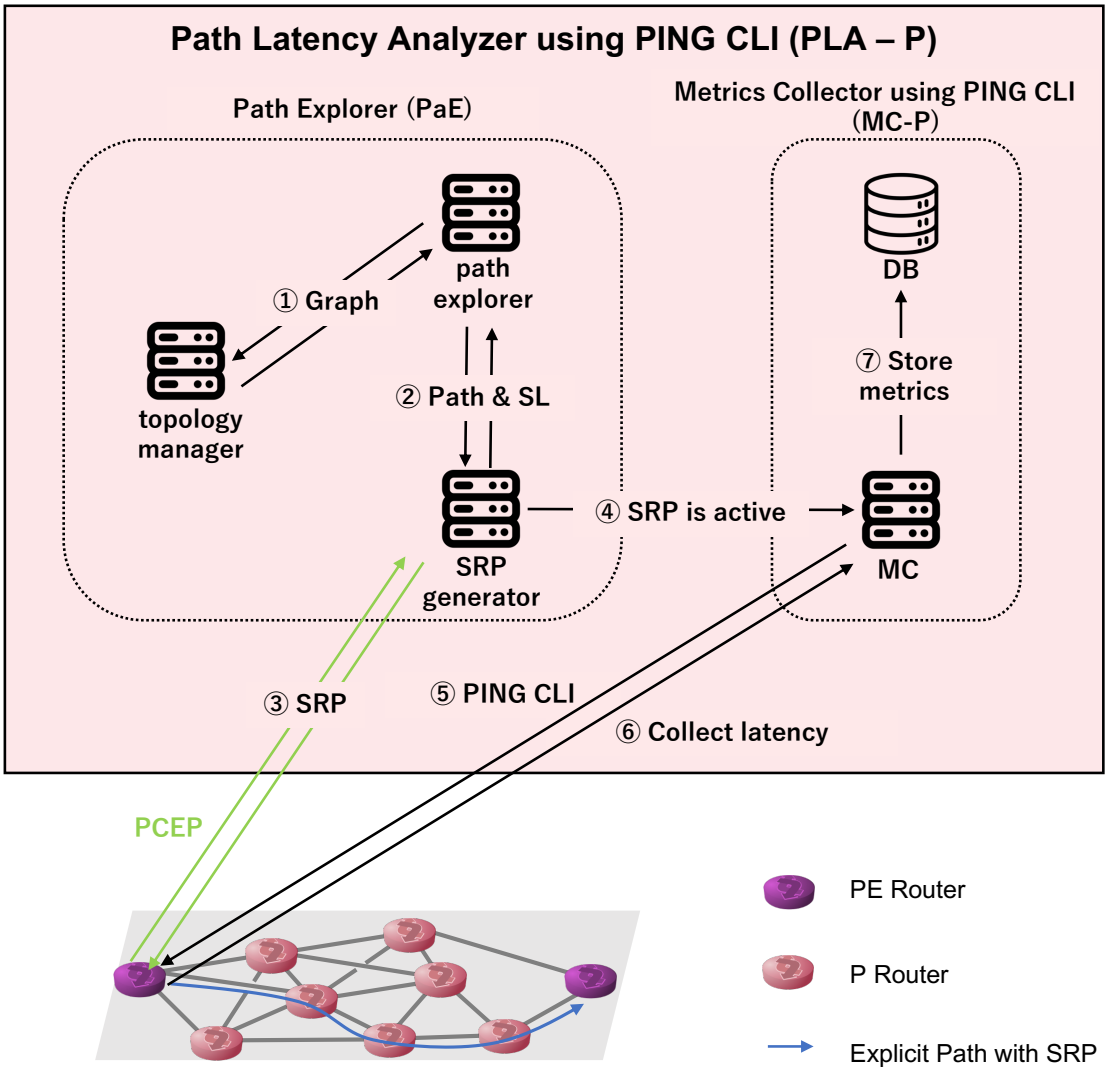


図 4.1: PLA-P のアーキテクチャ概要

Path Explorer (PaE)

PaE はネットワークトポロジーの管理機能とパスの探索機能，探索されたパスを候補パスとする SRP の生成機能，SRP をネットワーク上のノードに適用する機能の四つを有する．

- **ネットワークトポロジーの管理機能**

  図 4.1 の topology manager は，ネットワークトポロジーを管理する．対象のネットワーク上のノードとリンク情報，各ノードとインタフェースに結び付けられた SID 情報，保護箇所の情報を管理する．上記の三種の情報はパスの探索に使用される．

- **パス探索機能**

  図 4.1 の path explorer は，topology manager が管理するグラフを用いて，パスの探索をする．BPath の探索は，グラフに含まれる保護箇所に関する情報を用いて行われる．path explorer

は，パスを探索すると共に，発見されたパスに従ってパケットを転送するための SL も生成する．探索アルゴリズムの詳細は，第 4.1.3 節で説明する．

- **SRP の生成機能**
  図 4.1 の SRP generator は，path explorer が計算した SL を候補パスとする SRP を作成する．SRP は後述する MC-P で使用される．
- **ネットワーク上のノードとの通信機能**
  ネットワーク上のノードに SRP を適用し，SRP が有効に設定されたことを確認するために，PCEP を使用する．ノードに適用された SRP が有効である場合，MC-P に通達する．

Metrics Collector using PING CLI (MC-P)

MC-P はノード上で PING CLI を実行するためのリクエストを送信し，RTT を収集する機能，および収集した遅延メトリクス情報を管理する機能を持つ．

- **PING CLI の実行リクエストの送信機能**
  MC-P は PING CLI リクエストを生成し，gRPC 通信を通してノードにリクエストを送信する機能を有する．生成される PING CLI リクエストには，SR manager が生成した SRP が含まれており，SRP の候補パスに従って ICMP パケットを転送する．
- **RTT の収集機能**
  PING CLI の実行結果の RTT を，上述した PING CLI の実行リクエストのレスポンスとする．ある SRP に従う PING CLI を実行し，実行結果である RTT を取得するまで，一つの gRPC セッションで完了する．
- **収集した遅延メトリクス群を管理する機能**
  収集された複数の PING CLI の実行結果から，RTT の平均値，最大値，最小値ばらつき，パケットロス率を計算し，データベースに格納する．

4.1.3 パス探索アルゴリズム

本アルゴリズムは，最短パスではなく，全候補パスを探索するためのものである．本フレームワークでは，計算されたパスに対して測定を行い，遅延メトリクスを収集するため，最小遅延パスを計算できない．そのため，SRP と PING CLI を使って，全候補パスの遅延メトリクス群を収集する．

候補パスは深さ優先探索 (Depth-First Search, DFS) [54] とその応用によって計算される．DFS は，グラフやツリー構造を探索するためのアルゴリズムの一つで，「深さ」優先で進められ，可能な限り深く探索した後，戻りながら未探索の分岐を探索する手法である．本研究では，FaRe によって提供される BPath を考慮した候補パスを計算するために，訪問履歴の初期化の概念を導入した．本来の DFS は一度訪問したノードを再度訪れることはないが，探索の際に保護されたリンクを使用する場合，その時点におけるノードを始点として再度 DFS を実

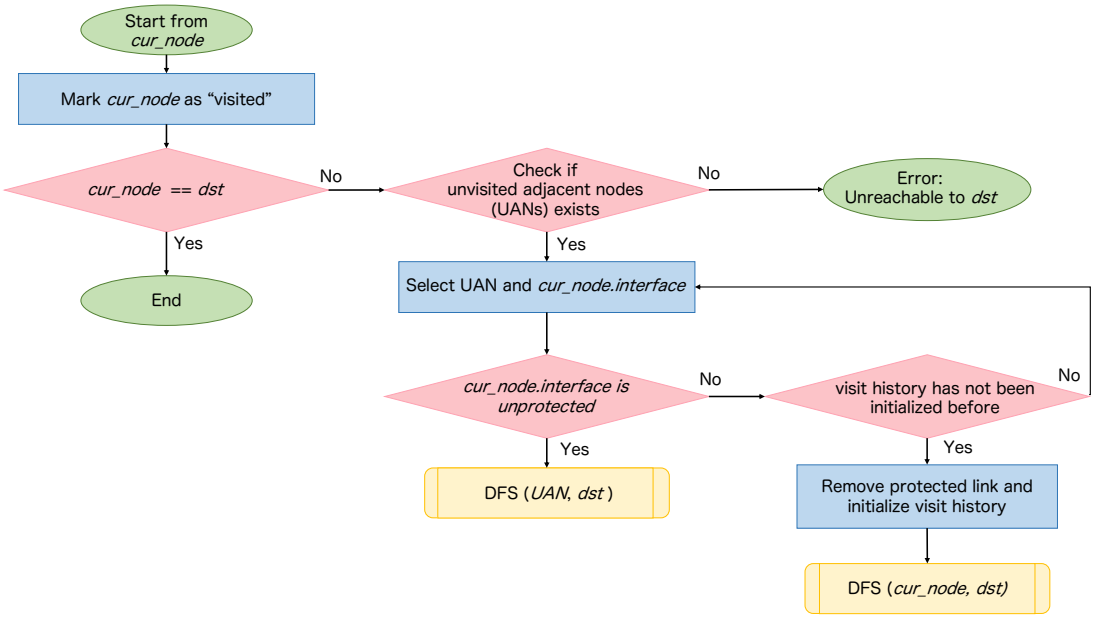


図 4.2: BPath を考慮したパス探索フロー

行することを一度のみ許可した．

訪問履歴の初期化を導入した BPath の探索アルゴリズムは図 4.2 のフローチャートに従って進められる．パス探索は送信元ノードを起点として開始される．基本的には，訪れたノードを”visited”と記録しながら探索し，宛先ノードに到着したら探索が終了する．ネクストホップとして選択するノードは”visited”の記録がない未訪問の隣接ノード (UAN) に限り，順に探索されていく．図 4.2 における変数 $cur\_node.interface$ は，現在訪問しているノード cur_node から UAN にホップする際に使用するインタフェースを指し，保護されていない (FaRe が未設定) ならば，通常の DFS と同様に現在のノードのネイバーを起点として更に探索が進められる．一方，$cur\_node.interface$ が保護されている場合，訪問履歴の初期化が行われたことがあるか確認する．初期化が行われていないならば，保護されたリンクをグラフから除いた上で，訪問履歴を初期化し，現在のノードを起点として探索を再開する．初期化が行われたことがあるならば，現在の探索を中断し，別の UAN を探索する．

4.2　Path Latency Analyzer using TWAMP (PLA-T)

4.2.1　アーキテクチャの概要

本フレームワークは，TWAMP を用いた遅延測定フレームワークであり，表 4.1 に示したプロトコルを用いる．PLA-T は，図 4.3 に示した構成を持つ．隣接するノード間で TWAMP のセッションを張り，リンクごとの遅延メトリクス群を測定する．測定されたメトリクスは IGP を用いてネットワーク全体に広報される．メトリクスの測定と広報は図 2.1 に示したフローに従って行われる．リンクステート情報 (遅延メトリクス群，トポロジー情報，SID 情報，

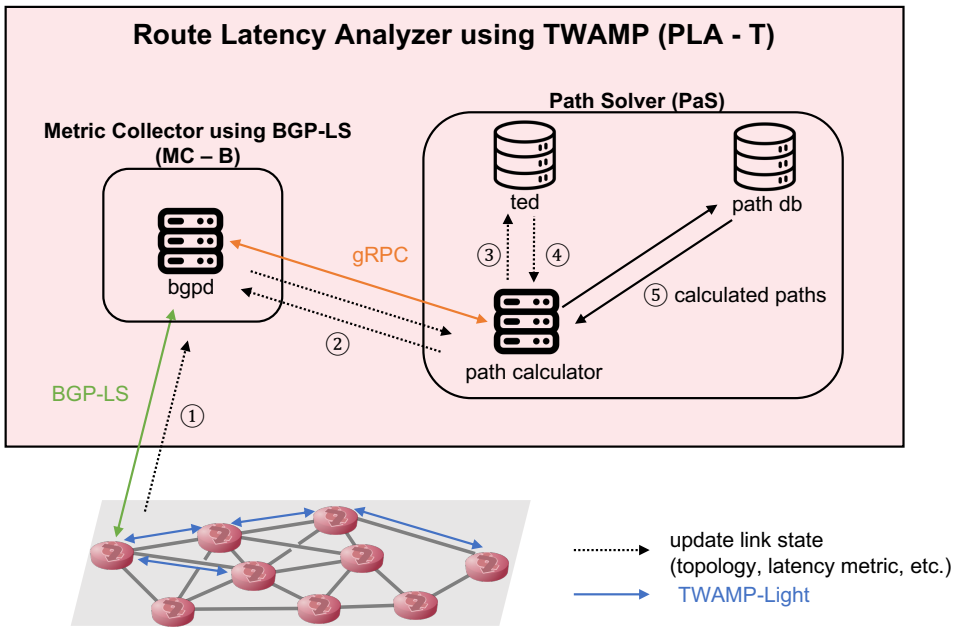


図 4.3: PLA-T のアーキテクチャ

FaRe により保護されているインターフェース情報) は BGP-LS を使ってメトリクス収集機能を持つ Metrics Collector using BGP-LS (MC-B) に集計される．上述したリンクステート情報が BGP-LS において，どのようなデータ構造で広報されるかを付録 C で追記する．

集計されたリンクステート情報は gRPC 通信を使って, パス計算機能を持つ Path Solver (PaS) に転送される．PaS は，トポロジー情報や保護インタフェース情報，遅延メトリクス群を用いてパスを計算する．

4.2.2 構成要素

PLA-T は MC-B と PaS の二つの要素で構成される．

Metrics Collector using BGP-LS (MC-B)

MC-B は，メトリクスの収集機能と PaS との通信機能を有する．これら二つの機能は，PLA-T が行う測定のリアルタイム追従性のボトルネックとなるため，高速な処理が必要である．

- **BGP-LS を用いたメトリクスの収集機能**
  MC-B は BGP デーモンとしての機能を有する．ルーターと BGP セッションを確立して，遅延メトリクス群を含むリンクステート情報を収集・保持する．ネットワークのリンクステート情報が更新されると，BGP-LS を介して，保持されたリンクステート情報も更新される．そのため，MC-B におけるリンクステート情報の更新頻度は，IGP と BGP-LS のメトリクス広報の周期に依存する．

- **gRPC を用いた PaS との通信機能**
  MC-B は gRPC サーバとしての機能を有する．PaS は gRPC クライアントとして，MC-B にリクエストを送信し，リンクステート情報を取得する．

Path Solver (PaS)

PaS は，gRPC を用いて MC-B からリンクステート情報を収集する機能，リンクステート情報を管理する機能，PPath と BPath の計算機能を有する．

- **gRPC を用いた MC-B との通信機能**
  PaS は gRPC クライアントとして，MC-B にリクエストを送信し，リンクステート情報を取得する．
- **リンクステート情報の管理機能**
  PaS は MC-B から取得したリンクステート情報を，自身の持つ TED に格納する．
- **パス計算機能**
  障害シナリオごとに，TED に格納されたリンクステート情報を用いて以下の四つのパスを計算する．
  1. IGP コストに基づいて計算された，最短の PPath とその遅延時間
  2. 対象の障害シナリオにおける，IGP コストに基づいて計算された，最短の BPath とその遅延時間
  3. 遅延メトリクスに基づいて計算された，最小遅延の PPath とその遅延時間
  4. 対象の障害シナリオにおける，遅延メトリクスに基づいて計算された，最小遅延の BPath とその遅延時間

  本研究では，パス 3, 4 を計算する際に，遅延メトリクス群のうち遅延メトリクスのみを使用した．パス計算の詳細は第 4.2.3 節で説明する．

4.2.3　パス計算アルゴリズム

パスの計算には，以下の 3 つの情報が必要である．

1. ネットワークトポロジー：各ノードの識別情報，ノード同士がどのように接続されているかの情報
2. 各リンクのコスト：各リンクにおける IGP コストと遅延メトリクスの値
3. 保護インタフェース：FaRe によって，障害発生時に迂回することが保証されているインタフェースの情報

最短経路は，ダイクストラ法 [55] とその応用によって計算される．ダイクストラ法は重み付きグラフにおいて，始点から全ての頂点への最短経路を効率的に求めるアルゴリズムである．本研究では，Algorithm 1 - 3 を用いて，IGP コストあるいは遅延メトリクスを重みとし

た，送信元ノードから宛先ノードまでの最短の PPath と BPath を算出した．Algorithm 1 は，ネットワークトポロジー情報とリンクコスト情報を使って，最短経路を計算するアルゴリズムであり，Algorithm 2 は，計算途中のノードから最小コストのノードを求めるアルゴリズムである．一方，Algorithm 3 は，Algorithm 1 で求められる最短経路情報を使って，送信元ノードから宛先ノードまでの経路情報，メトリクスの合計値，必要な遅延時間を抽出するアルゴリズムである．

初めに，PPath の計算方法について説明する．送信元ノードを *srcNode*，宛先ノードを *dstNode* として，Algorithm 1 を使用する．変数 *calcNodes* は，ダイクストラ法で計算対象となるノードの集合を管理するデータ構造である．*calcNodes* に含まれるノードのうち，*visited* が *true* の場合，最短経路が決定したことを示し，*false* の場合，計算途中を示す．変数 *localNode* は, Algorithm 2 に従って算出された, 計算途中のノードのうち, 最小コストのノードを示す (L4)．そして，*localNode* の隣接ノード *remoteNode* それぞれについて探索する (L11)．変数 *cost* は，パス 1, 2 を計算する場合，IGP コストを指し，パス 3, 4 の計算の場合，遅延メトリクスを指す (L12, 17, 22)．*remoteNode* が計算途中の場合 (L15)，*localNode* を経由した経路が経由した経路より低コストならば (L16)，*remoteNode* の情報 (*cost*, *latency*, *prevNode*) を更新する (L17,18)．*remoteNode* が初めて計算されるノードならば，新たに *calcNodes* に追加する (L22-24)．全ノードの計算が完了すると，srcNode を起点とした際の全ノードへの最短経路情報が格納された *calcNodes* が出力される．そして，*calcNodes* と宛先ノード *dstNode* として，Algorithm 3 を使用し，PPath の最短経路とその遅延時間を算出する．以上の手順で，パス 1, 3 が求められる．

一方，BPath は，Algorithm 1 に更なる条件を加えることで算出される．図 4.4 が示すように，計算された PPath が，保護されたインタフェースを用いる場合，BPath の計算プロセスが動作する．保護対象のリンクをトポロジーから除外した上で，送信元ノードから PLR まで，PLR から宛先ノードまでの二つの最短経路を計算する．そして，二つの経路の経路情報，遅延時間を統合したものを BPath として算出する．

Algorithm 1: ダイクストラ法を用いた最短経路探索アルゴリズム

Input: *srcNode*, *dstNode*

Output: *calcNodes*

```
1  srcNode.visited ← true;
2  calcNodes ← [srcNode];
3  while true do
4      localNode ← nextNode(calcNodes);
5      if localNode = null then
6          return Error;
7      end if
8      if localNode = dstNode then
9          break;
10     end if
11     foreach link ∈ localNode.links do
12         newCost ← localNode.cost + link.metric;
13         newLatency ← localNode.latency + link.latency;
14         remoteNode ← link.remoteNode;
15         if remoteNode ∈ calcNodes then
16             if newCost < remoteNode.cost then
17                 remoteNode.cost, remoteNode.latency ← newCost, newLatency;
18                 remoteNode.prevNode ← localNode;
19             end if
20         end if
21         else
22             remoteNode.cost, remoteNode.latency ← newCost, newLatency;
23             remoteNode.prevNode ← localNode;
24             calcNodes.Insert(remoteNode);
25         end if
26     end foreach
27     localNode.visited ← true;
28 end while
29 return calcNodes;
```

Algorithm 2: 次に探索するノードを決定するアルゴリズム

Input: *calcNodes*

Output: *nextNode*: The next node to process

```
1  nextNode ← None;
2  foreach curNode ∈ calcNodes do
3      if curNode.visited then
4          continue;
5      end if
6      if nextNode = None or nextNode.cost > curNode.cost then
7          nextNode ← curNode;
8      end if
9  end foreach
10 if nextNode = None then
11     return Error: ”Next node not found”;
12 end if
13 return nextNode;
```

Algorithm 3: 探索済みパスから最短経路と遅延時間を抽出するアルゴリズム

Input: *calcNodes*, *dstNode*

Output: *shortestPathInfo*

```
1  curNode ← dstNode;
2  while curNode ≠ srcNode do
3      metricList.Insert(curNode.cost);
4      latencyList.Insert(curNode.latency);
5      nodeList.Insert(curNode);
6      curNode ← curNode.prevNode;
7  end while
8  shortestPathInfo.nodeList ← nodeList;
9  shortestPathInfo.metricList ← metricList;
10 shortestPathInfo.latencyList ← latencyList;
11 return shortestPathInfo;
```

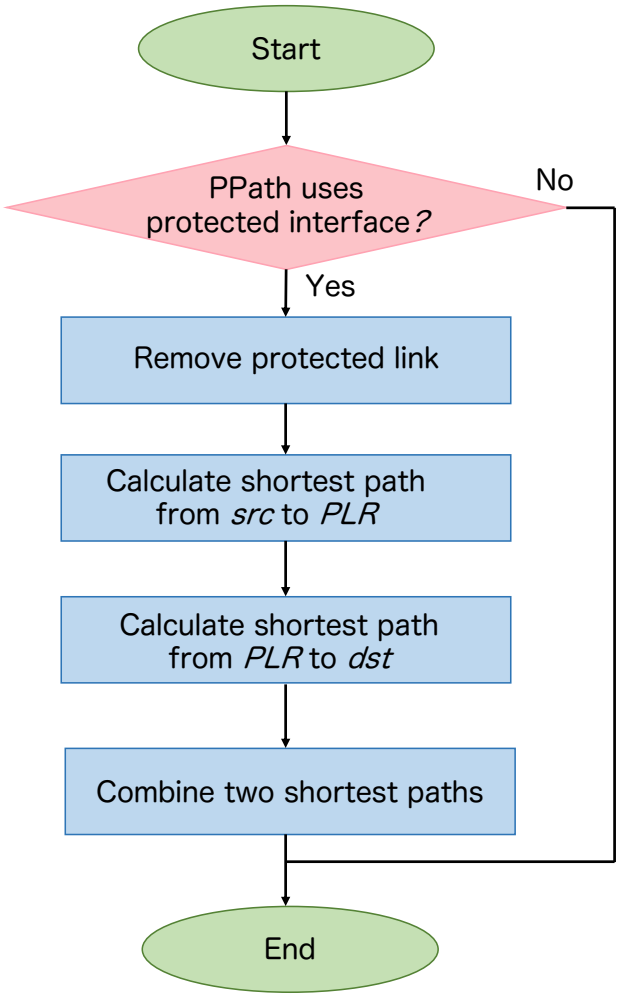


図 4.4: BPath の計算フロー

第 5 章

実装

本章では，第 4 章で述べた PLA-P と PLA-T の実装の詳細について説明する．

5.1　PLA-P の実装

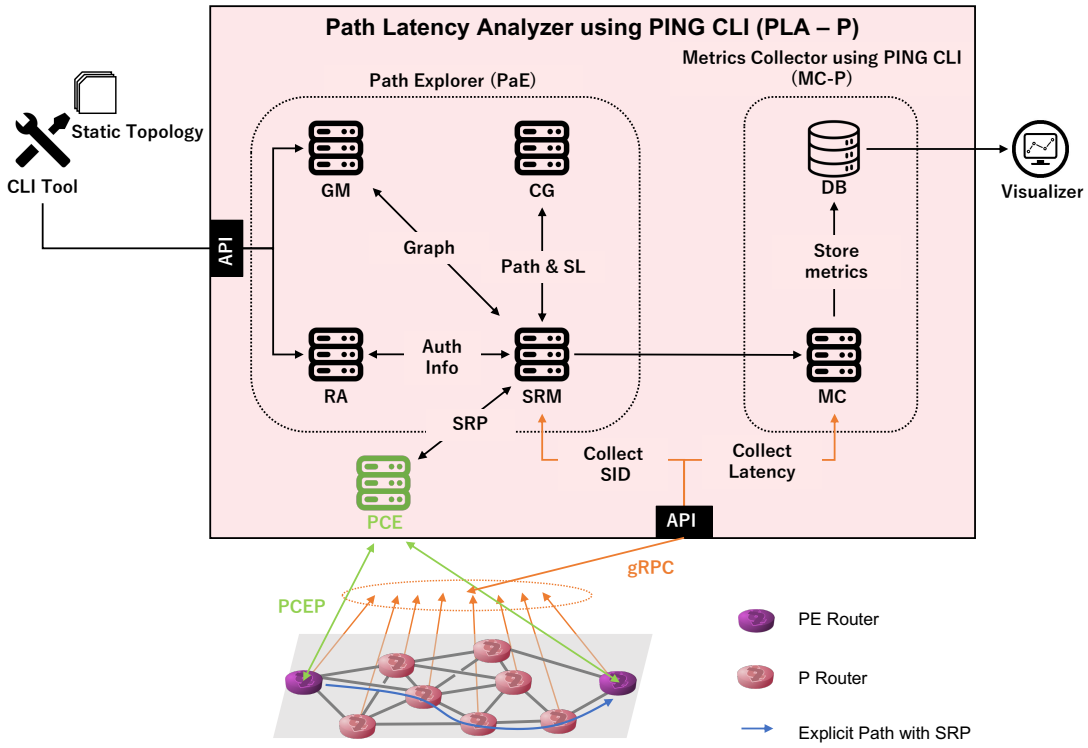


図 5.1: PLA-P の詳細実装

本節では，第 4.1 節で説明した PLA-P の実装内容について説明する．PLA-P の構成を図 5.1 に示す．

5.1.1　実装環境

表 5.1: PLA-P の実装に使用した環境・ソフトウェアの詳細

	バージョン	役割
OS	Ubuntu 20.04.6 LTS	実装環境
Golang	v1.22.2 linux/amd64	実装言語
Xrgrpc	v0.6.0	Cisco IOS-XR と gRPC するためのライブラリ
Pola PCE	v1.3.0	SRP をルーターに適用するための PCE
InfluxDB	v2.7.1	収集した遅延メトリクス群を格納するデータベース

PLA-P の実装には，表 5.1 に示される環境・ソフトウェアを用いた．本フレームワークは Golang を用いて実装され，ネットワーク上のルーターと gRPC 通信するために Xrgrpc [56] を，生成した SRP をルーターに適用するためのインタフェースとして Pola PCE [57] を利用した．Xrgrpc は Cisco 社の IOS-XR が動作するルーターと gRPC 通信するためのライブラリである．そのため，Xrgrpc を用いて実装された本フレームワークは，IOS-XR の OS が動作するルーターとの通信に限定される．

5.1.2　PaE の実装

PaE はネットワークトポロジーの管理，ルーターとの接続の認証管理，パス探索，SID の収集，SRP の生成，SRP のルーターへの適用の六つの機能を有し，それぞれの機能を担当する以下の四つの要素で構成される．

表 5.2: SID 収集時に実行されるコマンド

IGP	収集対象	実行コマンド
IS-IS	Node SID	show isis segment-routing label
	Adj SID	show isis segment-routing label adjacency persistent
OSPF	Node SID	show ospf sid-database
	Adj SID	show ospf segment-routing adjacency-sids configured

1. **Graph Manager (GM)**
   GM は，入力されたトポロジー情報をもとに，パス計算に使用されるグラフを生成する機能を有する．グラフの生成には，SRM から受け取った SID 情報も使用される．
2. **Router Authenticator (RA)**
   RA は，ネットワーク上のルーターと PLA-P 間の通信に必要な認証情報を管理する機能を有する．ルーターへの接続にはパスワード認証を使用している．

表 5.3: Pola PCE に送信される SRP の情報

Information	Description
PE Router Address	SRP を適用するルーターのアドレス． Pola PCE から接続し，PCEP セッションを張ることが可能．
Source Address	PING CLI の送信元アドレス．
Destination Address	PING CLI の宛先アドレス．SRP の Endpoint に使用．
SL	CG が発見した SL．Explicit Path として SRP の候補パスに登録．
Color	SRP に設定する Color．900 番台を使用し，SRP ごとに一意に決定．

3. **Candidate paths Generator (CG)**
   CG は，GM が生成したグラフをもとにパスを計算する機能を有する．パス計算は第 4.1.3 節のアルゴリズムに基づいて行われる．BPath の計算に必要な障害シナリオの情報は，GM が生成したグラフに含まれる．
4. **SR Manager (SRM)**
   SRM は，ネットワーク上のルーターから Node SID と Adj SID を収集し，GM に提供する機能と，CG が計算したパスを候補パスとする SRP を生成し，ルーターに適用する二つの機能を有する．
   SID は gRPC 通信を通して，表 5.2 に示されるコマンドが実行して収集される．実行されるコマンドは，対象のネットワークで OSPF あるいは IS-IS が動作している時，Node SID と Adj SID を表示するものであり，PaE は出力結果をレスポンスとして受け取る．
   SRM で生成される SRP は Pola PCE を介してルーターに適用される．Pola PCE には表 5.3 に示される五つの情報が送信される．ルーターに登録される SRP は，Color と Endpoint (Destination Address) により一意に定まり，MC-P で SRP を指定可能となる．また，Pola PCE は Stateful PCE であるため，発行済みの SRP の保持・管理が可能である．そのため，Pola PCE を通して，適用した SRP が有効であるかを確認できる．

5.1.3 MC-P の実装

MC-P は，SRM が生成した SRP に従って PING CLI を実行し，遅延メトリクス群を収集・管理する機能を有する．

MC-P が送信する PING CLI の実行リクエストは，図 5.2 に示すように，JSON [58]*形式で生成され，Yet Another Next Generation (YANG) [59]†というデータモデリング言語に変換されたのち，ルーターに送信される．図 5.2 のリクエストの例では，送信元アドレスを 11.11.11.11 とし，srte_c_900_ep_22.22.22.22 という名称の SRP で指定された経路に従って，ICMP パケッ

*JavaScript Object Notation の略称で，データモデリング言語の一種．
†ネットワーク機器の設定や状態管理，手続き呼び出しを人間が読みやすい形で表現する言語．

```
"Cisco-IOS-XR-mpls-ping-act:mpls-ping": {
    "sr-mpls": {
        "policy": {
            "name":"srte_c_900_ep_22.22.22.22",
            "lsp-endpoint":"22.22.22.22"
        }
    },
    "request-options-parameters": {
        "interval":1,
        "repeat":5,
        "source":"11.11.11.11"
    }
}
```

図 5.2: MC-P で生成される，SRP に紐づいた PING コマンド実行リクエストの例

トを 5 個，1 s の間隔で送信する．リクエストは 11.11.11.11 のアドレスを持つルーターに対して，gRPC 通信を使って送信される．

PLA-P は，送信元ルーター，宛先ルーター，障害箇所の組み合わせごとに，複数の候補パスを用意し，上述したリクエストを送信する．図 5.3 に示すように，複数のプロセスを並列に実行する．Pola PCE により，アクティブであることが確認された SRP 一つに対して，一つのプロセス上で PING CLI のリクエストを行う．各プロセスでは，SRP を使って図 5.2 にある PING リクエストの作成，リクエストの送信，遅延メトリクス群の収集が行われる．SRP が非アクティブになる，あるいはレスポンスが一定時間返ってこない場合，プロセスは終了する．

収集された遅延メトリクス群は DB に格納される．本研究では，InfluxDB [60] を採用した．InfluxDB は InfluxData 社が開発したオープンソースのデータベースで，時系列データの管理に優れている．

5.2　PLA-T の実装

表 5.4: PLA-T の実装に使用した環境・ソフトウェアの詳細

	バージョン	役割
OS	Ubuntu 22.04.5 LTS	実装環境
Golang	v1.22.7 linux/amd64	実装言語
GoBGP	v3.30.0	BGP デーモン

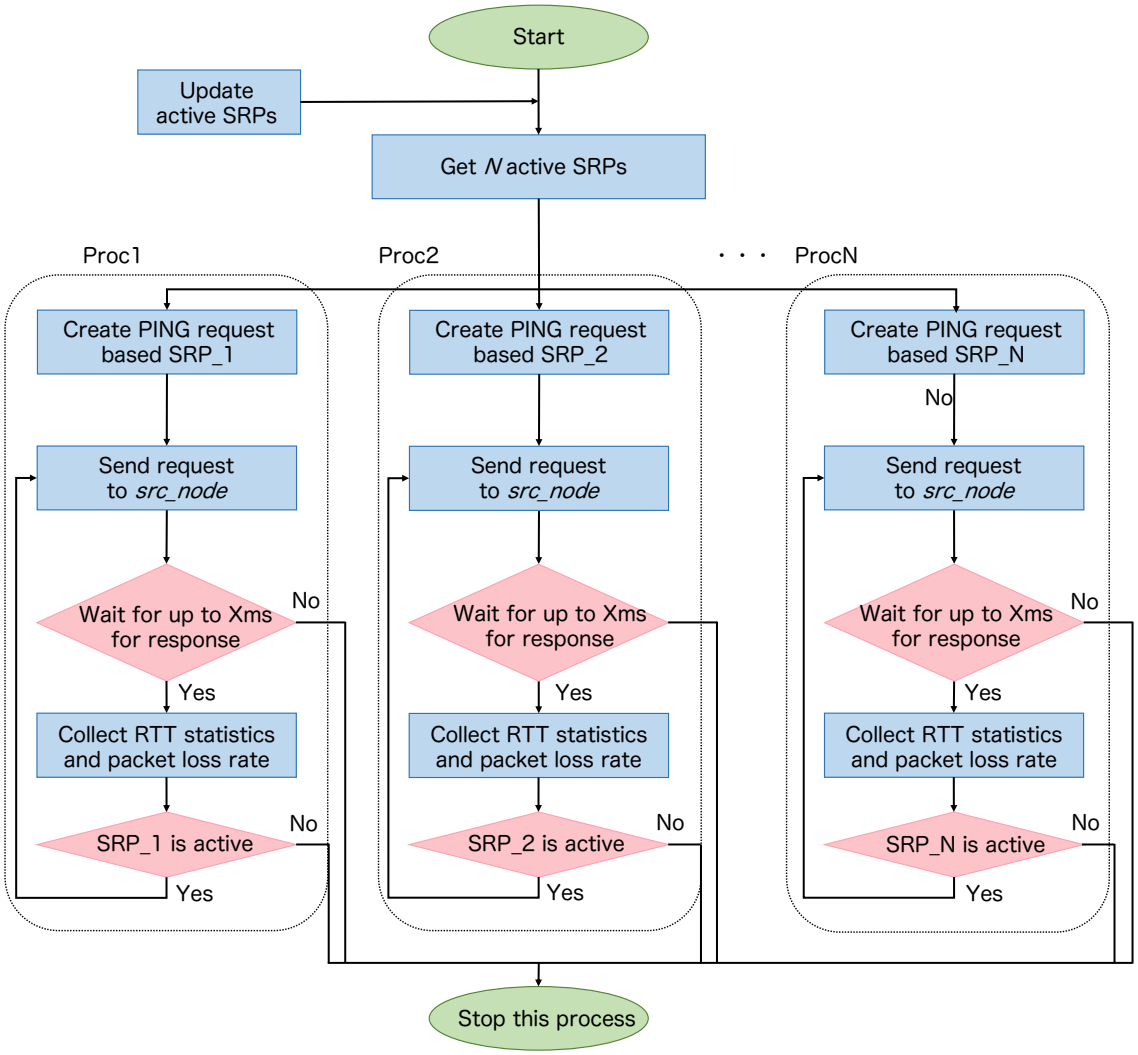


図 5.3: MC-P で動作する複数プロセスの並列処理

5.2.1 実装環境

PLA-T の実装は，表 5.4 に示される環境・ソフトウェアで行われた．本フレームワークは Golang を用い，その全てをユーザ空間で実装した．

5.2.2 MC-B の実装

MC-B の実装には，GoBGP [61] というオープンソフトウェアを用いた．GoBGP は Golang を用いて開発されており，柔軟で拡張性の高い BGP ルーターを構築するためのフレームワークとして広く使用されている．GoBGP は BGP-LS に対応しているが，RFC8571 [62] で定義される TE メトリック拡張のリンク属性 TLV は v1.31.0 の時点で実装されていない．そのため，単方向リンク遅延，最小/最大単方向リンク遅延，単方向遅延変動の三つの TLV の実装

を追加し[†]，TWAMP で測定した遅延メトリクス群を BGP-LS で受信できるようにした．また，FaRe の設定の有無を確認するために，BGP-LS Path Attribute に含まれる，Adjacency SID TLV の Backup Flag (B-Flag) を利用した [63,64]．B-Flag がセットされている場合，対象のリンクが保護されていると判断する．

5.2.3　gRPC を用いた通信

本フレームワークは，MC-B が gRPC サーバとして，PaS が gRPC クライアントとして動作する．gRPC で送受信するデータフォーマットにプロトコルバッファ [65] を採用した．プロトコルバッファは Google 社が開発した軽量なデータシリアライズ形式で，バイナリ形式でデータをエンコードすることで，JSON や XML といったデータ形式よりも高速かつ小型のデータ表現が可能である．gRPC とプロトコルバッファを用いることで，MC-B と PaS 間の通信を効率化した．

5.2.4　PaS の実装

TED の更新周期を 10 s に設定し, PaS は MC-B に対して 10 s おきにリクエストを送信し，最新のリンク情報を取得するように実装した．また，パス計算プロセスも TED の更新のたびに実行されるように設計した．この時，トポロジー上の全てのルーターを使って，送信元と宛先ルーターの組み合わせを試行した．パスは，IGP コストと遅延メトリクスの二つのメトリクスを用いて計算した．計算された PPath と BPath は，送信元と宛先ルーターの Router ID をキーとして管理された．

[†] https://github.com/Yosshi72/gobgp/tree/add-bgpls-advertisement-delay-metrics

第 6 章

評価

本章では，第 4 章で説明した二つの測定フレームワークの性能を評価するために行った実験について述べる．評価の指標として，下記式に示される，測定精度 E_{acc} と遅延変動応答性 δ_T の二つの指標を設けた．

$$E_{acc} = \frac{|\tau_{actual} - \tau_{probe}|}{\tau_{actual}} \tag{6.1}$$

$$\delta_T = T_{detect} - T_{event} \tag{6.2}$$

ここで，測定精度 E_{acc} は，実際にトラフィック転送に要する時間 τ_{actual} と測定フレームワークが測定した τ_{probe} の相対誤差を示す．τ_{actual} は，$\tau_{probe} - 0.5$ から $\tau_{probe} + 0.5$ の間のトラフィックの転送時間の平均値を示す．遅延変動応答性 δ_T は，ネットワークの遅延が変化してから，測定フレームワークがその遅延変動を検出するまでに要する時間である．また，測定値が，段階的に遅延の変動を追従した場合，実トラフィックの転送時間と測定値との誤差が 5 ms 以内になったタイミングを T_{detect} とする．E_{acc}, δ_T はいずれもその値が小さいほど，測定フレームワークの性能が良いといえる．

6.1 小規模ネットワークにおける測定フレームワークの評価

6.1.1 実験環境

表 6.1: 各ノードの OS

種別	OS
PE ルーター	Cisco IOS-XR v7.9.1, Junos 22.4R1.10
P ルーター	Cisco IOS-XR v7.9.1, Junos 22.4R1.10
CE ホスト，TCA	Ubuntu22.04

本節の評価は, 図 6.1 に示されるネットワークトポロジーを使って行われた．2 台の Provider Edge (PE) ルーターと 4 台の Provider (P) ルーター，2 台の Customer Edge (CE) ホストで構成

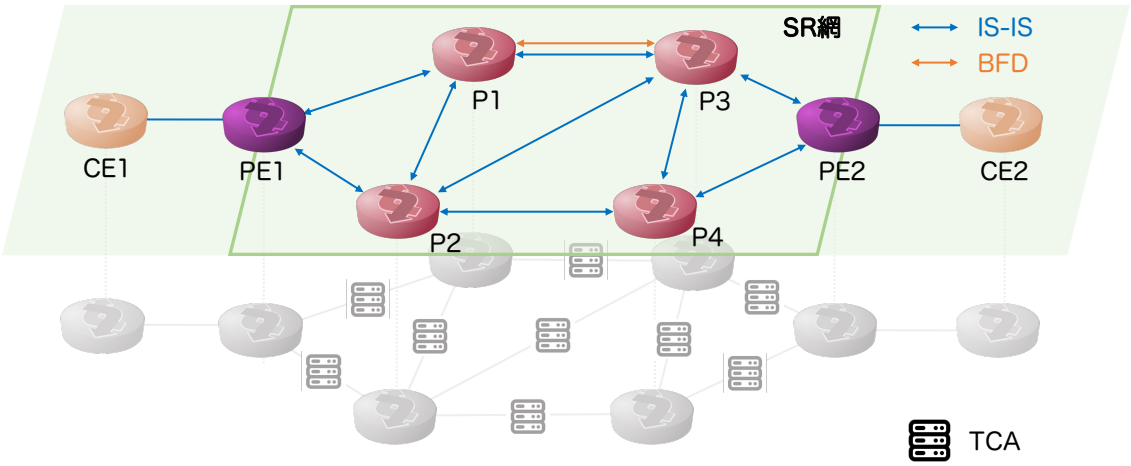


図 6.1: 実験で使用する小規模ネットワークトポロジー

表 6.2: 実験で使用したソフトウェア

ソフトウェア	バージョン
docker	v27.3.1
containerlab	v0.59.0
vrnetlab	v0.20.1

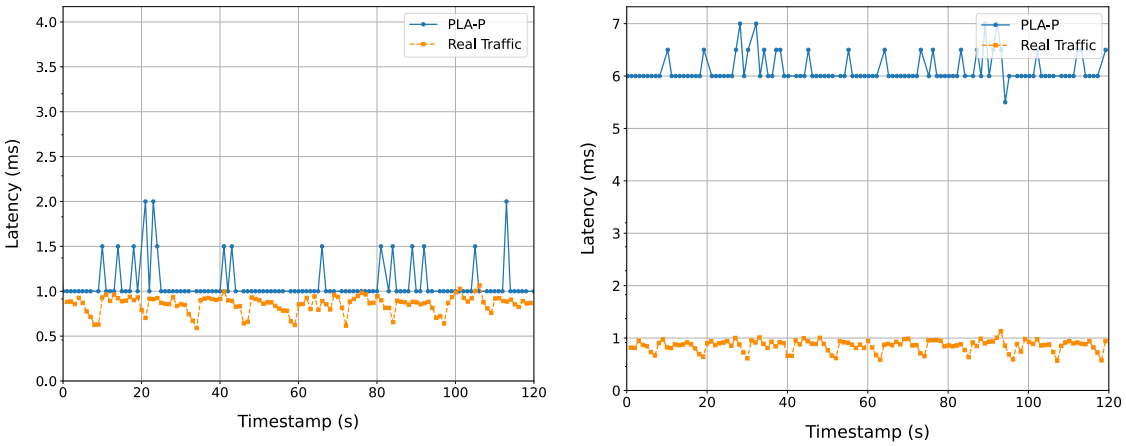
表 6.3: 小規模トポロジーにおけるネットワーク設定

IGP	IS-IS
障害シナリオ	P1, P3 間のリンク障害
BFD	P1, P3 間
PPath	PE1 → P1 → P3 → PE2
BPath	PE1 → P1 → P2 → P4 → P3 → PE2
ProInt [s]	5
ProCon [-]	10
AdsThr [%]	10

したネットワークを用いた．各ルーター間には，Traffic Control CLI を実行する Traffic Control Agent (TCA) を配置し，トラフィックの遅延とジッターを制御した．各ルーターとホストに使用した OS は表 6.1 に示すように，PLA-P の検証には Cisco IOS-XR v7.9.1，PLA-T の検証には Junos 22.4R1.10 を用いた*．全ての実験は仮想ネットワーク上で実施し，表 6.2 に示すソフトウェアを用いて構築した†．ネットワークの設定は表 6.3 に示した通りである．アンダー

*PLA-P における Xrgrpc の制約，PLA-T における TWAMP の制約のため．

†ノードは全て docker コンテナ化された．vrnetlab (`https://github.com/hellt/vrnetlab`) は Virtual Machine ベースの仮想ルーターイメージを docker イメージに変換するのに用いた．



(a) P4 - PE2に遅延なし　(b) P4 - PE2に10 msの遅延

図 6.2: 実験 P1 における PLA-P の測定結果

レイネットワークは IS-IS で構築し，想定する障害シナリオは，P1 と P3 を結ぶリンクの障害に限定した．障害検出には BFD プロトコルを採用し，障害の早期発見を試みた．さらに，PPath，BPath は表 6.3 に示す経路になるようにコストを調節した．ProInt, ProCon, AdsThr の設定は PLA-T の性能検証実験でのみ使用された．

評価に使用される τ_{actual} の測定には，PING CLI を用いた．時刻同期*された CE1, CE2 でパケットキャプチャをし，シーケンス番号が一致する ICMP Echo Request について，タイムスタンプの差分をとることで，CE1 から CE2 までの片道遅延時間を求めた．

6.1.2　PLA-P の性能評価

本節では，PLA-P の性能評価のためのいくつかの実験について，その手順と結果を説明する．PLA-P では，PING CLI により RTT が収集されるため，得られたメトリクスの値を半分にしたものを片道遅延時間とみなした．以後，PLA-P の遅延時間は全て測定した RTT を半分にしたもを指す．

表 6.4: 実験 P2 における遅延変動シナリオ

時間 [ms]	0 - 60	60 - 180	180 - 300	300 - 420	420 -
$TCA_{1,3}$ が設定した遅延時間 [ms]	N/A	30	50	10	N/A
$TCA_{1,3}$ が設定したジッター [ms]	N/A	1	10	1	N/A

P1. 遅延時間が一定の場合における性能評価

PLA-P が測定する遅延時間の精度を評価する実験を行った．本研究では，パス PE1 → P1

*docker ホストの/usr/share/zoneinfo/Asia/Tokyo とコンテナの/etc/localtime をファイルマウントした．

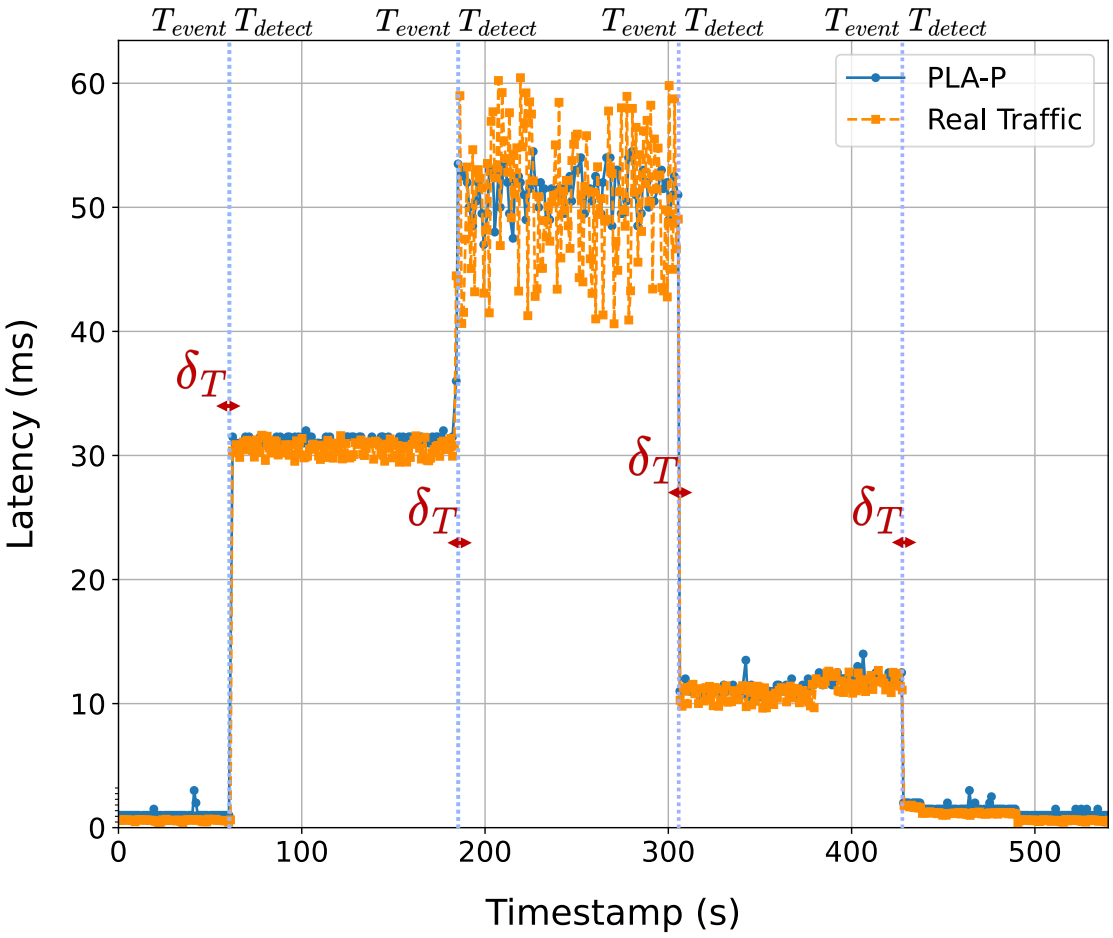


図 6.3: 実験 P2 における PLA-P の測定結果

→ P2 → P4 → P3 → PE2 について，PLA-P が測定した遅延時間と実トラフィックの遅延時間を比較した．

結果を，図 6.2 に示した．PLA-P という凡例は，PLA-P が測定した片道遅延時間を，Real Traffic という凡例は，実際のトラフィックのパケット転送に要した片道遅延時間を示している．図 6.2 (a) では，PLA-P は，サブミリ秒程度の測定誤差で遅延時間を測定できたが，(b) のように，P4 と PE2 間に配置された $TCA_{2,e2}$ で遅延を設定すると，測定精度は大きく下がった．また，PLA-P の測定結果には，1 s 程度のスパイクノイズが確認された．

P2. **遅延時間が変動する場合における性能評価**

P1 と P3 間に配置された $TCA_{1,3}$ により，遅延とジッターの変動が引き起こされる場合における，PLA-P の測定精度，変化への応答性を評価した．表 6.4 に示されるシナリオで遅延とジッターを変化させた．

結果は，図 6.3 に示した通りで，遅延の変化量に関わらず，PLA-P は 1 s 程度で変化を追従できた．また，Timestamp が 180 s から 300 s の間の遅延時間に注目すると，ジッターが PLA-P の測定に与える影響は，実際のトラフィックに与える影響よりも小さかった．

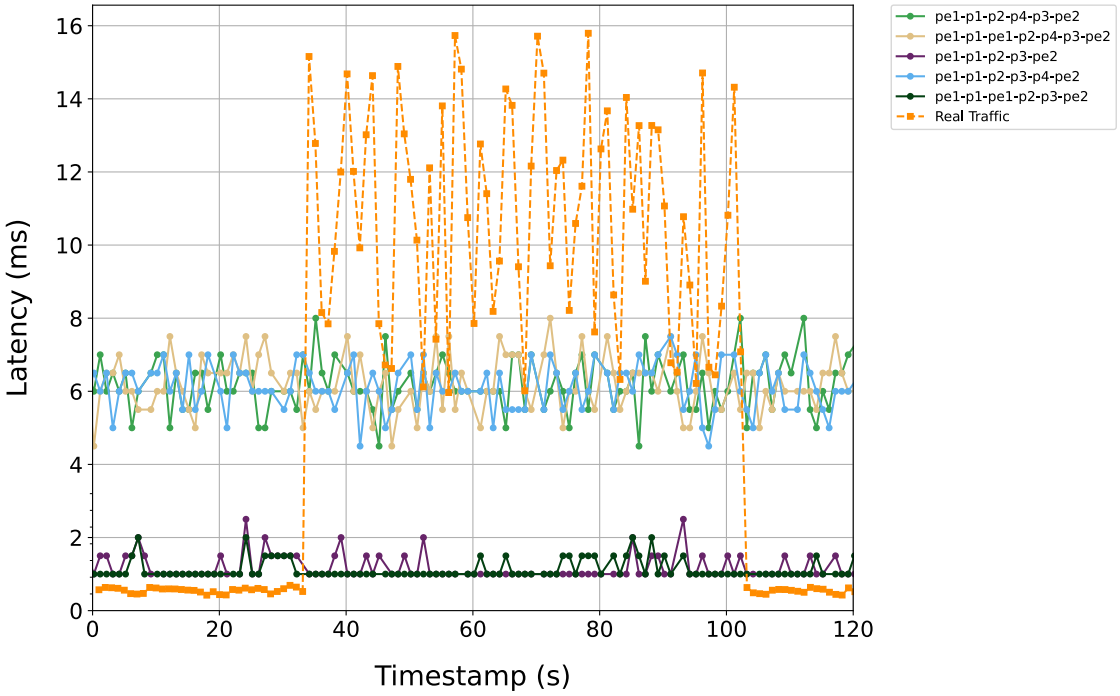


図 6.4: 検出された BPath の候補パスとその遅延時間

P3. **低遅延な BPath の検出実験**

P1, P3 間のリンクのリンク障害を想定した際に，PLA-P で検出される候補パスの遅延時間と，障害時に使用される表 6.3 に示された BPath の遅延時間を比較した．あらかじめ P2 と P4 間に設置された $TCA_{2,4}$ で 10 ms の遅延を発生させ，測定開始から 30 s 経過したら，P1 と P3 間のリンクをダウンさせ，更に 60 s 経過したらリンクをアップさせた．また，PLA-P が検出した候補パスは合計で 11 個あったが，本実験ではそのうち 5 つを測定対象とした．

結果は，図 6.4 に示す通りである．実トラフィックが使用する BPath は遅延が発生しているリンクを使用するため，障害発生により BPath に経路が切り替わると，遅延時間が 10 ms 程度増加した．一方，PLA-P が検出した 5 つの候補パスのうち，2 つはリンク P2-P4 を使用しないため，TI-LFA が提供する BPath よりも低遅延の BPath を検出することができた．

6.1.3　PLA-T の性能評価

本節では PLA-T の性能評価のためのいくつかの実験について，その手順と結果を説明する．

T1. **遅延時間が一定の場合における性能評価**

一定の遅延を発生させるために，$TCA_{1,3}$ で遅延とジッターを設定した．$TCA_{1,3}$ が発生させる遅延とジッター，および TWAMP が遅延メトリクスを広告する周期 AdsInt を調整

表 6.5: 実験 T1 で使用したパラメータ

実験番号	遅延 [ms]	ジッター [ms]	AdsInt [s]	AdsThr [%]
(i-a)	N/A	N/A	120	10
(i-b)	N/A	N/A	50	10
(ii-a)	30	3	120	10
(ii-b)	30	3	50	10
(iii-a)	30	9	120	10
(iii-b)	30	9	50	10

し，表 6.5 に示す 6 通りのパラメータの組み合わせを使い，一定の遅延が発生している状況における PLA-T の性能評価を行った．

図 6.5 はその結果を示している．PLA-T (TWAMP) という凡例は，PLA-T が測定した遅延メトリクスを示し，Real Traffic という凡例は，実際のトラフィックが要した片道遅延時間を示している．(i-a), (i-b) より，PLA-T の測定結果は，実トラフィックの所要時間よりサブミリ秒だけ大きな値を示すことがわかった．また，(ii-a), (ii-b), (iii-a), (iii-b) より，ジッターのような極めて短い周期での遅延変動に対しては，その変化量の大小に関係なく，PLA-T では追従できないことが読み取れた．

T2. **遅延時間が変動する場合における性能評価**

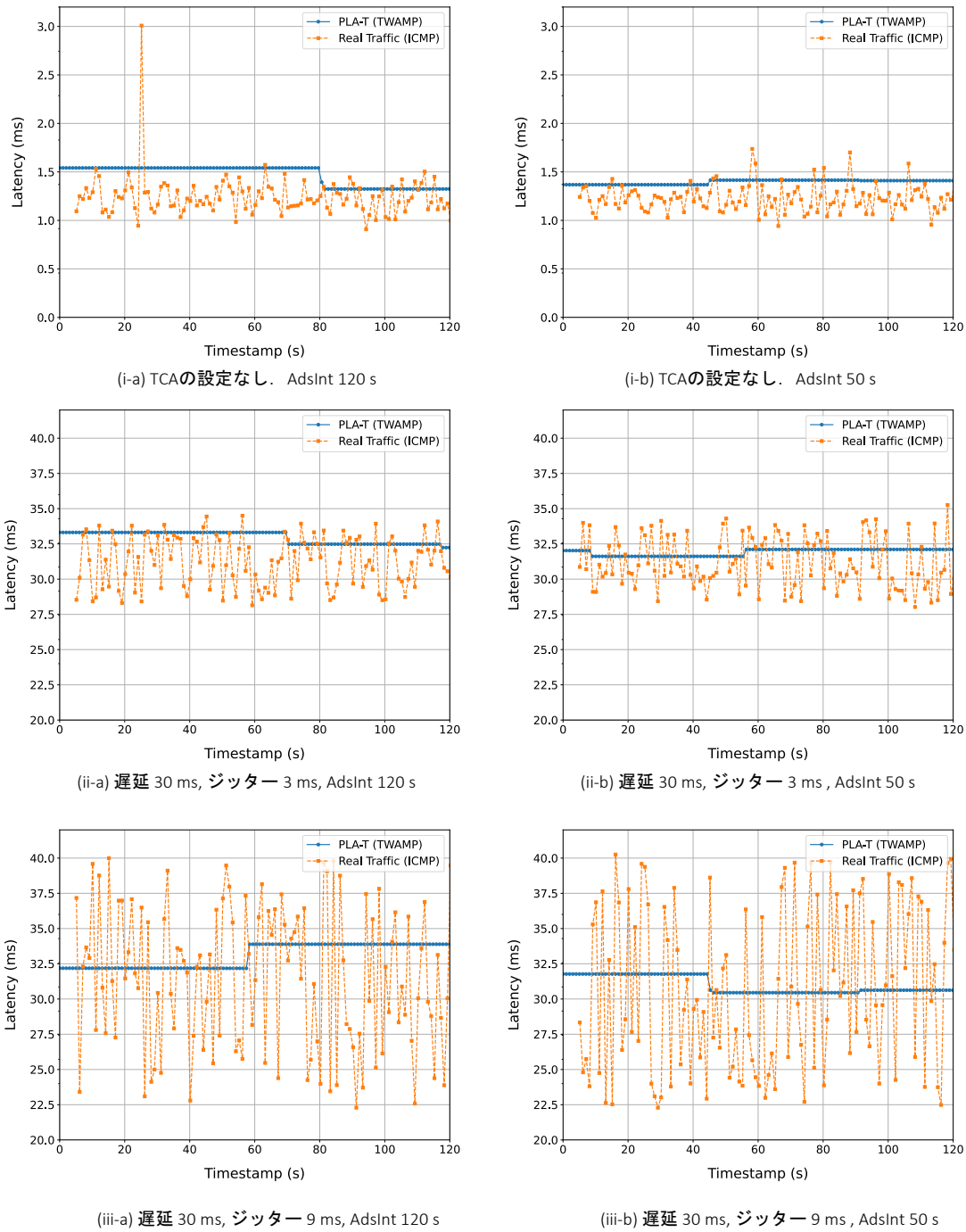
遅延の変化が長期的な間隔で行われる場合の，PLA-T の測定精度，変化への応答性を評価した．実験 P2 と同様に，表 6.4 に示されるシナリオで遅延とジッターを変化させた．また，TWAMP の設定は，(a) AdsInt が 120 s, AdsThr が 10 % の場合と，(b) AdsInt が 50 s, AdsThr が 10 % の場合の二つを試行した．

結果は，図 6.6 に示す通りである．(a), (b) を比べると，AdsInt ををを小さく設定することで，ネットワークの遅延変化の検出時間を短縮できることがわかった．また，(a) のように AdsInt の遅延メトリクスの更新周期と，実際のネットワークに生じる遅延変化の間隔がほとんど同じ場合，PLA-T は遅延の変化に徐々に追いつけなくなる様子が確認された．一方，(b) のように AdsInt の値が，遅延変化の間隔に比べて十分短い場合，PLA-T は遅延の変化を追従できており，段階的にその変化を追従していた．TWAMP では ProCon 個の測定値の統計量を遅延メトリクスとして扱うため，遅延変化前の測定値と変化後の測定値の平均を取った結果，段階的な遅延変化への追従が見られた．

T3. **BPath に対する性能評価**

表 6.3 で示した BPath に関する測定性能を評価した．測定の途中に障害が発生し，パスが BPath に切り替わった後のトラフィックの転送時間と，PLA-T が測定した BPath の遅延時間を比較した．以上の測定を 1 回として合計 10 回の測定を行った．

結果は，図 6.7 に示した通りである．Real Traffic という凡例は，トラフィックの転送時間を，PLA-T (PPath) と PLA-T (BPath) という凡例は，それぞれ PLA-T が測定した PPath



(i-a) TCAの設定なし．AdsInt 120 s

(i-b) TCAの設定なし．AdsInt 50 s

(ii-a) **遅延** 30 ms, **ジッター** 3 ms, AdsInt 120 s

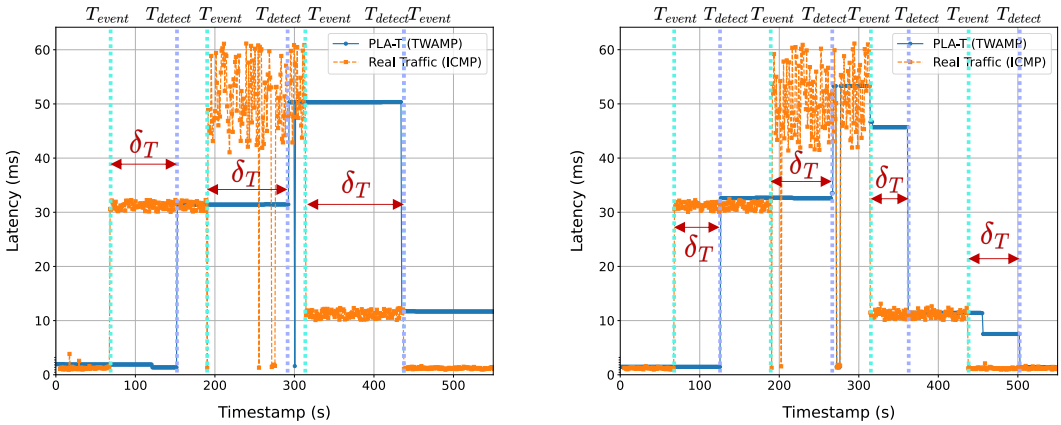
(ii-b) **遅延** 30 ms, **ジッター** 3 ms , AdsInt 50 s

(iii-a) **遅延** 30 ms, **ジッター** 9 ms, AdsInt 120 s

(iii-b) **遅延** 30 ms, **ジッター** 9 ms , AdsInt 50 s

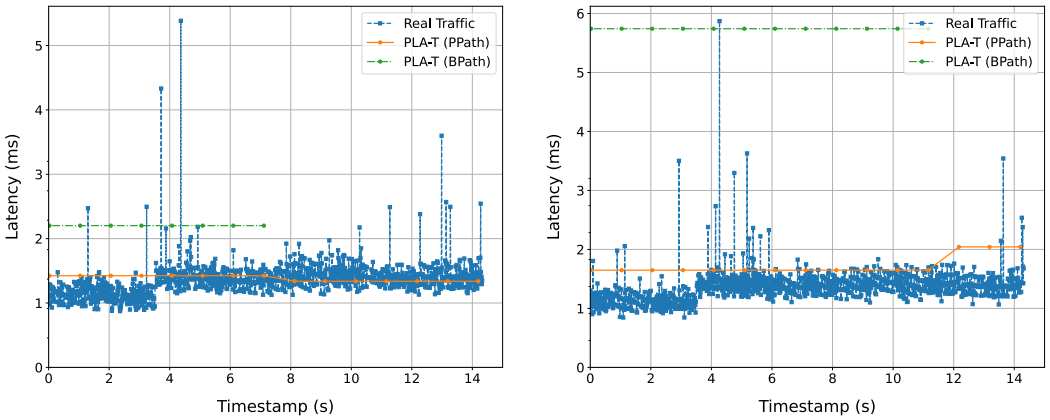
図 6.5: 実験 T1 における PLA-T の測定結果

と BPath の遅延時間を示している．まず，最も BPath の測定精度が良かった場合のグラフ (a) に注目すると，Timestamp が 3.5 s の時点で障害が発生し，パスが切り替わったことが Real Traffic のグラフから読み取れる．三つのグラフを比較すると，PPath に比べ，BPath の方が測定誤差が大きくなることがわかった．障害発生前の Timestamp が 0 s から



(a) AdsInt 120 s　　(b) AdsInt 50 s

図 6.6: 実験 T2 における PLA-T の測定結果



(a) 測定精度が最も良かった場合　　(b) 測定精度が最も悪かった場合

図 6.7: 実験 T3 における PLA-T による BPath の測定結果

latenceye cspf -m=igp -s=0000.0000.0011 -d=0000.0000.0022

```
0000.0000.0011 -> 0000.0000.0022
  Primary Path:
    0000.0000.0011 (StackedLabel: [16001 16003 16022])
    0000.0000.0001 (StackedLabel: [16003 16022])
    0000.0000.0003 (StackedLabel: [16022])
    0000.0000.0022 (StackedLabel: None, TotalCost: 30)
  Backup Path:
    0
      0000.0000.0011 (StackedLabel: [16001 16002 16004 16003 16022])
      0000.0000.0001 (StackedLabel: [16002 16004 16003 16022])
      0000.0000.0002 (StackedLabel: [16004 16003 16022])
      0000.0000.0004 (StackedLabel: [16003 16022])
      0000.0000.0003 (StackedLabel: [16022])
      0000.0000.0022 (StackedLabel: None, TotalCost: 55)
```

latenceye cspf -m=delay -s=0000.0000.0011 -d=0000.0000.0022

```
0000.0000.0011 -> 0000.0000.0022
  Primary Path:
    0000.0000.0011 (StackedLabel: [16001 16003 16022])
    0000.0000.0001 (StackedLabel: [16003 16022])
    0000.0000.0003 (StackedLabel: [16022])
    0000.0000.0022 (StackedLabel: None, TotalCost: 1427 us)
  Backup Path:
    0
      0000.0000.0011 (StackedLabel: [16001 16002 16003 16022])
      0000.0000.0001 (StackedLabel: [16002 16003 16022])
      0000.0000.0002 (StackedLabel: [16003 16022])
      0000.0000.0003 (StackedLabel: [16022])
      0000.0000.0022 (StackedLabel: None, TotalCost: 2589 us)
```

図 6.8: 実験 T4 における低遅延パスの検出結果

3.5 s 間では，Real Traffic と PLA-T (PPath) は誤差 0.2 ms 程度であるが，障害発生後の Timestamp 3.5 s から 7 s 間では，Real Traffic と PLA-T (BPath) は誤差 0.6 ms 程度まで大きくなった．一方，BPath の測定精度が最も悪かった場合 (b) のグラフでも，BPath の方が PPath の測定よりも精度が悪くなることが読み取れる．(b) では，BPath の測定結果と実際のトラフィック転送に必要な時間の誤差は 4.2 s 程度であった．

T4. **低遅延な BPath の検出機能**

図 6.8 に示すように，PLA-T は IGP のコストベースで計算された最短経路よりも，低遅延の BPath を検出することができた．通常の TI-LFA で計算される BPath は，表 6.3 に示される経路である．P2 と P4 間のリンクに遅延が発生した際には，この経路が最小遅延であるとは言えない．PLA-T は遅延メトリクスベースで BPath を計算することで，より低遅延な BPath (PE1 → P1 → P2 → P3 → PE2) を発見できた．

6.2 測定フレームワークのスケーラビリティに関する検証実験

本節では，測定フレームワークの持つスケーラビリティについて検証した．本研究で考えるスケーラビリティとは，測定対象のネットワークトポロジーの規模が大きくなっても，十分な測定性能を保てることである．

6.2.1 PLA-P のスケーラビリティ

第 5.1.3 節で述べたように，PLA-P は探索した候補パスの数だけ遅延測定を並列処理する．並列度が大きくなると，ネットワーク内のトラフィック量が増えるだけでなく，MC-P からのリクエストを受ける入口ノードの負荷も増大する．

第 6.1.2 節では，候補パスの数を 5 つに限定して遅延測定を行った．図 6.9 に示すように，候補パスの数を 10 に増やし，入口ノードの負荷を 2 倍にすると，測定に必要な時間は 50 ms 程度増加した．そのため，候補パスの数が増えるほど，測定に必要な時間が増加していくといえる．

6.2.2 PLA-T のスケーラビリティ

表 6.6: 大規模トポロジーにおけるネットワーク設定

IGP	IS-IS
障害シナリオ	P1, P6 間のリンク障害
BFD	P1, P6 間
PPath	PE1 → P1 → P6 → P10 → PE2
BPath	PE1 → P1 → P2 → P5 → P6 → P10 → PE2

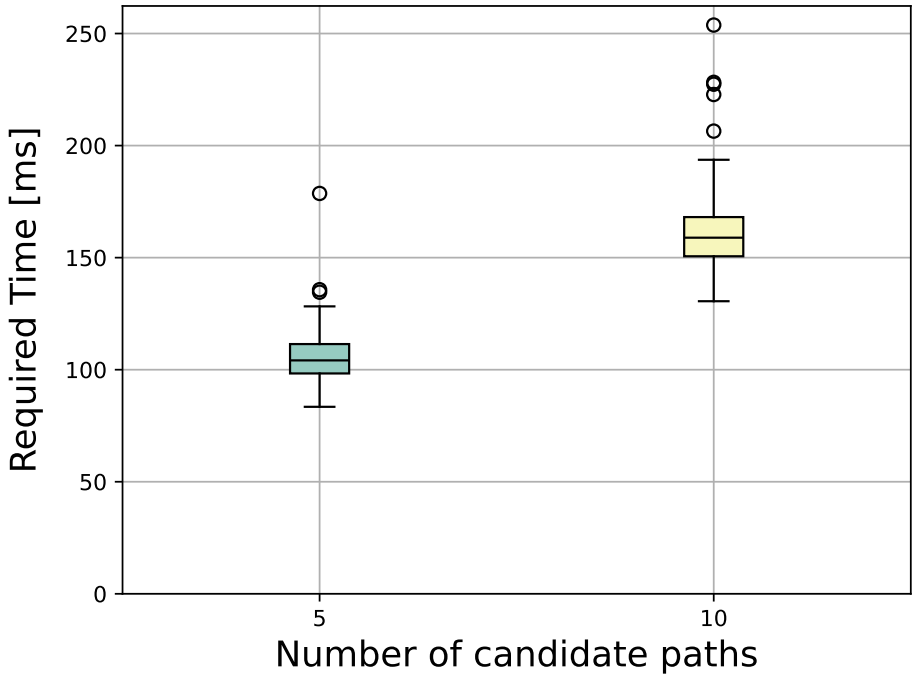


図 6.9: PLA-P における遅延測定の並列度と遅延測定に必要な時間の相関

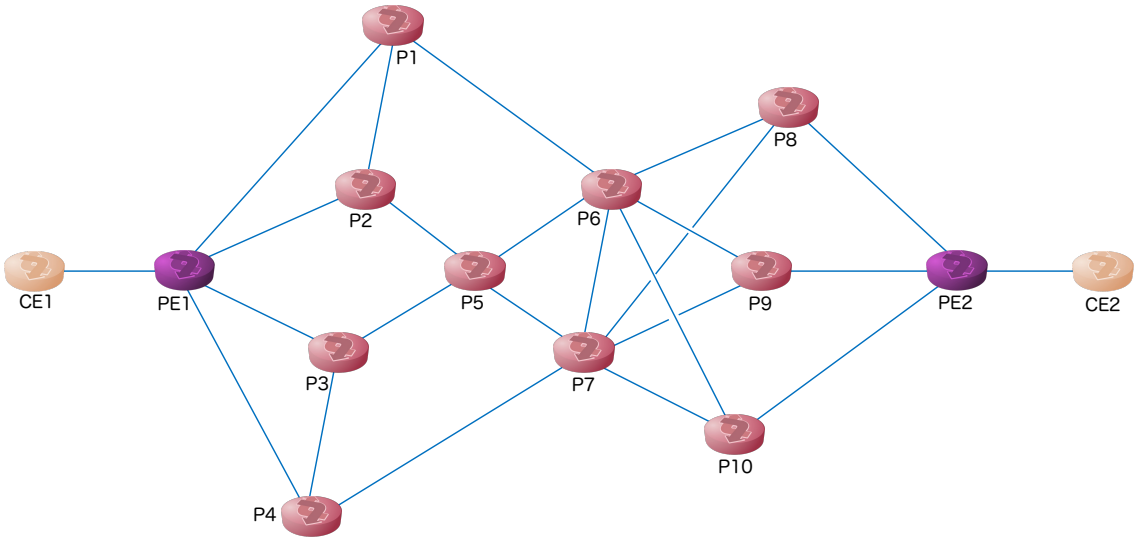
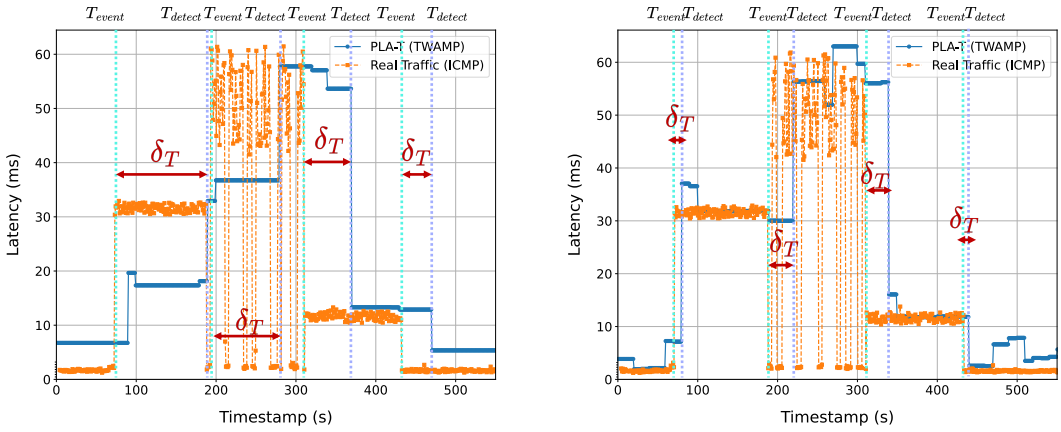


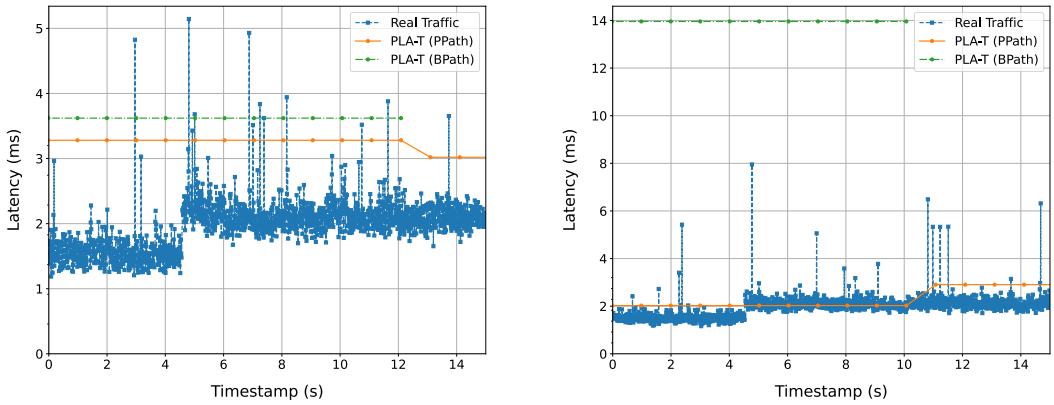
図 6.10: 実験で使用する大規模ネットワークトポロジー

PLA-T のスケーラビリティを検証するために，図 6.10 に示すトポロジーを使って実験を行った．実験環境は，第 6.1.1 節と同様に構築した．また，表 6.6 に示すようにネットワークを設定した．本節では，実験 T2 と T3 を図 6.10 に示すトポロジー上で行い，PLA-T の性能を評価した．遅延変動に対する応答性を評価するために，実験 T2 と同様の実験を行い，(a)



(a) AdsInt 50 s, ProInt 5 s, ProCon 10　(b) AdsInt 30 s, ProInt 1 s, ProCon 5

図 6.11: 実験 S1 における PLA-T の測定結果



(a) 測定精度が最も良かった場合　(b) 測定精度が最も悪かった場合

図 6.12: 実験 S2 における PLA-T による BPath の測定結果

AdsInt 50 s，ProInt 5 s，ProCon 10 と設定した場合と，(b) AdsInt 30 s，ProInt 1 s，ProCon 5 と設定した場合の二通りを試行した．さらに，測定精度を評価するために，実験 T3 と同様の実験を行い，計 10 回の測定を行った．

S1. **大規模トポロジーにおける遅延変動の応答性の評価**

図 6.11 (a) と図 6.6 (b) は，使用している TWAMP のパラメータは同じであり，トポロジーサイズが大きくなっても，遅延の変化への応答性には大きな差がないことが読み取れる．また，図 6.11 (b) に示すように，測定パケットの送信間隔や遅延メトリクスの広告の周期を小さくすることで，さらなる応答性の改善が可能であることがわかった..

S2. **大規模トポロジーにおける測定精度の評価**

図 6.12 は，計 10 回の測定実験を行い，BPath について最も測定誤差が小さかったものと

大きかったものを表している．BPath の測定誤差は，最も小さかった場合だと 1.6 ms 程度だったが，最も大きかった場合だと 12 ms も離れていた．図 6.7 と比較しても，トポロジーサイズを大きくすることで，測定精度が悪くなったことがわかる．

6.3　PLA-P と PLA-T の性能評価のまとめ

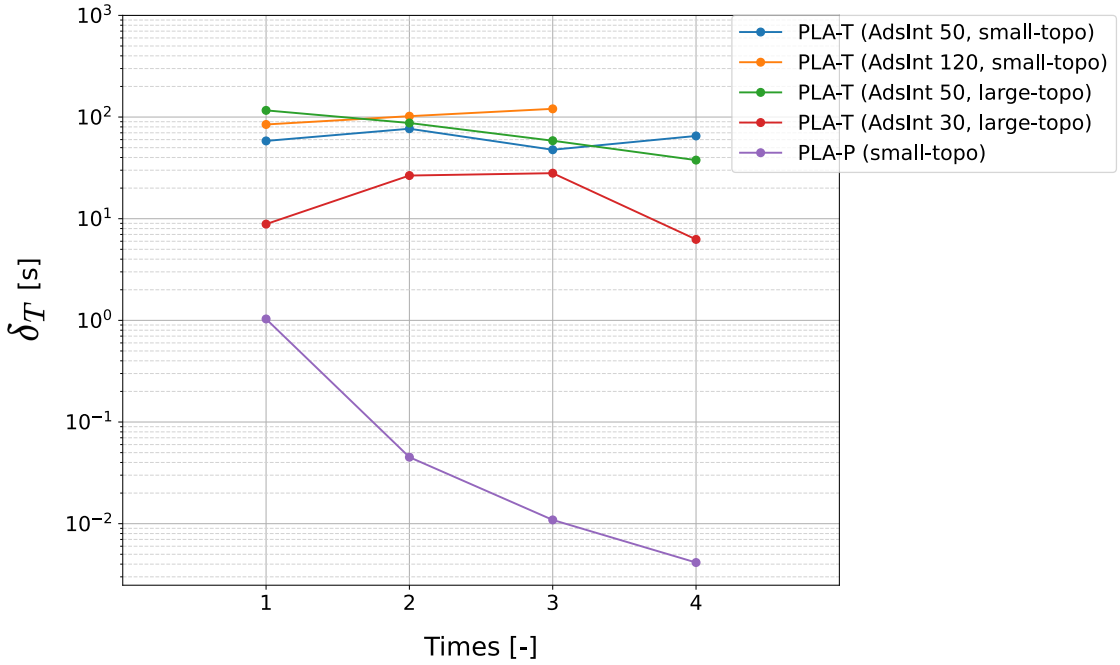


図 6.13: PLA-P と PLA-T の遅延変動応答性 δ_T の比較

図 6.13 と図 6.14 に，二つの測定フレームワークの遅延変動応答性 δ_T と測定精度 E_{acc} を示した．(PLA-P は，第 7 章で述べるように，測定パケットの帰りのパスを制御しなければ正確な測定ができないため，図 6.14 には示していない．)

図 6.13 について，横軸は遅延変動の回数を示し，縦軸は δ_T を示している．プロットされたデータは，実験 P2 の測定結果である PLA-P (small-topo)，T2 の二つの条件の測定結果である PLA-T (AdsInt 50, small-topo) と PLA-T (AdsInt 120, small-topo)， S1 の二つの条件の測定結果である PLA-T(AdsInt50, large-topo) と PLA-T (AdsInt30, large-topo) の合計 5 つである．PLA-P の δ_T の値は非常に小さく，遅延に変動が発生する度にその値を小さくしていった．一方，PLA-T は PLA-P に比べると，遅延変動応答性は大きな値をとった．PLA-T は TWAMP のパラメータを操作し，測定パケットの送信間隔と遅延メトリクスの更新周期を短くすることで，応答性が向上した．また，トポロジーサイズが δ_T に与える影響はなかった．

図 6.14 について，実際のトラフィックが流れるパスが BPath に切り替わるまでは，PLA-T が測定した PPath の遅延時間と τ_{actual} を使い，BPath に切り替わってからは，BPath の測定結果と τ_{actual} を使って E_{acc} を計算した．図 6.14 にプロットされたデータは，実験 T3 の二つの

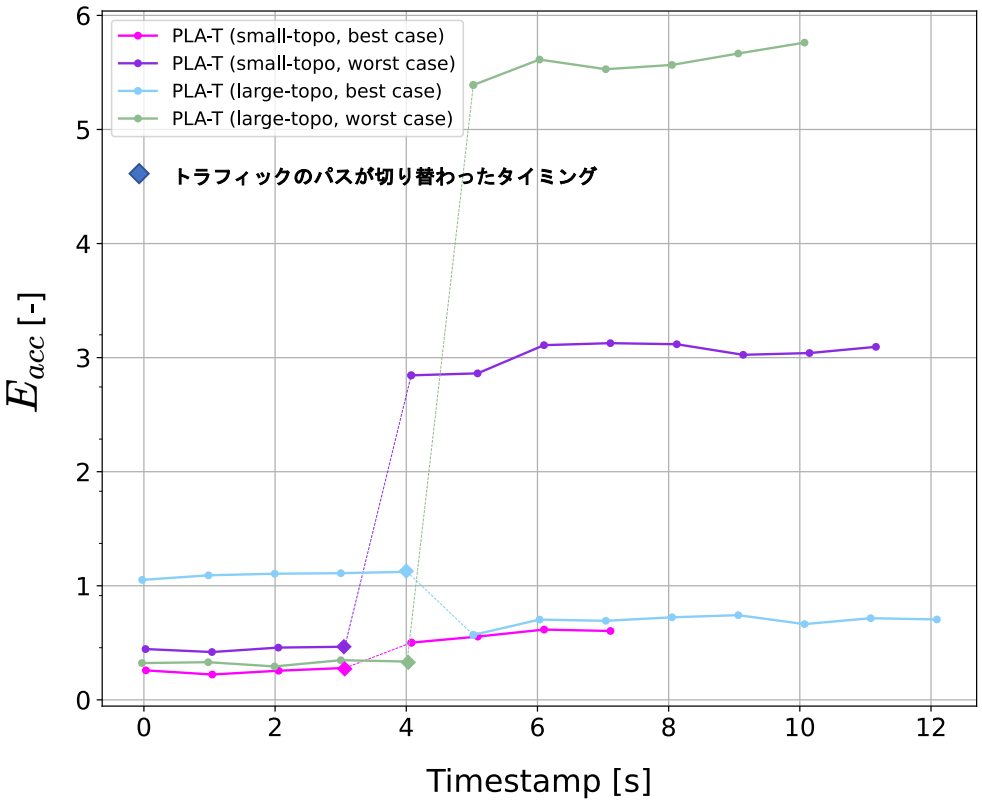


図 6.14: PLA-T の測定精度 E_{acc} の評価

条件の測定結果である PLA-T (small-topo, best case) と PLA-T (small-topo, worst case)，S2 の二つの条件の測定結果である PLA-T (large-topo, best case) と PLA-T (large-topo, worst case) の合計 4 つである．BPath の測定の Best Case において，トポロジーサイズによる E_{acc} に対する影響はほとんどなかったが，Worst Case では，トポロジーサイズが大きくなることで，E_{acc} の値がおよそ 2 倍に大きくなった．

第 7 章

考察

表 7.1: PLA-P と PLA-T の性能比較

	PLA-P	PLA-T
測定精度	×	○
変化への追従度	○	×
スケーラビリティ	×	△

PLA-P と PLA-T の性能を，表 7.1 に整理した．PING CLI ではミリ秒オーダーでの測定が，TWAMP ではマイクロ秒オーダーでの測定が可能であるため，使用している測定ツールの性能は PLA-T の方が良いと言える．さらに，PING CLI と TWAMP は共に双方向遅延時間を測定し，PING CLI を使用する PLA-P は E2E の遅延を測定するため，非対称性グラフでは正確な測定をすることができない．PLA-P では SRP を使い，指定した経路の測定を行ったが，SR は非対称性グラフを生むため，図 6.2 (b) が示すように，トポロジーに生じる遅延箇所によっては，大きく測定精度が損なわれてしまう．そのため，PLA-P を使用した測定では，帰りの経路を制御する工夫が必要であるといえる．一方，図 6.14 が示すように，PLA-T の測定は Worst Case における精度は低い．実トラフィックの転送時間が 1 ms 程度と小さかったため，TWAMP の測定による誤差の影響が強く出てしまったと考えられる．そのため，図 6.5 に示すような，トラフィックの転送時間 τ_{actual} が 10 ms 以上必要な状況下における E_{acc} の評価が必要である．

また，遅延変化に対する追従速度は PLA-P の方が上回っている．PLA-T では，更新された遅延メトリクスが MC-B に反映されるまで，IS-IS によるメトリクス広告の周期分だけ待つ必要がある．一方，PLA-P では，ルーターで PING が実行されるタイミングにおける遅延時間を測定するため，即座に追従することができた．

PLA-P は，図 6.9 に示すように，並列処理する候補パスの数が増加すると，一回あたりの PING コマンドの実行に必要な時間が増加した．候補パスの遅延時間の測定は，全て PE1 に対して PING のリクエストを送信することで行われた．並列度が増加することで，PE1 のような入口ノードに対する負荷が集中したことが原因と考えられる．第 6.1.2 節で行われた実験で

は 6 つのノードで構成された小規模のトポロジーであったため，候補パスの数は高々 10 個程度であったが，より大規模なトポロジーでは候補パスの数は更に増加し，測定に必要な時間も益々増加すると考えられる．そのため，トポロジーサイズが大きくなるほど，遅延変化の追従は遅くなるといえる．

一方，PLA-T はトポロジーサイズが大きくなっても，遅延変動応答性に変化は見られなかった．PLA-T における遅延変動応答性は，TWAMP のパラメータによる影響のみ受けると考えられる．測定パケットの転送間隔，遅延メトリクスの広告周期を短く設定すると，応答性は向上するが，TWAMP の測定パケットや IS-IS の更新パケットがネットワーク内に溢れることになるため，実トラフィックへの影響が大きくなると考える．また，トポロジーサイズが大きくなると，測定精度は悪くなることがわかった．PLA-T は，TWAMP で各リンクの遅延メトリクスを測定し，パスの遅延時間を計算するため，トポロジーサイズが大きくなり，パス長が増えると，TWAMP による測定誤差の影響が大きくなることが原因と考えられる．

最後に，二つの測定フレームワークのユースケースについて議論する．本フレームワークは，十分なリソースが確保できず，BPath を使った冗長性確保が必要な状況で，遅延時間の評価が必要なケースの使用が推定される．例えば，Wireless Ad Hoc Federated Learning (WAFL) [66] と呼ばれる，無線通信を使った協調学習の状況下では，限られたリソースの中でネットワークパフォーマンスが求められる．機械学習ではトラフィック通信量が多くなるため，十分なトラフィックが流れている状況における，測定精度の評価を追加で行う必要がある．

第 8 章

結論

8.1 まとめ

本研究では，FaRe が提供する BPath の遅延時間を測定するために，PING CLI と TWAMP を用いた二種類の測定フレームワーク PLA-P, PLA-T をそれぞれ提案・開発した．開発された測定フレームワークは，遅延時間の測定に加え，FaRe の BPath よりも低遅延の候補パスを検出する使用方法が想定される．

実験の結果，PLA-P は SRP を使用しているため，非対称ネットワークとなり，正確な測定は困難であることがわかった．しかしながら，PLA-P の遅延変動に対する応答性は PLA-T と比べて非常に強かった．また，PLA-T は，パス長が大きくなると，測定精度が落ちる傾向にあったが，ネットワークに遅延が発生している状況下では，測定誤差が微々たるものであった．さらに，PLA-T は，トポロジーサイズによらず，TWAMP のパラメータ次第で応答性を高めることができた．

8.2 今後の展望

1. **PLA-P の復路の経路制御について**
   PLA-P は，PING CLI を使用しており，非対称性ネットワーク下で正確な測定をすることができない．そのため，帰りの経路を制御する方法を検討する必要がある．PLA-P では，行きの経路を制御するために SRP を使用した．SRP は，Endpoint，Color，SL の三つで構成される．PLA-P では，SRP を Color で識別している．そのため，復路を制御する SRP と往路を制御する SRP の Color を紐づければ良い．そこで，BGP で広告した経路情報と Color 情報を紐付ける Automated Steering という技術に注目する．BGP を使い，入口ノード宛の経路と経路に紐付けられる Color を出口ノードに広告する．入口ノードで PING CLI を実行し，広告済みの Color に紐づいた SRP が指定する経路を，復路の経路として使用する．以上を実行することで，往路と復路が同じ経路を通るように制御できる見込みである．
2. **TWAMP パラメータ変更による PLA-T の応答性とネットワークに対する影響について**
   TWAMP のパラメータは，PLA-T の遅延変動応答性に影響する．応答性を向上させようと

すると，ネットワーク上を流れる測定用パケットおよび IGP の更新パケットが増加し，サービスに与える影響が大きくなる．このトレードオフの関係を分析するために，パラメータを調整して PLA-T の性能を詳細に分析する．

3. **十分な遅延が発生している状況下における PLA-T による BPath の測定精度評価**
   本研究では，PLA-T による BPath の測定精度を評価した．トラフィックの転送時間が数 ms 程度である場合，測定精度 E_{acc} が 5 を上回ることもあった．すなわち，測定誤差が 10 ms を上回ったということである．しかしながら，ネットワークに遅延が発生し，トラフィック転送時間が数十 ms 程度のときも同程度の測定誤差であれば，測定精度 E_{acc} は 1 を下回ることになる．そのため，ネットワークに遅延が発生している際の測定精度 E_{acc} の評価実験を追加で行う予定である．
4. **膨大なトラフィックが流れているネットワークにおける測定フレームワークの性能評価**
   本研究では，スケーラビリティの評価として，トポロジーサイズを変化させた実験を行った．しかしながら，スケーラビリティはトポロジーサイズに対してのみに限らず，ネットワーク上を流れるトラフィック量に対しても評価すべきである．そのため，サービス通信量が十分多く，パスの切り替えがネットワークに与える影響が大きい状況下で，測定フレームワークによる BPath の測定結果と，障害発生後，パスが切り替わった後のトラフィックの転送時間を比較し，その精度を評価する．

発表文献と研究活動

国際学会 [査読あり]

1. **Y. Ito**, J. Nakazato, R. Shiiba, K. Fukuda, H. Esaki, H. Ochiai,“Improving QoS in Failure Scenarios: Measurement System for Low-Latency Backup Path”, The 39th International Conference on Information Networking (ICOIN), Thailand, January, 2025.

口頭発表 [査読なし]

2. **伊藤 吉彦**, 中里 仁, 椎葉 瑠星, 福田 健介, 江崎 浩, 落合 秀也,“低遅延の代替パス提供のための測定フレームワークの提案”, マルチメディア，分散，協調とモバイルシンポジウム 2024 論文集 2024 1333-1339, 2024-06-19.
3. **伊藤 吉彦**, 中里 仁, 椎葉 瑠星, 福田 健介, 江崎 浩, 落合 秀也,“障害時における代替パスの測定システムの提供による QoS の改善”, 信学技報, vol. 124, no. 313, IA2024-55, pp. 16-23, 2024 年 12 月.

ポスター発表 [査読なし]

4. **Y. Ito**, H. Ochiai, J. Nakazato, R. Shiiba, K. Fukuda, H. Esaki,“Intent Based Protection using Segment Routing” WIDE Camp, 2024 年 3 月. [Best Poster Award 受賞]

その他発表

5. **Y. Ito**, H. Ochiai and H. Esaki,“Self-Organizing Hierarchical Topology in Peer-to-Peer Federated Learning: Strategies for Scalability, Robustness, and Non-lID Data”, 2023 IEEE Future Networks World Forum (FNWF), Baltimore, MD, USA, 2023.

参考文献

[1] Anja Feldmann, Oliver Gasser, Franziska Lichtblau, Enric Pujol, Ingmar Poese, Christoph Dietzel, Daniel Wagner, Matthias Wichtlhuber, Juan Tapiador, Narseo Vallina-Rodriguez, Oliver Hohlfeld, and Georgios Smaragdakis. The lockdown effect: Implications of the covid-19 pandemic on internet traffic. In *Proceedings of the ACM Internet Measurement Conference*, IMC ' 20. ACM, October 2020.

[2] Ministry of Internal Affairs and Communications. *2023 White Paper on Information and Communications*, chapter 2. Ministry of Internal Affairs and Communications, 2023.

[3] Cisco. Cisco annual internet report (2018–2023) white paper. Technical report, Cisco, 2020.

[4] Nusratullah Khan, Muhammad Usman Akram, Asadullah Shah, and Shoab Ahmad Khan. Important attributes of customer satisfaction in telecom industry: A survey based study. In *2017 4th IEEE International Conference on Engineering Technologies and Applied Sciences (ICETAS)*, pp. 1–7, 2017.

[5] Yanjiao Chen, Kaishun Wu, and Qian Zhang. From qos to qoe: A tutorial on video quality assessment. *IEEE Communications Surveys Tutorials*, Vol. 17, pp. 1126–1165, 2015.

[6] Nitin Karalkar. A comparative study on qos and qoe in modern networking. *International Journal of Advanced Research in Science, Communication and Technology*, 2022.

[7] Chang Wu, Yuang Chen, and Hancheng Lu. Statistical qos provision in business-centric networks. *ArXiv*, Vol. abs/2408.15609, , 2024.

[8] Eric S. Crawley, Raj Nair, Dr. Bala Rajagopalan, and Hal J. Sandick. A Framework for QoS-based Routing in the Internet. RFC 2386, August 1998.

[9] Zheng Wang and J. Crowcroft. Quality-of-service routing for supporting multimedia applications. *IEEE Journal on Selected Areas in Communications*, Vol. 14, No. 7, pp. 1228–1234, 1996.

[10] JP Vasseur and Jean-Louis Le Roux. Path Computation Element (PCE) Communication Protocol (PCEP). RFC 5440, March 2009.

[11] David Tipper. Resilient network design: Challenges and future directions. *Telecommunication Systems*, Vol. 56, pp. 5–16, 05 2013.

[12] Oleksandr Lemeshko, Maryna Yevdokymenko, Oleksandra Yeremenko, Ahmad M. Hailan, Pavel Segeč, and Jozef Papán. Design of the fast reroute qos protection scheme for band-

width and probability of packet loss in software-defined wan. In *2019 IEEE 15th International Conference on the Experience of Designing and Application of CAD Systems (CADSM)*, pp. 1–5, 2019.

[13] Alan Ford, Costin Raiciu, Mark J. Handley, Olivier Bonaventure, and Christoph Paasch. TCP Extensions for Multipath Operation with Multiple Addresses. RFC 8684, March 2020.

[14] Gee-Hwan Ahn, Jin-Soo Jang, and Wook-Hyun Chun. An efficient rerouting scheme for mpls-based recovery and its performance evaluation. *Telecommunication Systems*, Vol. 19, pp. 481–495, 2002.

[15] Arun Viswanathan, Eric C. Rosen, and Ross Callon. Multiprotocol Label Switching Architecture. RFC 3031, January 2001.

[16] Clarence Filsfils, Stefano Previdi, Les Ginsberg, Bruno Decraene, Stephane Litkowski, and Rob Shakir. Segment Routing Architecture. RFC 8402, July 2018.

[17] Jozef Papán, Pavel Segeč, Marek Moravčík, Jakub Hrabovský, Ľudovít Mikuš, and Jana Uramova. Existing mechanisms of ip fast reroute. In *2017 15th International Conference on Emerging eLearning Technologies and Applications (ICETA)*, pp. 1–7, 2017.

[18] Andrzej Kamisiński. Evolution of ip fast-reroute strategies. In *2018 10th International Workshop on Resilient Networks Design and Modeling (RNDM)*, pp. 1–6, 2018.

[19] Klaus-Tycho Foerster, Andrzej Kamisix0144;ski, Yvonne-Anne Pignolet, Stefan Schmid, and Gilles Tredan. Improved fast rerouting using postprocessing. *IEEE Transactions on Dependable and Secure Computing*, Vol. 19, No. 1, pp. 537–550, 2022.

[20] Kausik Subramanian, Anubhavnidhi Abhashkumar, Loris D'Antoni, and Aditya Akella. D2r: Policy-compliant fast reroute. In *Proceedings of the ACM SIGCOMM Symposium on SDN Research (SOSR)*, SOSR '21, p. 148–161, New York, NY, USA, 2021. Association for Computing Machinery.

[21] Venkat Mohan, Y Reddy, and Kalpana K K. Active and passive network measurements : A survey. Vol. 2, , 05 2012.

[22] John Klincewicz, James Schmitt, and Richard Wong. Incorporating qos into ip enterprise network design. *Telecommunication Systems*, Vol. 20, pp. 81–106, 05 2002.

[23] Juraj Frnda, Miroslav Voznak, and Lukas Sevcik. Impact of packet loss and delay variation on the quality of real-time video streaming. *Telecommunication Systems*, Vol. 62, pp. 265–275, 2016.

[24] Nurul Asyikin Mohamed Radzi, Wan Siti Halimatul Munirah Wan Ahmad, Fairuz Abdullah, M.Z. Jamaludin, and Mohd Nasim Zakaria. Recent trends in mpls networks: Technologies, applications and challenges. *IET Communications*, Vol. 14, , 01 2020.

[25] Antonio Cianfrani, Marco Listanti, and Marco Polverini. Incremental deployment of segment routing into an isp network: a traffic engineering perspective. *IEEE/ACM Transactions on Networking*, Vol. 25, No. 5, pp. 3146–3160, 2017.

[26] Clarence Filsfils, Ketan Talaulikar, Daniel Voyer, Alex Bogdanov, and Paul Mattes. Segment

Routing Policy Architecture. RFC 9256, July 2022.

[27] Rafal Stankiewicz, Piotr Cholda, and Andrzej Jajszczyk. Qox: What is it really? *IEEE Communications Magazine*, Vol. 49, No. 4, pp. 148–158, 2011.

[28] Jozef Babiarz, Roman M. Krzanowski, Kaynam Hedayat, Kiho Yum, and Al Morton. A Two-Way Active Measurement Protocol (TWAMP). RFC 5357, October 2008.

[29] Zhao Bao-hua. Research on ping technology. *Microcomputer applications*, 2007.

[30] Junos cli reference.

[31] Ahmad Vakili and Jean-Charles Grégoire. Accurate one-way delay estimation: Limitations and improvements. *IEEE Transactions on Instrumentation and Measurement*, Vol. 61, pp. 2428–2435, 2012.

[32] Matthew J. Zekauskas, Anatoly Karp, Stanislav Shalunov, Jeff W. Boote, and Benjamin R. Teitelbaum. A One-way Active Measurement Protocol (OWAMP). RFC 4656, September 2006.

[33] Djalel Chefrour. One-way delay measurement from traditional networks to sdn: A survey. *ACM Comput. Surv.*, Vol. 54, No. 7, July 2021.

[34] Spencer Giacalone, David Ward, John Drake, Alia Atlas, and Stefano Previdi. OSPF Traffic Engineering (TE) Metric Extensions. RFC 7471, March 2015.

[35] Les Ginsberg, Stefano Previdi, Spencer Giacalone, David Ward, John Drake, and Qin Wu. IS-IS Traffic Engineering (TE) Metric Extensions. RFC 8570, March 2019.

[36] Hannes Gredler, Jan Medved, Stefano Previdi, Adrian Farrel, and Saikat Ray. North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP. RFC 7752, March 2016.

[37] F. Paolucci, V. Uceda, A. Sgambelluri, F. Cugini, O. Gonzales de Dios, V. Lopez, L. M. Contreras, P. Monti, P. Iovanna, F. Ubaldi, T. Pepe, and P. Castoldi. Interoperable multi-domain delay-aware provisioning using segment routing monitoring and bgp-ls advertisement. In *ECOC 2016; 42nd European Conference on Optical Communication*, pp. 1–3, 2016.

[38] Grzegorz Rzym, Krzysztof Wajda, and Piotr Chołda. Sdn-based wan optimization: Pce implementation in multi-domain mpls networks supported by bgp-ls. *Image Processing & Communications*, Vol. 22, No. 1, p. 35, 2017.

[39] Alia Atlas and Alex D. Zinin. Basic Specification for IP Fast Reroute: Loop-Free Alternates. RFC 5286, September 2008.

[40] Stewart Bryant, Clarence Filsfils, Stefano Previdi, Mike Shand, and Ning So. Remote Loop-Free Alternate (LFA) Fast Reroute (FRR). RFC 7490, April 2015.

[41] Pushpasis Sarkar, Shraddha Hegde, Chris Bowers, Hannes Gredler, and Stephane Litkowski. Remote-LFA Node Protection and Manageability. RFC 8102, March 2017.

[42] Ahmed Bashandy, Stephane Litkowski, Clarence Filsfils, Pierre Francois, Bruno Decraene, and Daniel Voyer. Topology Independent Fast Reroute using Segment Routing. Internet-Draft draft-ietf-rtgwg-segment-routing-ti-lfa-19, Internet Engineering Task Force, Novem-

ber 2024. Work in Progress.

[43] Inc. Cisco Systems. *Topology Independent Loop-Free Alternate (TI-LFA)*. Cisco Systems, Inc., 2025. Accessed: 2025-1-4.

[44] Thomas Holterbach, Stefano Vissicchio, Alberto Dainotti, and Laurent Vanbever. Swift: Predictive fast reroute. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, SIGCOMM '17, p. 460–473, New York, NY, USA, 2017. Association for Computing Machinery.

[45] Dave Katz and David Ward. Bidirectional Forwarding Detection (BFD). RFC 5880, June 2010.

[46] Sepehr Javid, Miika Komu, and Jimmy Kjällman. Qos-aware congestion control using srv6. In *2023 IEEE Conference on Standards for Communications and Networking (CSCN)*, pp. 228–234, 2023.

[47] Eryk Schiller, Chao Feng, Rafael Hengen Ribeiro, Francesco Marino, Martin Buck, and Burkhard Stiller. Demo: Utilizing srv6 to optimize the routing behavior for tactical networks. In *2023 IEEE 24th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pp. 361–363, 2023.

[48] Ohmmar Min Mon and Myat Thida Mon. Quality of service sensitive routing for software defined network using segment routing. In *2018 18th International Symposium on Communications and Information Technologies (ISCIT)*, pp. 180–185, 2018.

[49] Zhenlin Tan, Zhuojun Huang, Peiyong Ma, Shuangfeng Lan, Yirong Zhuang, Yu Jiang, and Xiaobin Liang. A low-latency and high-reliability slice for ip backbone network based on srv6. In Qi Liu, Xiaodong Liu, Jieren Cheng, Tao Shen, and Yuan Tian, editors, *Proceedings of the 12th International Conference on Computer Engineering and Networks*, pp. 242–253, Singapore, 2022. Springer Nature Singapore.

[50] Vítor Pereira, Miguel Rocha, and Pedro Sousa. Traffic engineering with three-segments routing. *IEEE Transactions on Network and Service Management*, Vol. 17, No. 3, pp. 1896–1909, 2020.

[51] Liesbeth Roelens, Óscar González de Dios, Ignacio de Miguel, Edward Echeverry, and Ramón J. Durán Barroso. Performance evaluation of ti-lfa in traffic-engineered segment routing-based networks. In *2023 19th International Conference on the Design of Reliable Communication Networks (DRCN)*, pp. 1–8, 2023.

[52] Klaus-Tycho Foerster, Mahmoud Parham, Marco Chiesa, and Stefan Schmid. Ti-mfa: Keep calm and reroute segments fast. In *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 415–420, 2018.

[53] Tao He, Zhibo Hu, Huaimo Chen, Mehmet Toy, and Chang Cao. SRv6 Path Egress Protection. Internet-Draft draft-ietf-rtgwg-srv6-egress-protection-17, Internet Engineering Task Force, November 2024. Work in Progress.

[54] Robert Tarjan. Depth-first search and linear graph algorithms. In *12th Annual Symposium*

on Switching and Automata Theory (swat 1971), pp. 114–121, 1971.

[55] E.W. DIJKSTRA. A note on two problems in connexion with graphs. *Numerische Mathematik*, Vol. 1, pp. 269–271, 1959.

[56] Nicolas Leiva and Contributors. xrgrpc: grpc library for cisco ios xr. `https://github.com/nleiva/xrgrpc`, 2024. Accessed: 2024-12-27.

[57] NTT Communications Corporation. *Pola PCE Documentation*. NTT Communications Corporation, 2024. Accessed: 2024-12-27.

[58] Tim Bray. The JavaScript Object Notation (JSON) Data Interchange Format. RFC 8259, December 2017.

[59] Martin Björklund. The YANG 1.1 Data Modeling Language. RFC 7950, August 2016.

[60] Inc. InfluxData. *InfluxDB Documentation*. InfluxData, Inc., 2024. Accessed: 2024-12-27.

[61] Organisation Studies Research Group. *GoBGP Documentation*. Organisation Studies Research Group, 2024. Accessed: 2024-12-27.

[62] Les Ginsberg, Stefano Previdi, Qin Wu, Jeff Tantsura, and Clarence Filsfils. BGP - Link State (BGP-LS) Advertisement of IGP Traffic Engineering Performance Metric Extensions. RFC 8571, March 2019.

[63] Stefano Previdi, Les Ginsberg, Clarence Filsfils, Ahmed Bashandy, Hannes Gredler, and Bruno Decraene. IS-IS Extensions for Segment Routing. RFC 8667, December 2019.

[64] Stefano Previdi, Ketan Talaulikar, Clarence Filsfils, Hannes Gredler, and Mach Chen. Border Gateway Protocol - Link State (BGP-LS) Extensions for Segment Routing. RFC 9085, August 2021.

[65] Inc. Google. *Protocol Buffers Documentation*. Google, Inc., 2024. Accessed: 2024-12-27.

[66] Hideya Ochiai, Yuwei Sun, Qingzhe Jin, Nattanon Wongwiwatchai, and Hiroshi Esaki. Wireless ad hoc federated learning: A fully distributed cooperative machine learning, 2022.

[67] Clarence Filsfils, Darren Dukes, Stefano Previdi, John Leddy, Satoru Matsushima, and Daniel Voyer. IPv6 Segment Routing Header (SRH). RFC 8754, March 2020.

[68] Huanan Chen, Zhibo Hu, Huaimo Chen, Xuesong Geng, Yisong Liu, and Gyan Mishra. SRv6 Midpoint Protection. Internet-Draft draft-chen-rtgwg-srv6-midpoint-protection-14, Internet Engineering Task Force, February 2024. Work in Progress.

[69] Ketan Talaulikar. Distribution of Link-State and Traffic Engineering Information Using BGP. RFC 9552, December 2023.

謝辞

本研究を遂行するにあたり，多くの方々からご指導とご支援を賜りました．ここに深く感謝の意を表します．

まず，指導教員である落合秀也准教授には，研究室での 3 年間にわたり的確な指導と温かい励ましをいただきました．心より感謝申し上げます．また，江崎浩教授および塚田学准教授には，研究室全体の進捗報告会において鋭いご指摘を頂戴し，研究の深掘りに繋がる貴重な示唆を賜りました．中里仁特任助教には，日々のミーティングを通して研究の進め方や技術的な助言を多く頂きました．これらのご指導は本研究の進展に大きく寄与しました．

さらに，総合研究大学院大学の福田健介教授と椎葉榴生氏には，鋭いご指摘を交えたディスカッションを通して，本研究の質を高めていただきました．また，研究に専念できるよう定期的に時間を割いていただきましたことに心より感謝申し上げます．

外部では，株式会社ブロードバンドタワーの豊田安信氏，慶應義塾大学の澤田開杜氏をはじめとする WIDE プロジェクト vSIX ワーキンググループの皆様に，多くの技術的アドバイスをいただきました．これにより，実装スキルを向上させることができました．さらに，エヌ・ティ・ティ・コミュニケーションズ株式会社の三島航氏には，研究テーマの設定から技術的な相談まで幅広い支援をいただき，感謝の念に堪えません．

また，JANOG，ShowNet，ICTSC にてお世話になった方々とのネットワークに関する活発な議論を通じて，研究へのモチベーションを高めることができました．この場を借りて感謝の意を表します．

日常の研究活動を支えてくださった高橋富美秘書，岩井愛映子秘書にも深く感謝申し上げます．研究室の先輩である金谷光一郎氏，伊藤広記氏，山本桃歌氏には，ネットワーク技術に関して多くのご指導をいただきました．同期や後輩の皆様にも，互いに切磋琢磨しながら成長を共にできたことに感謝しております．

最後に，これまで支えてくださった家族，友人，恩師の方々，そして様々な機会を与えてくださったすべての皆様に，深い感謝を申し上げます．

付録 A

予備実験: FaRe の性能検証

A.1　実験環境・シナリオ

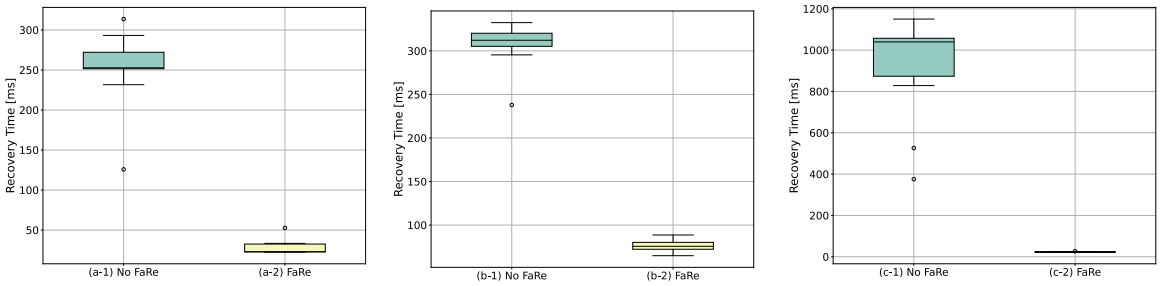
本章では，ネットワークに FaRe を導入した際における，耐障害性向上がどの程度であるかを検証した実験とその結果について説明する．FaRe の技術には TI-LFA を採用し，SR 環境における FaRe の検証を行った．

環境は第 6.1.1 節と同様に構築され，使用するネットワークトポロジー，想定する障害シナリオやパスは，全て同じものである．本実験では時刻を同期させた CE1 と CE2 を用意し，CE1 から CE2 に対して ICMP パケットを 2 ms 間隔で送信し，CE2 でパケットキャプチャした．ICMP Echo Request のシーケンス番号に注目し，CE2 に届いていないパケットを確認することで，障害復旧に要する時間を計測した．

表 A.1: FaRe の性能評価実験の条件一覧

実験番号	TI-LFA の有無	遅延の有無	再計算を始めるまでの時間
(a-1)	×	×	正常
(a-2)	○	×	正常
(b-1)	×	○	正常
(b-2)	○	○	正常
(c-1)	×	×	遅い
(c-2)	○	×	遅い

実験では表 A.1 に示す 6 つの場合について，ネットワークのダウンタイムを計測した．P1 ルーターに TI-LFA を設定した場合と設定しない場合，トポロジーの各リンクに 10 ms 程度の遅延を設定した場合と設定しない場合，各ルーターのパスの再計算が始まるまでの時間を 0.2 s から 1 s に増やした場合について試行した．



(a) 遅延なし，再計算を始める時間が正常　(b) 遅延あり，再計算を始める時間が正常　(c) 遅延なし，再計算を始める時間が遅い

図 A.1: TI-LFA の性能評価実験の結果

A.2　結果と評価

図 A.1 は，表 A.1 に示した 6 つの条件について，障害復旧に要する時間を測定した結果である．(a) - (c) のいずれの条件でも，TI-LFA を設定することで障害復旧に必要な時間が大幅に短縮されることが分かった．(a) のようにネットワークに遅延が発生せず，パスの再計算が行われる場合では，FaRe を使用しなければ数百 ms を要するところを，50 ms 以内で復旧できることが分かった．また，(b) のようにネットワーク全体に遅延が生じている場合でも 100 ms 以内で復旧できた．さらに，(c) のようにルーターでパス再計算に時間が要する状況を再現したところ，FaRe による障害復旧時間の短縮効果が最も大きかった．

以上より，障害発生後にパスの収束に時間を要するような状況であっても，FaRe を導入することで数十 ms 程度で障害復旧が可能である．FaRe はネットワークの状況にかかわらず，高速な障害復旧を実現できる．

付録 B

LFA 系列の FaRe の計算手法

本章では，LFA，rLFA，TI-LFA における BPath の計算アルゴリズムについて説明する．

B.1　LFA

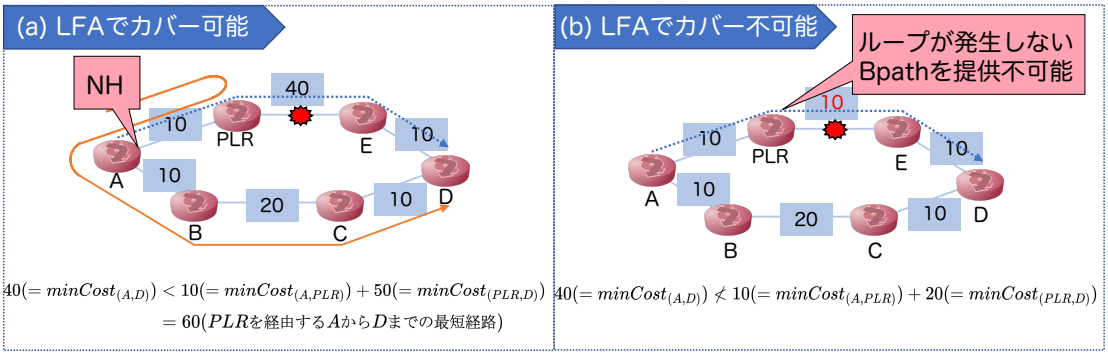


図 B.1: LFA における障害シナリオのカバー範囲

LFA [39] は，FaRe の計算手法の一つで，経路を BPath に切り替えることで生じる，ループの発生を防ぐことを目的としたメカニズムである．LFA では，式 (B.1) を満たすノード NH を，BPath における PLR からのネクストホップノードとする．NH から宛先ノード D までのコストの総和の最小値が，NH から PLR を経由して D まで到達するコストの総和の最小値より小さいならば，NH と PLR の間でループが発生することはない．

$$minCost_{(NH,D)} < minCost_{(NH,PLR)} + minCost_{(PLR,D)} \tag{B.1}$$

しかしながら，LFA では，ループフリーな BPath を提供できるかは，トポロジー次第である．図 B.1 (b) のようなトポロジーの場合，PLR は障害箇所の迂回のために NH に該当するノード A にパケットを転送しようとするが，A は障害を検出できず，PLR にパケットを転送

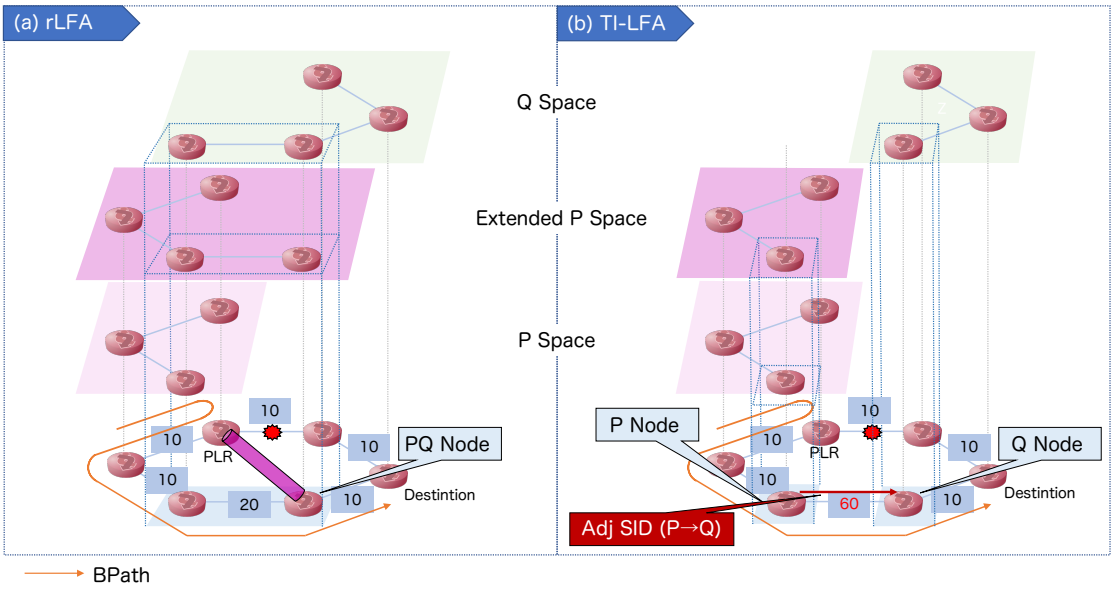


図 B.2: rLFA および TI-LFA を使ったループフリーな BPath の提供手法

しようとする．そこで，ループが発生しないノードまでパケットを転送することで，カバレッジを広げたものが，次節で説明する rLFA である．

B.2　rLFA

LFA では，障害箇所の隣接ノードしか利用しなかったが，rLFA [40, 41] はループが発生しない遠隔ノードを利用することで，障害シナリオのカバー範囲を拡張した．LFA でループフリーな BPath が見つからない場合に，リモートノードまでトンネリングすることで，障害に対してより高い適用性を提供した．rLFA は，次の二つの空間を使って設計されている．

1. **P-Space, Extended P-Space**
   P-Space とは，障害検出地点から到達可能な範囲を定義したものである．すなわち，PLR を起点として，障害が発生したリンクあるいはノードを迂回して安全に到達可能なノードの集合である．また，Extended P-Space は，より広範囲のノードを対象とするために，P-Space の概念を拡張したものである．P-Space では PLR を起点としたが，Extended P-Space は，PLR の隣接ノードを起点とした P-Space の和集合である．
2. **Q-Space**
   Q-Space とは，宛先ノードに到達可能なノードの集合であり，宛先まで安全にパケットを転送できる範囲を定義したものである．

以上を整理すると，PLR から P-Space あるいは Extended P-Space (以降，これら二つをまとめて P-Space と称する) に属するノードまで，最短経路パスに従ってパケットを転送することができ，Q-Space に属するノードから宛先ノードまで，最短経路パスでパケット転送が可能で

ある．そのため，P-Space と Q-Space の両方に属するノードを経由することで，ループフリーな BPath が利用可能になる．P-Space と Q-Space が重なる範囲に属するノードは PQ Node と称され，PQ Node をトンネルのエンドポイントとすることで，ループフリーな BPath を提供する手法が rLFA である．

しかしながら，rLFA においても，依然としてカバーできない障害シナリオが存在する．P-Space と Q-Space に共通範囲がない場合，トンネルのエンドポイントが存在しないため，BPath を計算することができない．

B.3 TI-LFA

TI-LFA [42] は，rLFA をさらに拡張し，中継ノードあるいはリンクにおける単一の障害シナリオに対して，完全な保護を保証する．SR の技術を利用することで，rLFA が対応できない，PQ Node が存在しないシナリオであっても BPath の計算が可能である．P-Space に属するノードのうち，最も Q-Space に近いノードを P Node，Q-Space に属するノードのうち，最も P-Space に近いノードを Q Node と称し，P Node から Q Node まで SR を使って転送する．

SR は，データプレーンに MPLS を指定した SR-MPLS と，IPv6 を指定した SRv6 の 2 種類に分類される [67]．SR-MPLS では，SR 領域に属する全てのノードで SID の処理をするが，SRv6 では，SID を認識せずに IPv6 パケットとして処理するノードが存在する．SID を処理するノードはエンドポイントノード，IPv6 パケットとして処理するノードはトランジットノードと称される．SRv6 Middle Protection [68] では，SRv6 環境において，SID で指定されたエンドポイントノードへの疎通性が失われた際の処理について説明している．IGP 収束前後の 2 段階に対応しており，エンドポイントノード間でトラフィックを効率良く迂回させる．IGP 収束前には，PLR で障害が起きたエンドポイントを迂回する．この時，障害箇所に対応する SID は SL から除外される．IGP 収束後には，障害箇所が Forwarding Information Base (FIB) のエントリから削除されているため，SID が指すエントリが存在しない場合，その SID をスキップし，次のエンドポイントに宛先を書き換えてパケットを転送する．

TI-LFA では複数箇所の障害シナリオに対応できない他，単一障害であっても，SR 領域の出口ノードの保護はできない．現在，TI-LFA の他にも，SR 環境における，あらゆる障害時の保護手法について，活発に議論がなされている [53].

付録 C

BGP-LS Path Attribute

BGP-LS は，OSPF や IS-IS などのリンクステート情報を BGP を通じて配布し，SDN コントローラによるトポロジー学習や SR-TE への活用に用いられる技術である．経路情報の柔軟な管理と伝達を可能にするために，Path Attribute と TLV 形式の構造が使われる．Path Attribute は，特定の経路に関する情報を伝達するために使用される属性データで，本研究では，MP_REACH_NLRI と BGP-LS Attribute [69] の 2 つを利用した．TLV は，情報の種類を識別するための Type，データ長を表す Length，データの値の Value の 3 つの部分で構成されるデータ形式である．

本研究において，BGP-LS を使って取得した情報を表 C.1, C.2 に整理した．表に示された情報は PLA-T の MC-B に伝達されたのち，PaS の TED に保管される．

表 C.1: Node Information

Data	Path Attribute	TLV / Sub-TLV
AS Number	MP_REACH_NLRI	Local Node Descriptors / Autonomous System
Router ID	MP_REACH_NLRI	Local Node Descriptors / IGP Router ID
Host Name	BGP-LS Attribute	Node Name
IS-IS Area ID	BGP-LS Attribute	IS-IS Area Identifier
SRGB Start Label	BGP-LS Attribute	SR Capabilities
SRGB Range	BGP-LS Attribute	SR Capabilities
NodeSID Index	BGP-LS Attribute	Prefix SID
IPv4 Loopback Address	MP_REACH_NLRI	IPv4 Router-ID of Local Node

表 C.2: Link Information

Data	Path Attribute	TLV / Sub-TLV
Local Node	MP_REACH_NLRI	Local Node Descriptors / IGP Router ID
Remote Node	MP_REACH_NLRI	Remote Node Descriptors / IGP Router ID
Local Node IPv4 Address	MP_REACH_NLRI	Link Descriptors / IPv4 interface address
Remote Node IPv4 Address	MP_REACH_NLRI	Link Descriptors / IPv4 neighbor address
IGP Cost	BGP-LS Attribute	Metric
AdjSID	BGP-LS Attribute	Adjacency SID
Protection Flag	BGP-LS Attribute	Adjacency SID (B-Flag)
Latency Metrics	BGP-LS Attribute	Unidirectional link delay, Min/Max Unidirectional link delay, Unidirectional Delay Variation