ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

TELECOM ParisTech

2017-ENST-0060

Doctorat ParisTech

T H È S E

pour obtenir le grade de docteur délivré par

TELECOM ParisTech

Spécialité - Informatique et Réseaux

présentée et soutenue publiquement par

Wenqin Shao

le 29/11/2017

Ingénierie du Trafic Inter-domaine Basée sur la Mesure

Directeur de thèse : **Jean-Louis Rougier**
Co-encadrement de thèse : **Luigi Iannone**

Jury

M. Philippe Owezarski, Directeur de recherche, LAAS - CNRS	Rapporteur
M. Chadi Barakat, Chargé de Recherche HDR, INRIA, Sophia Antipolis - Méditerranée	Rapporteur
Mme. Cristel Pelsser, Professeur, Université de Strasbourg - CNRS	Examinateur
Mme. Sandrine Vaton, Professeur, Telecom Bretagne	Examinateur
M. François Devienne, Co-foundateur, BORDER 6	Invité
M. Luigi Iannone, Maître de conférences, Telecom Paristech	Directeur de thèse
M. Jean-Louis Rougier, Professeur, Telecom Paristech	Directeur de thèse

TELECOM ParisTech
école de l'Institut Télécom - membre de ParisTech

T H È S E

MEASUREMENT-BASED INTER-DOMAIN TRAFFIC ENGINEERING

scalability, data interpretation and network event visibility

WENQIN SHAO

Department of Network and Computer Science
Télécom ParisTech

Thesis Committee:
M. Philippe Owezarski, *LAAS-CNRS*
M. Chadi Barakat, *INRIA*
Mme. Cristel Pelsser, *Université de Strasbourg, CNRS*
Mme. Sandrine Vaton, *Telecom Bretagne, IRISA*
M. François Devienne, *BORDER 6*
M. Luigi Iannone, *Telecom ParisTech*
M. Jean-Louis Rougier, *Telecom ParisTech*

Submitted in partial fulfillment of the requirement for the Degree of Doctor of Philosophy

keywords: Internet routing, traffic engineering, performance measurements, time series, changepoint analysis, congestion, interactive data visualization

CONDENSÉ DU MANUSCRIT

INTRODUCTION

Internet est une collection de réseaux gérés individuellement. Son système de gestion et de routage distribué lui permet de se développer rapidement. Cependant, la distribution du trafic sous-optimale à partir d'une vue globale est également d'originaire de ce comportement distributé. Ce problème se manifeste souvent par la congestion lorsqu'il existe encore de la capacité disponible. De nombreux efforts ont donc été consacrés à une meilleure performance de la transmission, en allégeant ou en évitant la congestion.

La congestion se produit sur un chemin Internet partagé par plusieurs flux lorsque la demande totale dépasse la capacité de liaison. En évitant la concurrence vicieuse qui finit par bloquer tous les flux, le mécanisme de contrôle de la congestion de bout en bout joue un rôle important. Il améliore les performances de transmission de manière distribuée. Il vise à 1) utiliser pleinement la bande passante tout en 2) introduisant un minimum de delai de transmission supplémentaire (longueur de file d'attente courte) et en 3) assurant un partage équitable des ressources [3, 6, 109].

Les performances de transmission peuvent être encore améliorées avec une capacité de liaison suffisamment importante pour satisfaire la demande de tous les flux. À cette fin, il est nécessaire de dimensionner régulièrement le réseau en fonction de la croissance des demandes de trafic [26]. Pourtant, la construction d'infrastructures tout seul ne suffit pas. Premièrement, le déploiement du réseau se déroule sur une période beaucoup plus longue que les fluctuations du trafic. Avant que la capacité supplémentaire ne soit déployée, certains liens peuvent devenir saturés alors que d'autres restent presque non-consommé en raison de la variation sporadique la demande de trafic. Deuxièmement, le surdimensionnement est coûteux, étant donné que les technologies futures réduiront considérablement le coût par unité de bande passante.

Par conséquent, des schémas de routage réactifs et flexibles sont nécessaires, complémentaires au dimensionnement du réseau. Ils maximisent la capacité de réseaux qui peut être réellement utilisée. Au sein d'un réseau (intra-domaine), l'administrateur peut en premier lieu estimer/modéliser la matrice de trafic qui traverse son réseau. Ensuite, chaque flux peut être divisé et cheminé sur plusieurs routes pour s'adapter au mieux à la capacité disponible [75, 83]. Parmi différents réseaux (inter-domaines), la marge d'amélioration des performances provient principalement de plusieurs chemins Internet entre

les réseaux source et ceux de destination. C'est parce que la capacité de bout en bout est potentiellement élargie avec des chemins Internet plus riches. Une telle diversité de chemins peut déjà être obtenue via multihoming sous Border Gateway Protocol (BGP). Avec le multihoming, un réseau achète l'accès au reste de l'Internet via plusieurs fournisseurs. De nombreuses propositions encouragent également la propagation de plusieurs chemins Internet, pour ne citer que quelques-un : BGP add-path [123], MIRO [40], NIRA [47], YAMR [63], Pathlet [56], IDRD [84], etc. Cependant, dans le routage inter-domaine, chaque réseau n'a toujours pas la visibilité en dehors de son propre territoire, par exemple, la capacité d'une liaison distante et la demande de trafic concurrente provenant d'autres réseaux sur ce lien. Ceci est particulièrement vrai avec BGP, le protocole de routage inter-domaine de facto qui ne va pas être obsolète prochainement. Par conséquent, il n'est pas possible pour les réseaux actuels de déterminer quels chemins disponibles offrent les meilleures performances vers une destination donnée. En d'autres termes, le défi est de savoir comment mieux utiliser la capacité Internet disponible avec un protocole agnostique de performance BGP. Nous entreprenons ce défi dans cette dissertation.

Une approche directe de rendre la décision de la route ayant conscience de la performance est la mesure. Par exemple, en connectant avec les autres réseaux, Facebook et Google utilisent des instruments intégrés dans leurs applications pour apprendre les aspects performances de bout en bout sur plusieurs chemins disponibles [118, 119]. Au fur et à mesure que le prix du transit diminue, le multi-homing devient plutôt une pratique courante dans de nombreux réseaux de petite et moyenne taille. Ces réseaux ont également un besoin immédiat d'amélioration des performances, de sorte que leurs buinesses puissent survivre.

Malgré le besoin concret d'ingénierie du trafic inter-domaine basée sur des mesures, de nombreuses questions restent sans réponse ou partiellement traitées. Un réseau typique peut communiquer régulièrement avec des destinations différentes ~ 100k. Mesurer continuellement la performance de toutes ces destinations est coûteux et n'est pas nécessaire. Quelles sont les destinations les plus importantes ? Est-ce que ces destinations changent au fil du temps ? Comment les identifier ? En outre des mesures de volume de traffic, comment les mesures de performance doivent-elles être traitées ? Ont-ils besoin de nettoyage ? Si oui, quels sont les problèmes potentiels de qualité des données ? Quelles sont les origines de ces problèmes ? Comment pouvons-nous mettre en valeur et atténuer leurs impacts sur la sélection des itinéraires ? En outre, afin de réagir dynamiquement aux changements de performance, comment détecter en premier lieu des changements significatifs dans les mesures de performance ? Comment le faire sans paramètres codés en dur ou d'une manière ad

hoc ? Comment atteindre un niveau de robustesse acceptable ? Enfin, une fois qu'un changement de performance est détecté, comment pouvons-nous en savoir plus sur les aspects réseau ? Où cela arrive-t-il ? Cela peut-il avoir un impact sur d'autres chemins Internet actuellement utilisés ? Dans cette thèse, nous cherchons à explorer les réponses aux questions posées ci-dessus, afin que la fabrication d'un système TE basé sur des mesures avance dans ces aspects : scalabilité du system de mesure, interprétation des données de performance et visibilité des causes de changement de performance.

LE CONTEXT

Nous mettons en scène les travaux de cette thèse sous BGP, tout en réalisant pleinement Locator/Identifier Separation Protocol (LISP), Software Defined Networking (SDN), etc. sont des pistes de recherche prometteuses. C'est parce que BGP sera encore le protocole de routage *de facto* d'Internet dans un avenir prévisible. Et le déploiement d'un nouveau mécanisme de routage doit être incrémentiel. Avant la prédominance de tout ce qui n'est pas BGP, BGP est ce qu'une majorité d'Autonomous System (AS) doit vivre avec. Il y a donc des besoins immédiats d'amélioration.

Nous nous concentrons sur la TE pour trafic sortant dans cette thèse. Car la TE entrante est intrinsèquement difficile sous BGP, en raison d'un manque de méthode efficace de guidage le trafic entrant.

Nous ciblons les AS stub (potentiellement en multi-homing). C'est parce que Content Provider (CP), Hosting Provider (HP) et Internet Service Provider (ISP), étant les principaux types de réseaux parmi les AS stub, sont ceux qui ont le plus besoin de TE. De plus, la re-sélection de chemin dynamique dans ces réseaux ne générera pas de problèmes de convergence de routes BGP sur l'ensemble de l'Internet.

Enfin, nous supposons que l'amélioration des performances de transmission est aujourd'hui la principale motivation pour la TE sortante. L'acheminement du trafic sur Internet fait désormais face à moins de contraintes monétaires grâce à la baisse du prix de transit [133, 148] et de la présence des Internet Exchange Point (IXP) de plus en plus dense dans le monde entier [134]. En revanche, en vu de la demande pour la diversité géographique et topologique de connexion [102], un défi de performance reste à relever.

Afin de réaliser réellement ce gain de performance provenant de multiples chemins d'Internet, une sélection de route dynamique basée sur des mesures de performance est requise, c'est-à-dire *TE basé sur la mesure*. Akella et al. [49] a montré une preuve de concept d'un tel system. Seulement 100 destinations sont émulées dans ce travail. Ce nombre est beaucoup moins par rapport à l'échelle réelle qu'un AS stub peut faire face sur une base quotidienne. Dans ce travail, le meilleur chemin pour chaque destination est choisi en fonction de

Exponentially Weighted Moving Average (EWMA) sur des mésures de Round-Trip Time (RTT) dans le passé. Les résultats montrent que la meilleure performance de transmission sur toutes les destinations est atteinte lorsque la décision d'itinéraire est prise selon uniquement la dernière mesure de RTT. Cependant, compte tenu de la nature bruissante des mesures RTT, une telle approche simpliste peut conduire à des changements de chemin extrêmement fréquents. En outre, traiter Internet comme une boîte noire pour les mesures de latence ne fournit pas une visibilité des événements de réseau sous-jacents, qui est utile et parfois nécessaire . Ces événements de réseau, par ex. les changements de chemin et la congestion sont les causes réelles d'une dégradation significative des performances et ainsi les raisons du changement de route.

Afin de répondre aux préoccupations ci-dessus prononcées et de réduire l'écart entre le concept et un système qui fonctionne [124], nous étudions dans cette thèse multiples types de mesure dans des vrai réseaux et l'Internet lui même pour améliorer la scalabilité du system qui collecte des mesures, l'interprétation des mesures et la visibilité sur des événements réseaux ayant impact sur la performance.

Une plate-forme TE (pour traffic inter-domaine) basée sur la mesure a deux éléments essentiels. Ils sont illustrés en noir sur la figure 3 : (i) le system qui mesure la performance des chemins utilisables et (ii) l'intelligence qui selectionne des chemins. La performances de bout en bout est ainsi mesurée, plus précisément la latence aller-retour Round-Trip Time (RTT), sur toutes les chemins disponibles vers en semble de préfixe de destination donné. Dans chaque préfixe de destination, quelques hôtes avec des ports ouverts, par ex. 80, 443, sont découverts puis utilisés comme destination de mesure. Une fois alimenté par les mesures de performance, le moteur de décision choisit pour chaque destination les meilleurs chemins à chaque instant et les implémente sur les routeurs BGP.

LA SÉLECTION DE PRÉFIXES LES PLUS IMPORTANTS

Un réseau client ayant besoin de Traffic Engineering (TE) inter-domaine est souvent de type Internet Service Provider (ISP), Hosting Provider (HP), Content Provider (CP). Il envoie du trafic vers un large éventail de destinations, en général entre 10k à 100k de prefixes BGP. Le système de mesure et de décision de chemin illustré sur la figure 3 fait donc face à un défi qui est de suivre et d'optimiser en temps réel les performances de transmission vers toutes ces destinations. Cependant, il est bien connu que la plupart du volume du trafic est généralement concentré sur une petite partie des préfixes BGP [7, 17, 31, 79]. Il est donc possible et raisonnable de se concentrer uniquement sur les préfixes de destination les plus importants, .

A cette fin, il faut prévoir quels préfixes correspondront aux volumes du trafic les plus importants dans un proche avenir. Deux blocs fonctionnels supplémentaires sont ainsi ajoutés à la figure 3) : (iii) la collecte de statistiques sur le volume de trafic ; (iv) la sélection de préfixe, qui identifie parmi l'ensemble des destinations celles les plus importantes (c'est-à-dire ceux ayant le volume le plus élevé dans un avenir prévisible) et communique l'ensemble de préfixes sélectionné aux composants qui effectuent les mesure de performance et décident les chemins.

Deux raisons nous obligent à *prédictivement* sélectionner des préfixes de volume important et à concevoir des mécanismes spécifiques pour cette tâche. Tout d'abord, le volume de trafic par préfixe évolue avec le temps, de même que l'ensemble des préfixes qui représentant un volume important. Walleriche et al. [39] ont montré que le classement de la bande passante des flux peut changer radicalement d'un moment à l'autre. Afin de maintenir un ensemble de préfixes d'importance, il faut donc prévoir répétitivement le volume de trafic pour chaque préfixe. À notre connaissance, aucune étude n'a fait l'objet d'une enquête approfondie sur l'évolution dans le temps du volume de trafic associé aux préfixes BGP.

Deuxièmement, en prévoyant le volume de trafic pour chaque préfixe individuel, des méthodes plus efficaces sont nécessaires. Les modèles bien établis Time Series Forecasting (TSF) et Artificial Neural Networks (ANN) ont déjà été utilisés dans la prédiction du trafic [31, 36, 85]. Ces travaux ciblaient le trafic inter-Point of Presence (PoP) pour les tâches hors ligne, telles que le dimensionnement de réseau. Ces modèles sont non seulement lourds en termes de calcul, mais ils nécessitent également des pré-traitements des données et du réglage des paramètres d'une manière trace par trace. Ces coûts rendent ces méthodes moins applicables dans le contexte de l'inter-domaine TE qui implique jusqu'à 100k préfixes. Par conséquent, des méthodes de prédiction moins complexes sont nécessaires.

Nous avons analysé dans cette partie les traces du trafic réel provenant de neuf réseaux différents situés dans cinq pays pour comprendre la distribution du volume de trafic associé aux préfixes BGP, ainsi que sa variation dans le temps. Nous avons observé que les préfixes les plus importants (représentant le plus grand volume sur une semaine) sont généralement stables dans le temps, avec de légères variations autour de leur volume moyen par heure. Sur la base de cette observation, nous avons proposé trois simples métriques (également faciles à calculer) pour sélectionner de manière proactive les préfixes ayant un volume de trafic prévisible important. Nous avons démontré que les métriques que nous avons proposées conduisent à une meilleure couverture des volumes par rapport aux solutions existantes. De plus, nous avons évalué les performances de transmission pour les préfixes de destination sélectionnés en utilisant plusieurs

fournisseurs de transit. Nous avons également simulé un algorithme qui selectionne des routes dynamiquement dans le temps. Les résultats ont montré que même avec un mécanisme assez basique, la performance RTT globale pourrait être améliorée de 20% dans certains réseaux étudiés par rapport au meilleur fournisseur de transit disponible.

LA QAULITÉ DES MESURES

À partir d'ici, notre étude se concentre sur les mesures de latence et de trajectoires. Ces mesures sont disponibles sur les platesformes TE chez les clients, de même que les données de volume étudiées dans le chapitre 3. Cependant, dans un souci de reproductibilité, nous avons décidé de passer aux mesures effectuées par RIPE Atlas, une plateforme de mesure mondiale offrant un accès ouvert aux données.

La complétude des données

En plus de la reproductibilité, la qualité des données est un autre aspect clé pour les recherches en métrologie. Grâce à des études antérieures [99, 103], il est maintenant connu que en case de RIPE Atlas, la charge a des impacts évidents sur la précision et l'ordonnancement des mesures. Nous nous concentrons sur la complétude des données, un autre aspect de la qualité des mesures qui a reçu moins d'attention jusqu'ici. Des mesures manquantes peuvent entraîner diverses conséquences indésirables. En dehors de l'élargissement de l'intervalle de confiance de l'inférence [110], il nécessite en général des adaptations méthodologiques, par ex. dans l'analyse spectrale [61, 93, 116], sinon l'estimation serait biaisée [62].

Une raison évidente de l'absence de données est que la sonde RIPE Atlas ne fonctionne pas (ou pas correctement), par ex. éteinte [143]. Tant qu'une sonde est alimentée, elle essaie de maintenir une connexion à un contrôleur pour soumettre les mesures et recevoir les allocations des tâches comme indiqué sur la figure 14. Par conséquent, l'activité de connexion de la sonde fournit une bonne indication de la disponibilité de la sonde et est utilisée dans les investigations menées par RIPE sur la stabilité du système d'exploitation de la sonde [129, 130, 149].

Afin de déduire l'existence possible d'autres causes, nous avons comparé les horodatages de mesure avec les moments où un sonde se connecte et se déconnecte du système de contrôleur d'Atlas. Si les mesures manquantes coïncident avec la déconnexion de la sonde, il y a de fortes chances que la raison de cette manque est que la sonde soit dysfonctionnelle, par ex. éteinte. Cependant, si les mesures sont perdues lorsque la sonde est bien connectée, il faut s'attendre à

quelque chose d'anormal, au-delà du problème connu du système d'exploitation de la sonde.

Dans notre analyse couvrant un grand nombre de sondes sur un mois, seulement 60% des sondes v3 Atlas ont des mesures complètes. D'environ 1/3 des segments de manques semblent étroitement liés à la période déconnectée. Le problème de stabilité du système d'exploitation de la sonde pourrait avoir contribué à de tels manques, comme le suggère la distribution à queue lourde de la longueur des segments manques.

Cependant, 2/3 des segments de manques restants se sont produits pendant que les sondes sont connectées. La moitié d'entre eux ne durent pas plus de 2 datapoints et sont donc susceptibles d'être causés par des problèmes d'ordonnancement. Cependant, environ 25% de cette catégorie dure longtemps ($\geqslant$ 1h).

Nous avons signalé la découverte à l'équipe d'ingénierie de RIPE avec un cas spécifique qu'ils pourraient examiner. La dernière réponse de l'équipe RIPE a confirmé que la cas que nous avons cité dans le rapport avait des problèmes de synchronisation de l'heure. Pour aider à faire avancer l'enquête, nous avons partagé avec l'équipe RIPE tous les segments de manque de données de longue durée que nous avons identifiés. Ces échanges peuvent être trouvés sur le forum RIPE Atlas à `https://www.ripe.net/participate/mail/forum/ripe-atlas`, avec le titre "Actual measurement interval much larger than planned".

Bien que le résultat final ne soit pas concluant ni révélateur en termes de mécanisme sous-jacent, cette étude a aidé à réaliser un problème concernant la complétude des données et à le traiter sérieusement. Ce problème peut être évité ou largement atténué, si les sondes sont correctement choisies comme source de données de mesure. Avec des données complètes et relativement complètes au fil du temps, de nombreuses étapes de nettoyage de données peu justifiables peuvent être évitées.

La variation supplémentaire dans les mésures de latence

Dans la TE inter-domaine, la qaulité de mesure depend aussi du fait *si la mesure RTT reflète principalement les caractéristiques des chemins AS*.

La fonction de sélection d'itinéraire dans la figure 3 repose sur les mesures de performance de chemin (plus de détails dans le chapitre 5). Toutefois, les mesures RTT peuvent être "polluées" par des facteurs non liés au réseau, par exemple des problèmes locaux tels que la surcharge du processeur, ou des problèmes de réseau au niveau sub-AS non qui ainsi ne sont pas pertinent, par exemple la congestion locale du réseau de la destination. Réagir à ces problèmes n'est pas l'objectif principal de TE dans l'inter-domaine, car ce der-

nier tout seul ne se suiffit pas. Cependant, aucun de ces travaux antérieurs [24, 49] n'a réalisé l'importance de ce problème.

Ce problème de qualité des données soulève une série de questions : *si nous mesurons un même chemin AS avec différents hôtes dans le préfixe de la destination, à quoi ressembleront ces différentes séries temporelles RTT ? Auront-ils des traits similaires ? Si non, comment pouvons-nous choisir celles qui conviennent le mieux aux fins de TE dans l'inter-domaine ?* Nous essayons de répondre à ces questions en effectuant des regroupements sur un tel ensemble de séries temporelles de RTT. Sans connaissance préalable ou hypothèse, l'étude vise à révéler automatiquement les structures inhérentes de ces séries temporelles de RTT.

Dans cette étude, nous avons analysé des séries temporelles RTT entre deux AS. Nous avons découvert que ces séries temporelles de RTT recueillies dans cette étude démontrent diverses formes de variation, bien qu'un chemin d'AS en commun soit mesurée. Il a confirmé que les mesures RTT doivent être "nettoyée". Nous avons regroupé ces séries temporelles RTT en extrayant plusieurs caractéristiques comme leur représentation. Les clusters résultants ont réussi à séparer les traces bruyantes des traces lisses selon l'intuition humaine et l'expertise. De plus, nous avons localisé l'emplacement qui cause la plupart des variations dans les mesures RTT de bout en bout, en appliquant les méthodes de regroupement aux premiers sauts des mesures traceroute. Nos résultats ont confirmé le bon sens que la plupart des variations proviennent du réseau d'accès.

DÉTECTER LES CHANGEMENTS DANS LES SÉRIES TEMPORELLES DE RTT

Les mesures RTT interviennent dans TE inter-domaine basé sur la mesure à deux phases. Premièrement, les mesures RTT révèlent les moments où la resélection de l'itinéraire est nécessaire. Deuxièmement, les mesures RTT servent de matériel de prise de décision dans la sélection de la route. Nous nous reportons sur le délai mesuré, les coûts de transmission et les politiques de routage, etc., pour décider quel chemin/fournisseur de transit est le meilleur choix pour atteindre chaque destination. Nous discutons dans cette partie l'emploi des mesures RTT pendant cette première phase.

Les moments où la re-sélection de route est nécessaire sont essentiellement lorsque la performances sur certains chemins AS changent. Le défi de la détection de changement de performance provient principalement de deux aspects. Premièrement, les mesures RTT sont bruyantes. De nombreux facteurs le long du trajet mesuré peuvent contribuer aux variations du délai de bout en bout, par ex. fluctuation de la charge sur l'hôte final, l'arrivé soudaine de traffic en grand volume, etc. Cela nécessite des méthodes de détection du changement

qui tolèrent des bruits tel que des diviation de courte durée, tout en restant sensible aux événements qui comptent vraiment, tels que la congestion persistante. Deuxièmement, les caractéristiques de latence sur les différents chemins peuvent se différer beaucoup. Il est donc souhaitable de détecter les changements pour ces séries temporelles sans emploi de paramètres qui dépendant du chaqu chemin/destination. Beaucoup de pratiques courantes ne satisfont pes les exigences énumérées ci-dessus.

Il est généralement admis que les changements de routage inter-domaines ont un impact important sur le niveau de RTT. Pucha et al. [44] ont montré que les changements de routage inter-domaines entraînent une plus grande variation sur le médiane de RTT que les changements intra-domaine. Rimondini et al. [94] ont confirmé que 72.5% des changements de route BGP dans leur étude sont associés aux changements RTT. Des observations similaires ont été faites dans un grand Content Delivery Network (CDN), où les changements de routage inter-domaine sont responsables de plus de 40% de dégradation de l'expérience utilisateur sévère [81].

Les événements intra-domaines ne sont pas moins importants. Pucha et al. [44] ont découvert que les changements de chemin intra-domaine peuvent provoquer des changements RTT d'amplitude comparable à ceux inter-domaine. En outre, ils ont souligné que ce sont les changements de chemin intra-domaine, et non la congestion, qui sont responsables de la majorité (86%) des changements de RTT. Une découverte différente a cependant été faite par Schwartz et al. [66]. Ils ont observé que la plupart des variations RTT se situaient plutôt sur les chemins (c'est-à-dire en raison de la congestion) que parmi les chemins (c'est-à-dire en raison des changements de chemin).

Les conflits dans les travaux précédents pourraient être causés par les différents emplacements d'où les mesures ont été lancées. Par exemple, Chandrasekaran et al. [100] ont observé que les changements de chemin d'AS n'ont qu'un impact marginal sur RTT dans le noyau d'Internet. Cependant ces travaux précédents [44, 66] incluent aussi des réseaux d'accès. Les résultats pourraient aussi changer avec le temps. Par exemple, la topologie Internet "aplatie", la quantité croissante de trafic dans les CDN au cours de la dernière décennie [65, 73] pourrait avoir modifié les caractéristiques du changement de chemin et aussi de la congestion, et par conséquent, leur impact sur RTT.

En gardant cela à l'esprit, nous aimerions souligner les efforts sur les méthodes et les outils. Ils permettent une analyse itérative dans le temps, au-delà de l'observation ou de l'analyse ponctuelle sur un ensemble de données spécifique.

La discussion et la découverte des travaux précédents sont éclairantes, mais leurs méthodes de traitement de mesure RTT peuvent difficilement être appliquées à la TE intra-domaine. Dans [44, 66, 100],

les mesures RTT sont d'abord groupées par des chemins sous-jacents ; l'impact des changements de route sur RTT est ensuite estimé par la comparaison des statistiques de RTT associées, par ex. des centiles. Cependant, dans un système TE esquissé dans la figure 3, les RTT sont mesurés avec une fréquence plus élevée que les chemins. Ainsi, on n'est pas tout le temps sûr sur le chemin emprunté par une mésure RTT à un moment donnée. Les raisons sous-jacentes sont triples. Premièrement, les mesures RTT sont en général moins coûteuses. Compte tenu du nombre potentiel de destinations à surveiller (voir section 3), les mesures de chemin sont mieux limitées. Deuxièmement, un RTT plus petit est l'objectif de TE, nous avons donc l'incitation à suivre de près son évolution. Cependant, le chemin est juste un des résultats de l'optimisation. Troisièmement, les changements RTT se produisent généralement plus fréquemment que les changements de chemin. Une raison importante est la congestion. Le regroupement des mesures de RTT par des changements de route ne peut pas éclairer la présence de tels événements. Par conséquent, nous devons explorer des méthodes qui permettent d'identifier les changements RTT inhérents, au lieu de s'appuyer sur des mesures externes telles que les changements de chemin pour décrire la variation des performances de transmission.

Parmi les études approfondies sur les méthodes de détection des changements et leurs applications dans divers domaines [46, 48, 53], Rimondini et al. [94] sont parmi les premiers à utiliser la détection de changement dans l'analyse de mesure de RTT. Cependant, ils ont réglé la sensibilité de détection de telle sorte que les changements détectés correspondent le mieux aux changements de route BGP vers le préfixe de destination mesuré parmi d'autres préfixes choisis au hasard. Cette approche risque d'ignorer les changements RTT dus aux changements de chemin dans intra-domaine et à la congestion. De plus, un tel réglage est potentiellement nécessaire pour chaque destination individuelle, donc difficile à mettre à l'échelle. Pour obtenir une approche plus générale et découplée des mesures de chemin, nous proposons dans cette partie un cadre d'évaluation pour la sélection et l'étalonnage des méthodes de détection des changements dans leur applications sur les mesures RTT.

Quelle méthode (parmi les nombreuses propositions existantes) est la plus appropriée pour les séries temporelles de RTT dans l'Internet n'est toujours pas prononcée. De plus, de nombreuses méthodes de détection de changement sont paramétriques. Identifier les meilleurs paramètres pour les entrées de type RTT reste aussi obsure. L'absence d'un cadre d'évaluation est un problème fondamental pour résoudre les problèmes susmentionnés.

Un cadre d'évaluation quantifie la performance d'une certaine méthode de détection sur un ensemble de données de référence. Avec l'évaluation quantifiée, différents paramètres d'une même méthode

ou différentes méthodes peuvent être comparés et réglés afin de fournir les meilleurs résultats de détection. Naturellement, un cadre d'évaluation devrait être composé de deux parties : 1) un ensemble de données de "vérité terrain", 2) une méthode de notation.

Cet ensemble de données de "vérité terrain" n'est pas seulement un ensemble de séries temporelles RTT représentatives des caractères de latence sur Internet. Il devrait aussi porter des étiquettes indiquant les moments de changement. Nous ne sommes pas au courant d'un tel ensemble de données qui soit publiquement disponible à ce jour. Nous expliquons dans la section 5.5.2 comment nous le construisons avec beaucoup de soin. Quant à la méthode de notation, elle quantifie la similarité/différence entre la "vérité terrain" et les points de changement détectés par les méthodes. Nous expliquons dans la section 5.5.1 que la classification classique vrai/faux positif est trop rigide pour l'étiquetage manuel et la détection de changement dans les séries temporelles de RTT. Nous explorons et relevons les défis de la comparaison entre deux ensembles d'horodatages avec une tolérance de décalage temporel.

Avec le cadre d'évaluation prêt, il ouvre la porte à l'exploration de la méthode de détection de changement les plus performants pour les mesures RTT. Pour la famille de détection présentée dans la section 5.4, deux principaux paramètres doivent être définis : la pénalité et la fonction de coût (qui depend de l'hypothèse de la distribution). Nous considérons la combinaison entre tous les critères d'information introduits (AIC, BIC, MBIC et Hannan-Quinn), et tous les types de distribution supportés (Gaussiane, Poisson, Exponentiel et Gamma), y compris l'approche non-paramétrique basée sur la distribution empirique.

Avec quelques tests préliminaires, nous avons rapidement réalisé que la détection avec la distribution normale tend à être sur-sensible, sous tous les configuration possibles de pénalité. Beaucoup de variations de courte durées et insignifiants au term d'amplitude sont marqués comme des changements. C'est parce que la moyenne et la variance de la distribution normale sont indépendamment contrôlées par deux paramètres distincts, ce qui augmente les chances de s'adapter aux changements subtils. D'un autre côté, les distributions exponentielle, Gamma et de Poisson sont trop engourdies. La moyenne et la variance de Poisson et de la distribution exponentielle sont couplées par un paramètre, qui restreint leur flexibilité d'ajustement [1]. La distribution gamma est confrontée au même problème, mais avec une histoire plus compliquée. Une distribution gamma peut être décrite par deux paramètres α et β : mean = $\frac{\alpha}{\beta}$, variance = $\frac{mean}{\beta}$. [91], l'implémentation que nous utilisons, nécessite une entrée *a priori* pour α, qui détermine en fait la sensibilité globale. Seul β est ajusté pour la

1 Poisson, mean = variance = λ; exponentielle, moyenne / variance = λ, moyenne = $1/\lambda$.

détection de changement. Avec un plus grand α, un β plus grand est nécessaire pour maintenir la même estimation moyenne pour un segment donné. Une moyenne fixe avec un plus grand β impose une plus petite tolérance de variation, donc plus susceptible de diviser le segment donné en raison de changements de variance plus petits. En bref, un plus grand α conduit à une détection plus sensible. L'option par défaut définit α à 1, ce qui dégénère la distribution Gamma en distribution exponentielle. Nous avons également essayé α de 1 à 100, avec l'étape égale à 1. Aucun d'entre eux ne surpasse les meilleurs paramètres affichés plus tard. Nous ne considérons donc plus la distribution Gamma.

En supposant une distribution exponentielle et de Poisson, nous remarquons que le niveau moyen d'une série temporelles de RTT décide d'une certaine manière la tolérance de variation. Par exemple, pour un chemin incluant des liaisons transpacifiques, nous nous attendons à un RTT minimum supérieur à $80msec$. Dans ce cas, la distribution de Poisson correspondante pourrait facilement tolérer des déviations RTT de $20msec$, ce qui est déjà non négligeable. Cependant, le fait d'avoir une moyenne et une variance couplées peut aussi être une caractéristique souhaitable. Nous avons observé au cours de l'étiquetage que le niveau d'un segment RTT et sa variance sont souvent positivement liés au cours des périodes de congestion.

Pour tirer parti de la caractéristique décrite ci-dessus et augmenter la sensibilité de détection, nous proposons pour la distribution exponentielle et de Poisson une *transformation de données* : d'abord soustraire la série temporelle de RTT par sa valeur minimum (baseline) pour abaisser son niveau global ; les changements sont ensuite détectées. Ce paramètre est noté `cpt_poisson` et `cpt_exp` respectivement. Pour l'intérêt de comparaison, nous considérons également la distribution de Poisson **sans transformation de données** et nous l'indiquons comme `cpt_poisson_naive`. La distribution normale `cpt_normal` et l'approche non-paramétrique `cpt_np` sont appliquées directement sur les séries temporelles initiales.

Toutes les combinaisons entre les types de distribution et les choix de pénalité sont evaluées. Pour chaque type de distribution, nous montrons seulement son paramètre de pénalité le plus performant en termes de score F_2 pondéré dans la figure **??**.

Plus de 75% de changements, en termes de poids, peuvent être détectés pour plus de la moitié des séries temporelles avec peu n'importe quel distribution. Tous ces types de distribution ont montré une meilleure performance en considerant le F_2 pondéré que le F_2 classique, indiquant que certains changement non-détectés sont en effet de peu d'importance opérationnelle. Cependant, il semble avoir un grand espace d'améliorations. Des efforts sont particulièrement nécessaires pour augmenter la précision de detection.

La précision de `cpt_normal` est particulièrement mauvaise. Cela confirme que `cpt_normal` est en effet sur-sensible sur les données de type RTT. Au contraire, le rappel de `cpt_normal` est remarquable parmi tous les candidats. Cependant, ses scores F_2 sont les plus faibles parmi tous les candidats. La pauvre performance globale de la détection souligne l'importance de trouver un juste équilibre entre la sensibilité et la pertinence. Pour les autres méthodes, leurs performances sont relativement proches. Par rapport à `cpt_poisson_naive` (sans transformation de données), `cpt_poisson` obtient un rappel plus élevé sans sacrifier de façon évidente la précision. Par conséquent, `cpt_poisson` présente un léger avantage sur les performances globales. En fait, sans transformation de données, en supposant la distribution exponentielle ne détecte aucun changement pour une grande partie des séries temporelles dans l'ensemble de données de vérité terrain. Ce sont toutes des preuves que la transformation de données proposée améliore la performances de détection pour la distribution de Poisson et exponentielle.

Nous détectons aussi les changements de chemin rencontrés par les mesures RTT collectées. Les changements de chemin de niveau AS et IP sont pris en compte. Ils sont connus pour avoir un impact potentiel sur RTT. Le but de la détection de changement de chemin n'est pas de répéter des études précédentes, tel que quel type de changement de chemin contribue le plus au changement RTT. Cela aide plutôt à améliorer la compréhension de la détection de changements pour les mesures RTT.

Le tableau 6 détaille le nombre de correspondances entre les changements de chemin et ceux de RTT. Chaque cellule indique le nombre de correspondances entre la ligne (type de changement de chemin) et la colonne (méthode de détection de changement RTT). La dernière colonne contient le nombre total de changements de chemins de chaque type. De même, la dernière ligne fournit le nombre total de changements RTT détectés par les deux méthodes.

La fraction des changements de chemin d'AS correspondant aux changements de RTT dans cet étude est beaucoup plus faible que ce taux de 72.5% dans [94]. Il semble que les changements de chemin AS ont un impact moins significatif sur RTT que la compréhension précédente. Y a-t-il quelque chose de particulier dans nos données ou nos méthodes ? De plus, le nombre de changements de chemin d'AS correspondant à un changement de RTT par `cpt_np` n'est que d'une moitié de celui par `cpt_possion`. Pourquoi ? Tous ces phénomènes sont très intrigants. Nous explorons les raisons sous-jacentes dans la section 5.9. Dans ce condensé, nous expliquons uniquement une de ces observations : pourquoi il y a une grande partie de changement RTT correspondent à aucun changements de chemin.

D'abord, nous n'avons pas été en mesure de mesurer le chemin de reoutr avec RIPE Atlas. Par conséquent, il est impossible de détecter

les changements de chemin sur cette partie-là. Cependant, ces changements de chemin pourraient avoir contribué aux changements de RTT. Notamment dans le contexte du routage inter-domaines où les chemins dans les deux sont souvent asymétriques. Cela implique que les changements RTT provoqués par les changements de chemin sur le sens de retour sont probablement différents de ceux provoqués par les changements de chemin sur le sens d'aller.

Deuxièmement, la congestion. La congestion peut être indépendante des changements de chemin et causer des variations significatives sur RTT. La figure 44b donne un exemple typique de changements RTT probablement causés par la congestion. Il y a trois augmentations transitoires de RTT qui peuvent être visuellement remarquées dans la sériee temporelle illustrée. Nous disons que les deux derniers sont probablement de la congestion. Premièrement, ils ne correspondent à aucun changement de chemin, au moins sur le sens d'aller. Deuxièmement, ces augmentations sont probablement causées par le remplissage des files d'attente le long du chemin. Parce que, les valuers RTT ne sont pas "plates". Sur un chemin légèrement chargé, nous nous attendons à des mesures RTT relativement constantes. C'est parce que les files d'attente sont presque vides, donc pas de place pour la variation de délai. La valeur de RTT est dominée par la latence de propagation. Cependant, les variations de RTT à l'intérieur de la bosse est probablement une manifestation de l'évolution de la demande de trafic, et comment ce dernier change la longueur des files d'atteinte. Les deux bosses/congestion s'écartent de la value de base (la latence de propagation) et durent plusieurs heures. Ils ont donc un impact significatif sur les performances de transmission. Nous les avons détectés avec succès avec les méthodes de détection de changement étudiés dans cette thèse. Cependant, une telle détection n'est pas possible avec la méthode précédemment proposée [93]. Il effectue une analyse spectrale sur les séries temporelles de RTT pour identifier des congestions périodiques. Une telle congestion persistante est normalement due au manque de capacité du réseau. Alors, la congestion transitoire dans la figure 44b est plus probablement causée par la variation de trafic soudaine. La TE basé sur la mesure vise à éviter les deux types de congestion lorsqu'il existe des chemins alternatifs avec la capacité suiffisante.

Troisièmement, la détection sur-sensible. Si nous supposons hardiment que les changements de chemin sur le sens de retour provoquent une quantité comparable de changements de RTT comme le font les changements de chemin sur le sens d'aller, il y a encore beaucoup de changements RTT non appariés. Certains d'entre eux pourraient être effectivement attribués à la congestion, comme expliqué ci-dessus. Les modifications RTT non appariées restantes sont manifestement le résultat d'une détection trop sensible. Nous relevons, à partir d'une vue macroscopique dans la section 5.7 et la section 5.10.1,

que `cpt_poisson` a tendance à surestimer le nombre de changement lorsque la série temporelle de RTT est bruyante. Des exemples individuels sont données dans la figure 44 pour illustrer la nuance de sensibilité à partir d'une vue microscopique. Dans la figure 44a, `cpt_np` a détecté toute la congestion périodique de petite amplitude. C'est en fait assez impressionnant, car ces changements sont à peine visibles pour les experts humains. Les changements marqués par `cpt_np` ont en effet mis en évidence leur présence, et les ont rendus plus faciles à remarquer visuellement. Dans la figure 44b, les deux méthodes ont identifié les deux grandes bosses près de la fin de la série temporelle. La différence est que`cpt_poisson` a également marqué les changements de niveau intermédiaire. Ces changements intermédiaires ne sont pas corrélés aux changements de chemin. En plus de cela, ils sont également redondants en informant la congestion qui se produisait à ce moment-là. La raison d'une telle sur-sensibilité était due à son incompétence de l'ajustement de la sensibilité de détection en fonction du niveau de variance. Ce problème est exploré et expliqué dans la section 5.10.2.

Pour résumer le travail de cette section, nous avons proposé un cadre d'évaluation pour la détection des changements sur les séries temporelles RTT. Le cadre est robuste avec l'ensemble de données étiqueté manuellement et pondère les changements RTT en fonction de leur importance dans l'opération du réseau. Nous avons en outre conçu une transformation de données adaptée aux mesures RTT pour améliorer la sensibilité de détection de certaines méthodes de détection. Enfin, nous corrélons les changements RTT et path détectés en établissant une correspondance entre eux. Nous avons étudié la distinction de sensibilité entre différentes méthodes de détection des changements.

DÉDUIRE L'EMPLACEMENT RESPONSABLE DE CHANGEMENTS RTT

Cette idée provient d'abord de l'étude de cas dans la section 4.5 sur les changements RTT partagés par plusieurs séries temporelles RTT traversant des différents chemins AS. Nous avons réalisé que ces changements RTT ne sont pas exclusifs aux mesures sur un chemin Internet spécifique, mais impactent plutôt plusieurs séries temporelles RTT simultanément, comme le montre la figure 29. Optimiser le routage inter-domaine contre la cause de tels changements de RTT serait alors une approche plus fondamentale et plus efficace que de traiter chaque préfixe individuel et chaque chemin vers eux.

Afin de déduire l'emplacement des causes, une hypothèse raisonnable est que de tels changements RTT partagés sont plus probablement causés par les parties communes de ces chemins, au lieu d'être la conséquence d'une synchronisation parfaite entre plusieurs problèmes dispersés dans divers endroits. Avec cela, il est alors possible

d'affiner la portée des causes possibles avec des mesures ayant à la fois des parties communes et des parties divergentes. Un exemple de jeu d'inférence est donné dans la figure 46. Avec l'hypothèse, nous pouvons d'abord étendre la cause aux liens 1 et 4, aux nœuds 2 et 5. Comme il y a une mesure sans changement de RTT, celui traversant lien 1 et le nœud 2. lien 1 et le nœud 2 sont ainsi moins probable d'être la cause. En consequence, le lien 4 et le nœud 5 sont alors plus susceptibles d'être la cause.

Nous sommes intéressés à identifier l'endroit dans l'Internet, aussi précise que possible, qui est responsable des changements RTT détectés. Ceux que nous avons en entrée sur la plates-formes TE chez les clients sont 1) les mesures RTT provenant de sources multiples vers des destinations multiples pour les utilisations TE ; 2) les chemins AS sous-jacents pour ces mesures RTT. Les sources de mesures sont les plates-formes clientes et les destinations sont les préfixes auxquels les clients envoient leur trafic. La source des mesures peut être multiple si nous fusionnons des mesures provenant de plusieurs plates-formes clientes ou si le client a plusieurs sites avec différentes options de fournisseur.

La figure 47 décrit les composants pour l'inférence de la cause de changement de RTT. La détection de changement (section 5) transforme les séries temporelles de RTT en séquences d'evènement de change. La construction de la topologie construit un graph pour les AS et les liaisons traversés par les mesures RTT. Ce graphe de topologie est une étape intermédiaire dans la conception d'une métrique d'inférence pour chaque nœud et chaque lien présents dans la topologie. Comme on le voit sur la figure 46, le fait que si un nœud/lien est la cause du changement RTT dépend non seulement des mesures qui le traversent, mais aussi de celles à coté. L'identification de tels ensembles de mesures pour chaque nœud/lien nécessite des connaissances sur la topologie. De plus, le graphe de topologie sert aussi à visualiser l'endroit des cause inférés. La valeur exacte de ces métriques d'inférence à chaque instant est calculée à partir de séquences d'événements de changement RTT. Ensuite, l'inférence de cause est effectuée pour chaque nœud et chaque lien en fonction de la valeur de leurs métriques d'inférence. La sortie de l'ensemble du système indique les liens et les noeuds responsables des changements RTT à un moment donné.

Afin d'initier l'inférence, nous avons fait deux hypothèses.

Assumption 1-*cause unique*. *Pour chaque changement RTT détecté, il n'y a qu'une seule cause, nœud ou lien, sur le chemin mesuré.*

C'est une hypothèse courante faite dans les études de congestion de TCP [6, 109]. S'il y a une congest le long du chemin, il se stabilisera finalement sur le lien avec la bande passante la plus faible. Nous étendons cette hypothèse aux liens inter-AS et AS pour s'adapter à la granularité d'inférence.

Assumption *2-parties communes.* *Si les mesures sur plusieurs chemins subissent un changement RTT partagé, les parties communes de ce chemin mesuré sont plus susceptibles d'être la cause.*

Il décrit une façon possible de satisfaire l'hypothèse précédente lorsque plusieurs mesures RTT avec des intersections sur leur chemins sont considérées. C'est simplement un cas plus probable que d'avoir simultanément plusieurs parties éloignées dans l'Internet provoquent un changement significatif à la même instant. Nous utilisons cette hypothèse pour concevoir des ensembles de mesures spécifiques (métrique d'inférence) pour chaque nœud et chaque lien dans le graph de topologie. Ensuit, nous examinons chaque lien et chaque nœud sur le graph à l'intervalle de 10 minutes.

Pour vérifier si un nœud (AS) est la cause exclusive d'un changement RTT partagées, nous concevoir pour elle un ensembles de mesures (métrique d'inférence), dans lesquels le seul élément commun est le nœud lui-même. Si une majorité des mesures dans un tel ensemble vivre en même temps un changement de RTT, nous pouvons alors le localiser au nœud en question, selon l'hypothèse *2-common parts*.

Après inférence de cause pour chaque nœud, nous pouvons exclure tous les liens avec l'un de ses deux nœuds inférés comme cause, selon l'hypothèse *1-single cause*. Ensuite, si un lien provoque effectivement un changement de RTT, nous nous attendons à ce qu'une majorité de mesures traversant ce lien subissent simultanément (dans un même intervalle de temps) un changement de RTT, condition nécessaire mais non suffisante pour la responsabilité de lien. En d'autres termes, en violant cette condition, le lien peut être exempté d'être cause du changement RTT.

Pour les liens restantes, leur responsabilité dépend des liens adjacents et non adjacents. Imaginez qu'un incident se produise au cœur d'Internet ou à un grand IXP, un large éventail de mesures traversant des liens périphériques peut être potentiellement impacté. Pourtant, les liens périphériques ne sont pas responsables du changement RTT. Pour retracer la véritable cause du changement, différents critères sont nécessaires pour les liens dans différentes configurations topologiques.

Nous avons implémenté un outil de visualisation interactif pour inspecter le nombre normalisé d'événements de changement RTT et le résultat d'inférence de chaque lien/nœud sur le graph de topologie. La figure 52 est un capture d'écran de l'outil qui montre l'intensité de changement RTT sur chaque lien.

Cette thèse est développée autour d'une poursuite de *mieux utiliser les diverses mesures de réseau dans l'ingénierie du trafic sortant en inter-domaine pour les AS stub*.

Nous avons souligné la nécessité de se concentrer sur les destinations les plus importantes. Grâce à l'étude du dynamisme temporel des volumes par préfixe BGP, nous sommes arrivés avec des méthodes simples qui sélectionnent efficacement l es préfixes de destination avec des volumes de trafic importants. La scalabilité du système de mesure peut ainsi être améliorée.

Plus tard, nous nous sommes concentrés sur les mesures de latence. Nous avons présenté et diagnostiqué certains problèmes de qualité des données. Des lignes directrices pour atténuer leurs impacts sur le traitement des données et la sélection de route ont été discutées.

Afin de mieux interpréter les mesures de performance, nous avons introduit la detection de changement dans le traitement des séries temporelles RTT. Ces méthodes détectent des changements significatifs sur la performance du chemin et servent de déclencheur pour la re-sélection de route. Pour permettre et encourager les efforts futurs, nous avons construit un cadre d'évaluation sur la détection de changement pour les mesures RTT dans l'Internet.

Enfin, avec l'aide de la detection de changement, nous avons été en mesure de qualifier le pourcentage d'un groupe de mesures qui subissent des changements RTT au même moment. Nous avons en outre inféré les endroits dans l'Internet qui ont potentiellement provoqué ces changements. Cette visibilité permet d'optimiser le routage ver des préfixes que nous n'avons pas pu mesurer directement. Afin de mieux illustrer le processus d'inférence et les causes identifiées, nous avons conçu des outils de visualisation interactifs afin de tracer les métriques d'inférence et aussi les résultats d'inférence sur un graphe de topologie au niveau AS.

ABSTRACT

As transit price continues to drop, mulithoming has now become a common practice among many medium and even small size networks. Yet, improving the transmission performance over multiple Internet paths remains challenging. One major difficulty comes from the current Internet routing protocol Border Gateway Protocol (BGP). It is not performance-aware in propagating and choosing routes. On top of that, BGP is not going to obsolete shortly.

To bypass the limitations of BGP, some previous studies and industrial solutions suggest regular measurement of transmission performance over all available paths. Then, the best routes are chosen for each destination considering not alone policies but as well measurements. That is the main idea of measurement-based Traffic Engineering (TE). In transferring this idea into designs/systems that can cope with real network requirements, plenty of issues are still left open.

First, measurement-based TE has to deal with the huge number of potential destinations. This heavy measurement load is further multiplied by the number of available paths/providers. Instead of covering the entire address space, it is more resource efficient to focus on several important destinations. To verify the feasibility of that intuition, we studied working traffic traces from real networks. The results showcased that most traffic is indeed concentrated on a small fraction of destinations. Based on these findings, we devised simple methods to predict those 'heavy-hitter' destinations.

Second, measurement-based TE requires insightful measurement interpretation. In this work, we mainly cared about round-trip latency on Internet paths. We first identified and diagnosed several data quality issues that were previously unattended. Guidelines to mitigate their impacts were discussed. Further, we tried to cluster latency time series with similar characters, e.g. overall variation level, a particular shape at a given moment.

We encountered difficulties in meaningfully clustering latency measurements. These difficulties led us to the detection of moments of significant changes for individual latency time series. Moments of performance change can be regarded as a compact data representation of latency time series. They therefore have the potential to facilitate the grouping/clustering operation. Ultimately, these moments are when route re-selection is potentially needed for the measured destinations. Because otherwise traffic toward these destinations might suffer from avoidable performance degradation. To that end, we applied *change-point analysis* methods to latency time series. We devised an evaluation framework to quantify the robustness and sensitivity of diverse

detection methods. With the open-sourced evaluation method, we aimed at encouraging as well further efforts on methodological improvements.

Last but not the least, we tried to infer the network locations that are responsible for significant latency changes. This visibility allows performance-aware route selection for certain destinations that can not be measured directly. When paths toward these destinations traverse change causes, we reasonably assume similar performance changes on these paths as well. We since developed a series of inference procedures to attribute the cause of latency changes to Autonomous System (AS) or inter-AS links. Change detection methods previously studied were employed to first detect performance changes and then to group paths that underwent a same performance change. To better illustrate the inference process and the identified causes for latency change, we built two interactive visualization tools to plot the results on a topology graph.

In this dissertation, we tackled some of the most pronounced challenges in measurement-based TE for interdomain routing. Contributions are brought to measurement scalability, interpretation of performance data and visibility on causes of performance changes.

RÉSUMÉ

Avec la baisse du prix de transit, multihoming a maintenant devenu une pratique courante parmi les réseaux de moyenne même petite taille. Toutefois, il reste difficile de réellement améliorer la performance de transmission via ces multiple chemin désormais disponibles. Une des difficultés est d'originaire du protocole de routage employé dans ce contexte : Border Gateway Protocol (BGP). Le protocole ne prend pas en compte les éléments de performance lors de la sélection et la propagation des routes. De plus, ce protocole va probablement continuer à dominer les routages d'Internet.

Pour pouvoir contourner les contraints du BGP, des études dans le passé et des solutions industrielles suggèrent de mesurer régulièrement les multiples chemin Internet. Puis, la meilleure route à chaque destination est sélectionnée selon les données de ces mesures. C'est ce que nous appelons l'ingénierie du trafic (TE) alimenté par la mesure. En réalisant cette idée et satisfaisant les requis des vrais réseaux, pleine de problèmes sont laissés ouverts.

D'abord, un system comme tel doit pouvoir gérer énorme de destinations. Cela pesse lourdement sur la fonctionnalité de mesure. En plus, cette charge est multipliée par le nombre de chemins disponibles. Au lieu de couvrir la totalité des destinations, il est clairement plus sage de se focaliser sur certains unes les plus importantes. Afin de vérifier que ce soit faisable, nous avons étudié des vrais trafics sur les vrais réseaux à profils variés. Le résultat montre qu'effectivement une grande partie de trafic se concentre sur une petite collection de destinations. De plus, les trafics associés aux ces destinations sont relativement plus facile à prédire que le reste. S'appuyant sur ces découverts, nous avons identifié des moyens simples mais efficaces à capter ces destinations d'importance.

Deuxièmement, les données récoltées par le system de mesure restent à être interpréter. Dans cette thèse, nous nous occupons principalement la latence d'aller-retour comme l'indicateur de performance d'un chemin. Nous avons commencé par identifier et analyser certains problèmes liés à la qualité de données. Nous avons également discuté comment atténuer ces impacts. En outre, nous avons essayé de grouper ensemble des séries temporelles de latence qui partage des traits similaires, par exemple vécus un changement au même moment.

En donnant des sens pratiques aux groups résulté, nous avons rencontré des difficultés. Ils nous ont dirigé vers la détection de changement pour des séries temporelles de latence. Les moments de changement significatif se peuvent être regardés comme un représentation compacte pour les données de performance. Ils nous facilitent

ainsi l'opération de regroupage plus tard. Finalement, ces changements servent également comment déclencheur à la réévaluation des routes sélectionnées. Car quand un changement de performance est survenu sur l'un des chemins mesurés, il indique soit une dégradation potentielle soit des espaces à l'amélioration. S'en rendant compte, nous avons mis en pratique des méthodes de *changepoint analysis* sur des séries temporelles de latence. Nous avons même conçu un mécanisme d'évaluation de la qualité de détection pour données de type latence Internet. Les implémentations de cet outils et données labélisées sont rendu publique, dans l'objectif d'encourager les efforts à venir sur la méthodologie de détection.

Finalement, nous avons essayé d'inférer les endroits dans l'Internet qui sont susceptibles d'être la cause des changements de performance détectés. Cette visibilité permet de réagir aux accidents pour certain destinations que nous n'arrivions pas à mesurer directement. Quand certains chemins vers ces destinations traversent une cause de changement, il est probable que ces chemins non mesurés ont vécu le même changement que les autres mesurés. Basant sur cette hypothèse, nous avons développé une sérié de logique qui attribue la cause de chaque changement de performance à un réseau sur le chemin ou un bout de lien entre deux réseaux. Les méthodes de changpoint analysis étudiées plutôt nous aident d'abord à identifier les changements de performance sur chaque chemin mesuré, et puis à regrouper les chemins par les changements qu'ils ont vécu. Pour mieux illustrer le processus et les résultats de l'inférence, nous avons développé des outils interactifs projetant les outputs de chaque étape sur une graphe de topologie.

Dans cette thèse, nous avons entrepris des défis les plus remarqués concernant l'ingénierie du trafic alimenté par la mesure dans routage Internet. Nous avons apporté des contributions sur la scalabilité, l'interprétation des données et la visibilité sur la cause de variation de performance.

PUBLICATIONS

[1] Wenqin Shao, Francois Devienne, Luigi Iannone, and Jean-Louis Rougier. "On the use of BGP communities for fine-grained inbound traffic engineering." In: (2015). arXiv: 1511.08336. URL: http://arxiv.org/abs/1511.08336.

[2] Wenqin Shao, Jean Louis Rougier, François Devienne, and Mateusz Viste. "Improve round-trip time measurement quality via clustering in inter-domain traffic engineering." In: *Proceedings of the NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*. AnNet. 2016, pp. 1105–1108. ISBN: 9781509002238. DOI: 10.1109/NOMS.2016.7502970.

[3] Wenqin Shao, Jean-louis Rougier, François Devienne, and Mateusz Viste. "Missing measurements on RIPE Atlas." In: *CoNEXT Student Workshop*. 2016. arXiv: arXiv:1701.00938v1.

[4] Wenqin Shao, Luigi Iannone, Jean Louis Rougier, Fran??ois Devienne, and Mateusz Viste. "Scalable BGP prefix selection for effective inter-domain traffic engineering." In: *Proceedings of the NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*. NOMS. 2016, pp. 315–323. ISBN: 9781509002238. DOI: 10.1109/NOMS.2016.7502827. arXiv: 1511.08344. URL: http://arxiv.org/abs/1511.08344.

[5] Wenqin Shao, Jean-louis Rougier, and Antoine Paris. "One-to-One Matching of RTT and Path Changes." In: *29th International Teletraffic Congress (ITC 29)*. 2017. URL: https://arxiv.org/abs/1709.04819.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

ACM	Association for Computing Machinery
NAT	Network Address Translation
LISP	Locator/Identifier Separation Protocol
SDN	Software Defined Networking
RIB	Internet Routing Information Base
FIB	Forwarding Information Base
TE	Traffic Engineering
BGP	Border Gateway Protocol
AS	Autonomous System
MED	Multi-exit Discriminator
RTT	Round-Trip Time
CDN	Content Delivery Network
LB	Load Balancing
TSF	Time Series Forecasting
ANN	Artificial Neural Networks
ISP	Internet Service Provider
CP	Content Provider
HP	Hosting Provider
IXP	Internet Exchange Point
IRP	Intradomain Routing Pattern
EWMA	Exponentially Weighted Moving Average
ARIMA	Autoregressive integrated moving average
PoP	Point of Presence
RIPE	Réseaux IP Européens
PAM	Partitioning Around Medoids
ASW	Average Silhouette Width

PCA Principal Component Analysis

UDM User Defined Measurements

DTW Dynamic Time Wrapping

ED Eucilidean Distance

NOC Network Operation Center

NDA Non-disclosure Agreement

DM Divergent Measurement

FM Feature Measurement

EXT Extension Link

iEXT Impacted Extension Link

1 INTRODUCTION

Internet is a collection of individually managed networks. Its distributed management and routing scheme allows it to grow rapidly. However, the other side of the coin is sub-optimal traffic distribution from a global view. This problem is often manifested as congestion when there is still available capacity. Many efforts has thus been devoted to a better transmission performance, by alleviating or avoid congestion.

Congestion takes place on an Internet path shared by multiple flows when the total demand exceeds the link capacity. In avoiding vicious competition that eventually blocks all flows, end-to-end congestion control mechanism plays an important role. It improves transmission performance in a distributed manner. It aims at 1) fully utilizing bottleneck bandwidth while 2) introducing minimum additional transmission delay (short queue length) and 3) ensuring fair share of resources [3, 6, 109].

The transmission performance can be further improved with big enough link capacity to satisfy the demand of all flows. To that end, it is necessary to regularly dimension the network according to the growth of traffic demands [26]. Yet, infrastructure-building alone is not enough. First, network deployment happens on a much longer time span than traffic fluctuations. Before extra capacity is deployed, some links may become saturated while others remain almost idle, due to changes in traffic demand. Second, over-dimensioning is costly, given that future technologies will drastically reduce the cost per bandwidth unit.

Therefore, reactive and flexible routing schemes are needed, complementary to network dimensioning. They maximize the actual usable capacity. Within a network (intradomain), the administrator may in first place estimate/shape the traffic matrix that comes across its network. Then, each flow can be split onto multiple paths to best fit in the available capacity [75, 83]. Across different networks (interdomain), the room for performance improvement comes mainly from multiple Internet paths between source and destination networks. It is because the end-to-end bottleneck capacity is potentially enlarged with richer Internet paths. Such path diversity can already be obtained through multihoming under current Border Gateway Protocol (BGP). With multihoming, a network purchases the access to the rest of the Internet via multiple providers. Many other propositions as well encourage the propagation of multiple Internet paths, to name a few BGP Add-path [123], MIRO [40], NIRA [47], YAMR [63], Path-

let routing [56], IDRD [84], and etc. However, in interdomain routing, each network generally does not have the visibility outside its own territory, e.g. capacity of a remote link and competing traffic demand from other networks on that link. This is particularly true with BGP, the de facto interdomain routing protocol that is not going to obsolete shortly. Therefore, it is not possible for current networks to figure out which available paths offer the best performance toward a certain destination. In other words, the challenge is how to make better use of available Internet capacity with a performance agnostic protocol BGP. We undertake this challenge in this dissertation.

One straightforward way to make the route decision performance-aware is through measurement. In engineering their interdomain peering edge, both Facebook and Google employ instrumentation embedded in their applications to learn end-to-end performance over multiple available paths [118, 119]. As transit price drops, multihoming becomes rather a common practice of numerous medium and small size networks. These networks as well have the immediate need for performance improvements, so that their business could survive.

Despite the concrete need for measurement-based interdomain traffic engineering, many questions remain unanswered, or partially addressed. A typical network can communicate with ~ 100k different destinations regularly. Continuously measuring the performance to all that many destinations are costly and not necessary. What are the most important destinations to focus on? Do these destinations change over time? How to identify them? Besides volume measurements, how should performance measurements be processed? Do they need cleaning? If yes, what are the potential data quality issues? What are the origins of these issues? How can we showcase and mitigate their impacts on route selection? Further, in order to dynamically react to performance changes, how to detect in first place significant changes in performance measurements? How to do so without hard-coded or ad hoc parameters? How to achieve at the meantime an acceptable level of robustness? Finally, once a performance change is detected, how can we learn more about this event from a network perspective? Where does it happen? Does it potentially impact other Internet paths currently in use? We aim to explore answers to the above questions in this dissertation, so that the building of measurement-based route selection system advances in these aspects: measurement scalability, interpretation of performance data and visibility on causes of performance changes.

2

BACKGROUND, SCOPE AND ROADMAP

2.1 INTERDOMAIN ROUTING

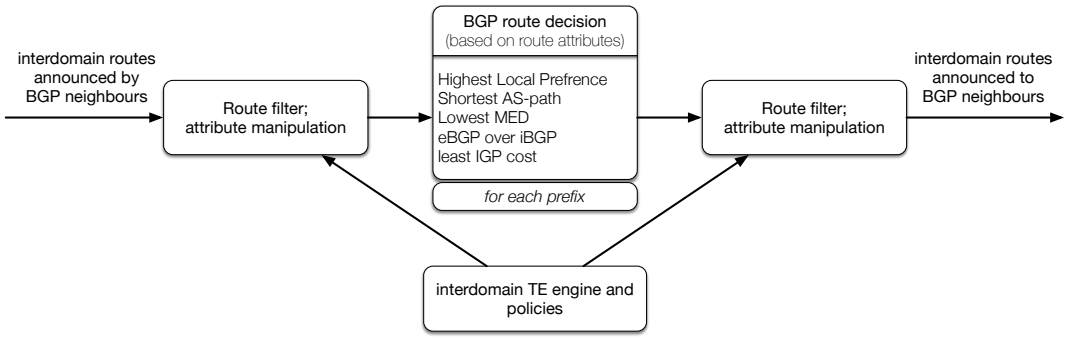


Figure 1: Workflow of Border Gateway Protocol (BGP) route selection and propagation within an Autonomous System (AS).

Interconnection of tens of thousands of independently manged networks forms Internet. Each individual network is as well called an Autonomous System (AS). Routing that happens among those ASes is referred to as *interdomain routing*. In order to exchange interdomain routes, each AS uses Border Gateway Protocol (BGP) [140], a path vector routing protocol, to communicate with other ASes. Each AS announces its own routes (routes towards its own prefixes) along with other routes to its BGP neighbors. For each prefix as destination, one single best route is selected by the BGP route decision process. The selection considers the BGP attributes attached to each route, as illustrated in Figure. 1. According to configured TE polices, each route can be filtered or altered based on its attributes [10, 21]. Such operations can take place before BGP route decision and as well before route advertisement to its neighbors.

Fig. 2 illustrates the propagation of interdomain routes to prefix `137.194.0.0/16` of AS1712 via BGP exchanges. Each AS inserts its own AS number when announcing the route to other ASes, thus forming an AS path at the receiver side. After AS3333 learns the routes to `137.194.0.0/16` and AS1712 learns routes to `193.0.0.0/21` in a similar way, the two ASes can exchange traffic across Internet using these two prefixes.

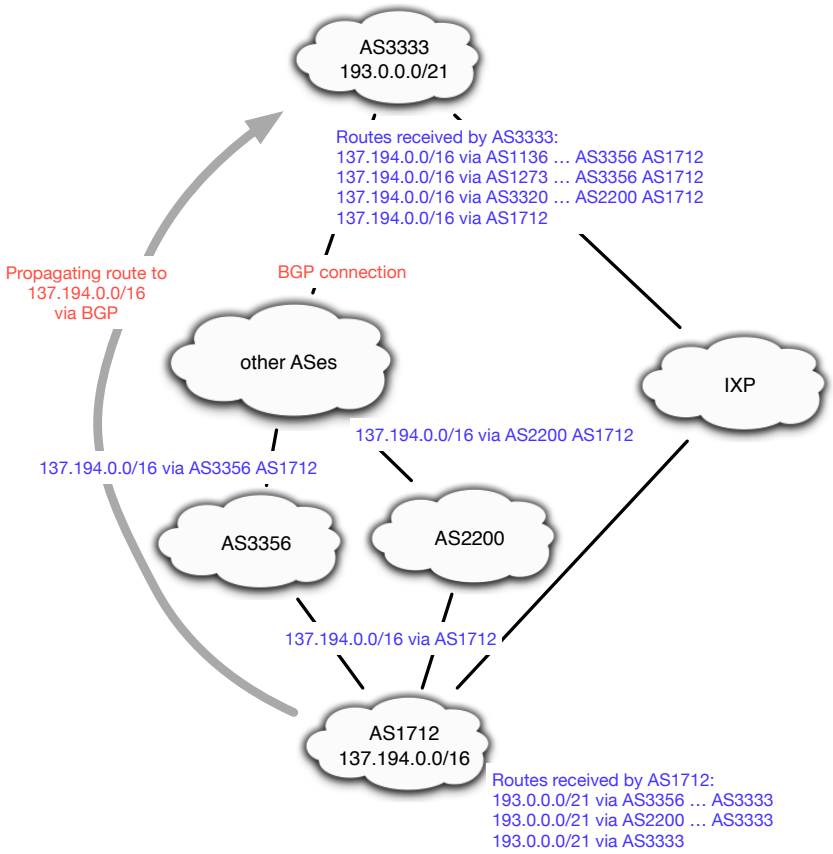


Figure 2: Interdomain route propagation via BGP, an example of `137.194.0.0/16`. Networks illustrated are fictional.

There are two types of ASes in the above BGP exchanges: *transit provider* and *stub AS*. Transit provider is to stub as router to end host. Transit provider refers to ASes that offer to forward/reroute traffic that is not originated from nor sent to itself, as a commercial service. For that purpose, a transit provider announces to its clients all the interdomain routes its learns and to its other BGP neighbors the routes to its clients. In Figure 2, AS3356 and AS2200 are transit providers of AS1712. They help announce the route to prefix `137.194.0.0/16` to the rest of Internet. On the contrary, a stub AS only cares about sending out its own traffic and becoming reachable to others. Accordingly, it only announces to its transit providers its own routes. AS1712 is a stub AS in Figure 2, as it does not relay interdomain routes for other ASes. For example, AS1712 will not announce to AS3356 the route to `193.0.0.0/21` learnt from AS2200. Consequently no traffic toward ASes other than itself shall arrive at it.

Besides the relationship described above, peering is another type of exchange under BGP. Two ASes in peering relationship exchange

with each other their own routes, so that they can directly communicate without employing a transit provider. In Figure 2, AS1712 and AS3333 directly peer at an IXP. IXP is a collocation facility that eases establishment of peering relationship, thus is transparent to BGP route exchanges.

2.2 INTERDOMAIN TE

Thanks to transit and peering relationship, AS3333 and AS1712 in Figure 2 may receive multiple routes to send out traffic. Meanwhile, they may as well receive traffic from multiple neighbors. Such diversity in route brings up two questions: 1) which routes are the best; 2) how to route corresponding traffic on the desired paths. The efforts spared in answering these two questions are referred to as *interdomain TE* [17, 21, 32].

The first question deals with the objective of TE. In general, a network aims at optimizing the cost and/or performance of transmission. The second question explores the method to steer traffic. We summarize the current practices, their limitations and challenges for incoming and outgoing traffic separately in this section.

2.2.1 *Outbound interdomain TE*

In sending traffic to a destination prefix, an AS has total control over the routes to be employed. One common practice is to tune *local preference* BGP attributes [52] before BGP route decision (Figure 1). Therefore the challenge is rather on the composition of best routes in terms of cost and performance.

The cost of interdomain transmission depends on the 95^{th} percentile bandwidth consumption on links purchased from transit providers, i.e. transit links [120]. Meanwhile, whether these transit links are congested impacts in return the transmission performance. Hence, outbound TE resolves in dynamically calculating the appropriate amount of traffic to be routed on each transit link. The objective of such traffic re-distribution is to lower the overall transit cost, under the constraint of not saturating any transit link at any instant (if possible).

Goldenberg et al. [24] formulated this quest as a minimum-cost multi-commodity flow problem. Uhlig and Bonaventure [28] fulfills the same goal while minimizing the number of route changes by predicting the traffic volume. Zhu et al. [96] avoids congestion on transit links by including border router queue length in route decision.

Performance-wise, it is clearly sub-optimal to greedily saturate the cheapest transit link while other transit links remain idle. However, there are other factors that may as well put transmission performance in danger. The minimum delay of Internet transmission is dominated by the physical length traversed by a route. However, neither AS path

length nor transit cost reveals/correlates to the underlying distance. On top of that, transient events like congestion can as well happen remotely [15, 93], independent of traffic load on transit links. To identify performance difference across multiple routes, end-to-end measurements are indispensable. They cumulatively reflect the contribution along the entire path, including the transit links.

2.2.2 *Inbound interdomain TE*

Inbound TE takes care of the incoming traffic distribution on available transit links. Through Figure 1 and 2, we learn that the paths that incoming traffic takes are decided by remote senders. For instance, AS3333 decides which routes to use to send traffic to `137.194.0.0/16` of AS1712. What AS1712 can do to influence the route decision of AS3333 is to tune its route advertisement. Shown in Figure 1, an AS can filter routes or change certain BGP attributes before advertisement. Some common practices are: selective announcement, more specific announcement, AS path prepending, setting Multi-exit Discriminator (MED) [52].

These approaches are not perfect. Selective announcement introduces reachability risks. More specific announcement gives rise to Internet Routing Information Base (RIB) inflation. AS path prepending is shown to be feeble in avoiding a specific ingress link [32]. BGP community [50, 106] and redistributed communities [14] allow finer grained operations with better certainty. However, it requires the support from transit providers.

Under BGP, it appears to be very difficult to have a fine control over the paths/ingress links of incoming traffic. In this context, many efforts were focused on steering mechanism for incoming traffic. Traffic ingress point can be dictated by setting the source address of outbound traffic to that of the desired ingress interface. Such address 'spoofing' can be achieved through encapsulation [51] or Network Address Translation (NAT) [107]. However, outgoing traffic could be dropped, for security considerations, when the it bears source addresses different from provider's address delegation[127]. LISP pushes such approach to a revolutionary level by introducing a separate addressing space in the core of Internet that enables various TE operations that are impossible with mere BGP [137]. Studies show fine-grained and dynamic inbound optimization is feasible with LISP [43, 45, 74]. Yet, the protocol deployment remains limited.

2.2.3 *Software Defined Networking and interdomain TE*

Recently, Software Defined Networking (SDN) brings as well new possibilities to interdomain TE. One idea is to delegate the TE tasks of an AS to a third party. The third party performs route decision and traffic

steering in a centralized manner, in accordance to SDN design philosophy. It is advocated that the interdomain TE can hence be done in a more cooperative way. Since conflicts of interest involving multiple ASes are solved centrally, an overall optimality can thus be achieved. Kotronis, Dimitropoulos, and Ager [78] advances that such AS clusters under same TE service provider can form and expand in a gradual way thanks to network effect. Gupta et al. [89] focuses on the application of SDN in a more specific network environment, IXP. The members of an IXP by nature forms a cluster of ASes. They exchange their routing information along with their TE polices within one centrally managed facility. Application specific peering, e.g. only peer for the exchange of video traffic, is made possible under this framework.

2.3 SCOPE OF THIS THESIS

We stage the works of this thesis under BGP, while fully realizing LISP, SDN, etc. are promising directions to pursuit. It is because BGP is still going to be the *de facto* routing protocol of Internet in the foreseeable future. And the deployment of any new routing mechanism must be incremental. Before the takeover of anything non-BGP, BGP is what a majority of ASes have to live with. There are thus immediate needs for improvements.

We focus on outbound TE in this thesis. Inbound TE has been shown to be inherently difficult with BGP, due to a lack of effective traffic steering method.

We target stub ASes (potentially multi-homed). It is because CP, HP and ISP, being major network types among stub ASes, are those who need most outbound TE. Moreover, dynamic route re-selection in those networks will not cast Internet-wide BGP route convergence issues.

Finally, we assume improving transmission performance is nowadays the major motivation for outbound TE. Routing traffic across Internet now faces fewer monetary constraints thanks to decreasing transit price [133, 148] and high IXP growth rate worldwide [134]. Instead, through demand for geographical and topological connection diversity [102], a performance challenge remains to be addressed.

2.4 MEASUREMENT-BASED TE AND MOTIVATIONS

BGP route decision mechanism is unaware of the performance characteristics of candidate paths [34], as shown in Figure 1. Yet, the transmission performance toward a destination varies over different interdomain routes. Akella, Seshan, and Shaikh [15] pointed out that bandwidth bottleneck can be within certain transit providers or on the links between remote ASes. This observation suggests that the choice of transit is of relevance to transmission performance. Further,

Akella et al. [16] revealed a 30% potential performance gain that an AS could achieve with multi-homing.

In order to actually realize this performance gain, dynamic route selection based on performance measurements is required, i.e. *measurement-based TE*. Akella et al. [49] presented a demo implementation of a measurement-based TE system. Only 100 destinations are emulated in this work. This number is far less from the actually scale that a stub AS might face on a daily base. In the work, the best route for each destination is chosen based on the EWMA of past Round-Trip Time (RTT) measurements. The results show that the best transmission performance over all destinations is achieved when route decision is made upon last single measurement. However, considering the noisy nature of RTT measurements, such a simplistic approach can lead to overwhelmingly frequent path changes. Moreover, treating Internet as a blackbox for delay measurements fails to provide useful and sometimes necessary insight into the underlying network events. These network events, e.g. path changes and congestion, are the actual causes for significant performance degradation and the reasons of route change.

In order to address the above concerns and narrow down the gap between the concept and a working system [124], we study in this thesis traffic volume, delay and path measurements to improve the scalability, measurement interpretation and performance visibility of measurement-based interdomain TE.

2.5 ROADMAP

2.5.1 *Building blocks of measurement-based TE*

A measurement-based interdomain TE platform has two essential building blocks. They are illustrated in black in Figure 3: (i) path performance measurement and (ii) route decision. The platform measures the end-to-end performance, more specifically Round-Trip Time (RTT), over all the available routes towards a given destination prefix. Within each destination prefix, a couple of hosts with open ports, e.g. 80, 443, are discovered and then used as probing destination in active delay measurements. Once fed with performance measurements, route decision engine dictates for each destination the best routes at each moment and imposes them on BGP border routers.

2.5.2 *Prefix selection: focus on most important destinations*

The above design faces a scalability issue. A stub AS can exchange with up to around 100k destination prefixes. Continuous performance monitoring to all these destinations over all available paths could be prohibitively costly. Feamster, Borkenhagen, and Rexford [17] already

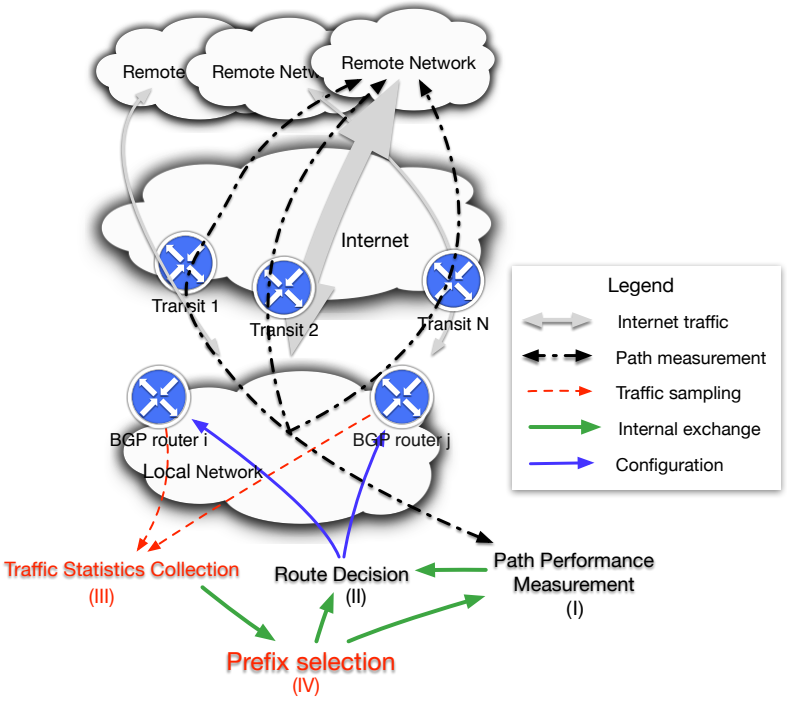


Figure 3: Building blocks of measurement-based inter-domain TE system.

realized this issue and proposed to focus on popular destinations. However, no exact solution was given. In Chapter 3, we tackle the selection of prefixes associated with important traffic volume. Through study on their temporal dynamism at different time resolutions, we arrive at very simple yet efficient mechanisms. As a part of the proposition, two additional building blocks are added to the system architecture (highlighted in red in Figure 3).

2.5.3 *RTT measurements with RIPE Atlas*

Studies in Chapter 3 base solely on the measurement data collected by a proprietary TE platform [124] from client networks. Working trace from real networks increases the credibility of our observation. However, it brings as well reproducibility concerns. In Chapter 4, we first justify our choice of using RIPE Atlas [142] as source for path and delay measurements in later researches. We then discuss a data quality issue originated from the measurement platform.

Further, we investigated another data quality issue that is specific to interdomain TE. We employ unsupervised learning methods to reveal the inherent structure of a group of delay measurements on a same AS path.

During the above study, we noticed an interesting case where several RTT time series exclusively share a similar shape at about the same moment. We thus find it promising to infer the actually location of shared RTT changes by grouping RTT time series of similar shapes. To that end, we study the application of time series clustering methods to RTT measurements and discuss their limitations.

2.5.4 *Change detection for RTT measurements*

Difficulties in grouping RTT time series with similar shapes leads to further studies in Chapter 5 and 6. Chapter 5 studies the application of *changepoint analysis* methods to RTT measurements. These methods aim at detecting significant changes in time series. We regard the resulted change moment as an expressive way to simplify the representation of RTT time series. Meantime, we realize that the detected moments of change can serve as an informative and robust trigger for route re-selection.

To quantify the change detection performance on RTT measurements, we build an evaluation framework and benchmark several candidate methods. The temporal correlation between RTT changes and routing events are as well studied and illustrated.

2.5.5 *Inferring the location of RTT changes*

With simplified data representation enabled by changepoint detection, we try to infer the location of detected RTT changes in Chapter 6. Knowing the location of RTT changes are of significance in measurement-based interdomain TE. It is especially useful when we hope to optimize the routing to destinations that we are not able to measure directly. Such network visibility allows avoiding certain problematic paths when end-to-end measurements are absent.

For that purpose, we first group RTT time series undergoing same RTT changes with the help of changepoint analysis. We come up with a series of inference logic to attribute RTT change to ASes and inter-AS links. Visualization tools are provided to illustrate the inference process and the inferred locations of change on an AS-level topology.

3

SCALABLE PREFIX SELECTION

ABSTRACT

The growing size of global Internet Routing Information Base (RIB) seems to pose a challenge to measurement-based interdomain Traffic Engineering (TE). However not all destination prefixes are important to a client network at a given moment. Generally, merely 0.1% ~ 1% of them are used in forwarding each hour. Moreover, some of them are responsible for much more traffic than the rest. A natural consequence is thus to perform TE only for those prefixes that matter. However, traffic volume associated to a prefix varies over time. We have little knowledge on the characteristics of the traffic variations across Border Gateway Protocol (BGP) prefixes. On top of that, it is not trivial identifying, in a predictive manner, prefixes of significance among the crowd. Resource available for traffic volume processing is in general limited. The calculation is normally done on commodity hardware dedicated for TE usages out of cost considerations. In this context, sophisticated methods predicting volume for each single prefix will not scale.

We revealed in this chapter the relationships among prefix volume importance, stability and predictability using working traffic traces from 9 networks of various profiles. With these findings, we proposed three resource-efficient metrics to predictively select prefixes of important volume. The proposed metrics yielded both satisfying volume coverage and pretty low prefix churn. Furthermore, we showcased that the performance in terms of RTT could differ a lot among different transit providers, which calls for fine-grained dynamic route selection mechanism to drain this gain. We simulated a route selection algorithm and fed it with RTT measurements from real networks. On certain networks, the algorithm achieved a 20% performance gain compared to the usage of a single best transit provider.

3.1 PREFIX SELECTION: A PROBLEM OF SCALABILITY

A client network in need of measurement-based interdomain Traffic Engineering (TE) are often of type Internet Service Provider (ISP), Hosting Provider (HP), Content Provider (CP). It sends out traffic to a wide range of destinations, from several 10k to 100k BGP prefixes. The measurement and route decision sub-system illustrated in Figure 3 thus faces a scalability issue tracking and optimizing in real time the transmission performance to these destinations. However, it is well-known that most traffic volume-wide is generally concentrated on only a fraction of the BGP prefixes [7, 17, 31, 79]. It is thus possible and reasonable to focus only on those important destination prefixes in measurement and route re-selection.

Confine measurement-based TE to important prefixes requires predicting which prefixes will correspond to the most important traffic volumes in the near future. To this end, two additional function blocks are added to the system design in Figure 3): (iii) traffic volume statistics collection; (iv) prefix selection process, which selects the set of the most important prefixes (i.e., the ones with the highest volume in the foreseeable future) and communicates the selected prefix set to measurement and route decision function blocks.

Two reasons oblige us to *predictively* select prefixes of important volume and devise specific mechanisms for that task. First, traffic volume per prefix evolves over time, so does the set of prefixes representing important traffic volume. Walleriche et al. [39] showed that the bandwidth ranking of a 5-tuple flow can change drastically from one moment to another. In order to maintain a set of prefixes of importance, one thus has to predict traffic volume for each prefix repeatedly. To our best knowledge, no study has given an in-depth investigation on the evolution in time of the traffic volume associated with BGP prefixes.

Second, in predicting traffic volume for each individual prefix, more efficient methods are needed. Well established Time Series Forecasting (TSF) models and Artificial Neural Networks (ANN) have been used previously in traffic prediction [31, 36, 85]. These works targeted on highly aggregated inter-Point of Presence (PoP) traffic for off-line tasks, such as network dimensioning. These models are not only computationally heavy, but also require data pre-processing and parameter tuning on a per trace base. These overheads make those methods less applicable in the context of inter-domain TE that involves upto some 100k prefixes. Therefore, less complex prediction methods are needed.

The rest of this chapter is organized as follows:

- Section 3.2 reviews some related works on Internet traffic dynamism, Forwarding Information Base (FIB) caching and the performance gain of interdomain TE.

- Section 3.3 mainly studies the temporal dynamism of Internet traffic at BGP prefix scale with working traffic traces from networks of diverse profiles (ISP, HP and CP) located in different countries (France, Germany, Poland, Spain, UK and USA). It quantitatively describes the burstiness of the traffic from these networks.

- Section 3.4 proposes three metrics for the prediction of important prefixes based on the findings established in Section 3.3. These methods are evaluated and compared to Grey model employed in one previous work on FIB caching. Results show that the proposed schemes out-perform the Grey model in terms of volume coverage and prefix churn.

- Finally, the end-to-end path performance through each available transit provider are measured from the nine client networks toward their own selected prefixes of volume importance. Through transit provider performance evaluation, the significant differences among transit providers reported in former studies are confirmed. This highlights the potential gain of measurement-based interdomain TE.

3.2 RELATED WORK

Some previous works [17, 24, 49] acknowledged the importance of performing inter-domain TE only for important destinations. However, no general solution was given to predictively select the BGP prefixes representing most traffic volume.

Some other works leveraged the skewed distribution of Internet traffic to downsize FIB. They aim at installing only a small part of the Internet routes in forwarding table, out of some software or hardware limitations [43, 54, 58, 79, 80, 104]. In these works, traffic dynamism on much smaller time scales, e.g. seconds and minutes, is of relevance. In our work, we used traffic traces of coarser time resolution over longer period of time. Such configuration adapts better to the complicated operations involved in inter-domain TE, especially path performance measurements. Nonetheless, we have compared as well our prefix selection methods to these works in the hope that our study could be of value for their problem as well.

When it comes to the understanding on traffic dynamism, Zhang et al. [80] assumed that the stability and popularity of Internet traffic is positively correlated without verification. Papagiannaki et al. [25] showed that this correlation is not evident for 5-tuple flows on 5 minute interval. Our work investigated this relationship for BGP prefixes using working traffic traces from diverse real networks.

Regarding the performance benefit of actually taking advantage of multiple paths in multi-homing, Akella et al. [16] quantified the per-

	SA	SB	SC	SD	SE	SF	SG	SH	SI
Type	CP	ISP	HP	HP	CP	CP	ISP	CP	HP
Vol.	133	528	6.7	1129	1871	5.1	0.2	29.9	6.2

Table 1: Average traffic volume per hour (in GB) for the different measured networks

formance gain using traces from a large CDN network in 2003. In a latter work [49] in 2008, they evaluated a dynamic route selection system on a testbed, but with only 100 destinations. We evaluated this gain by performing delay measurements from real client networks toward important prefixes predictively selected with approaches proposed in this article.

3.3 CHARACTERS OF INTERNET TRAFFIC OVER BGP PREFIXES

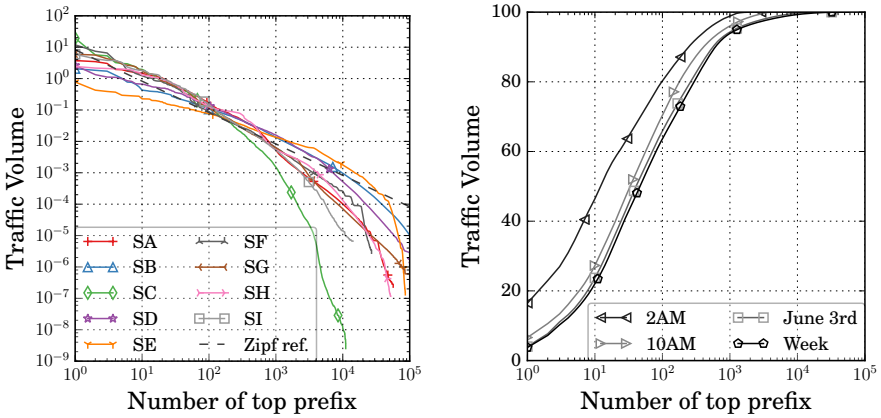
We base our study on working traffic traces collected from 9 client networks of very different profiles listed in Table 1. They are either CP, HP or ISP. Traffic traces covers a time period of two entire weeks, from May 25th, 2015 to June 8th, 2015, from all networks except SB and SD. On SB and SD, the data only covers the second week (starting from June 1st, 2015). These traces were sampled from real traffic, similarly to previous works concerning FIB caching [58, 80]. It has been shown that the bias introduced by sampling is negligible, especially when we focus on large volume prefixes.

Traffic entries toward each BGP prefix are first bucketed in 1 hour bins. The traffic volume in each bin are then summed up. In comparison, studies concerning FIB caching [79, 80] use shorter time bins, from 1 second to 10 minutes, to capture instant variations of traffic characters. In our case, we argue that 1 hour is an appropriate update interval for prefix selection. It is because each prefix selection round is followed by time consuming operations, e.g. finding reliable hosts that can be actively probed within a newly selected prefix.

3.3.1 *Traffic distribution over BGP prefixes*

As outlined earlier, the feasibility of prefix selection is based on the assumption that most traffic concentrates on a few popular destinations. Some previous works have shown this property with their own datasets [7, 17, 39]. We demonstrate here that our dataset as well has this phenomenon of uneven traffic distribution across destination prefixes.

In Figure 4a, the volume share (percentage in unit, on the y-axis) of each BGP prefix is plotted for the week starting from June 1st, 2015. Prefixes are decreasingly sorted along the X-axis according to their



(a) Week volume share, log-log. (b) CDF, different time spans, SA.

Figure 4: Traffic distribution among BGP prefixes.

cumulative volume fraction over the week. The X-axis labels indicate rankings of prefixes. We observe that the week volume associated with BGP prefixes can be approximately described by a reference Zipf's distribution with $N = 10^5, s = 1$ (dashed line).[1] This figure shows that Internet traffic (within our dataset) is indeed highly concentrated on a few prefixes.

Figure 4b compares the traffic distribution (in CDF) of SA at different time resolutions: prefix volumes within one hour time (at 2AM and 10AM), traffic accumulated over 24 hours (on June 3rd) and the traffic throughout the full week. The uneven traffic distribution is not unique on week time scales (Figure 4a), but as well demonstrates over shorter time ranges. Moreover, the level of traffic concentration over BGP prefixes actually varies within a day. The 1^{st} ranking prefix at 2AM represents almost 20% of all traffic, while at other time or time spans, this ratio is much lower. This observation leads to the study in the following section. This change in time leads to the study in the following section.

3.3.2 *Temporal dynamism of traffic over BGP prefixes*

We are interested in understanding how traffic volume evolves over time, and how this dynamism is reflected in uneven traffic distribution.

3.3.2.1 *Coefficient of Variation*

In this section we mainly use Coefficient of Variation (c_v) to characterize the traffic volume volatility over time for each BGP prefix. The

1 Zipf's law defines that the k^{th} most popular element among total N elements has an occurrence share of $f(k, s, N) = \frac{1/k^s}{\sum_{n=1}^{N} 1/n^s}$.

study uses the data over one entire week starting from June 1st, 2015. In order to facilitate the discussion, we first introduce some notations.

Each destination prefix P ever active during a week is associated with a volume time series $v(P) = \{v(P)_h\}_{h=1,\ldots,168}$ that stores its traffic volume over the week at hour interval. We calculate the Coefficient of Variation (c_v) for prefix P according to

$$c_v(P) = \frac{\delta(v(P))}{\mu(v(P))},$$

where δ stands for the calculation of standard deviation and μ for mean.

c_v can be regarded as a measure of traffic volume variation in relation to its hourly mean over a week. A larger $c_v(P)$ value suggests that the $v(P)$ tend to take values in a bigger range that is normalized by its average level. Consequently it becomes more difficult to anticipate the traffic volumes for this prefix P [29].[2]

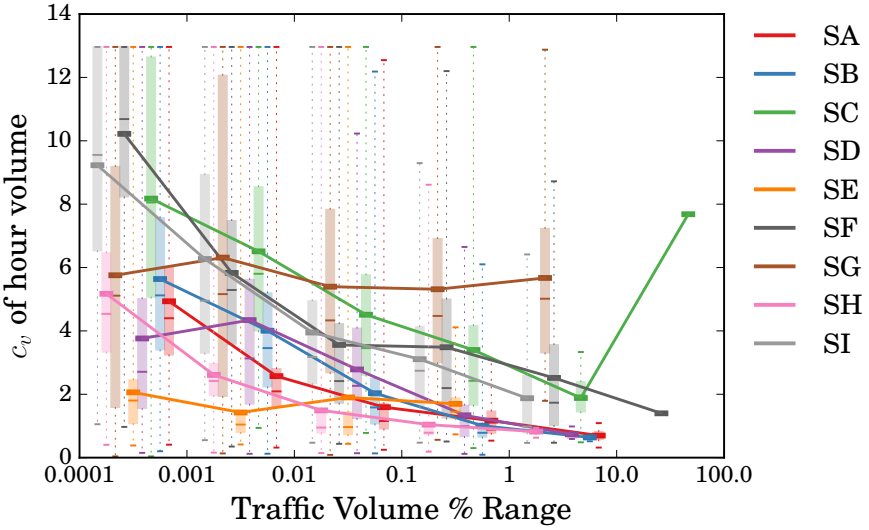


Figure 5: Relation between $\{c_v(P)\}_P$ and week volume share for all BGP prefixes P.

Further, we are interested in knowing how the c_v value is related to prefixes of different volume importance. For example, are prefix with big volumes tend to be more stable or volatile over time? To answer such question, we visualize the relationship between c_v and prefix volume in Figure 5. Each prefix is sorted along the X-axis according to its accumulated volume share over the week. Prefixes with large volume share are located on the right side. To compact the visual representation, we further group all the prefix into six bins, representing

2 By construction, the maximum c_v for a hourly volume series of 168 in length is $\sqrt{167}$, corresponding to the case where the prefix in question is active during only one single hour throughout the entire week.

different discrete level of volume importance. These bins can be identified with the X-axis labels. They are $[10^{-4}, 10^{-3})$, $[10^{-3}, 10^{-2})$, all the way to $[10, 100)$. For all the prefixes within in same bin, we summarize their c_v using a vertical boxplot. The two ends of the box represent 25^th and 75^th percentile of c_v values of prefixes within the bin. The thin line in the middle stands for median, while the thick one for mean. The whiskers are min and max separately. In order to outline the relationship between c_v and prefix volume importance, the mean value of the six bins are connected with a thick line.

From this Figure, we observe that for all the networks except SG and SE, c_v of large volume prefixes tends to be smaller in average and constrained in a narrower box. On the contrary, the prefixes with smaller week volume share tend to have larger c_v values. It allows us to capture a significant part of the overall traffic (represented by those stable and large prefixes) by simply picking the prefixes with large average hour volume.

3.3.2.2 *Core presence intensity*

We continue to explore traffic dynamism from another perspective: for each hour h, we define the $core_h$ as the prefix set containing top ranking prefixes that represent 95% of total traffic. Imagine that we were to identify prefixes representing that much traffic, then $core_h$ will be the smallest set that we could arrive at. One essential feature of the *core* set is its size. It gives a rough picture on how many prefixes that a network might want to consider in measurement-based TE. Table 2 lists 1) the average number of prefixes included in the *core* (*core* size can change over time), 2) the average *core* size percentage with regard to the average number of active prefixes each hour and 3) the maximum size of the *core* set over the week. For a big part of the networks in our dataset, *core* set only represents a small fraction of active prefixes. This observation is in line with the uneven traffic distribution demonstrated earlier.

Name	Avg. prefix #	% w.r.t active prefix	Max prefix #
SA	629	17.87	1051
SB	5264	9.45	13934
SC	73	4.59	177
SD	2481	17.35	3757
SE	15501	53.61	20900
SF	377	30.73	772
SG	570	7.76	1766
SH	965	19.42	1731
SI	175	21.00	415

Table 2: Core prefix set statistics.

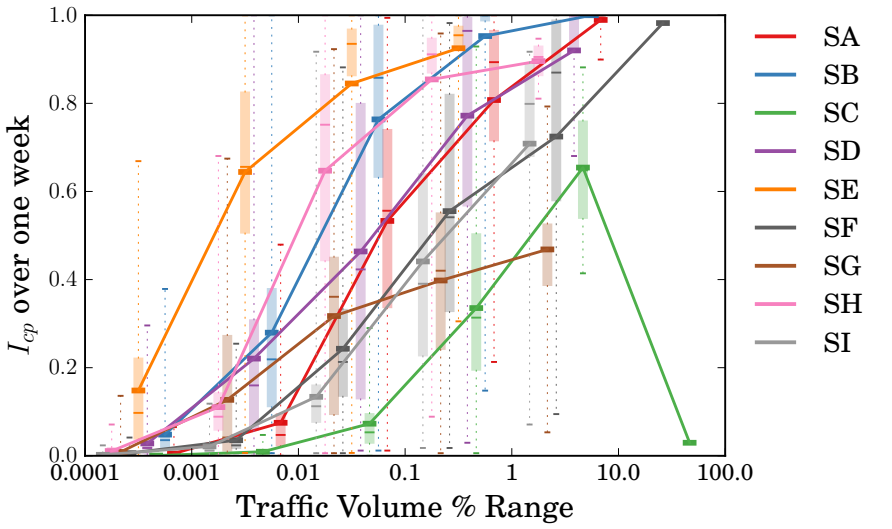


Figure 6: Relation between I_{cp} over the week and week volume fraction of BGP prefixes.

If a prefix is inside the *core* at a certain hour, it can be regarded important for bringing a significant amount of traffic. We thus define the "*core* presence" for a prefix P at each hour h as:

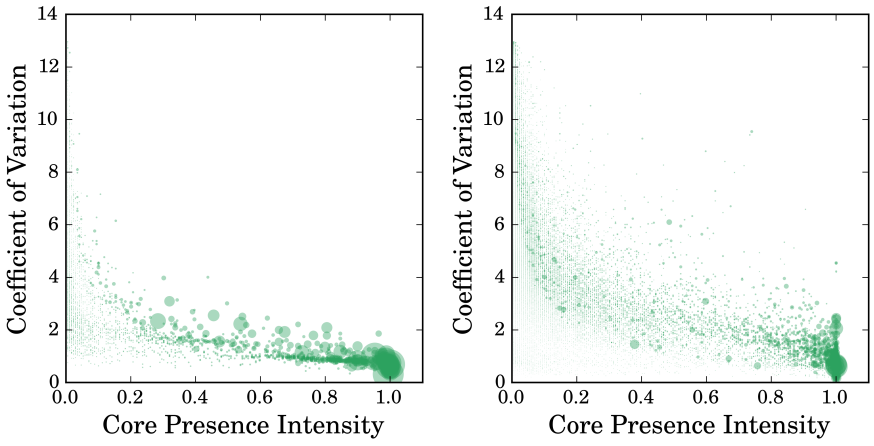
$$cp(P)_i = \begin{cases} 1 & \text{when } P \in core_i, \\ 0 & \text{otherwise,} \end{cases}$$

The *core* presence intensity $I_{cp}(P)$ is then defined as the frequency of *core* presence over the week: $I_{cp}(P, 168) = \frac{1}{168} \sum_{i=1}^{168} cp(P)_i$. Intuitively, I_{cp} can be regarded as an indicator of fitness that a prefix shall be continuously monitored in measurement-based TE. Again, we are interested in knowing how the value of I_{cp} relates to the overall traffic volume importance. Similarly, we visualized this relationship in Figure 6, using the same visual language employed in Figure 5.

For all the networks, we can see that prefixes with bigger week volume share are more likely to have a high I_{cp} over the week, i.e. they appear frequently in the *core*. We can conclude that by focusing on prefixes that intensively appear in the *core* throughout the week, we will be able to capture a large part of the prefixes associated with important traffic volume over the week.

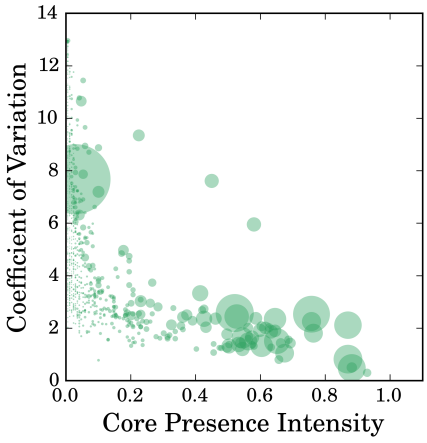
3.3.2.3 *Relationship between* I_{cp} *and* c_v

For the sake of comparison, we study as well the correlation between these two metrics: c_v and I_{cp} in Figure 7. In the figure, each circle represents a prefix and its radius is proportional to the prefix's week volume share. The biggest circles mostly concentrate in the lower right corner (small c_v and large I_{cp}) of each sub-graph, which corresponds to the remarks made previously. However, some exceptions

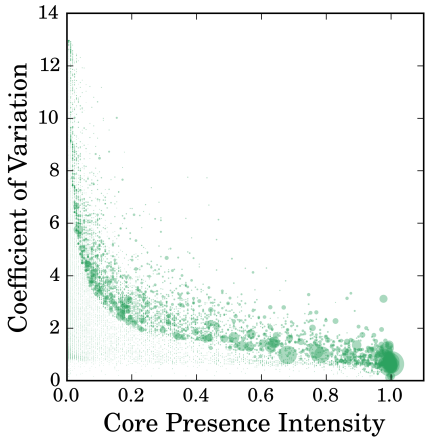


(a) SA, 56684 prefixes

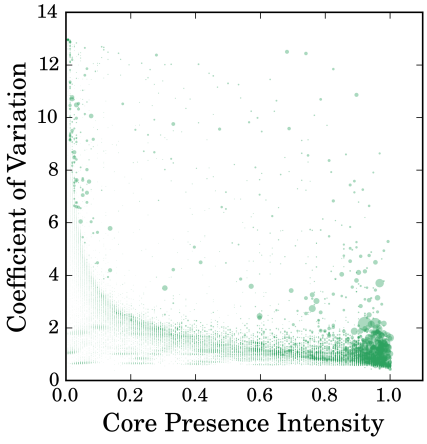
(b) SB, 504707 prefixes



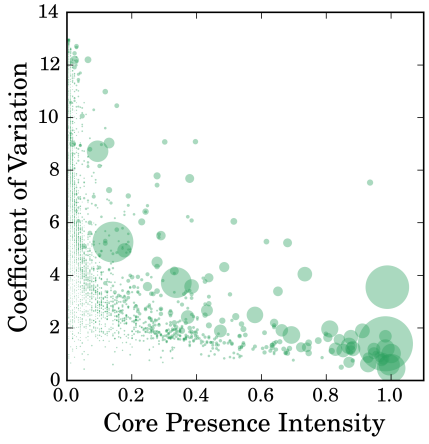
(c) SC, 11065 prefixes



(d) SD, 140700 prefixes

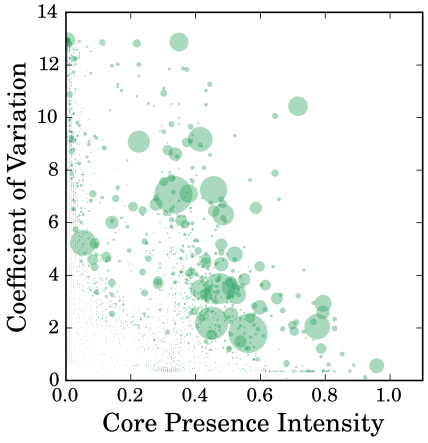


(e) SE, 86014 prefixes

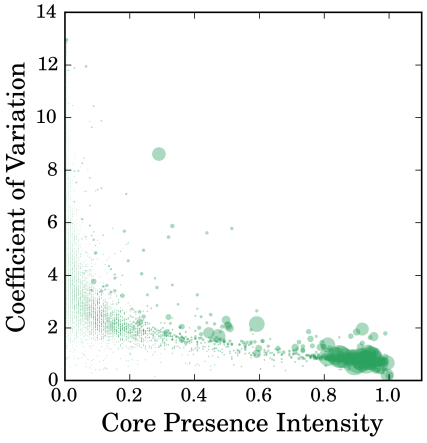


(f) SF, 26907 prefixes

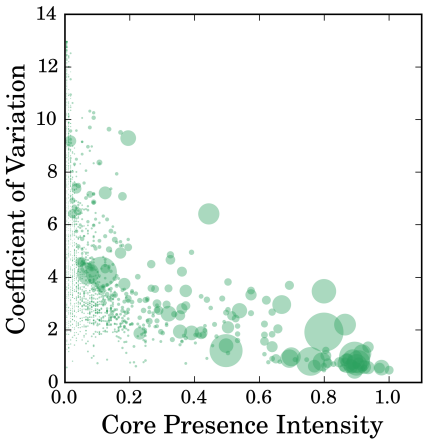
Figure 7: Relation between I_{cp} over one week and c_v of hour volume over the week from June 1st, 2015. Each circle stands for a prefix. Number of active prefixes plotted is each sub-graph throughout the week is also given. Circle size is proportional to the week volume fraction of the prefix the circle represents and is of the same scale for all networks.



(g) SG, 104425 prefixes



(h) SH, 52061 prefixes



(i) SI, 14189 prefixes

Figure 7: (cont.) Relation between I_{cp} over one week and c_v of hour volume over the week from June 1st, 2015.

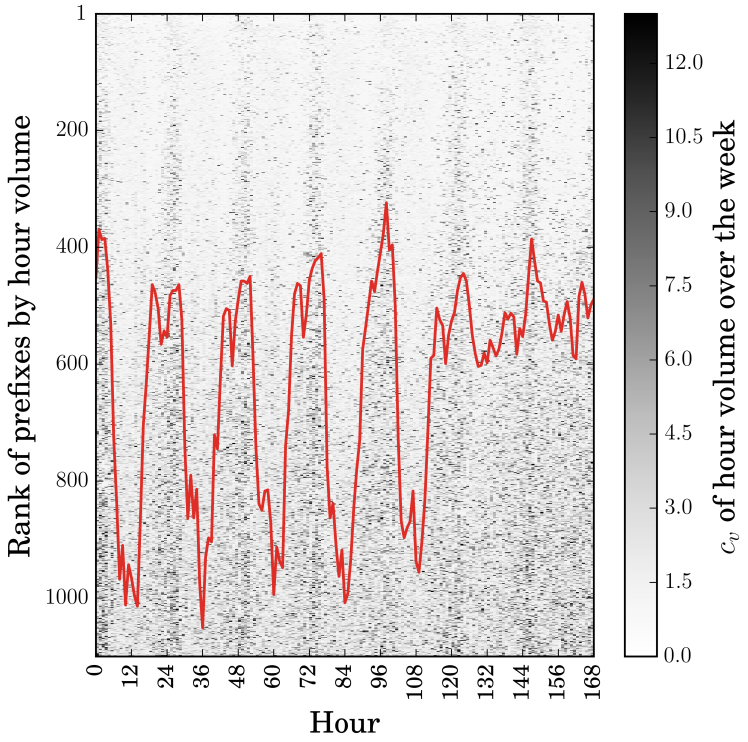
exist especially on SC (but also SF, SG and SI). On corresponding sub-graphs, we notice big circles having low I_{cp} and relatively high c_v. These prefixes bring significant week volume share within a short duration, which makes predictive prefix selection difficult. This observation leads the discussion on traffic burstiness later on. Finally, it's not a surprise to see that the c_v of prefixes with big week volume is, to a certain extent, inversely correlated to their I_{cp}.

3.3.2.4 *A view at hour interval*

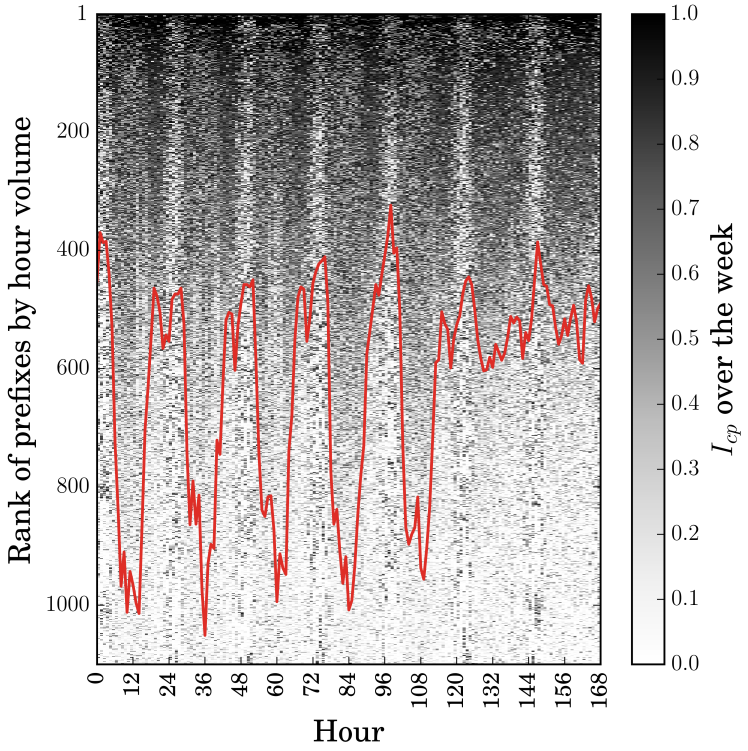
The above analyses at week scale conclude that most prefixes that are associated with a big week volume tend to be stable in hourly variation and present often in *core*. Here, we explore the prefix volume dynamism at hour resolution by visualizing each prefixes weekly c_v and I_{cp} in Figure 8. In the figure, each column (every hour in the week has it column) corresponds to top prefixes ranked by their **hour volumes**. Large volume prefixes are at the top of the graph with small ranks. Within the column, a grey scale tile is assigned to each prefix to portrait its c_v or I_{cp} over the week. Prefixes above the red line are those composing *core* at each hour. Only graphs for SA are shown. They presents most informative patterns, since less bursty traffic is from SA. We notice that the *core* size varies regularly on a daily base. During peak hours, the *core* size can be twice its minimum value. This diurnal pattern is another illustration of different traffic concentration level at different moments, a phenomenon first revealed with Figure 4b. Moreover, the *core* size seems to show a different pattern then the rest of the days. Since these two days are actually a weekend, this change suggests probably a potential hebdomadal traffic cycle.

In the sub-graph for c_v (Figure 8a), the area above the red line has observably lighter tone than the lower part. This implies that in each hour, prefixes in the *core* have more stable hour volume variation, i.e. closer to its mean value, than those outsiders with little volume significance. Moreover, the figure demonstrates as well a time-of-day pattern for c_v values. During late night and early morning, prefixes in the *core* have deeper color, thus more volatile, than those in the day time. This indicates that important prefixes composition are not quite the same during these two periods. The above described phenomenon are sometimes a little bit more difficult to perceive on networks with more bursty traffic. Anyhow, for those 'bursty' networks, the tone of the *core* area is not visibly darker. It means the hourly volume time series for large prefixes are not obviously less stable. We should all the same be able to pick them out using their mean value. The lack of clear diurnal pattern for some networks is possibly related to their business type and client activity, which is out of the scope of this work.

When it comes to I_{cp} (Figure 8b), it is true for all networks that top ranked prefixes each hour are more likely to frequently appear



(a) c_v



(b) I_{cp}

Figure 8: c_v and I_{cp} over one week for top ranked prefixes at each hour on SA. Prefixes are ranked by their hour volume along the column (big prefixes at the top). Their c_v or I_{cp} over the week are represented by the grey scale. Red line indicates the number of prefixes in *core* prefix set each hour.

in the *core* prefix set. That is deeper color at the top of the graph. However, we can witness light spots each hour in the upper part of the graph, which corresponds to bursty traffic. Time-of-day pattern from I_{cp} values can also be clearly observed. It confirms again that prefixes active over night are different from those during the day time.

3.3.3 *Quantitative index of traffic burstiness*

Previous explorations have shown that the predictability of traffic volume from a client network is related to its traffic burstiness. In order to describe burstiness in a quantitative manner, we define the index $\beta(P)_h$, for a prefix P at certain hour h as:

$$\beta(P)_i = \begin{cases} -\log(I_{cp}(P)) \times vp(P)_i & \text{if } I_{cp}(P) > 0 \\ 0 & \text{if } I_{cp}(P) = 0, \end{cases}$$

where $vp(P)_h$ is the hour volume percentage (among all active prefixes) of prefix P at hour h. The logarithmic term applied on I_{cp} aims at amplifying the volume contribution of prefixes with rare *core* presence, and attenuating the influence of prefixes being intensively in the *core*. A large β value indicates that it is harder to predict the volume associated with this prefix while representing a significant hour volume.

In order to estimate the overall burstiness of all prefixes at hour h for a network, we sum up the $\beta(P)_h$ for each P inside the *core* of that hour: more formally,

$$BI_h = \sum_{P \in core_h} \beta(P)_h.$$

For all the networks, we estimate their traffic burstiness with the mean and maximum of BI value series over the week. The results are given in Table 3. The maximum β, contribution from a single prefix, throughout the week is also given.

Mean BI over the week measures the general burstiness of traffic, while maximum BI describes the degree of burstiness in worst cases. In accordance to the observation made from Figure 7, there are big volume prefixes with fairly low I_{cp} on SC, SF, SG and SI, whence the much bigger maximum β value. What is less evident in Figure 7 is that SD suffers actually a lot from bursty traffic, even more than SC on average. For the rest networks, i.e. SA, SB, SE and SH, their mean BI over the week is around 30 or lower. Their corresponding subgraphs in Figure 7 manifest as well much less big circles on the left side where I_{cp} is low. More specifically, SB suffers more from bursty traffic than SA, therefore bigger value in both maximum and mean BI over the week. This is however not easy to tell directly from Figure 7. Nonetheless, the maximum β on SA is larger than that on SB, which is due to the fact that traffic on SB is more evenly distributed among

Network	Mean BI	Max BI	Max β
SA	14.61	37.79	7.44
SB	31.09	46.85	4.08
SC	40.57	145.07	145.05
SD	42.14	69.34	18.10
SE	20.91	44.30	20.17
SF	44.10	98.69	78.77
SG	51.21	125.41	102.05
SH	15.91	35.06	16.29
SI	38.59	85.47	56.56

Table 3: Traffic burstiness.

active prefixes (see Figure 4). Therefore the faction of traffic associated to each prefix is generally smaller. In short, we found this simple metric capable of describing the traffic burstiness of the networks studied.

3.4 PREDICTIVE PREFIX SELECTION

3.4.1 *Candidate prediction metrics*

Based on the previous observations, several approaches naturally emerges for the prediction of traffic volume "importance" of a prefix.

MEAN VOLUME With this metric, we predict that at hour $h+1$, the volume importance of prefix P is indicated by its mean hourly volume over the last L hours, $MV(P,L)_{h+1} = 1/L \times \sum_{i=h-L+1}^{h} v(P)_i$. It is based on the observations from Figure 5, that top prefixes over the week tend to have smaller hourly volume variation around their mean volume.

CORE PRESENCE INTENSITY The prediction could use $I_{cp}(P,L)_{h+1} = 1/L \times \sum_{i=h-L+1}^{h} cp(P)_i$, i.e. the *core* presence intensity of the prefix over the last L hours. It derives from the observation from Figure 6, that top prefixes by their week volume are more likely to have intense *core* presence.

CORE VOLUME Finally, the prediction could be based on $CV(P,L)_{h+1} = 1/L \times \sum_{i=h-L+1}^{h} cp(P)_i \times v(P)_i$, a combination of MV and I_{cp}. CV has the potential to be more resource thrifty compared to MV, as it is calculated only for those prefixes ever appeared in the *core* over the last L hours — while MV is computed for all active prefixes. According to Table 2, the *core* size each hour is only about 5% to 50% of all active prefixes.

3.4.2 *Grey model as reference method*

In previous work by Zhange et al. [80] on FIB caching, a grey differential model GM(1,1) [4] is employed to predict which BGP prefixes will represent the biggest packet counts. It is by far more computationally efficient (O(L)), compared to TSF methods such as ANN (O(L ∗ M)) and Autoregressive integrated moving average (ARIMA) ($O(L^2)$), where L is the length of the time series, M is the number of hidden nodes in the neural network. For the sake of comparison, we implemented the GM(1,1) model to predict big volume BGP prefixes.

A brief introduction to this model and how we apply it in our context is given below. Mathematical details of this model can be found in the work by Deng [4]. Instead of hour volume, GM(1,1) predicts the cumulative hour volume v^1 :

$$v^1(P,L)_i = \sum_{j=i-L+1}^{i} v(P)_j,$$

where $v^1(P,L)_i$ is the cumulative hour volume of prefix P over last L hours at hour i. The purpose is to derive $v(P)_{i+1}$, i.e. the volume in the following hour, from estimations of cumulative volumes of hour $i+1$ and i:

$$\hat{v}(P)_{i+1} = \hat{v}^1(P,L)_{i+1} - \hat{v}^1(P,L)_i,$$

where $\hat{v}$ and $\hat{v}^1$ are all estimation values given by the model. GM(1,1) predicts that the cumulative hour volume at hour $i+1$ equals:

$$\hat{v}^1(P,L)_{i+1} = (v(P)_{i-L+1} - \frac{b}{a})e^{-aL} + \frac{b}{a},$$

where a and b are parameters that can be estimated with least square method (symbol with hat are all estimations).

$$\hat{\mathbf{a}} = \begin{bmatrix} \hat{a} \\ \hat{b} \end{bmatrix} = (\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\mathbf{Y},$$

where

$$\mathbf{B} = \begin{bmatrix} -0.5(v^1(P,L)_{i-L+2} + v^1(P,L)_{i-L+1}) & 1 \\ -0.5(v^1(P,L)_{i-L+3} + v^1(P,L)_{i-L+2}) & 1 \\ \cdots & \cdots \\ -0.5(v^1(P,L)_i + v^1(P,L)_{i-1}) & 1 \end{bmatrix},$$

$$\mathbf{Y} = \begin{bmatrix} v(P)_{i-L+2} \\ v(P)_{i-L+3} \\ \cdots \\ v(P)_i \end{bmatrix}.$$

We can see that for each single prefix, two parameters are to be estimated, a and b, at each hour based on a new volume series that slides over time. Computationally, estimation with $GM(1,1)$ is much heavier than the three metrics proposed above.

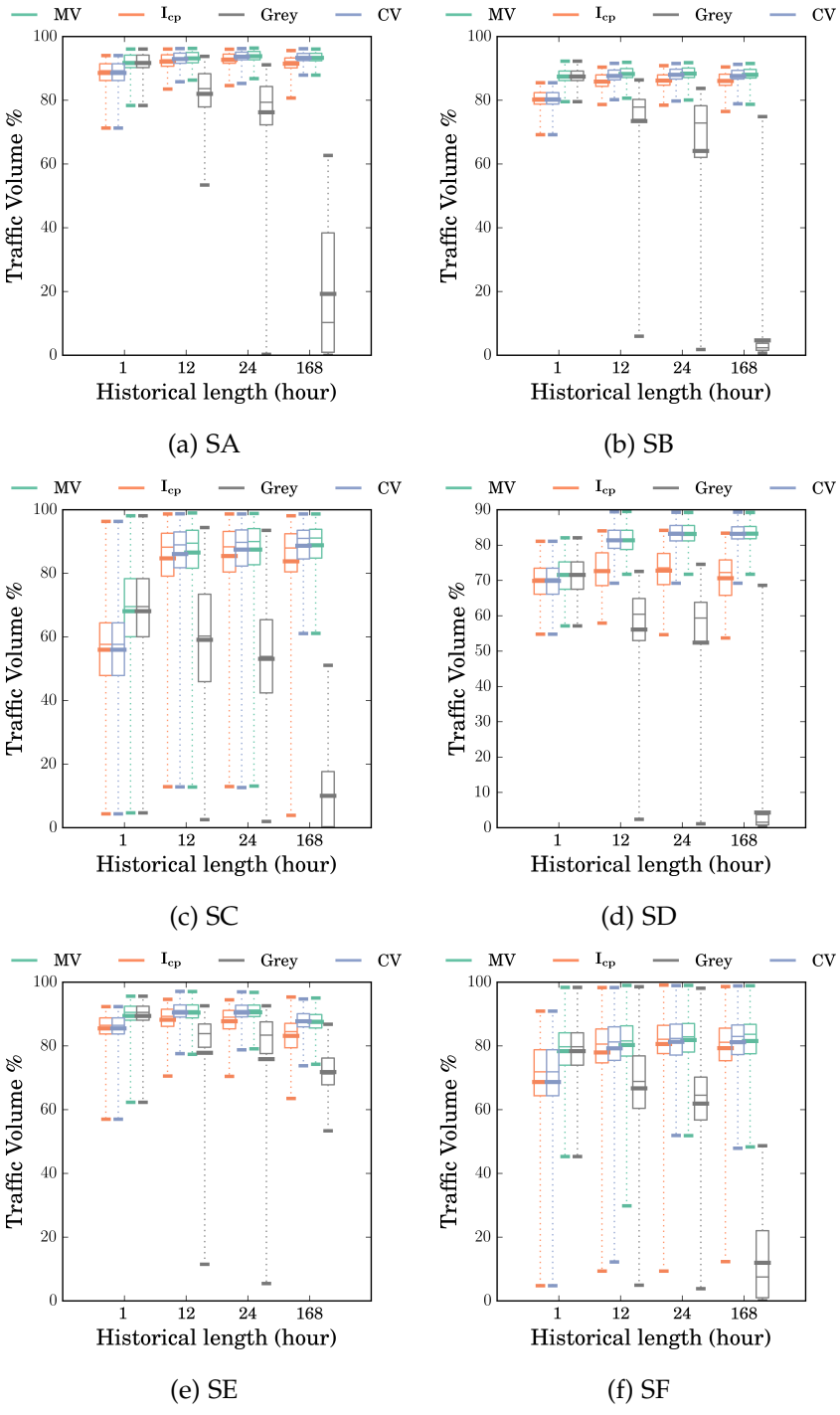
3.4.3 *Prefix selection evaluation*

In evaluating and comparing the performance of these methods, we fixed the selection set size to the maximum *core* size over the week. Figure 9 illustrates the hour volume coverage by the four methods in the form of box-plot, representing the minimum, maximum, 25th and 75th percentile, medium and mean values (thicker bar in the middle of the box). Among proposed metrics, we find that the CV is very close to MV in terms of volume coverage, proving that it is a good approximation of the later.

Basing solely on the last 1 hour records, all methods yield already a mean volume coverage $> 80\%$ on SA, SB, SE and SH, which implies a strong continuity in prefix volumes between two consecutive hours. On SC and SG, however, using records of last 168 hours, i.e. a week, offers much better minimum volume coverage than shorter records. This is due to the fact that at certain hour, SC and SG undergo a great amount of bursty traffic (as observed previously, e.g. in Table 3 and Figure 7). By increasing the historical length, the selection metrics are able to have better visibility into the past and capture some of these bursty prefixes — finally improving the minimum coverage. The gain in minimum volume coverage by using long historical records can actually be observed on all networks, between 1 hour and 12 hours, also between 12 hour and 24 hour. However from 24 hour to 168, this gain doesn't necessary happen on all sites, which is due to the fact that the total volume brought by some highly bursty prefix is diluted by the long time span using MV and CV metrics. In order to capture them, a larger selection set size is needed, which inevitably includes more prefixes of few significance.

On the other hand, the hour volume coverage by grey model drops as we increase the historical records length and is in general worse than the metrics proposed in this work. In order to understand the underlying reason, we used $GM(1,1)$ model described above to dynamically predict the total hour volume of all prefixes, which is normally much more regular and smoother than the volume series of individual prefixes, Figure 10.

For site SB and SD, we miss the hour volume data for the week starting from May 25th. $GM(1,1)$ model using records of last 168 suffers a lot from data missing and converges extremely slowly to actual traffic volume. However, such sudden change in value is not unexpected, since bursty prefixes can bring huge amount of traffic within a short duration and then remain silent over days. Such prefixes are

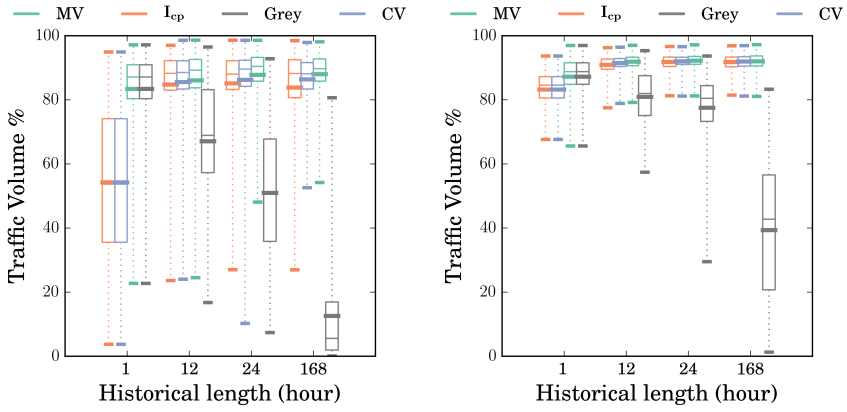


(a) SA (b) SB

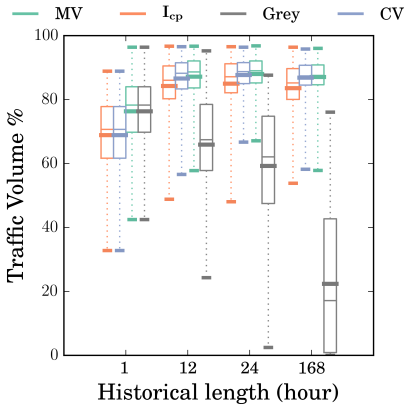
(c) SC (d) SD

(e) SE (f) SF

Figure 9: Hour volume fraction covered by prefixes predictively selected using historical records of different lengths. The selection set size of each network is set to the maximum *core* size over the week starting from June 1st, 2015, see in Table 2.

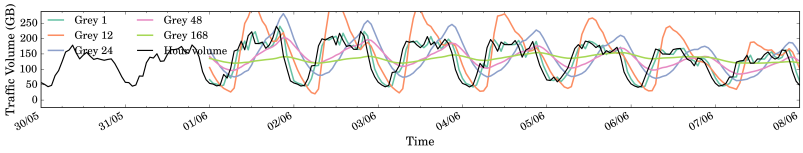


(g) SG (h) SH

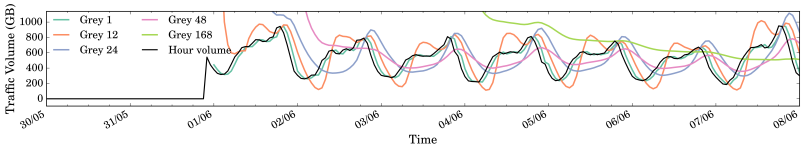


(i) SI

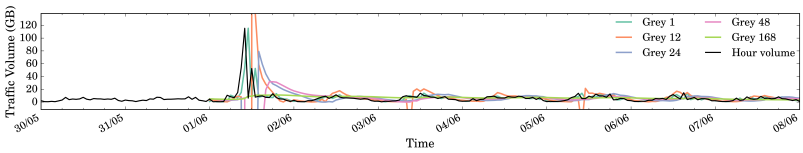
Figure 9: (cont.) Hour volume fraction covered by prefixes predictively selected using historical records of different lengths.



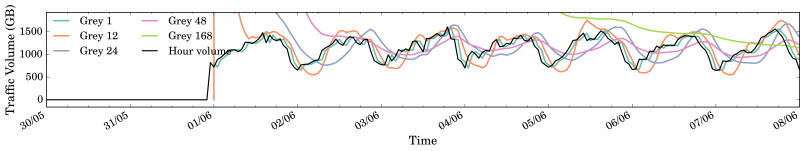
(a) SA



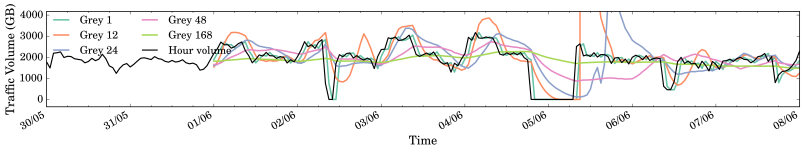
(b) SB



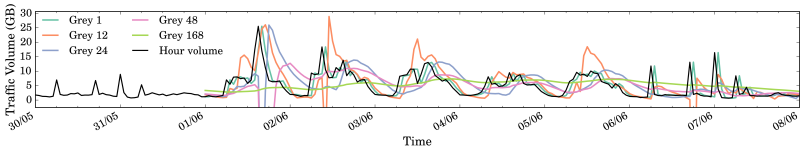
(c) SC



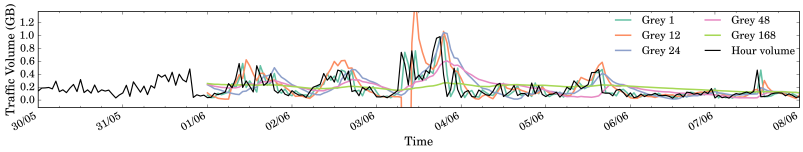
(d) SD



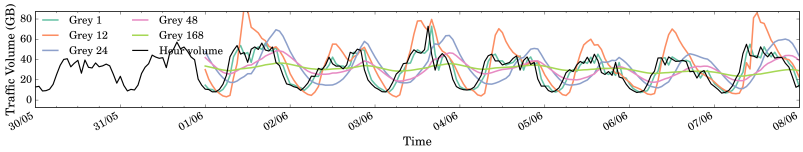
(e) SE



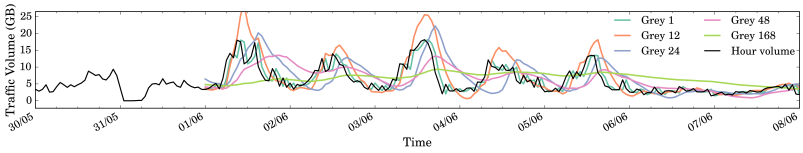
(f) SF



(g) SG



(h) SH



(i) SI

Figure 10: Predict total hour volume using GM(1, 1) with different historical record lengths for the week starting from June 1st, 2015.

commonly seen on SC, SF, SG judging from their burstiness index in Table 3. This explains why grey model leads to fairly low volume coverage in Figure 9c.

Furthermore, in Figure 10b, Figure 10c and as well in others, we can see that grey model reacts to volume variations in an obviously delayed manner. Longer the historical records length, longer grey model delays the variations of actual trace. This behavior could be fatal in the presence of bursty prefixes. We can observe considerably large pikes in both directions several hours after the volume burst in Figure 10c, which might greatly disturb the prefix selection during those periods and consequently cause low volume coverage.

Compared to grey model, metrics proposed in our work do not directly predict hour volumes of each prefix but rather are indirect measures of its importance in terms of volume. And analysis in Section 3.3.2 showed that the overall coverage using these metrics are guaranteed by the traffic character itself.

Figure 11 gives the results concerning the churn of the selection prefix set (in box-plot representation). The churn is defined as the difference (i.e. number of new and deleted prefixes) between the new predicted set and the previous one. A high prefix churn is especially unwanted by the measurement sub-system, which aims at continuously monitoring and providing historical records of important destinations. Sarrar et al. [79] also argued that small prefix churn is very important in network architecture with decoupled forwarding and control planes, such as SDN (Software Defined Networking), for the preference over small communication overhead.

As expected, a clear drop of the churn value can be seen when the historical length increases — as opposed to what happens with the grey model. In that sense, using long historical records can be a wise choice in practice. Furthermore, for networks with relatively few bursty traffic, e.g. SA, the mean volume coverage with last 168 hour records is extremely close to that with last 24 hour records, shown in Figure 9a. Finally, for networks with highly bursty traffic, SC, using long records has the potential to obviously improve worst-case volume coverage.

For the purpose of lowering churn, Sarrar et al. [79] proposed selecting top prefixes over time bins of different lengths (ranging from 1 second to 10 minute in their FIB-caching environment). In our context, we found that the difference in mean volume coverage using record lengths larger than 1 hour is marginal, thus little gain can be expected from this method.

3.4.3.1 *Relation between volume coverage and traffic burstiness*

The mean/minimum coverage achieved with CV metric is showed as a function of the BI index in Figure 12. We can see that the mean (resp. minimum) coverage is inversely proportional to the mean (resp. max-

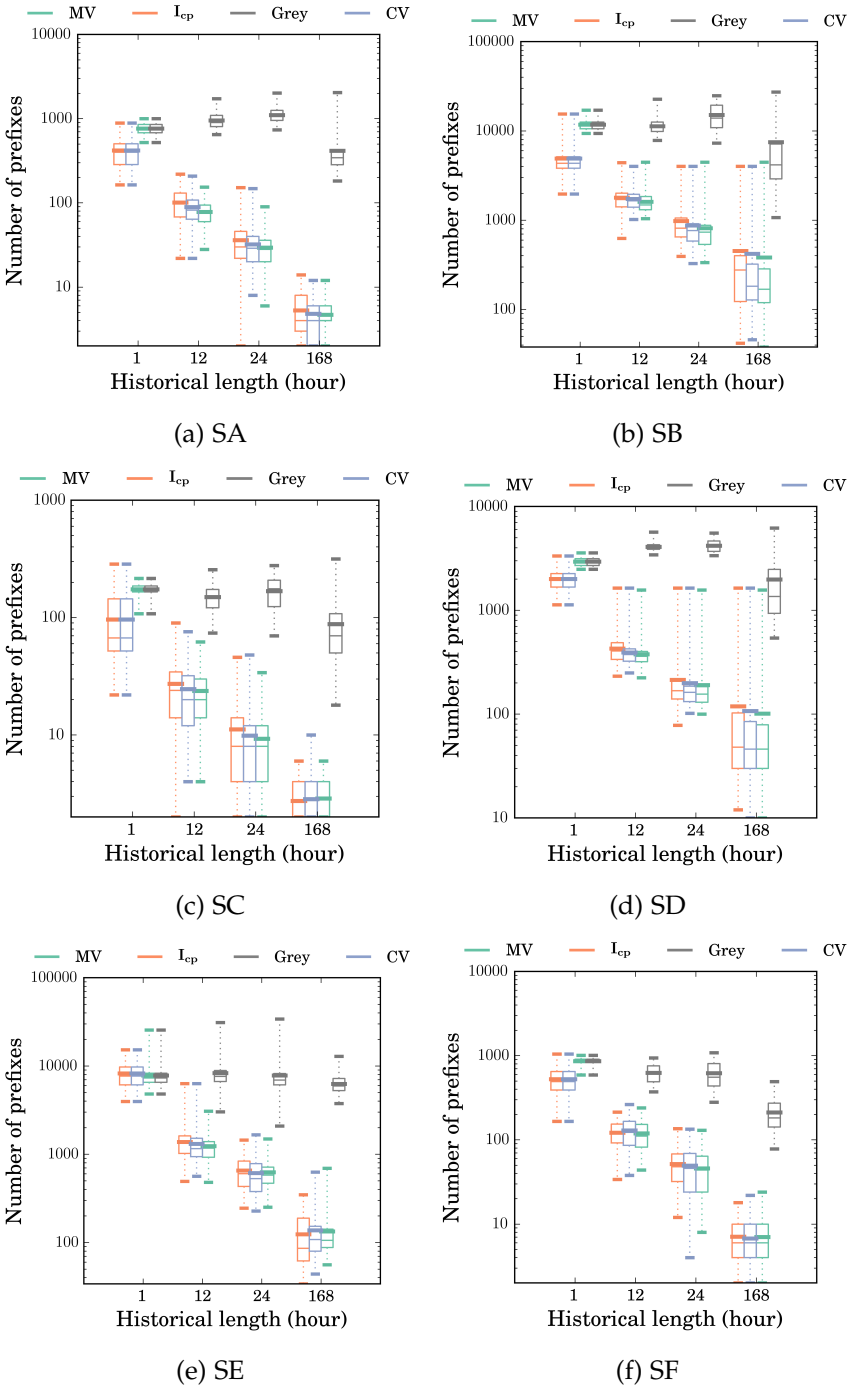
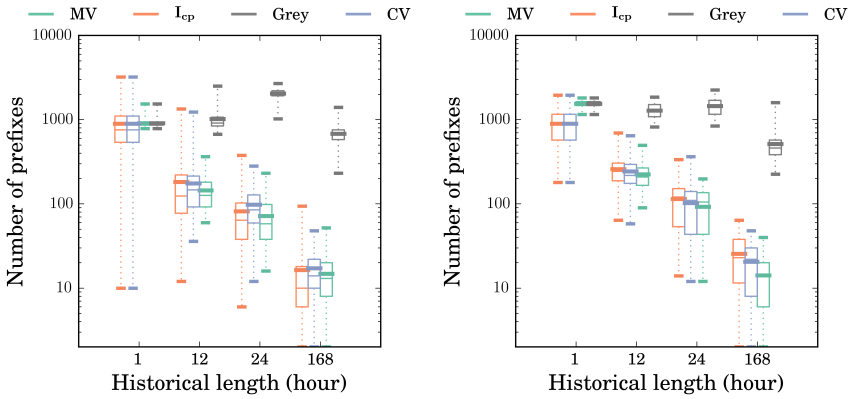
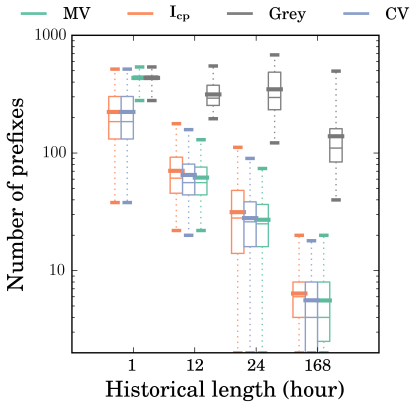


Figure 11: Hour churn of the prefix set predictively selected using historical records of different lengths. The selection set size of each network is set to the maximum *core* size over the week starting from June 1st, 2015, see in Table 2.

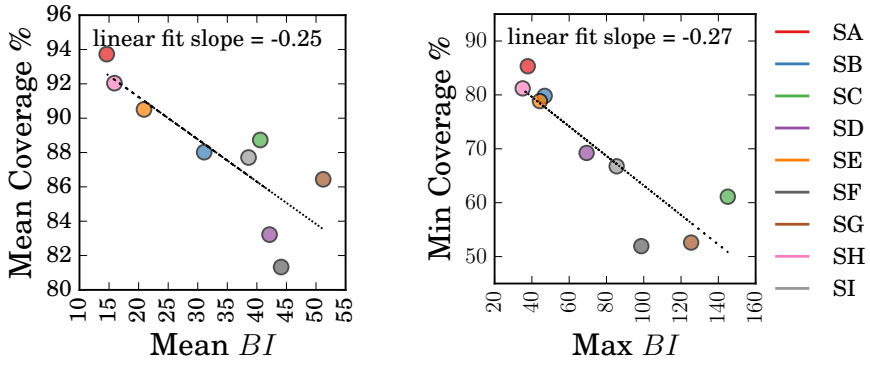


(g) SG (h) SH



(i) SI

Figure 11: (cont.) Hour churn of the prefix set predictively selected using historical records of different lengths.



(a) Average level

(b) Worst case

Figure 12: The relationship between burstiness index BI and traffic volume coverage of selected prefix using *CV* metric.

imum) BI index. This relation highlights the difficulty to cover a big fraction of traffic volume for networks with more bursty traffic. However, the quasi-linear curve obtained shows that BI is a very meaningful metric to identify sites with bursty trafic. For network with large BI values, it is worthy of choosing a larger prefix set size (which was previously fixed to the weekly maximum core size), if possible.

3.5 TRANSIT PROVIDER PERFORMANCE EVALUATION

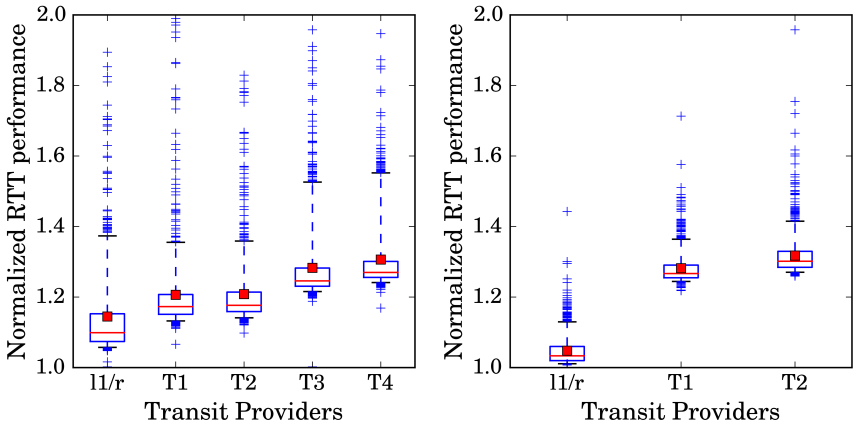
In this section, we explore the potential performance gain that client networks can potentially achieve with measurement-based TE.

In order to evaluate the performance gain, we continuously measured the Round-Trip Time (RTT) towards selected prefixes (using *MV* metric with $L = 168$) via all available transit providers using TCP SYN scan [59] over the week starting from June 1st, 2015, i.e. the second week of our observation. The probe traffic is steered by means of explicit routing. For a pair of selected destination prefix and available transit provider, a probe is scheduled at 240 second interval in average with 30% randomization w.r.t. the interval in timing.

We quantify the performance level of a transit provider with a metric proposed by Akella et al. [16]. This metric first normalizes the RTT via the chosen transit over the smallest RTT measurement across all available transit providers at the same probe round. This is done for each individual selected prefix. It then averages this normalized RTT over the entire selected prefix set. More formally the evaluation is done as the following:

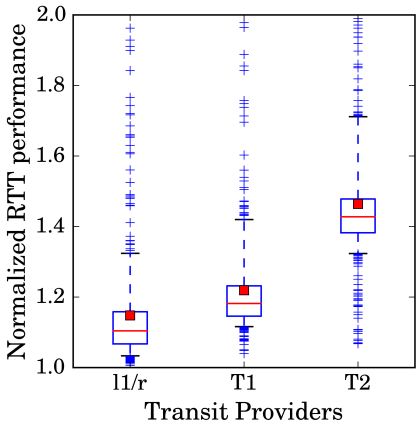
$$NP_{t_i}^{Tx} = \frac{1}{|SP|} \sum_{P \in SP} M_{t_i}^{Tx}(P) / \min_{T_j \in T} M_{t_i}^{T_j}(P)$$

where $NP_{t_i}^{Tx}$ is the normalized RTT performance for transit provider Tx at probe round t_i, and $M_{t_i}^{Tx}(P)$ denotes the RTT measured toward

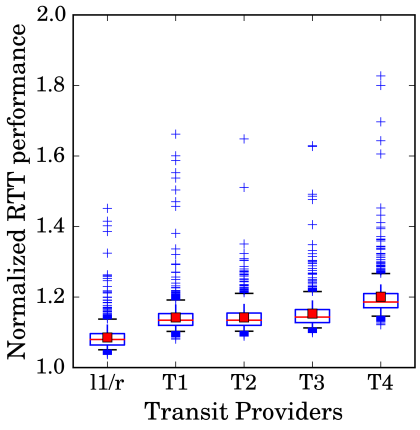


(a) SA, 866 prefixes, 85.69% traffic

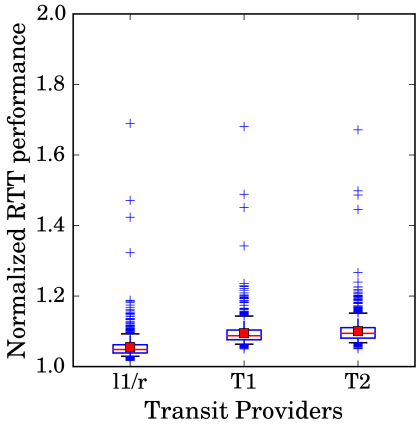
(b) SB, 935 prefixes, 48.49% traffic



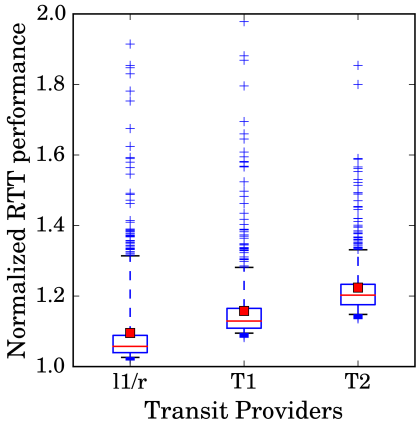
(c) SC, 762 prefixes, 44.23% traffic



(d) SD, 2841 prefixes, 49.18% traffic

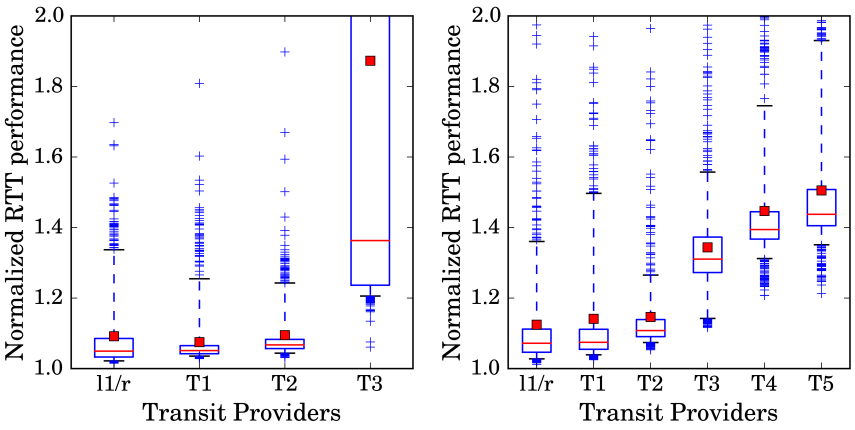


(e) SE, 2287 prefixes, 38.59% traffic

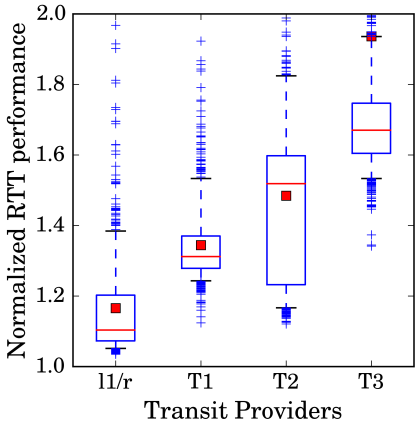


(f) SF, 879 prefixes, 78.49% traffic

Figure 13: Normalized RTT performance with active probing. Average number of prefixes probed and average traffic volume fraction represented by these prefixes each hour are given.



(g) SG, 924 prefixes, 63.08% traffic (h) SH, 73 prefixes, 2.83% traffic



(i) SI, 642 prefixes, 67.51% traffic

Figure 13: (cont.) Normalized RTT performance with active probing. Average number of prefixes probed and average traffic volume fraction represented by these prefixes each hour are given.

selected prefix P via Tx. If a transit provider offers the smallest RTT to all selected prefixes, it should have a normalized RTT performance equaling to 1. On the other hand, a large NP value indicates that the overall transmission performance using that transit provider is far from ideal.

If the egress transit provider for real traffic is chosen dynamically for each individual destination prefix, the route selection mechanism can be regarded as a virtual transit provider. At each probe round, the virtual transit provider corresponds to the choice physical transit providers for each destination prefix.

We implemented a virtual transit provider, denoted as l1/r, first appeared in [49] for the sake of comparison. l1/r selects the transit provider that provides the smallest RTT in last probe round for each destination prefix (l1 in l1/r). When RTT measurement data is not available (e.g. prefix newly selected, measurement timeout), it chooses randomly (r in l1/r). This choice is based on the hypothesis that RTT of a path demonstrates temporal locality and thus is closely related to its most recent measurement.

Figure 13 gives the results of transit performance evaluation in a box-plot. The box stands for 25^{th} and 75^{th} percentiles. The red line tells the medium value, while the red square indicates the mean. Finally, the whiskers represent 5^{th} and 95^{th} percentiles (values below and beyond, marked by a + symbol, can be regarded as outliers). Along the X-axis, available transit providers are aligned from left to right increasingly according to their mean normalized RTT performance (marked by a red square).

We observe that the performance differences among different transit providers is particularly evident on SC. This confirms that multihoming can still provide significant performance improvement nowadays. However this gain in performance is not inherently given in the context of BGP, as it it is performance agnostic in route decision. Even for transit provider that offers the best best mean NP, there exist moments where its performance deviates far above 1, which means that traffic toward some selected prefixes are suffering from RTTs much larger than other available transit providers. This implies that a network can not arrive at optimal performance by using a single transit providers in a static and indistinguishable manner for all the destination prefixes.

For all networks except SG, the virtual transit provider outperforms all physical transit providers. On SB, the performance metric NP of l1/r is 20% lower than that of the best available transit provider. Still, the NP value of this virtual provider varies within a wide range, which calls for further investigation into the characters of RTT variation in time, see Section 4 and 5.

Finally, we missed RTT measurement for quite a few selected destination prefixes on some client networks, especially SH, SC, SD and

SE. Reasons for such massive lack of measurement are explained in Section 6. A possible solution is given to allow TE for those prefixes without measurements.

CONCLUSION

This chapter tackled the problem of controlling a majority of data traffic via a small subset of BGP prefixes, by exploiting the uneven Internet traffic distribution. One of the challenges in addressing this problem was to select in a scalable manner the prefixes that will carry most traffic volume in the forthcoming time.

We analyzed real traffic measurements from nine different networks located in five different countries to understand the distribution of traffic volume associated with BGP prefixes, as well as its variation in time. We observed that the most important prefixes (representing the largest volume over a week) are generally stable in time, with small hourly variations around their mean of hour volume. Based on this observation, we proposed three simple metrics (also easy to compute) to proactively select prefixes with important foreseeable traffic volume. We demonstrated that the metrics we proposed lead to better volume coverage compared to the existing solutions. Furthermore, we evaluated the transmission performance for selected destination prefixes using multiple transit providers. We simulated as well a dynamic route decision algorithm. The results showed that with even a fairly basic mechanism, the overall RTT performance could be improved by 20% compared to the best available transit provider in some networks studied.

In order to further improve prefix selection methods, we have shown that capturing bursty prefixes is the key. To this end, we could group prefixes by their activity profiles. For each group, selection method is adapted to its traffic dynamism. When dealing with prefixes with regular volume patterns and small hourly variation, the simple metrics proposed in this work perform already sufficiently well. Nevertheless, for bursty prefixes, we might need a more sophisticate model that extracts additional activity features, for instance long term periodicity. The burstiness index β proposed in this chapter, shown to be very expressive, could be potentially used in prefix characterization and classification and thus is worthy of future work.

4

INTERNET MEASUREMENT WITH RIPE ATLAS

ABSTRACT

Starting from this chapter, our study is focused on delay and path measurements. These measurements are readily available on client TE platforms, same as volume data studied in Chapter 3. However, for the sake of reproducibility, we decided to switch to measurements conducted by RIPE Atlas, a world-wide measurement platform offering open data access. We justify this choice by succinctly introducing its design philosophy and comparing it to alternative platforms.

Besides reproducibility, measurement data quality is also crucial to research credibility. We hence studied the missing datapoints in measurements scheduled regularly by RIPE Atlas. Contrary to common belief, a big part ($\sim$ 60%) of continuous datapoint loss happened when the probe remained connected to the measurement infrastructure.

Further, we explored a data quality concern that is specific to TE applications. Through unsupervised learning, results showcased that a part of the RTT measurements on a same AS path were likely subject to local congestion. Avoiding this kind of delay variations in transmission is not really the objective of inter-domain TE. They were hence regarded as noises. The finding confirms the need for data cleaning, a process often neglected in previous practices.

Finally, we experimented several time series clustering methods to group RTT measurements with similar shapes, i.e. undergoing same RTT changes. Such RTT measurement groups, along with path measurements, can help reveal where the RTT changes come from. We show later in Section 6 that this visibility is particularly useful to interdomain TE.

4.1 REPRODUCIBILITY

In Section 3, we collected traffic volume and delay data from real client networks. All the studies concerning prefix selection was developed on that dataset. Having access to real client data increases the credibility of the discoveries made, and enhances the relevance of proposed schemes basing on these findings. The other side of coin is that such private dataset hinder the reproducibility, a paramount feature in metrology researches.

The Association for Computing Machinery (ACM) offers definitions for various terms referring to different degrees of research repoducibility [121]. The degree ranges from repeating the same result by the same team to reproducing the same result with independent implementation of proposed methods or measurement system. The way the measurement data is generated, stored and accessed is one key element for all these degrees of reproducibility.

Previous client data come from measurements performed by a proprietary TE platforms [124]. By nature, it is against the company's benefit to reveal the technical details on how measurements are conducted. The collected data contain sensible information, e.g. the destination prefixes that the client network talked to. They are required to remain on client owned platforms otherwise permission required. Due to capacity limitations and decreasing utility of old data, these measurements will not stay forever available on client servers. This prevents future verifications on the same dataset. If measurements are allowed to be retrieved, researchers are then responsible for the storage and security of these data. Anonymization is required, if open access were to be granted. The process is not trivial, since an appropriate balance between privacy and interpretability is hard to hit. Moreover, for better representativeness and statistical confidence, Internet measurement researches stress on large dataset over long period. Maintaining the access to these large dataset is clearly not without cost.

Realizing above limitations using private data, we looked for public measurement platforms that can alleviate the burden in measurement execution, storage and public access.

4.2 RIPE ATLAS

RIPE Atlas is not the only Internet measurement platform that provides open data access [99]. We justify this choice by first introducing RIPE Atlas. We then summarize and highlight its features that qualify it as a plausible option for our research, along with comparison to alternative measurement platforms. This introduction as well saves repetitive efforts on how measurements are conducted and collected in later studies.

4.2.1 *Overview of RIPE Atlas*

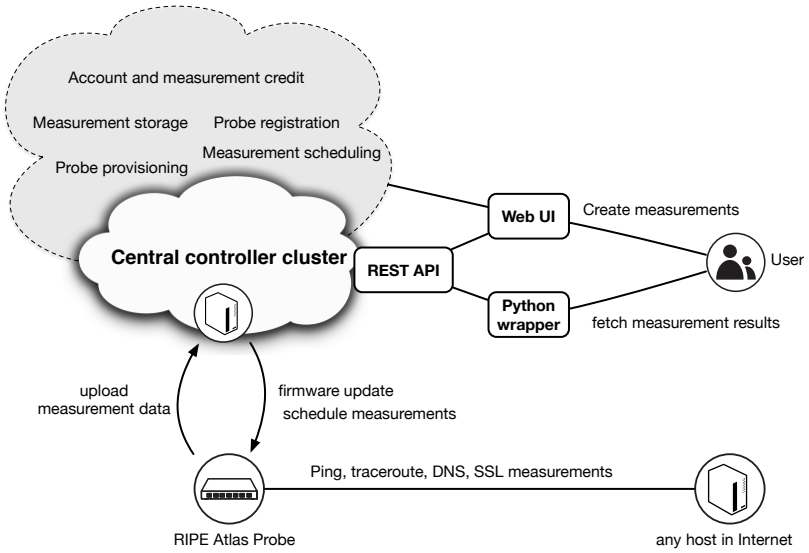


Figure 14: Building blocks of RIPE Atlas.

Réseaux IP Européens (RIPE) Atlas is a measurement platform centrally managed by the European Internet register. Figure 14 sketches the architecture of the platform. Probes are dedicated devices from which measurements are launched. The operation system on the probes are tailored by RIPE engineers for Internet measurements [147]. As of this writing (July 11, 2017), 19448 probes have been sent out and 9854 of them remain active. All these probes, hosted in 3511 IPv4 ASes and 1286 IPv6 ASes across 181 countries, can be commanded by any user to measure any destination in the Internet.

As a platform user, one does not have to connect to all these probe by him/herself to 1) create specific measurements; 2) fetch measurement results. One only has to interface with RIPE via programming API [141] or web page `https://atlas.ripe.net` to fulfill the above essential tasks along with other helpful functions such as measurement data visualization. To that end, RIPE collects in quasi-realtime measurements from all the connected probes and stores them in its server clusters.

4.2.2 *Measurement types*

RIPE Atlas supports a wide range of standardized Internet measurements with configurable parameters: ping, traceroute, DNS, SSL. Ping and traceroute measurement offer the Internet delay and path information that are required in measurement-based TE.

Another way of classifying the measurements is via the entity of measurement creator. User of the platform enjoys a great degree of liberty in specifying the destination, sources and time range of supported measurements. These are User Defined Measurements (UDM). Once a user defines a measurement, i.e. User Defined Measurements (UDM), the central controller clusters schedules it to corresponding probes and collects the measurement results.

There exists another category of measurements called *built-in measurements* [142]. These measurements are automatically executed by the probes without the need for controller scheduling. These measurements, originated from all probes, are mainly ping, traceroute and DNS measurements to DNS root servers and RIPE infrastructures. In later studies, we heavily rely on these built-in measurements given 1) their world-wide footprint, 2) super long history records (dating back to the first connection of each probe) and 3) low additional measurement costs.

4.2.3 *Describe, identify and fetch measurements*

Besides measurement type specific parameters, such as the protocol type for traceroute, following three elements are as well fundamental in describing a RIPE Atlas measurement: 1) participant probes; 2) the single measurement destination per measurement; 3) the time span of the measurement.

Both UDM and built-in measurements can be identified by a unique measurement ID, with which one learns the measurement meta-data, accesses data visualization provided by RIPE and eventually fetches the raw measurement results.

For example, with `https://atlas.ripe.net/measurements/3742863/#!openipmap`, one can have access to the path visualization of measurement #3742863, where 100 probes word-wide are selected to perform one-time traceroute toward `www.sigcomm.org`. With `https://atlas.ripe.net/api/v2/measurements/3742863/results/?start=1462147200&stop=1462233599&format=json`, anyone can easily download the entire raw measurement records of this measurement.

4.2.4 *Advantages*

RIPE Atlas is a measurement platform designed to facilitate reproducible researches. RIPE takes care of all the engineering challenges of 1) measurements scheduling to geographically distributed probes; 2) reliable and continuous data storage; 3) public access to data; 4) simple syntax for describing, identifying measurements; 5) well documented open-source programming tools for data manipulation.

The advantages of RIPE Atlas go beyond reproducibility. Compared to perfSONAR [150], PlanetLab [139] and DIMES [128], probes of

RIPE Atlas, with dedicated hardware and firmware for measurement tasks, are supposed to deliver measurements that are less impacted by probe local resource sharing issues and thus better reflect the network characteristics alone.

Moreover, RIPE Atlas is rapidly gaining popularity among many non-academic networks, such as ISP, CP and IXP, thanks to a wide range of monitoring applications henceforth enabled, to name a few, performance monitoring [94, 126], anomalies detection [105, 110, 135], peering and IXP measurements [122, 138] etc. Increasing number of commercial networks host RIPE Atlas probes, providing a much richer and realistic network profile from which measurements can be initiated, compared to other alternative options.

4.3 MISSING MEASUREMENTS ON RIPE ATLAS

Data quality is another key issue to metrology researches besides reproducibility. Through previous studies [99, 103], it is now known that load has obvious impacts on measurement precision and scheduling. We focus on data completeness, another aspect of measurement quality that received less attention so far. Missing measurements can cause various undesired consequences. Apart from widening confidence interval of inference [110], it requires in general methodological adaptations, e.g. in spectrum analysis [61, 93, 116], otherwise biased estimation would be expected [62].

One obvious reason of missing measurements is that the probe is not running (properly), e.g. power off [143]. As long as a probe is powered, it tries to maintain a connection to a controller to report measurements and receive assignments as shown in Figure 14. Therefore the probe connection activity provides a good indication of the probe availability, and is used in investigations conducted by RIPE on probe OS stability [129, 130, 149].

In order to infer the possible existence of other causes, we compared the measurement timestamps with the moments probe connects to and disconnects from the Atlas controller system. If measurement missing coincides with the probe disconnection, chances are that the probe is dysfunctional, e.g. power off, during the missing. However, if measurements are lost while the probe is well connected, something 'abnormal' should be expected, beyond the known probe OS issue.

4.3.1 *Data collection*

We observed the RIPE Atlas platform for one month, from 2016-06-01 to 2016-07-01 UTC. All the v3 probes first connected before the beginning date (11613 of them) are considered. Connection events (measurement ID 7000) and built-in Ping measurements to DNS b-

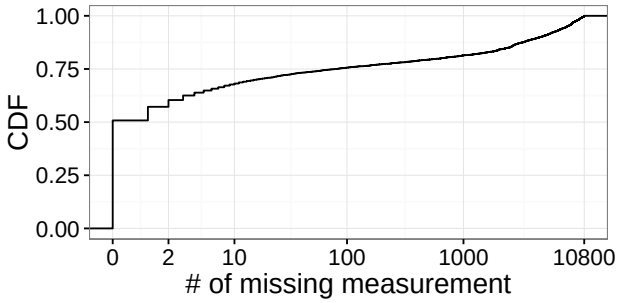


Figure 15: CDF of total missing length per probe.

root (measurement ID 1010), a highly available destination, are collected [125]. Controllers and the ping destination are not within the same network. Controller logs the moments at which probes connects to and disconnects from it. The built-in ping measurement is scheduled on every probe at 4min interval. 10800 ping results are thus expected from each probe over the month. 7353 probes, out of the available 11613, had Ping measurements during this period.

4.3.2 *Missing measurements at first glance*

We deem that there are missing measurements when the time interval between two neighboring measurements are abnormally longer than the planned value (> 150%). Interestingly, such long gaps turn out to be very close to integer times of planned interval. It is because as a cron-like mechanism is used to run measurements at regular interval and it retakes the previous phase after interruption [131, 143]. This character allows quantifying the length of missing segment (a period of continuous lack of datapoints) by the number of measurements skipped. 4440 probes (60.4% out of 7553 probes with data) miss no more than 2 datapoints. Such slight data incompleteness is totally legitimate, as random jitter is added to each single measurement to avoid synchronization within a probe and among different probes. For the rest, the missing length spans a wide range according to Figure 15. The graph depicts the distribution over probes the number of missing datapoints. 1358 (18.5%) probes miss more than 10% of the total measurements (i.e. 72 hours over a month).

4.3.3 *Cross missing measurements with connection events*

The basic idea of this study is to simply juxtapose measurement timestamps and connection events, and try identify unexpected patterns suggesting unknown issues.

Several reasons may contribute to the disconnection of a probe to its controller: 1) probe not working (properly), e.g. power off; 2) network issues preventing the connection, e.g. no available route; 3) con-

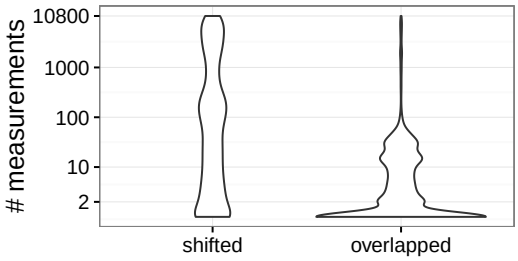


Figure 16: Missing length distribution.

troller not available, e.g. during maintenance [144]. Meanwhile, the last two reasons shall not prevent a probe from performing built-in measurements. It is because the built-in measurements are by default configured in all probes. There is no need for network connectivity or controller command to install them. Moreover, the measurements, though their result code being 'network unreachable' or 'timeout', can be store locally on the probe, under these two circumstances [132]. When they connects again to the controller cluster, the probe will upload in batch the local data. With that, we can conclude that *missing measurements do not necessarily occur when a probe is disconnected, but are unexpected while the probe is connected.*

4.3.3.1 *Overlap with connected period*

The 'abnormal' behavior pattern that we look for is thus missing measurements during the connected period. To that end, we count, for each missing segment, the number of missing datapoints that overlaps with connected period. We define the *overlap ratio* of a missing segment as the ratio between this count and the entire length of missing segments. The distribution of this overlap ratio is concentrated at the two ends, 0 and 1. This means that a missing segment is generally either mostly shifted/dislocated form connected periods, either largely overlapped with connected periods. For the convenience of illustration, we classify missing segments into two groups. One with overlap ratio $\leqslant 0.5$, denoted as *shifted*, the other with the rest, denoted as *overlapped*. The *overlapped* groups contains missing segments that are 'unexpected'.

The two groups demonstrate very different length distribution profiles, Figure 16. Totally, 15391 missing segments are observed. 10292 (66.87%) missing segments are overlapped with connected period. They are mostly short in length. 5560 of them last no more than 2 measurements. One possible explanation is that these measurements are skipped due to scheduling or load issues [103, 143]. Meanwhile, 2490 of them are equal to or longer than 1 hour in length, involving only 620 probes, for which we believe that the previous explanation hardly applies.

Missing segments shifted from connected period are more likely to be long. This is possibly due to the v3 probe OS stability issue still under investigation. It is known to be responsible for long term probe disconnection due to dysfunction. It requires manual operation to recover probes under such condition [129, 130, 132, 149].

4.3.3.2 *Temporal correlation between missing measurements and connection events*

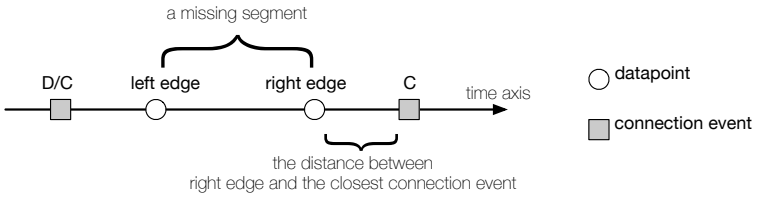


Figure 17: Illustration of *left edge* and *right edge* of a missing segment. A possible temporal relationship of the two edges with connection events is as well depicted.

To obtain a close-up view of this 'abnormal' behavior, we seek to find out: *when do measurements begin to be lost and when are they recovered? Are these moments close to connection events?*

A missing segment itself does not have any timestamps to mark its beginning and end. We thus use the timestamps of available measurements around it to describe the time range of a missing segment. We define the *left edge* of a missing segment as the last measurement before it, and the *right edge* the first measurement after it. We then measure how far the two edges are from the closest connection events. We as well identify the nature of these connection events. We denote 'D/C' for disconnection and 'C' for connection. These terms are illustrated in Figure 17.

1284 missing segments locate at the beginning or the end of the observation period. They are unavailable for this analysis as only one edge can be observed. For the rest with both edges, 5793 missing segments' left edge is closer to a disconnection event and the right edge is closer to a connection event. As we can imagine, with the help of Figure 17, these are probably missing segments that are shifted from connected periods. Therefore, we separate them from the rest by giving them a specific notation (D/C, C).

In Figure 18, we visualize the distribution of distance in time from the two edges to their separate closest connection event. (D/C, C) missing segments and the rest are plotted separately in two subgraphs. Each missing segment is a dot on the graph, colored accord-

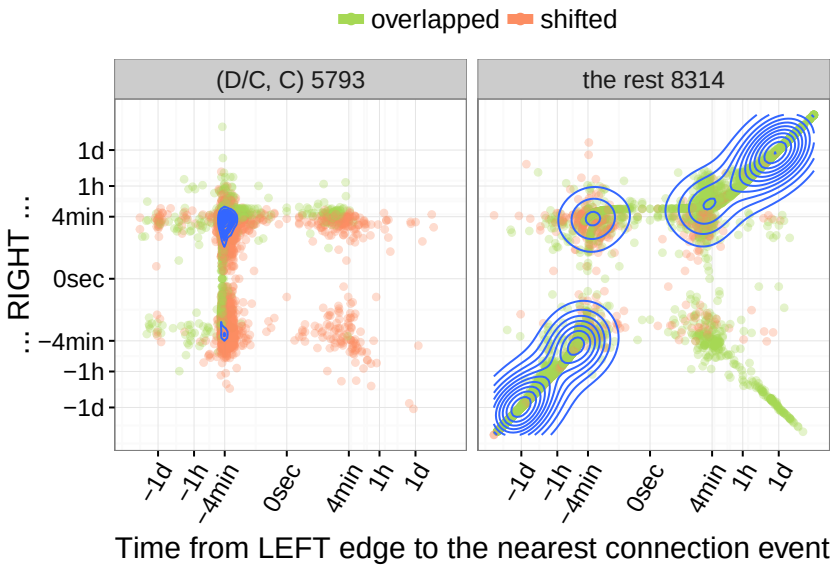


Figure 18: (D/C, C) stands for missing segments more closely correlated with disconnected period. Number of concerned missing segments is given in the title. Negative time distance means the edge happens before the connection event and vice verse.

ing to its overlap ratio. If a distance takes a negative value, it means the edge of a missing segment precedes the connection event, and vise verse. Since many dots overlap with each other on the surface, we indicate their 2-dimensional density using contour lines (estimated with MASS:kde2d package in R). The most inner contours sit in the densest areas.

For missing segments of (D/C, C) type, the densest area is around the $(-4\mathtt{min}, 4\mathtt{min})$. This means the left edge of a (D/C, C) missing segment most likely precedes a disconnection event by a Ping interval (4min), and its recovery (the right edge) tends to take place 4 minute after the connection event. Such strong correlation with probe disconnected period indicates that probe dysfunction is probably the cause for these missing segments. As a matter of fact, most dots in the (D/C, C) subgraph of Figure 18 are indeed 'shifted' ones according to their overlap ratio classification.

However, the beginning (left edge) of (D/C, C) missing segment can as well be far ahead of a disconnection event. At the left end of the subgraph, measurements begin to lost long before the closest disconnection. These missings are probably an early sign of some internal issues that finally prevents the probe connecting to the controller. For dots are the right side of the same subgraph, measurements only begin to lost a while after the probe is disconnected. One possible explanation is that measurements are first stored locally after disconnection from controller [132]. Then new measurements only begin to lost after the local storage is full.

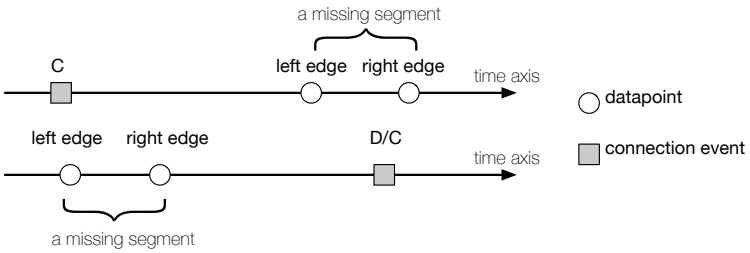


Figure 19: Illustration of event placements in time for missing segments having positively correlated distance from its two edges to the closest connection events

Concerning missing segment other than the (D/C, C) type, the majority of them actually overlaps with connected periods. Plus, them tend to concentrate along one diagonal direction, where the inner contours can be found. For these missing segments, the distances from left and right edge to connection events are highly correlated, suggesting that both left and right edges are on the same side of a same connection event. We illustrated such event placement in Figure. 19. The missing segment is either to the right side (after) of a connection event, or to the left side (before) a disconnection event. For a few missing segments that fall in the middle of a connected period, they appear on the diagonal line in the down right area (quadrant IV) of the corresponding subgraph. These missing segments are mostly short in length. If they were closer to any of the two extremities of the connection period, they will then become the case illustrated in 19. For the group of missing segments in the right upper part (quadrant II) of the subgraph, they basically wrap within in them a connection period. Depending on the length of this connection period, they can be either classified as overlapped or shifted. Since the distance from edge to connection event is most likely 4min according to the density contour, only when the connection period in middle is very short, the missing segment is labeled as shifted.

All the temporal placements of missing segment and connection events that are discussed here above are just the most likely/typical cases. They don't speak for the many specific cases that disagrees with the mainstream.

Long lasting (>1h) missing segment yet overlapped with connected period take place in both subgraphs. Most of them have at least one edge far from connection events. This suggests that the underlying mechanism of these missing segment is not necessary related to probe connection events. Moreover, these missing segments in general have pretty uneven absolute distance from their two edges to connection events. This is is a coupled effect of 1) its long length and 2) again

lack of correlation to connection events. Apart from the above observations, we were not able to arrive at a more conclusive description/-explanation for those cases.

Wrap-up

In our analysis covering a large number of probes over one month, only 60% of v3 Atlas probes have complete measurements. Around 1/3 missing segments appear to closely correlated to disconnected period. The probe OS stability issue might have contributed to such missings, as suggested by the heavy tail of the missing lengths.

However, the remaining 2/3 of missing segments occurred while probes are connected. Half of them are no more than 2 measurements in length, and are thus likely to be caused by scheduling issues. However, around 25% of this category lasts long($\geqslant$ 1h).

We reported the discovery to RIPE engineering team along with a specific case that they could look into. The last reply from RIPE team confirmed that the probe we mentioned in the report had "time synchronization issues". To help advance the investigation, we shared with RIPE team all the long missing segments identified along with corresponding probe IDs. These exchanges can be found on the RIPE Atlas forum at https://www.ripe.net/participate/mail/forum/ripe-atlas, with title "Actual measurement interval much larger than planned".

Though the final result is not conclusive nor revealing in terms of the underlying mechanism, this study did help realize a data completeness issue and treated it seriously. This issue can be mostly avoided or largely alleviated, if the probes are properly chosen as source of measurement data. With complete relatively complete data over time, a lot of poorly justifiable data cleaning steps can hence be avoided.

4.4 SAME AS PATH MEASURED BY DIFFERENT PROBES

A specific data quality concern in measurement-based interdomain TE is *whether the measurement RTT reflects mainly the characteristics of AS paths that could be used by real traffic*.

Route selection function in Figure 3 relies on path performance measurements as input (more details in Chapter 5). However, RTT measurements might be 'polluted' either by non-network factors, say host-local issues such as CPU overload, or non-representative sub-AS level network issues, say local congestion within the destination prefix. Avoiding these issues is not the main objective of interdomain TE, since it is mostly unfeasible with interdomain routing alone. However, none of these previous works on measurement based inter-domain TE [24, 49] has realized the importance of this problem.

This data quality issue gives rise to a series of questions: *if we measure a same AS path with different hosts in the destination prefix, what will*

these RTT time series look like? Will they have similar characters? If not, how can we pick out the ones that fit best for interdomain TE purposes? We try to answer these questions by performing clustering over a such set of RTT time series. Without prior knowledge or assumption, the study aims at automatically revealing the inherent structures of these RTT time series.

4.4.1 *Data collection*

We emulated a typical RTT measurements between two ASes, i.e. one local client AS (one host) and one destination AS (multiple hosts), with RIPE Atlas built-in measurements #1006 and #5006. They are respectively IPv4 ping and traceroute measurements from all Atlas probes to m.root-servers.net (202.12.27.33). Ping measurement is scheduled at 240 seconds interval, while traceroute at 1800 seconds. 120 RIPE Atlas probes within AS3215 are selected to construct our database [1]. These probes hit the same DNS root server clusters via a same AS path: AS3215, AS5511, (PARIX), AS7500 [2]. Time window for the data collection ranges from 2015-09-28 10:00:00 UTC to 2015-09-29 12:00:00 UTC.

We cleaned the data collected, with following steps:

- Remove probes with unstable connection to the Atlas platform. (Short total length, $<$ 95% expected length; multiple missing segments 4.3);
- Remove probes suffering from obvious hardware or local network issues. (high packet loss, $>$ 5% measurement data are timeout; any error flag found in measurement results).

We then consider only 100 common probes in both ping and trace traces remained after cleaning. All valid probes considered, the average IP hop number to the destination, m-root server, is 9. For the traceroute data set, we decided to concentrate on the first 3 hops (which should cover the access network). As a consequence, the traceroute data set are further cut into 3 parts, where each contains the RTT time-series till the corresponding hop.

To sum up, we fabricated four RTT time series data sets, each with 100 RTT time series.

- `pingData`, end-to-end RTT time series, 391 datapoints per trace;
- `traceData1`, RTT series till the first hop, 53 datapoints per trace;

1 The 120 probes are the same ones in user-defined measurement #2427397. The metadata of the that measurement are accessible to everyone.

2 Some of these probes are now moved to a different network. For example, probe 2036 is now within AS35540, probe 877 is hosted in AS12322, etc. Therefore, if the same tracerouts are repeated now or afterward, it will be normal to expect some probe to employ a different AS path.

- `traceData2`, RTT series till the second hop, 53 datapoints per trace;
- `traceData3`, RTT series till the third hop, 53 datapoints per trace;

4.4.2 *Clustering RTT series in feature space*

Generally speaking, a time series clustering approach can be decomposed into three parts: data representation, distance measure and clustering algorithm [97]. Due to its high dimension, time series is seldom used in its raw form [3]. Therefore, a time series needs to be transformed and represented with fewer datapoint that still captures its essence. Common approaches include dimension reduction [82], pattern extraction [108], etc. Distance measure quantifies the similarity/dissimilarity of two time series with their new data representation. A well known practice is Euclidean distance that we usually apply on a Cartesian coordination system. Finally, clustering algorithm defines the procedure of grouping time series based on the distances calculated on their data representation space. For each of these components, multiple possibilities exist. However, it is not clear which ones in combination could be the best fit for RTT measurements.

In this section, we extracted a set of features listed below from each individual time-series and used in actual clustering. The advantage of such data representation is that it first largely reduces the data dimension. Thanks to that, data set is more suitable to classic partitioning clustering methods, like k-means and k-medoids (also known as Partitioning Around Medoids (PAM)) [20]. Second, it is able to depict the data set from multiple aspects that are not evident with the raw form. Before clustering, each feature is z-normalized (zero mean and unit variance).

Following features are used in this work:

POWER SPECTRAL DENSITY is calculated using PyEgg [69] and cut into three bins relative to sampling/measurement frequency, $(0, 1/12]$, $(1/12, 1/6]$ and $(1/6, 1/2]$. Each of the bins individually functions as a feature.

SAMPLING ENTROPY proposed by Richman and Moorman [8] quantifies the regularity or predictability of a time series. In calculation [69], we used an embed dimension of 2 and a tolerance of $15msec$.

NUMBER OF CHANGES counts the number of times where the difference between two consecutive RTT measurements is greater than $15msec$.

3 The curse of dimensionality: https://en.wikipedia.org/wiki/Curse_of_dimensionality.

RANGE is the difference between the maximum and the minimum values of an RTT series.

MODE is the value most frequently present in a time-series.

MEAN the first-order moment, describes the overall RTT level.

STANDARD DEVIATION is derived from the second-order moment, describes how close measurements are to the mean.

SKEWNESS , the third-order moment, describes the lack of symmetry of RTT values observed around the center point of histogram, mode.

KURTOSIS , the fourth-order moment, describes whether the histogram of observations are peaked or flat relative to a normal distribution.

With RTT time series represented in the above feature space, we simply use Eucilidean Distance (ED) to measure the distance between two RTT time series.

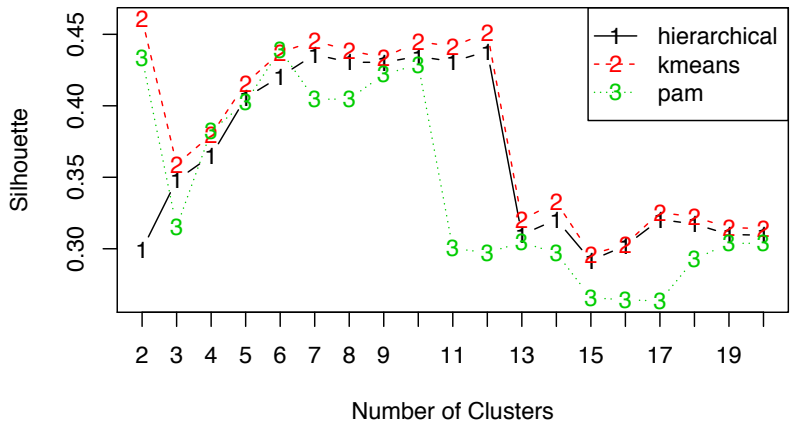


Figure 20: ASW achieved on `PingData` using different clustering algorithms when varying number of clusters.

Several clustering algorithms are available. In order to choose the one that fits best to our data set and decide the most appropriate cluster number, we use Average Silhouette Width (ASW) [2] to evaluate the quality of the resulted clusters.

For each datapoint i, we denote the average distance to other datapoints within the same cluster as $a(i)$. It can be regarded as an indication of how well datapoint i is clustered to this cluster. Then we find the i's closest neighbouring cluster, which is the one that has the smallest average distance to i. We then calculate the average distance from i to the members of its closest neighbouring cluster, and denote it as $b(i)$. It is an indicator of how well datum i is matched to its

neighbouring cluster. The silhouette of i is hence defined using $a(i)$ and $b(i)$ as follows:

$$s(i) = \begin{cases} 1 - \frac{a(i)}{b(i)} & \text{if } a(i) < b(i), \\ 0 & \text{if } a(i) = b(i) , \\ \frac{b(i)}{a(i)} - 1 & \text{if } a(i) > b(i). \end{cases}$$

$s(i)$ takes value in range $[-1, 1]$. When $a(i) \ll b(i)$, $s(i)$ approaches 1, which indicates that data-point i fits well in its own cluster. On the contrary, if $s(i)$ draws near -1, it means that the its closest neighbour cluster seems to be a better fit for datapoint i than where it belongs now. The average $s(i)$ over all data-points within a cluster is a measure of how tightly grouped the members of this cluster are. And the average $s(i)$ over the entire dataset is a measure of how appropriately the clusters are formed. ASW as well takes value in $[-1, 1]$.

Traditionally, the most appropriate cluster number k under a certain clustering algorithm can be identified by varying k to achieve the biggest ASW over the entire dataset. Same logic can be applied to the comparison of different clustering algorithms using a same distance metric. The algorithm with better clustering result shall lead to a bigger overall ASW value. This is how we compare the fitness of hierarchical clustering with Ward linkage, k-means and PAM. Dataset-wide ASW for `PingData` are visualized in Figure 20. We can see that k-means algorithm with $k = 2$ offers relatively confident clustering results. This observation holds as well true for traceroute data sets. Hence, this settings are used in the rest of this study: feature space, Eucilidean Distance (ED), k-mean, $k = 2$s.

4.4.3 *Clustering result interpretation*

# cls	`pingData`			`traceData1`			`traceData2`			`traceData3`		
	size	dist.	AWS	size	dist.	AWS	size	dist.	AWS	size	dist.	AWS
1	25	4.37	0.23	93	3.98	-0.21	76	3.02	0.41	31	4.35	0.07
2	75	2.61	0.54	7	2.17	0.35	24	4.58	0.08	69	2.90	0.41
AWS	0.46			-0.17			0.33			0.30		

Table 4: Summary of clusters characters on `PingData` feature space. Clusters are formed from each corresponding dataset. Meanwhile the intra-cluster distance, the intra-cluster ASW and the overall ASW are calculated over `PingData` distance. It is a compatibility test for clusters from traceroute datasets with `PingData`.

4.4.3.1 *Characteristics of achieved clusters*

Table 4 describes achieved clusters with several metrics: size of cluster, intra-cluster distance, intra-cluster ASW, and overall ASW at the bottom. Focusing on the column for `PingData` in this section (otherwise it would be confusing), we notice that two clusters of unbalanced size are obtained. Cluster 2 is much larger in size, yet with smaller intra-cluster distance. In accordance, the cluster algorithm is more confident of this cluster than the other, as the cluster ASW is much higher. This indicates that the members within cluster 2 demonstrate strong common features and are thus more closely placed to each other in the feature space.

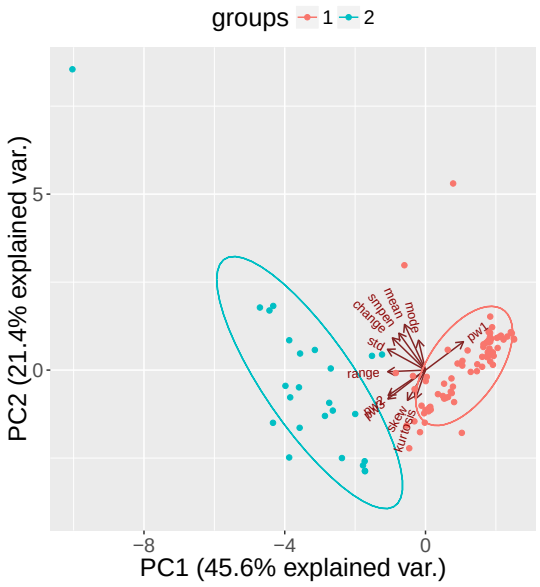
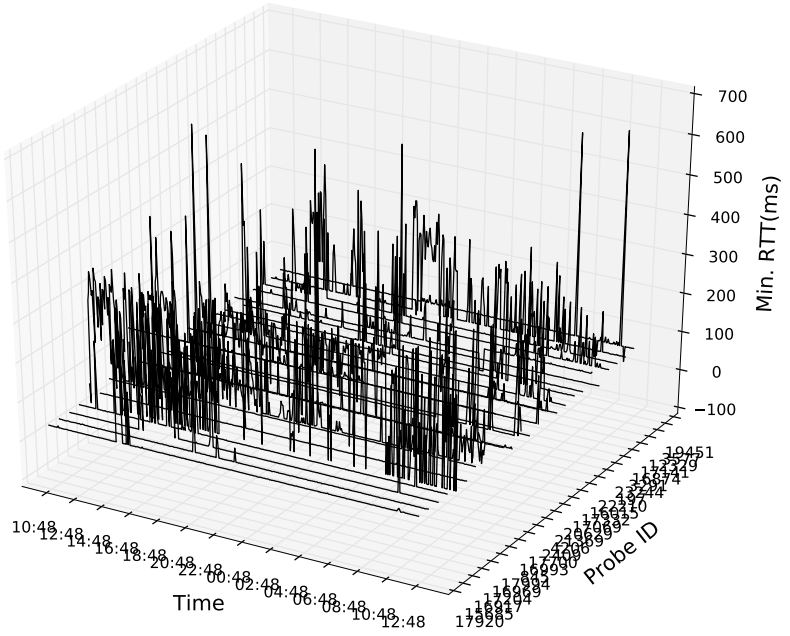


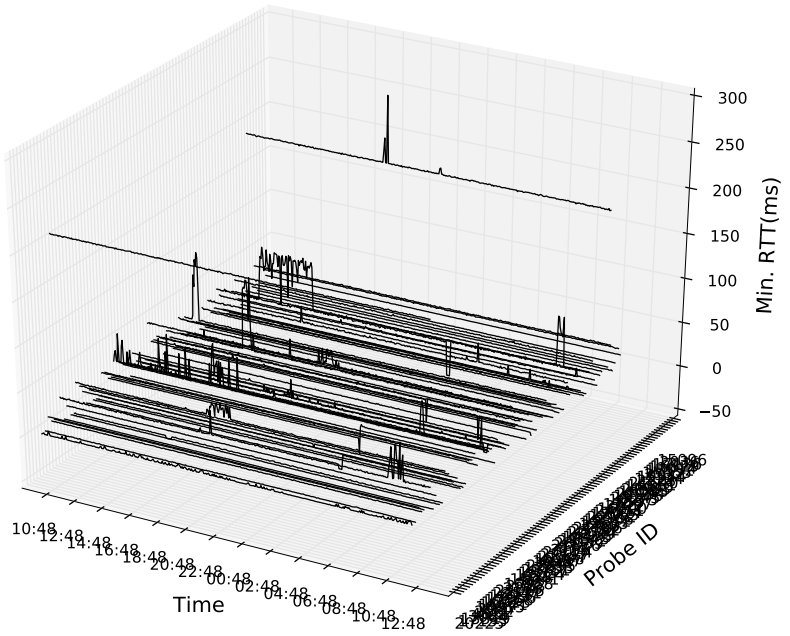
Figure 21: Projections of clusters on PCA features, `PingData`.

This "guess" can be straightforwardly observed from the projection of clusters on Principal Component Analysis (PCA) surface, shown in Figure 21. The contribution of each feature is as well indicated on the graph. We notice that selected features point to different directions on the PCA surface. This suggests that they form together a non-redundant (or not that much) description of the date set. As a consequence, no single feature takes dominant position in forming the clusters. Still, the tendency is that cluster 2 includes data points with little changes, small entropy and small standard deviation.

Figure 22 plots the RTT time series of these two clusters. Cluster 2, as expected, contains mainly time series with only a few variations and spikes. On the other hand, cluster 1 is composed of traces with large variations. This explains why the members of cluster 2 are more closely located to each other. It is because for a time-series to be smooth, there is only one form. while for it to be full of variations, there could be many possibilities.



(a) cluster 1.



(b) cluster 2.

Figure 22: End-to-end RTT series of cluster members from `pingData` dataset.

4.4.3.2 *Advantage of clustering*

What is also worthy of noticing is that cluster 2 actually tolerates traces with a few spikes and variation of small amplitude. It is actually an interesting feature. An alternative approach than clustering to

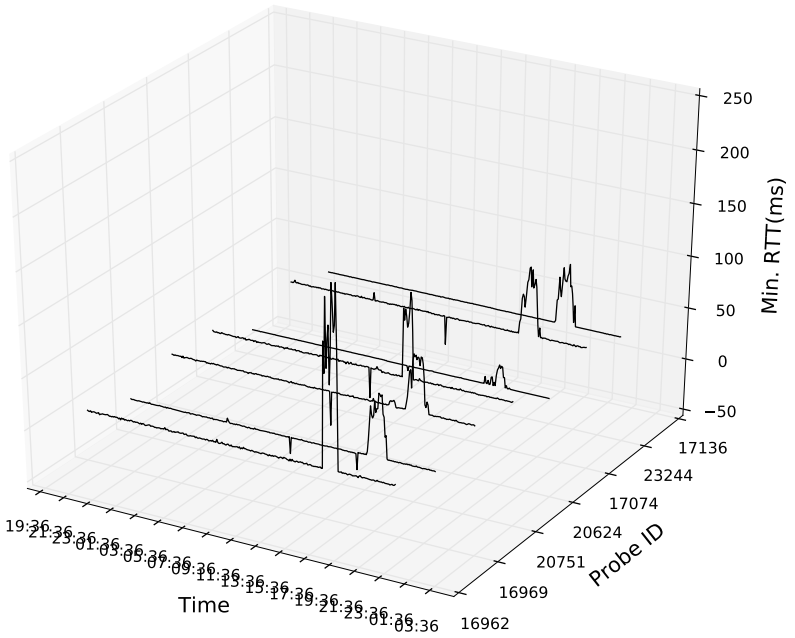


Figure 23: One cluster achieved when k = 12, `PingData`.

filter out "noisy" traces would be ranking these time series with one single metric. For instance, we could rank the RTT traces by their entropy and assume top x traces are "noisy" ones. The shorting-coming of such approach is two-fold. First, single feature might "wrongly" rank certain RTT time series. For example, entropy [71] and power spectral density are very sensitive to spikes. In networking uses, we believe that one or a few spikes in an RTT measurement shall not deteriorate an entire measurement time series, since RTT measurements can be impacted by many aspects other than path quality, e.g. load on end host or routers, within a short duration. Second, one has to define threshold values, i.e. the x, to deliberately separate data set into two or more parts. While with clustering, multiple features can be considered at the same time. Plus, the most appropriate number of clusters changes automatically with the input and reveals the inherent structure of the dataset. By setting a larger cluster number, one could achieve finer grained clustering results that convey subtler information. An example is given in Figure 23, where several traces of similar variation shape is grouped together.

4.4.3.3 *Implications for interdomain TE*

The two clusters of `pingData` are particularly meaningful to interdomain TE, where one might measure one single AS path with multiple end hosts. It shows that the resulted RTT time series can take very different shapes, even though the same AS path is concerned. One possible explication is that the IP-level path toward these end hosts

undergo various local issues. And some of these issues have obvious impact on end-to-end latency. If there were an AS-level issue, for instance congestion on an inter-AS link, it will be shared by all the RTT time-series. If certain RTT change is not common, it is then probably related to local issues that can not be surely avoided by changing an egress transit provider in sending out the traffic. Therefore, when comparing multiple RTT time series toward a same destination AS/prefix, the ones with fewer variations are supposed to be less impacted by these local conditions. They shall thus offer a more faithful view on the AS-level path performance. One another advantage of using smoother RTT time series in route selection is that the resulted routes are supposed to be more stable. Fewer RTT variations should potentially trigger fewer route changes.

Take-away

When clustering RTT time series in feature space, we arrive at two clusters that tell noisy traces apart from smooth ones without prior knowledge/assumption on the dataset structure. This reveals the inherent structure of the dataset that describes a same AS path with multiple RTT time series. A majority of these time series resembles each other and demonstrates least variations possible, while a handful of "outliers" may contain diverse additional variations. Considering the source of our data, our assumption is that some of the RIPE Atlas probes are continuously suffering from some local issues.

4.4.4 *Where do the additional RTT variations come from?*

pingData	traceData1		traceData2		traceData3	
	1	2	1	2	1	2
1	25	0	8	17	20	5
2	68	7	68	7	11	64

Table 5: Comparing cluster members resulted from different datasets. The number in each cell represent the number of common members share by the two clusters.

In this part, we tried to find out the origin of these additional RTT variations in cluster 1 of `PingData`, and their practical meanings in networking. To this end, we clustered, with the same method, the three datasets that are constructed from the first 3 hops in traceroute measurements. We test the compatibility of these cluster members with clusters of `pingData` in Table 5.

We observed that for `traceData2` (hop 2) and `traceData3` (hop 3), not only the cluster sizes arrived are close to those from `pingData`, but also a majority of cluster members overlaps. That is to say, us-

ing RTT time series till the 2nd hop or the 3rd hop, we ended up with similar clustering results as with end-to-end RTT measurements. This observation is further confirmed by Table 4, which describes the clusters from traceroute datasets on `pingData` feature space. Clusters from `traceData1` have the poorest overall ASW indicating that its clusters are not a good fit for end-to-end RTT time series. However, clusters resulted from `traceData2` and `traceData3` seem to be quite compatible with `PingData`, since the overall ASW is not far away from that achieved with `PingData` itself. Given that the clustering results of `pingData`, `traceData2` and `traceData3` are similar, we deduce that most variations in `pingData` actually comes from the link between the 1st and the 2nd hop.

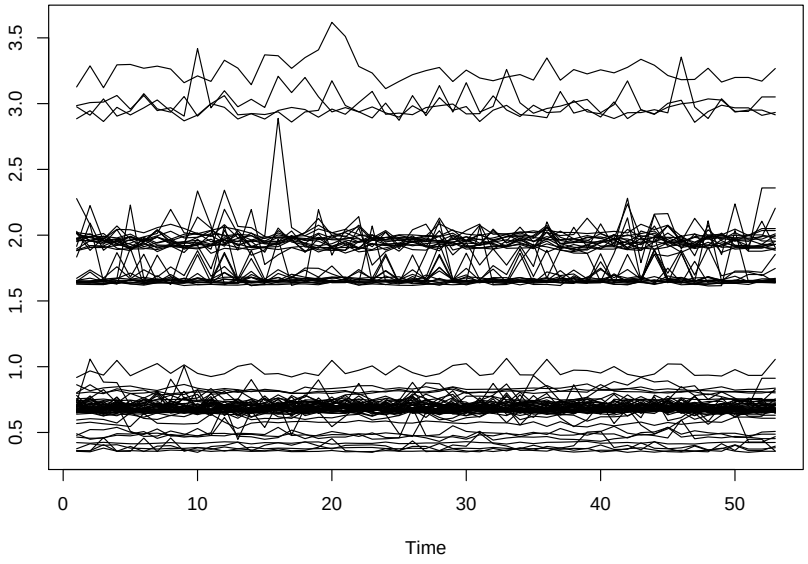


Figure 24: RTT (in msec) till first hop. The first hop is assumed to be the hop till the home router. Three different baselines are observed. This might due to differences in connection methods, hardware and firmware versions of Atlas probe and ISP home router.

In this specific case, the probes involved are mostly hosted in the residential network of an ISP AS3215. The first hop are generally the home router. This guess is indirectly confirmed by the observation that most traceroute records has `192.168.1.1` as the first hop address. As expected, RTT traces till the first hop are all pretty smooth according to Figure 24. We manually searched information regarding second hop addresses, e.g. `80.10.127.143` in `https://db-ip.com/80.10.127.143`. Results returned show that they are all access equipment of the ISP. It is thus logical to find most variation between the 1st hop (the home router) and the 2nd hop (ISP access equipment), as it corresponds to the ISP's access network.

Such finding is of course not a huge surprise.However, it suggests that with our clustering method, we are able to get rid of RTT measures that undergo severe access network problems.

Wrap-up

In this study, we datamined RTT time series between two ASes. We found out that RTT time series collected in this study demonstrate diverse variation shapes though one common AS path is measured. It confirmed that RTT measurements need to be "cleaned". We clustered these RTT time series by extracting several features as their data representation. Resulted clusters successfully separated noisy traces from smooth ones according to human intuition and expertise. Furthermore, we located the occurring location of most variations in the end-to-end RTT measurements by applying the clustering methods to the first hops of traceroute measurements. Our results confirmed the common sense that most variations come from the access network.

4.5 MULTIPLE RTT TIME SERIES WITH SYNCHRONIZED CHANGES

An RTT time series cluster, illustrated in Figure 23, contains multiple RTT measurements undergoing a similar shape RTT variation at the same time. This RTT change is not observed on other RTT measurements within the dataset. The implications is that a common part exclusive to these RTT measurements in Figure 23 could have caused this change. Inferring the location responsible for RTT changes in Internet is a an intriguing topic in its own right. Moreover, it contributes to better route selection logic, if achievable with only end-to-end delay and path measurements.

Clustering in feature space (Section 4.4.2) is however not the best option for the identification of RTT time series with similar shapes. It is because the features extracted, summarizing the entire time series, do not have the expressiveness over temporal structure. Therefore, we study clustering approaches where the data representation of RTT time series remain time series.

4.5.1 *Data collection*

In order to increase the chance of identifying RTT time series with shared change or shape, We collected ping and traceroute measurements from 170 RIPE Atlas probes hosted in European datacenters to DNS b-root from Jan. 18 to Jan. 24, 2016[4][5]. We cleaned the dataset by removing measurements with plenty of missing segments and time-

4 DNS b-root had single single instance at that moment.

5 IDs of these probes can be found at https://www.dropbox.com/s/6ai0aooxnubufma/pbid.txt?dl=0.

out measurements (see the Section 4.4.2 for detailed cleaning criteria). 128 probes remained. They are from 17 countries, 117 ASes and 120 prefixes. All these probes are equipped with the lasted v3 hardware.

The measurements from different Atlas probes are not strictly aligned. This brings inconvenience in comparing RTT changes in time across probes. We therefore tried to find a new set timestamps that minimizes the distance from the initial measurement timestamps to it, across all probes. The resulted average distortion in time is $55.01sec$ per datapoint, being much smaller than the measurement interval, i.e. $240sec$. We aligned all the RTT time series in the dataset by enforcing this new set of timestamps. For several rare moments when no measurement data was available, we padded them with the closest RTT measurements. Till this point, we fabricated a dataset of RTT time series with aligned timestamps and equal length.

4.5.2 *Data representation*

In order to capture the temporal structure of time series, we tired following data transformations that result still in time series.

RAW RTT The aligned and padded raw RTT measurements are used as them are for the sake of comparison. This representation is denoted as **RTT** later on.

SEGMENTS BY CHANGEPOINT DETECTION In the purpose of filtering unnecessary variations in the raw RTT measurements, we applied changepoint detection to them. The operation cuts each RTT time series into segments of different characteristics. We then simplify each RTT time series with the mean value of each segment without changing its total length. So achieved time series are denoted as **Seg**. The changepoint detection is performed with R package *changepoint*, version 2.2.1 [91]. The min RTT is first subtracted from each RTT time series. Then we assume Poisson distribution during the change detection. We later on dedicate Chapter 5 to the change detection for RTT measurements, where more technical details can be found.

Z-NORMALIZED TIME SERIES This process produces zero mean unite variance time series. It is a very common data pre-processing in time series clustering [27, 108]. It helps remove level differences that is less relevant to Time series shapes. However, under this representation, RTT time series with large and frequent changes resemble a lot those with small variations, to an extent that they become hardly distinguishable. Due to this undesired feature, the actual clustering result with z-normalized time series is relatively meaningless. Therefore, we no longer discuss this data representation.

MODE-CENTERED AND PIECE-WISE SCALED TIME SERIES In this transformation, we first center each RTT series around its mode by:

$$x_m = x - \tilde{x},$$

where $\tilde{x}$ is the mode of time series x. Then we scale the x_m time series by the following piece-wise function:

$$F_s(x) = \begin{cases} 0 & 0 \geqslant |x| < 10, \\ x - 10 & 10 \geqslant |x| < 60\ , \\ 10 \times \log_2(|x_i|) - \beta & |x| \geqslant 60, \end{cases}$$

where $\beta = 10 \times \log_2 60 - 50$. It is simply a term to make the $F_s(x)$ continuous in value. The intuition behind these operations are 1) since networks tend to have a dominant configuration and are most of the time free of congestion, its is the variations around the most popular value that defines the shape of the RTT time series; 2) for little variations less than 10 msec, we would like to consider them as insignificant noises and filter them; for moderate deviations, say in the range of $[10msec, 60msec)$, are changes that we would like to take into account during clustering, and are thus conserved linearly; while for super large changes, sometimes outlier values, greater than 60 msec, we would like to suppress their impact on clustering results. We refer to the resulted time series as **MP**.

4.5.3 *Distance measure*

Wang et al. [88] performed a comprehensive study on distance measures for time series data. They showed no distance measure is significantly better (in terms of the accuracy finding 1 closest neighbour) than Dynamic Time Wrapping (DTW) on a majority of data studied [101]. Meanwhile ED is of similar performance as Dynamic Time Wrapping (DTW) when training sample are big enough. Therefore, we consider these two well studied distance measures.

EUCLIDEAN DISTANCE is a non-elastic distance measure, which can only be applied to two time series of same length. All the above presented data representation contain time series of same length, thus applicable.

DYNAMIC TIME WRAPPING is an elastic distance measure, which allows it to handle time series of different length. Yet, Ratanamahatana and Keogh [33] showed that there is no significant difference in terms of classification accuracy for DTW when handling time series of different length and re-interpolated series of same length. Therefore, padding shall bring few impact when employing DTW. Different

from ED, DTW stretches or compress the two time series so that one resembles the other as much as possible. That is, one value in a time series can be aligned/matched to one or multiple values of arbitrary positions in the other time series, so that the accumulated distance between all aligned/matched points are the smallest. Constraints can be as well specified in finding such alignments, or in other words wrapping path. We give only intuitive explanations here. More details of DTW and window constraints can be found in [1, 30, 55].

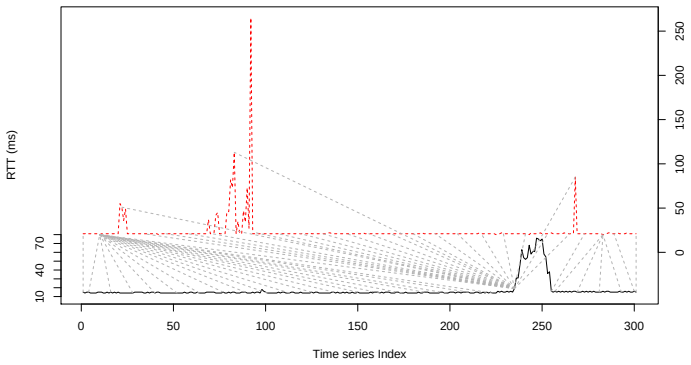


Figure 25: Matching/aligning of two RTT series. The black line is the query series, probe id 16969; the red dashed line is the reference series, probe id 16987. Dashed lines between these two series illustrates how values in query is matched to ones in the reference. The distance resulted is 3457.

Figure 25 is an example of how DTW aligns a query RTT series to a reference series [6]. We can see matches between RTT datapoints that are far away from each other.

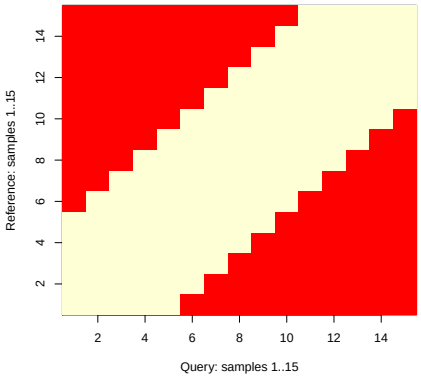


Figure 26: A Sakoechiba window [1] of 4 in size. Yellow part is the allowed alignment/matching area.

6 DTW is not a symmetric distance measure, in the sense that if query and reference time series are swapped, a different distance value can be produced.

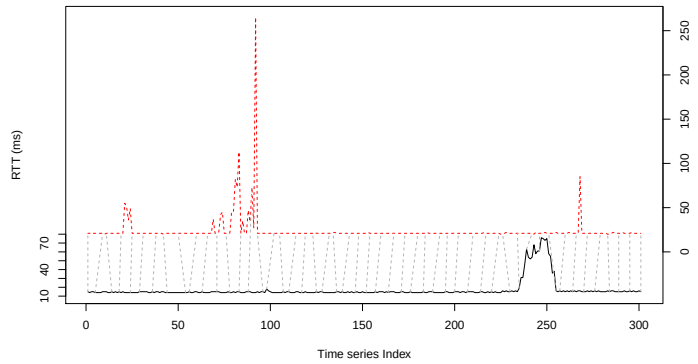


Figure 27: Matching/aligning of two RTT series with window. The black line is the query series, probe id 16969; the red dashed line is the reference series, probe id 16987. Dashed lines between these two series illustrates how values in query is matched to ones in reference. The distance resulted is 5181.

However, matching variations far away in time, such as the case in Figure 25 is not helpful for the search of synchronized RTT changes across probes. Therefore we apply a window restricting the range of alignment/matching between two RTT time series. The window used is visualized in Figure 26. With window size equaling 4, a data point in one time series can no longer be match to a datapoint 5 index or further away in another time series. 5 index in our RTT measurement settings means 20 min, which is already a large range. With the window restriction, the new alignment for the two time series in Figure 25 is now given in Figure 27. Not surprisingly, the distance becomes larger, which we regard as a more faithful measure of the dissimilarity of the two time series.

4.5.4 *Clustering results*

We used PAM instead of k-means in this study because 1) PAM is more robust than k-means, for its clustering result is less impacted by outlier observations [7]. 2) the mean 'value' of a set of time series does not have any practical meaning, while the medoid is a plausible choice [97].

The clustering result using ED is not very meaningful, and will not be discussed later on. The resulted cluster size is highly uneven. It forms always, regardless k setting, 1)one cluster containing a majority of RTT time series without observably similarity among them, and

7 The only difference between PAM abd k-means lies in the fact that k-means uses the mean value of cluster members as the prototype of that cluster, around which the cluster member are iteratively adjusted. Meanwhile, PAM, a.k.a. k-medoids uses the cluster member that minimizes the total distance between itself and all other cluster members as the cluster prototype.

2) several small clusters having strictly matched RTT spikes. We believe it is because the distance definition is too stiff. Strict match of big values lowers significantly the distance that otherwise would be accumulated.

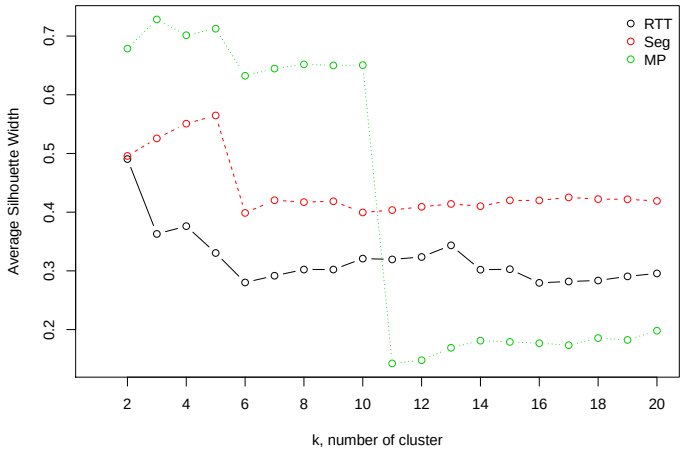


Figure 28: ASW over the entire data set with varying k.

We thus mainly compare the clustering results using DTW and PAM over the three data representations: **RTT**, **Seg** and **MP**. The overall ASW with varying cluster number k is given in Figure 28. The curve of **MP** demonstrates a very typical step-wise shape, evidence of strong cluster structure when $k \leqslant 10$ and how this structure is broken when more clusters are enforced. Among the three representations, the clustering algorithm is most confident of the clusters identified using **MP**.

To understand the resulted clusters with the three data representation, we first look at the raw RTT time series of these cluster members. With **RTT**, we found that clusters are essentially formed according to by their raw RTT level. This suggests that the mean value of raw RTT time series dominates the distance calculation. The resulted clusters of **MP** and **Seg** have pretty similar members when $k = 5$ and the overall ASW is high. One cluster contains exactly the same members under both representations. The raw RTT time series are plotted in Figure 29. It is obvious that all these RTT time series experienced a very alike RTT change over a same period. Both representation methods achieved an almost 0.8 ASW under DTW. This implies that the clustering algorithm is extremely certain about this result. The presence of this cluster is a strong evidence that the methods experimented in this study are indeed capable of identifying groups of time series with same RTT changes. Other clusters of **MP** and **Seg** are as well of relevance, suggesting again that they are very effective ways of extracting temporal structures in time series.

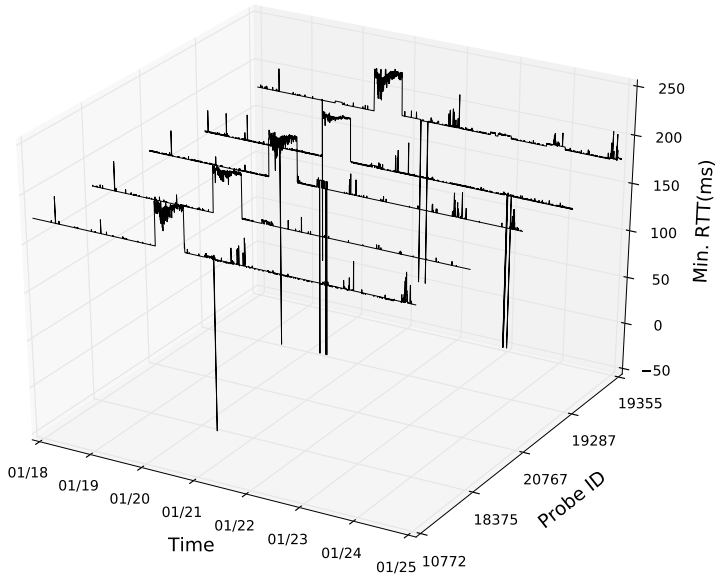


Figure 29: A common cluster in **MP** and **Seg** when $k = 5$.

4.5.5 *Network implications of shared RTT variations*

The 5 RTT time series in Figure 29 experienced a big RTT increase of at least 40msec for nearly a day on Jan. 20th, 2016. We seek to find out what happened from a network aspect.

COMMON AS HOP The only intermediate AS hop shared by all the 5 probes is AS6939. Meanwhile, other 94 probes in the dataset that pass through AS6939 didn't experience this RTT change. Therefore, we can not conclude that AS6939 is the cause. 3 of the 5 probes in the cluster pass through IXP AS8674, right before entering AS6939. The other 2 have unannounced IP addresses interconnecting AS6939.

ROUTING CHANGES No AS path change is witnessed during this period. However, there are IP level path changes within AS6939 that happened at the same moment as the shared RTT changes [8]. RTT till the hop entering AS6939 experienced the most significant increase during the end-to-end delay increase.

POSSIBLE SCENARIO Our guess was that IXP AS8674 changed the next-hop in reaching AS6939 out of some reason and the impacted traffic fell on a congested inter-AS link. The Network Operation Center (NOC) team of AS8674 kindly replied to our query and told us that they did "not see any special disturbance". Another possibility is that due to Non-disclosure Agreement (NDA), they were not allowed to re-

8 We are capable of distinguishing real IP path changes and those due to LB, more technical details in Section 5.

veal any information to non-member entities. In both cases, it turned out to be very difficult to identify which part of the Internet causes the shared change, using merely end-to-end measurements. Yet, capable of clustering in first place RTT time series with similar shapes does help narrow down the scope of possible causes.

4.5.6 *Limitations of time series clustering*

Time series clustering with **MP**/**Seg** and PAM offered very interesting results and helped us chasing down a specific case of RTT change shared by multiple RTT time series. However, it is still not ideal in the context of measurement-based TE for following reasons.

First, it won't scale. In order to obtain a distance matrix among n time series, $O(n^2)$ distance calculation is needed. The computation cost for clustering could thus be prohibitively high when there are thousands of RTT time series, one for each destination prefix.

Second, the interpretation of clustering results remain ad hoc, and thus hard to automatize. ASW of each cluster can serve as an indicator of cluster quality/relevance. Yet, what level of ASW should be considered significant enough is hard to justify. In previous analyses, we always resorted to raw RTT time series to judge if the clusters are indeed meaningful or not, before carrying out more specific analysis. On top of that, the clustering results do not state when RTT changes shared by the cluster members actually happen, preventing as well systematic investigation of relating network events at these moments.

Third, it is hard to transform clustering into an online process. Along the time, different parts of the Internet might cause RTT changes, and impact different sets of RTT measurements. Therefore different different clusters are expected over time. This requires the clustering methods to update the resulted clusters as new measurements flow in. The clustering methods studied in this chapter are clearly inapt.

To address the above issues, we found simplifying RTT time series via changepoint detection promising. Its power has already been demonstrated with the **Seg** data representation. It first outputs the moments when each RTT time series experiences significant changes. Based on this change-or-not binary status, we can then easily form groups of time series sharing the same change. With evolving groups, we can further track the causes for different RTT changes over time in the Internet. We develop this idea in Chapter 5 and 6.

5 CHANGE DETECTION FOR RTT MEASUREMENTS

ABSTRACT

There are two major motivations for the study on change detection for RTT measurements, both of which contributes to measurement-based intradomain TE. First, moments of important changes in RTT measurements can serve as trigger for route re-selection. Second, it helps group RTT time series undergoing same changes. This part will be covered in Chapter 6.

Change detection methods is a competent tool for RTT measurement processing. It is devised to detect significant changes in time series. Many domains has benefited from these methods, yet few effort was put on RTT measurements. It is thus unclear how well such methods work on RTT time series, and which method works the best. In this chapter, we first presented an evaluation framework for change detection on RTT times series, consisting of:

1. a carefully labelled 34,008-hour RTT dataset as ground truth;
2. a scoring method specifically tailored for RTT measurements. Furthermore, we proposed a data transformation that improves the detection performance of existing methods.

Finally, we investigated the change detection performance with regard to network events learnt through path measurements.

5.1 RTT CHANGES: THE TRIGGER FOR INTERDOMAIN TE

RTT measurements intervene in measurement-based interdomain TE at two phases. First, RTT measurements reveal the moments when route re-selection is needed. Second, RTT measurements serve as decision making material in route re-selection. We discuss in this chapter the usage of RTT measurements during the first phase.

Moments when route re-selection is needed are basically when the performance on certain AS paths change. The challenge for performance change detection mainly comes from two aspects. First, RTT measurements are noisy. Many factors along the measured path may contribute to the variations of end-to-end delay, e.g. end-host load fluctuation, bursty traffic, etc. This requires change detection methods to tolerate noises such as short living spikes, yet remain sensitive to events that really matter such as persistent congestion. Second, the delay characteristics on different paths can differ a lot. It is thus desirable to detect changes for these time series without path/destination dependent parameters. Many commonly seen practices fail to meet the above listed requirements.

PREFERENCE OVER SMALLER AVERAGE RTT Repeated RTT measurements over time can tell which transit provider offers the smallest delay in average toward a specific destination. Employing this transit provider for that destination can thus ensure a good overall performance. However, this approach falls short in handling transient RTT augmentation during congestion.

RTT THRESHOLD One can react to transit RTT increase by comparing the instant RTT measurements to a hard-coded threshold. Once the RTT surpasses the threshold value, route re-selection can be triggered to look for a better performance path at that specific moment. The drawback of this method is that the appropriate threshold value depends on each individual destination, and are thus not trivial to configure.

THRESHOLD OF RTT CHANGE AMPLITUDE In order to bypass the RTT threshold configuration that may vary among destinations, one can instead apply a single global threshold for the amplitude of RTT changes. This change amplitude threshold could be absolute values such as 30 msec or proportional to the RTT level, say 30% of previous measurement. The shortcoming of this approach comes from the fact that RTT measurements (via ICMP or TCP) could be pretty noisy. Shorting living variations with large amplitude are not rare. Their presence might lead to unnecessarily frequent route re-selection.

SMOOTHING THE RTT MEASUREMENTS One straightforward way to be robust when handling noisy RTT measurements is to smooth it before usage. EWMA can be applied over a couple of past measurements [49]. However, the application of such filters introduces additional parameters to be tuned, and likely in an ad hoc manner.

5.2 RTT CHANGE, NETWORK EVENTS AND TE

In this section, we summarize previous studies on RTT variations and their relation to network events. We explain why certain RTT analysis method adopted in the mentioned works is not a good fit for TE uses.

It is generally agreed that inter-domain routing changes impact the RTT level greatly. Pucha et al. [44] showed that inter-domain routing changes cause larger median RTT variation than intra-domain ones. Rimondini et al. [94] confirmed that 72.5% BGP route changes in their study are associated with RTT change. Similar observations were made in a large CDN, where inter-domain routing changes are responsible for more than 40% of severe user experience degradation [81].

Intra-domain events are no less important. Pucha et al. [44] discovered that intra-domain path changes can cause RTT changes of comparable amplitude as inter-domain ones. Moreover, they pointed out that it is intra-domain path changes, not congestion, that are responsible for the majority (86%) of RTT changes. A different claim was however made by Schwartz et al. [66]. They found out that most RTT variation is rather within paths (i.e. due to congestion) than among paths (i.e. due to path changes).

Conflicts in previous works could be caused by the difference in locations from where measurements were launched. For instance, Chandrasekaran et al. [100] observed that AS path changes only have marginal impact on RTT in the core of Internet, while previous works [44, 66] include as well access networks. Results might as well change over time. For instance, the "flattened" Internet topology, the increasing amount of traffic in private CDN over the last decade [65, 73] might have changed the characteristics of path change and congestion, and consequently how they impact RTT.

Bearing this in mind, we would like to emphasize the efforts on methods and tools. Beyond one-shot observation or analysis on a specific dataset, they allow iterative analysis over time.

The discussion and discovery of previous works are enlightening, yet their methods for RTT measurement processing can hardly be applied to intra-domain TE. In [44, 66, 100], RTT measurements are first grouped by underlying paths; impact of path changes are then estimated through comparison of associated RTT statistics, e.g. percentiles. However, in a practical TE system sketched in Figure 3, RTT are measured with higher frequency than paths. The underlying rea-

sons are threefold. First, RTT measurements are in general less costly. Considering the potential number of destination to be monitored (see Section 3), path measurements are better limited. Second, smaller RTT is the objective of TE, we thus have the incentive to follow its evolution closely. Meanwhile, path is just a result of optimization. Third, RTT changes generally happen more frequently than path changes. One important reason is congestion. Grouping RTT measurements by path changes can not shed light on the presence of such events. Therefore, we need to explore methods that can identify inherent RTT changes, instead of relying on external measures such as path changes to describe the variation of transmission performance.

5.3 CODE SPACE AND DATA COLLECTION

The main code space for work in this chapter is made public on Github with documentation: `https://github.com/WenqinSHAO/rtt`. The implementations of proposed methods are decoupled from the context of this project, and thus can easily be employed elsewhere.

We applied our methods to RIPE Atlas built-in measurements [142] and performed data analysis. These measurements are openly available so that the results of this work can be reproduced by other researchers or compared to alternative approaches. We collected RIPE Atlas built-in ping and traceroute measurement toward DNS b-root (measurement ID 1010, 5010) from 6029 v3 probes located in 2050 different ASes, 153 countries from 2016-10-01 to 2017-01-01 [1]. 184,358,516 ping and 23,507,910 traceroute measurements are collected and analyzed. The traceroute measurements flowed through 3036 ASes, 120 IXPs, containing 10,720 different AS paths.

5.4 CHANGEPOINT DETECTION

In this section, we introduce changepoint detection and its previous application on RTT measurements.

5.4.1 *A primer on changepoint detection*

The moments that cut a time series into segments of different characteristics are called *changepoints*. The problem of detecting the most appropriate changepoints is known as changepoint detection or change detection. With such method, changes in RTT measurements can be detected and quantified without the help of path measurements.

One common approach to changepoint detection is to translate the quest of finding the best changepoints into the following optimiza-

1 Measurements to other destinations might as well do. The fact whether the destination is anycast or not is of few importance in this work. The focus is on the method rather than on a specific dataset.

tion problem [2]. Assume we are given a sequence of measurement data, $y_{1:n} = (y_1, y_2, ...y_n)$. We expect changepoint detection method to produce m ordered changepoints, $\tau_{1:m} = (\tau_1, \tau_2, ...\tau_m)$. τ_i is the position of i^{th} changepoints and takes value from in $1, .., n-1$. These changepoints are given in an ordered way such that $\tau_i < \tau_j$ if and only if $i < j$. We define $\tau_0 = 0$ and $\tau_{m+1} = n$. Together with the m detected changepoints, they cut $y_{1:n}$ into $m+1$ segments, with the i_{th} segment containing $y_{\tau_{i-1}+1:\tau_i}, 1 \leqslant i \leqslant m+1$. For each segment, a cost is calculated. The cost can be seen as an inverse measure of the appropriateness of the the segment. The detection method seeks to minimize the sum of all the segments' cost (so that the produced $m+1$ segments/m changepoints are the most relevant):

$$\sum_{i=1}^{m+1} [C(y_{\tau_{i-1}:\tau_i-1})] + \beta f(m).$$

Here C is the cost function that measures the fitness of each segment. $\beta f(m)$ is a penalty function to prevent over-fitting. These two functions are the major parameters to be tuned.

One commonly used cost function is the minus of the maximum log-likelihood of the segment following a certain distribution [5, 9, 77]:

$$C(y_{s:t}) = -\max_{\theta} \sum_{i=s}^{t} \log f(y_i|\theta).$$

Here $f(y|\theta)$ is a density function with distribution parameter θ. For example, if we assume a Normal distribution, then we have $\theta = (\mu, \delta^2)$: $f(y|\theta) = f(y|\mu, \delta^2) = \frac{1}{\sqrt{2\delta^2\pi}} e^{-\frac{(y-\mu)}{2\delta^2}}$. The θ over the segment is obtained through maximum-likelihood estimation, assuming the distribution type. A smaller cost indicates the segment is more likely generated with the specified generative model (distribution type). It is the parameter change, not the model change, that the change detection actually identifies. With the above formulation, the choice of cost function is restrained to the choice of distribution types. Currently supported ones in [91] are: Normal, Exponential, Gamma and Poisson. However, in practice the exact distribution type could be unknown a priori, or different from the supported ones. Or the generative model itself (not just the parameters) can change over time. A recent progress thus proposes a cost function that is based on empirical distribution likelihood, where the specification on distribution type is not necessary. It is thus a non-parametric method [112].

When it comes to the penalty function, $f(m)$ is generally increasing and linear to the number of parameters introduced by m change-

2 Other formulations exist. A wider literature can be found in [70, 112]. We focus on this approach since it has well maintained libraries that prevent potential issues regarding the implementation [91, 112].

points: $m + (m+1)\dim(\theta)$ (m changepoints and $m+1$ segments) [3]. Common choices of β are information criteria, such as Akaike's Information Criterion (AIC) with $\beta = 2$, Schwarz Information Criterion (SIC, also known as BIC) with $\beta = \log n$, Hannan-Quinn Information Criterion with $\beta = 2\log\log n$, and Modified BIC (MBIC) with $\beta f(m) = -\frac{1}{2}[3f(m)\log n + \sum_{i=1}^{m+1}\log(r_i - r_{i-1})]$, where $r_i = \tau_i/n$ [48]. We have MBIC > BIC > Hannan Quinn. Note that larger penalty value leads to less sensitive detection.

5.4.2 *Application of changepoint detection to RTT measurements*

Among the extensive studies on change detection methods and their applications in various domains [46, 48, 53], Rimondini et al. [94] are among the first to employ change detection in network RTT measurement analysis. However, they tuned the detection sensitivity in a way that the detected changes correlate best to the BGP route changes toward the measured destination prefix among other randomly selected prefixes. This approach risks ignoring the RTT changes due to intra-domain changes and congestion. Plus, such tuning is potentially required for each individual destination, thus hard to scale. To achieve more general approach decoupled from path measurements, we propose in next section an evaluation framework for the selection and the calibration of change detection methods over RTT measurements.

5.5 EVALUATION FRAMEWORK FOR CHANGEPOINT DETECTION ON RTT MEASUREMENTS

Which method (among the wide variety of existing ones) is the most appropriate for Internet RTT time series is still not stated. Moreover, many changepoint detection methods are parametric. Identifying the best settings for these methods remains as well challenging. One fundamental issue in addressing the above problems is the lack of an evaluation framework.

An evaluation framework quantifies the performance of a certain detection method over a reference dataset. With quantified evaluation, different settings of a same method, or different methods can be compared and tuned to delivery the best detection results. Naturally, an evaluation framework should be composed of two parts: 1) datasets of "ground truth", 2) a scoring method.

Dataset of "ground truth" is not only a set of RTT time series that are representative of the delay characters over Internet. It should as well carry labels indicating the moments of change in these time series. We are not aware of any such dataset that is publicly available as

3 $\dim(\theta)$ is the dimension of θ. In the case of Normal distribution, $\dim(\theta) = 2$.

of this writing. We explain in Section 5.5.2 how we construct a ground truth dataset with great care.

As for the scoring method, it quantifies the similarity/difference between the "ground truth" and the detected changepoints. We explain in Section 5.5.1 that classic true/false positive classification is too rigid for both manual labeling and changepoint detection. We further explore and address the challenges of comparing two sets of timestamps with time shift tolerance.

5.5.1 *Scoring methods*

5.5.1.1 *True/False positive for changepoint detection*

A natural way to assess the difference/similarity of two set of moments in time is to classify each detected moment of change into two categories: true positive, if the moment is as well a ground truth changepoint; or false positive if otherwise.

The ratio between true positive and the total number of detected changepoints is called precision. It is a measure of the relevance of the detection results with regard to ground truth. The ratio between the true positive and total number of ground truth changepoints is called recall. It is a measure of the coverage of the actual changes that are successfully identified by the detection method. Precision and recall, both metrics are important to describe the performance of a detection method, since there is often a trade-off between them when tuning the detection sensitivity of a certain method. If a method is over-sensitive (extreme case: all the moments in a time series are marked as change), it might achieve a good recall, however the precision must be fairly poor. On the other hand, if the detection method is too prudent, the precision could be satisfying, meanwhile it probably leaves a lot of actual changes go undetected, thus poor recall. A good detection method thus shall excel in both metrics.

Above is the basic idea of true/positive scoring method. It is often used in classification problems. Changepoint detection can as well be regarded as one. The ground truth data set labels each timestamps of a given time series as a changepoint or not. Meanwhile, a changepoint detection method classifies each timestamps into the same two categories: change or not. And the evaluation by true/false positive compares the labels, the ground truth one and the detected one, for each timestamps. Therefore, the described scoring method naturally fits in and is indeed expressive.

5.5.1.2 *Tolerance of time shift*

Yet, changepoint detection is still a bit different from classification, just as timestamps are not like categorical data. In changepoint detection for time series, we seek to mark the moments of change in

time. If the marked moment falls exactly on a same timestamp of a ground truth changepoint, that is perfect. Categories match for the same timestsamp across the two sets. If the marked moment is too far away from any ground truth moment of change, then it shall be considered as a false alert. If the marked moment is just a few datapoints away from a certain ground truth changepoint, that is actually still OK. Such tolerance in time comes from two aspects. From a practical point of view, a shifted detection within a reasonable time range is still useful for TE. It is still more relevant than a false alarm far away from any true changes, as well as more informative than a miss alarm saying nothing about the occurring change. From a methodological point of view, slightly shifted detection just happens when the exact moment of changes are blurred by noises. Plus, when generating ground truth labels, we will see in Section 5.5.2, it is very hard to completely avoid slight errors, such as shift by several datapoints when putting the labels, due to human factors. This suggests that even the ground truth dateset is not extremely strict or certain about the position of change moments. Therefore, a certain level of time shift tolerance is actually required in evaluation, avoiding underestimate potential meaningful detection. Such tolerance in time implies that two timestamps labeled as changepoint across the two sets shall be allowed a match, as long as the distance in time between them is within a reasonable range.

5.5.1.3 *Optimal matching between ground truth changepoints and detected moments of change*

However, there are still ambiguity concerning how to introduce a tolerance window to true/false positive scoring. In order to facilitate the demonstration, we introduce following notions.

We assume ground truth $T_{1:k}$ containing k positions in $y_{1:n}$ indicating moments of actual change, while $\tau_{1:m}$ is the output of changepoint detection. A classic True Positive (TP) is a $\tau_j, \exists T_i \in T_{1:k}, T_i = \tau_j$. If we were to introduce a window of size w, we have to modify the definition of TP. But how?

One possibility is to apply the window to each ground truth changepoint. That is for each T_j, all the detected moments of change $\{\tau_i | T_j - w \leqslant \tau_i \leqslant T_j + w\}$ shall be considered TP for providing useful information (help detecting T_j) according to the definition of tolerance window. However the definition of recall becomes confusing when there are multiple detected changepoints classified as TP for a single ground truth changepoint. It is because the total number of timestamps in all the TP sets no longer strictly represents (is actually equal or larger than) the total number of ground truth changepoints that are successfully detected. The total TP timestamp numbers could even be larger than the total number of ground truth, which leads to a recall value larger than one. Meanwhile, it is unnecessary in practice to have

multiple alarms informing the occurrence of one same change. Therefore, it is not ideal having multiple τ matched to a T for the sake of tolerance window.

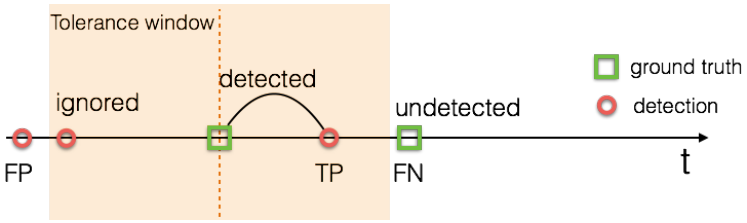


Figure 30: A matching dilemma between detected moments of change and two ground truth changepoints.

One cure to the above approach is to pick only the τ that minimizes the temporal distance to each T, if multiple are within the tolerance range. The remaining shall be regarded as false positives for not being the most relevant to any ground truth changepoints. Another issue occurs when the tolerance range of two ground truth changepoints overlaps. And a τ falls in the tolerance range of both ground truth timestamps. Which one should this τ detect? A such dilemma is illustrated in Figure 30. The tolerance range of the first ground truth moment is indicated in light orange in the graph. Two detected moments of change fall in this range. The later one is regarded as a TP for being the closest match to the first ground truth moment. Meanwhile, this changepoint from detection is as well in the tolerance range of the second ground truth changepoint. However, it can no longer be used to detect the later ground truth event, thus leaving the later undetected. It is not necessary an optimal arrangement of matching between ground truth and detection events. Because both ground truth has at least one detected moment of change within their window, yet one of them is left undetected.

The realization of the above dilemma made us question what is an optimal matching from an overall view. An reasonable formulation could be one that first maximizes the number of TP, i.e. matching (one-to-one) between ground truth and detection timestamps as long as the tolerance window allows. Then the it minimizes the total/average distance between these matched pair of timestamps to maximize the relevance of the matching. More formally, an optimal mapping between $T_{i:k}$ and $\tau_{i:m}$ with shift tolerance w is defined as $MP = \{(T_x, \tau_y)\} \mid |T_x - \tau_y| \leqslant w\}$, as the one that first maximizes $|MP|$, the size of MP, and then minimizes $\sum_{(T_x,\tau_y)\in MP} |T_x - \tau_y|$.

A convenient way to find the MP is through the construction of a bipartite graph $G = (V \cup W, E)$, illustrated in Figure 31. $V \cup W$ are the vertices of the graph. They are made of the ground truth

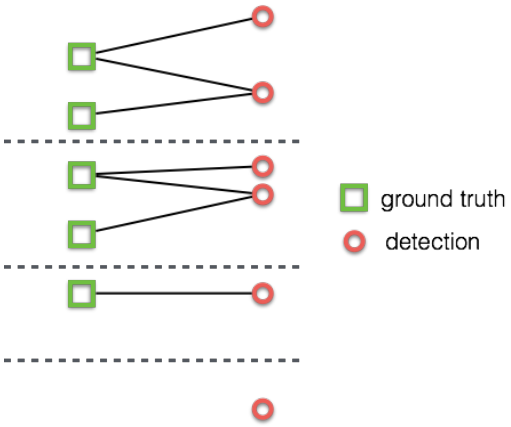


Figure 31: Constructing a bipartit graph from ground truth and detected changepoints. When the distance between a ground truth and detected changepoint is equal to or smaller than the defined tolerance window size, there is an edge connecting them. The weight of this edge equals to the distance in time between the two nodes.

and detected changepoints, i.e $V = T_{i:k}$ and $W = \tau_{1:m}$. Edge E is composed of all ground truth and detected change moments pairs the distance between which are no larger than the tolerance window, i.e. $E = \{(p, q) \mid |p - q| \leqslant w, p \in V, q \in W\}$. The cost of each edge is defined as the distance/shift in time between its two extremities $C(e) = |p - q|, e \in E$. Once the graph is constructed, the problem of finding the optimal mapping MP is translated into a well studied task of finding the *minimum cost maximum-cardinality matching* of G.

The Hungarian matching algorithm, as well known as the Kuhn-Munkres algorithm, is an established way of solving *minimum cost maximum-cardinality matching* problem [4]. The complexity of the algorithm is cubic to the number of the vertices in the graph. When there are a large number (e.g. > 100) of potential changes to be detected, it can thus take a while to compute. A performance boost is though possible due to the presence of the tolerance window. An edge is not going to connect two vertices, each from one part of the graph, that are far away in time. This feature allows us to cut the graph into much smaller disconnected sub-graphs. The cutting is illustrated in Figure 31 by the dotted lines. For positive numbers, the sum of cubic is always smaller than the cubic of number sum. Applying the Hungarian algorithm on disconnected sub-graphs can therefore reduce the amount of calculation.

4 An explanation to how the algorithm works would be lengthy and deviate the discussion to a direction that is less original. Interested reader can find a well written and illustrated guide to it via this link: https://brilliant.org/wiki/hungarian-matching/.

Once the MP is identified for G, the definitions for TP, precision and recall all come up naturally. For each detection τ_j, if $\exists m \in MP, \tau_j \in m$, it is regarded as a TP, otherwise as a FP (false positive). All the ground truth without matched detection $\{T_i \mid \nexists m \in MP, T_j \in m\}$ contributes to FN (false negative). Precision of the changepoint detection method, defined as $\frac{TP}{TP+FP}$, can be interpreted as the fraction of detection that is relevant or useful. Recall, defined as $\frac{TP}{TP+FN}$, can be regarded as the fraction of all ground truth change points that the method can successfully detect.

5.5.1.4 *Ground truth changepoints with weights*

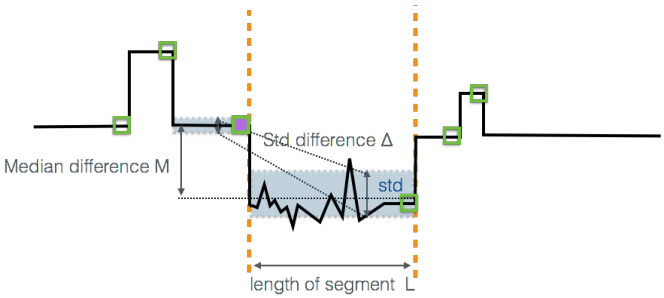


Figure 32: An illustration of how each ground truth (green square) moment of change is weighted. The example focuses on the while with a purple filling.

Each detected change may be treated equally (e.g. followed by root cause analysis, route re-selection), yet not all ground truth RTT changes are equally important in practice. Some changes could be for example relatively short living or small in amplitude. We thus propose to weight each ground truth changepoint T_i according to the follow three elements illustrated in Figure 32 for the purple filled ground truth changepoint:

1. the length of RTT segment following T_i, i.e. $T_{i+1} - T_i$;
2. the RTT level difference across T_i, denoted as M_i;
3. the RTT volatility difference across T_i, denoted as Δ_i.

More formally for each $T_i \in T_{i:k}$, with $T_0 = 1, T_{k+1} = n$, we define:

$$M_i = |Median(y_{T_{i-1}+1:T_i}) - Median(y_{T_i+1:T_{i+1}})|.$$

We use median instead of mean in the purpose of reducing the impact of abnormally large RTT measurements. We define:

$$\Delta_i = |Std(y_{T_{i-1}+1:T_i}) - Std(y_{T_i+1:T_{i+1}})|,$$

as the measure for variance change across the changepoint. We define empirically the weight associated to each $T_i \in T_{1:k}$ as:

$$\Omega_i = MAX(\log_2 \frac{T_{i+1} - T_i}{\rho}, 0) \times (M_i + \Delta_i).$$

Here ρ is a threshold for RTT segment length. If T_i leading to an RTT segment shorter than ρ, we ignore it in calculating $Recall$. The intuition behind this weighting is that RTT changes of large level or volatility are in practice regarded as more important. ρ and w tolerance window are set to $8min$ in this work, corresponding to two ping measurement intervals.

We can henceforth formulate a 'weighted' version of the $Recall$ metric to better reflect the operational importance of detected RTT changes: $Recall_W = \frac{\sum_{i,T_i \in TP} \Omega_i}{\sum_{j=1}^{k} \Omega_j}$.

Precision and recall are both important in evaluating the performance of a detection method. To consolidate them into one single metric, we used the notion of F_2 score. It weights recall twice as important as precision: $F_2 = (1 + 2^2) \times \frac{Precision \times Recall}{2^2 Precsion + Recall}$. The practical implication of this choice is that handling some FPs is less unwanted than missing out some important RTT changes.

5.5.2 *Ground truth dataset*

5.5.2.1 *Dataset quality control*

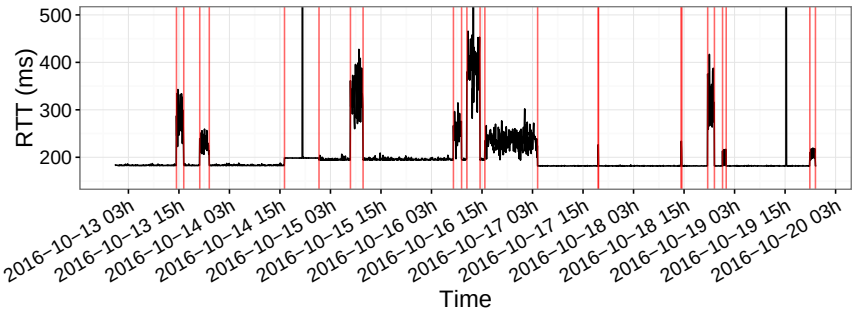


Figure 33: First 2500 Datapoints of an artificial RTT time series (one datapoint every 4min). Red vertical lines correspond to generated changes.

In order to determine which detection method works the best over RTT timeseries, a dataset with *a priori* labeled moments of RTT change is required, serving as ground truth. It's quality is essential to the credibility of evaluation results.

There are two approaches to fabricate a labeled ground truth dataset: 1) artificially generated data; 2) real data with labels manually put on. Obviously, real data is more representative of the Internet delay character. It is thus the preferred choice. However, it can only be labeled

by humans with domain knowledge. Manual labeling is tedious and error-prone. Without due care, it can undermine the quality of the entire ground truth dataset.

Therefore, it is important to verify the labeling quality. To that end, we rely on an artificial dataset. The idea is to generate some synthetic RTT timeseries with known changepoints. These artificial timeseries are then mixed up with real RTT timeseries during human labeling. Once the labeling is done, we compare the human-labeled changepoints over the artificial dataset with its the generated changepoints, using the scoring method presented in Section 5.5.1. That is to evaluate the detection performance of human labellers using generated changes as ground truth.

Synthetic RTT timeseries are after all not real RTT measurements. Human labellers performing perfectly over generated traces doesn't necessary indicate that they would do as well on real RTT measurements. Therefore it is highly desired that the artificial RTT timeseries resemble real ones as much as possible. Following steps are taken to generate a synthetic RTT time series.

First, we randomly generate several phases/segments of different delay baselines. Different RTT baselines corresponds to the different physical lengths on different Internet paths. For each phase of RTT baseline, we add noises representing micro queue length changes, waiting time due to router load and scheduling policies etc. Finally for each phase of RTT baseline, we generate relatively long during congestion with its own Markov process. Random parameters appropriate to each baseline segment decide the chance of getting into and out of a congestion period. The generated changepoints are moments when there are baseline delay changes and congestion entrance/exit. Short living spikes (length $\leqslant$ 2 datapoints) are considered noises rather than real changes. For the detailed generative models and model parameters of each step, please refer to the documentation in the code space available at `https://github.com/WenqinSHAO/rtt_gen.git`.

20 synthetic RTT timeseries, with 6485 datapoints each on average, are generated with the fabricated tool. The time interval between datapoints is 4min, in line with RIPE Atlas built-in ping measurement. They represent 8646 hours of RTT measurements with 935 generated changepoints. An example of these synthetic RTT trace is shown in Figure 33. The red vertical lines, indicating generated changepoints, are moments when there is a base line RTT change, or a congestion period starts/ends.

5.5.2.2 *Labeling*

As for the real RTT measurements, a great amount of real RTT traces of various characters are selected from RIPE Atlas to construct the ground truth dataset. Some are full of fluctuations; some contain peri-

odic congestion, some have many step-wise changes, etc. 50 real RTT timeseries containing 408,087 datapoints are selected. They represent more than 34,008 hours, i.e. 1417 days, of RTT measurements.

How the labeling is done by humans over a large data set as is crucial to the data quality. Looking at all these timeseries and marking changepoint timestamps with a text editor is clearly not a good idea. To enable intuitive (what humans are good at) labeling, we fabricated an interactive tool to visualize RTT time series `https://github.com/WenqinSHAO/rtt_visual.git`. With this tool, the labellers can easily pan, zoom RTT time series and verify the positions of marked changepoints [5].

1047 changepoints were identified by the labellers for these real RTT timeseries. All the labelled RTT traces, synthetic and real, along with the generated changepoints are all available in the main project repository specified in Section 5.3. Only the human-labeled real RTT time series are regarded as ground truth dataset. During the labeling process, the labellers all found that they were not sure whether a given RTT timeseries is real or synthetic. We therefore judge that the synthetic dataset are good enough. After revealing the identity of each time series, the labellers concluded that the synthetic RTT timeseries seemed to have less rich variation patterns compared to the real ones. There are moments when the labellers were not quite sure if a changepoint were to be placed or not, or where exactly to place a changepoint. These suspicious moments all turned out to be within real RTT timeseries. The presence of such difficulty is exactly one reason that time shift tolerance is needed in detection performance evaluation.

5.5.2.3 *Performance of human change detector*

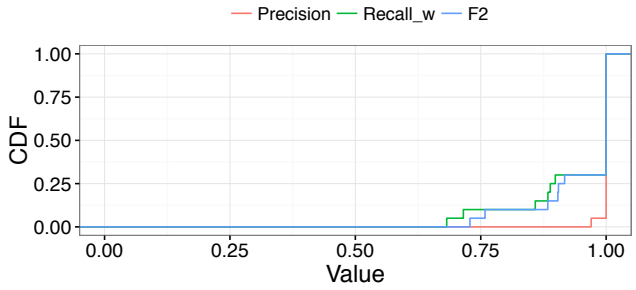


Figure 34: Presicion, Recall_W and weighted F_2 of human labellers on synthetic dataset.

The detection performance of human labellers on the synthetic dataset is shown in Figure 34. Human labellers have 100% for both Precision and Recall on 14 traces. For the rest, the Precision remains

5 The labellers are researchers/graduate students in networking.

high. A few changes are miss out, but their total weight remain limited. This suggests that the human labellers, with diligence and the help of the visualization tool, are indeed capable of high quality labeling.

5.6 EVALUATING CHANGEPOINT METHODS

5.6.1 *Candidate changepoint methods*

With the evaluation framework ready, it opens the door to the exploration of best performing changepoint detection method for RTT measurements. For the detection family presented in Section 5.4, there are two major parameters to be set: penalty and cost function/distribution. We discuss in detail these parameters in this section, while leaving the examination of other methods for future work.

We consider the combination between all the information criterion introduced (AIC, BIC, MBIC and Hannan-Quinn), and all the supported distribution types, including the non-parametric approach based on empirical distribution.

With some preliminary tests, we quickly realized that detection with Normal distribution tend to be over-sensitive, under all the penalty settings. Many short living and insignificant noises are marked as changepoints. It is because the mean and variance of Normal distribution are independently controlled by two separate parameters, which increases the chance of fitting subtle changes either in level or volatility. On the other hand, Exponential, Gamma and Poisson distribution are too numb. The mean and variance of Poisson and Exponential distribution are coupled by one parameter, which restrains their freedom of adjustment [6]. Gamma distribution faces the same issue, but with a more complicated story. A Gamma distribution can be described by two parameters α and β: mean=$\frac{\alpha}{\beta}$, variance=$\frac{mean}{\beta}$. [91], the implementation we use, requires an *a priori* input for α, which actually decides the overall sensitivity. Only β is adjusted in detecting changepoints. With a larger α, a larger β is needed to maintain the same mean estimation for given a segment. Fixed mean with larger β imposes a smaller variation tolerance, thus more likely to split the given segment due to smaller variance changes. In short, larger α leads to more sensitive detection. The default option sets α set to 1, which degenerates Gamma distribution to Exponential distribution. We further tried α from 1 to 100, at unit step. None of them outperforms the best settings shown later on. We therefore no longer consider Gamma distributions.

When assuming Exponential and Poisson distribution, we notice that the average level of an RTT time series somehow dictate the variation tolerance. For instance, for a path including trans-Pacific

6 Poisson, mean=variance=λ; Exponential, mean/variance=λ, mean=$1/\lambda$.

links, we shall expect a minimum RTT above 80$msec$. In this case the corresponding Poisson distribution could easily tolerate several RTT deviations of 20$msec$, which is already non-negligible. However, having coupled mean and variance can as well be a desired feature. We observed during labeling that the level of an RTT segment and its variance are often positively related during congestion periods.

To leverage the above described feature and as well as boost the detection senstivity, we propose for Exponential and Poisson distribution a *data transformation*: subtracting the RTT time series by its minimum value (baseline) to lower down its overall RTT level [7]. Changes are then detected for the baseline-removed RTT time series when assuming Poisson and Exponential distribution. Such setting is denoted as `cpt_poisson` and `cpt_exp` respectively. For the sake of comparison, we also consider Poisson distribution **without data transformation** and denote it as `cpt_poisson_naive`. Normal distribution and non-parametric approach are applied directly on initial RTT measurements. They are denoted as `cpt_normal` and `cpt_np` accordingly.

5.6.2 *Evaluation of candidate methods*

Before evaluating the methods presented above, one might wonder 1) whether the RTT segments in the ground truth dataset follow principally a specific distribution, and 2) whether that distribution assumption leads to the best detection performance.

We performed distribution test for 813 RTT segments longer than 20 datapoints against each of the discussed distribution types. We require a significance level of 0.05 in distribution test. Shapiro-Wilk test is used for Normality test; Chi-squared test for Poisson; Kolmogorov-Smirnov test for Exponential. Distribution parameters are estimated through Maximum Likelihood Estimation. 71 follow Normal distribution, 13 follow Poisson distribution, 11 follow Exponential distribution [8]. None of these distributions seems to have a dominant popularity among the labeled RTT segments. Still, Normal distribution seems to be the most compatible.

All combinations between distribution types and penalty choices are employed to detect changepoints in the ground truth dataset. For each distribution type, we only plot its best performing penalty setting in terms of weighted F_2 score in Figure 35. In later discussion, when we mention a specific distribution type say `cpt_poisson`, we actually refer to a configuration including the distribution itself, the corresponding data transformation plus its best performing penalty choice.

7 Note that timeout measurements are set to 1000ms. For Poisson distributions, RTT values are rounded to the closest integer.

8 It is possible that a segments passes the test for multiple distributions. Thus there are overlaps in the numbers.

More than 75% of changes, in terms of weight, can be detected for more than half of the timeseries with any of these distributions. All these distribution types have better weighted F_2 than classic F_2, indicating some changepoints missed out are indeed of little operational importance. However, it seems to have a big space for improvements. Efforts are especially needed to raise the precision of the detection results.

The precision of `cpt_normal` is particularly poor. This confirms that `cpt_normal` is indeed over-sensitive for RTT type of data [9]. On the contrary, the recall of `cpt_normal` is outstanding among all the candidates. However, its F_2 scores are the poorest [10]. The poor overall detection performance highlights the importance of striking a proper balance between sensitivity and relevance. It suggests as well that the goodness of fit is not necessary a guarantee for detection performance. For rest methods, their performances are relatively close. Compared to `cpt_poisson_naive` (without data transformation), `cpt_poisson` achieves higher recall and weighted recall without obviously sacrificing precision. Consequently, `cpt_poisson` stands a slight advantage in overall performance. As a matter of fact, without data transformation, assuming Exponential distribution detects no changepoint for a big part timeseries in the ground truth dataset. These are all evidences that the proposed data transformation improves the detection performance for Poisson and Exponential distribution.

9 Note that `cpt_normal`'s best penalty is MBIC, the largest adaptive penalty setting. This means that the detection sensitive is already suppressed to its maximum.

10 With F_2, recall is already weighted twice as important as precision.

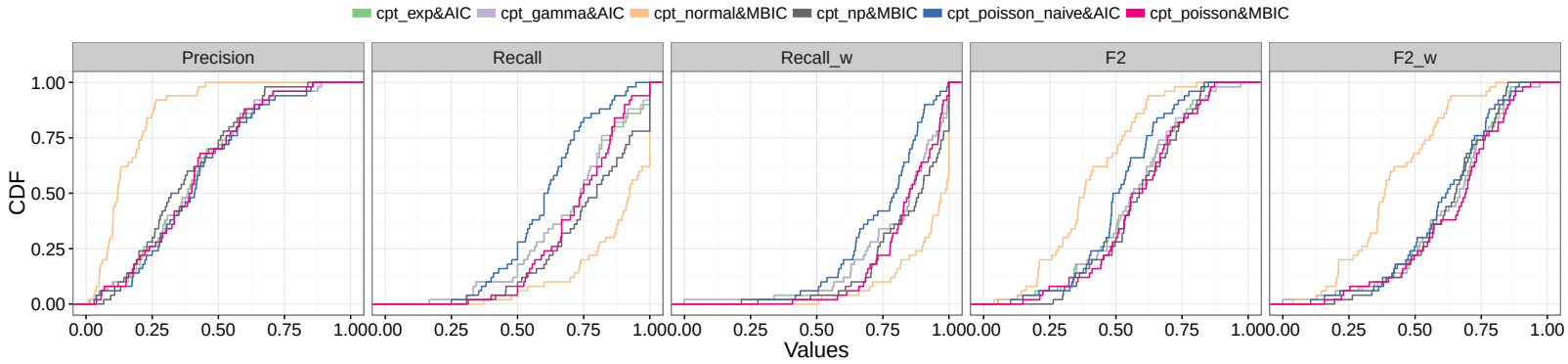
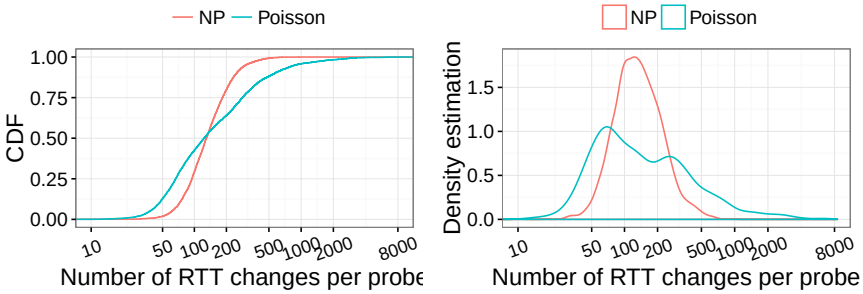


Figure 35: Precision, Recall, Recall_W, F_2 and F_2W with weighted recall on real RTT traces.

5.7 CHARACTERS OF RTT CHANGES

`cpt_poisson` and `cpt_np` with MBIC are used to detected RTT changes for all the 6029 collected ping timeseries spanning over three months. Please refer to Section 5.3 for more details concerning the collected data. We consider `cpt_poisson` as it is the best performing one, though by a small margin. `cpt_np` is included as it performs well and its cost function follows a different construction principal.



(a) CDF. (b) Density.

Figure 36: RTT changepoints number distribution with different detection methods under MBIC.

Figure 36 shows the distribution of RTT change numbers per probe trace. 4,844 probe traces each containing more than 30,000 ping measurements are considered in the figure. 854,626 RTT changepoints are detected by `cpt_np`. `cpt_poisson` almost doubled this number with 1,638,858 RTT changes. Interestingly, the median change numbers for both methods is the same, 122. Figure 36b shows that the change number by `cpt_poisson` spreads over a much wider range. With `cpt_poisson`, 711 probes traces (11.86%) have more than 500 changepoints, while only 35 (0.58%) with `cpt_np` experienced that many changes. This is probably because the cost function of `cpt_np` bases on the estimation of quantiles (by default 10 quantiles used, more can be set) of empirical distribution. The dimension of θ is much larger than Poisson and Normal distribution. The penalty value increases hence much faster for `cpt_np` when new changepoint is added, which prevents extremely large number of changepoints per probe trace.

We describe a detected changepoint from two aspects: 1) the difference of median RTT across the changepoint, and 2) the difference of RTT variance level across the changepoint. More specifically, they are M and Δ defined from ground truth changepoint weighting in Section 5.5.1. Figure 37 visualizes the two features of RTT changepoints detected by the two methods. Since there is a huge amount of changepoints that overlaps each other on the M and Δ surface, it becomes impossible to interpret if we were to explicitly plot each one of them. Therefore, we summarize all the changepoints using a 2-dimensional

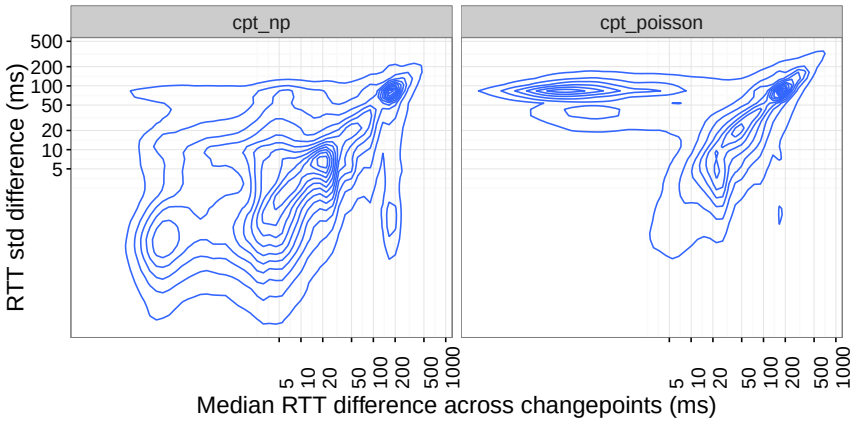


Figure 37: Density estimation of RTT changepoints characteristics.

density estimation (with MASS:kde2d package in R). A same practice has been conducted in Section 4.3 for Figure 18. A contour line represents a trajectory on which the changepoint density is the same on the M and Δ surface. Inside a contour, the density becomes higher.

On the M and Δ surface, changepoints in the top right corner are important ones with serious networking implications. Both methods result a pretty high changepoint concentration in this area. This suggests that there were indeed significant performance changes on the measured path over the three-month time. `cpt_poisson` and `cpt_np` both succeed in identifying some of these changes. What seems to be weird is that there is a cluster of changepoints detected by `cpt_poisson` on the top left panel of the the M and Δ surface. These changepoints, more precisely 319,541 in number (19.49%) with $M < 5msec, \Delta > 50msec$, imply big variance yet small level changes. They are almost absent in the detection results of `cpt_np`. After investigation over these chagnepoints, we discovered that they are mostly caused by frequent timeouts, and are associated with short RTT segment length. For example, probe 20854 had 2308 timeout measurements dispersed in the entire trace. Meanwhile, `cpt_np` handled these timeouts gracefully without emitting all the time change alarms. `cpt_poisson` appears to be a bit troubled by these short and frequent deviations.

Changepoints in the down left corner are relatively subtler. They in general won't lead to significant network performance issues and can thus be safely ignored in TE practices. Reasonably, a small fraction of changepoints detected by `cpt_poisson`, 87,021 (5.31%), are with M and Δ both smaller than $5msec$. Correspondingly, most part of this area in the figure is already outside the most outer contour. This suggests `cpt_poisson` tolerates well these small changes. On the other hand, `cpt_np` appears to be quite reactive in such case. As a matter of fact, 331,062 (38.71%) such changes are detected by `cpt_np`.

We later on relate the detected RTT changes to path changes. With that, we further discuss the detection sensitive and relevance of these two methods with regard to network events.

5.8 DETECTING PATH CHANGES

In this section, we detect path changes experienced by the collected RTT measurements. Both AS and IP level path changes are attended. They are known to have potential impact on RTT. RIPE Atlas built-in path measurement uses a rotating Paris ID setting. This brings confusion between IP path changes caused by load balancing and those due to intradomain routing changes. We address this issue in this section. The purpose of path change detection is not to repeat some of the studies sumeraized in 5.2, such as which kind of path change contributes most to RTT change. It rather helps to enhance the understanding on changepoint detection for RTT measurements.

5.8.1 *Routing change and Load Balancing (LB)*

Schwartz et al. [66] regarded all paths between a source-destination pair as "parallel paths" and found out that RTT measurements over these paths were mostly overlapping. However, there are two kinds of transitions among "parallel paths" that need to be distinguished. They are 1) IP path changes caused by intradomian routing protocol dynamics, such as route recalculation after link failure or configuration update, and 2) those caused by LB mechanisms. Intradomain path changes before the era of LB haven been shown to be responsible for important RTT changes [44]. On the other hand, LB paths are of equal/close administrative cost, hence similar performance characteristics [68]. This suggests that these two kinds of IP route changes could have very different impact on RTT, and thus need to be distinguished.

Trivial as it may sound, detecting IP path changes is challenging for RIPE Atlas built-in traceroute measurements. The difficulties come from two aspects: 1) the wide deployment of IP-level LB; 2) RIPE Atlas uses Paris traceroute with different Paris IDs every other measurement (incremented by 1, recycling between 0 and 15) [35, 86]. IP paths taken by two neighboring measurements can thus naturally differ – load-balanced on different available paths with different Paris IDs. From this angle, plain IP path changes doesn't mean that there were topological or configuration changes that lead to any real routing change. On the other hand, having different Paris IDs every time can also be helpful in this context. If traceroute were locked on a single Paris ID, it would then be unlikely to detect routing changes that only affect paths corresponding to other Paris IDs.

5.8.2 *Intradomain Routing Pattern change*

When a different IP path is measured with a same Paris ID, there is potentially a routing change. We call this kind of IP path change an Intradomain Routing Pattern (IRP) change. In the example below, the IRP change happens when Paris ID 2 begins to take IP path E instead of B. We refer to the two measurements with same Paris ID but different IP paths as *conflicting* measurements.

```
                              | IRP change
Paris ID: 0 1 2 3 4 .. 15 0 1|2 3 ..
IP Path:  A B B A A .. C  A B|E E ..
       A measurement series  | boundary -> forward
```

IRP changes can thus be identified by constructing a series of measurement sequences. Each sequence shall not contain conflicting measurements. Yet, combing any two neighboring sequences, there shall be as least one pair of conflicting measurements, as otherwise they can be merged. This can be done by moving the boundary of measurement sequences *forward* to include non-conflicting measurements, till a conflict is encountered, as shown in the above example. We call this approach *forward inclusion*.

The drawback of *forward inclusion* is that it potentially delays the detection of actual IRP changes. This is because, when including non-conflicting measurements forwardly, a measurement sequence always has the chance to absorb measurements till it experiences all the possible Paris IDs. However an actual IRP change could happen before that moment. An example of possibly delayed IRP change is given right below:

```
        !Possible position of actual IRP change
.. 1|2 3!4 5 .. 15 0 1|2 3 4 5 ... 15 0 1 2 3 4 5 ..
.. B|B A!A C .. C  A B|E E A C ... C  A B E E A C ..
                     | IRP change forward inclusion
                     | backward <- boundary
```

With *forward inclusion*, an IRP change will be detected at the 2nd appearance of Paris ID 2. While the actual change probably happens at the 1st appearance of Paris ID 4. It is because starting from the first Paris ID 4, all the measurements are non-conflicting with the later measurement series. The 1st appearance of Paris ID 2 and 3 are in fact a short deviation from a popular IRP.

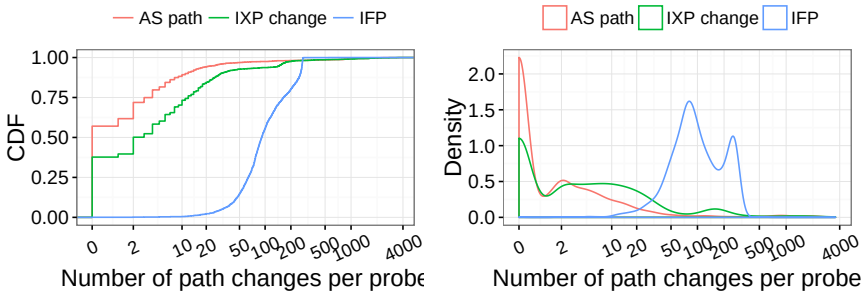
Cases like this are highly possible, because networks tend to have some stable configurations that lead to a few dominant paths over time [44, 100]. Deviations from dominant/popular IRPs are thus likely to be short living. With RIPE Altas built-in measurements, they probably won't last long enough to experience all the Paris IDs [11]. To better

11 It takes at least 450min (30min * 15) to go through all the 16 Paris IDs used in RIPE Atlas built-in traceroute.

reflect the presence popular IRPs, we push backwardly the boundary obtained by *forward inclusion* if 1) the latter measurement sequence is longer than the previous one; 2) the latter measurement sequence experiences all the Paris IDs at least twice. We refer to this approach as *backward extension*. We show later on in Figure 39 that IRP changes detected by *backward extension* have a much larger chance matching with RTT changes.

5.8.3 *Characters of detected path changes*

AS-level path changes are as well detected after translating IP hops to ASN hops [146]. We didn't consider third-party address [18, 67] and IP alias techniques[57, 64] in this operation. It is because the focus is to detect changes instead of constructing an accurate Internet topology. We did detect the presence of IXPs using the heuristics proposed by traIXroute [115]. Studies have shown that IXP could be involved in large RTT changes [136].



(a) CDF. (b) Density.

Figure 38: Distribution of path change times per probe. One probe with most complete traceroute measurement is chosen for each AS. 2050 probes/ASes are included in the graph.

We consider only AS path changes where the difference starts from a hop position involving public ASNs in both AS paths. Difference due to temporal presence of non-responding hops are ignored. IXP change happens when the difference starts from a position involving at least one IXP hop in the two consecutive AS paths. IRP changes are detected with *backward extension*. Those overlap with any AS path/IXP changes are excluded, since they are potentially caused by intradomain routing changes.

The distribution of the number of path changes per probe trace is illustrated in Figure 38. One probe with the most complete traceroute measurements is selected for each of the 2050 source ASes. 1170 (57.07%) of them experienced no AS path changes over the period of three months, indicating that the AS paths are in general very stable over time. Still, 51 (2.49%) probes underwent more than 100 AS path changes. 717 probes (34.98%) didn't have any IXP change. 140

probes experienced frequent (> 100) IXP changes. IRP changes are much more frequent than the other two path changes. Half of the selected probes experienced more than 90 IRP changes. We investigate the nature of these path changes, together with their potential impact on RTT, in Section 5.9.

5.9 MATCH BETWEEN RTT AND PATH CHANGES

If a pair of RTT and path change on a same Internet path is close in time, chances are that the RTT change is caused by the path change. We say that these two changes are correlated or matched. However, there is no straightforward way matching the two types of change, as the measurement intervals are different: 30min for traceroute while 4min for ping.

Again, *minimum cost maximum-cardinality matching* appears to be a reasonable formulation of the correlation between RTT and path changes. We therefore borrow the concept of optimal matching in changepoint evaluation (Section 5.5.1). The shift tolerance window is set to the interval of traceroute measurement. It is because causal relationship between the RTT and path change is possible (though not necessary) within that range. A pair of RTT and path changes are correlated/matched if they are within the so produced optimal matching. The notion precision, introduced in section 5.5.1, is now interpreted as the fraction of path changes that are matched to an RTT change. Recall now means the fraction of RTT changes that can be explained/matched to a certain type of path change.

We compare separately AS, IXP and IRP path changes to RTT changes detected with `cpt_np` and `cpt_poisson`. The matching is calculated for the 2050 probes mentioned in Figure 38, each from a distinct AS.

5.9.1 *Forward or backward?*

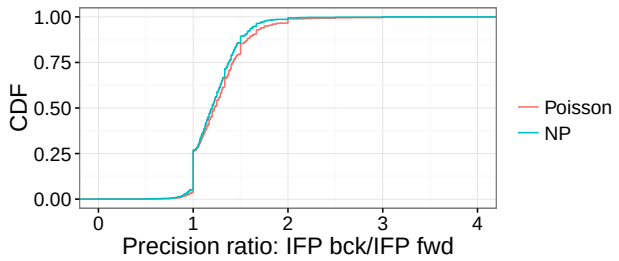


Figure 39: Precision ration between IRP changes detected by *backward extension* and *forward inclusion*.

In Section 5.8, two methods were presented to detect IRP changes: *backward extension* and *forward inclusion*. Do the IRP changes detected with these two methods have any difference in terms of matching

with RTT changes? To answer this question, we compare for each probe the matching precision of these two methods. The precision ratio between the two methods is given in Figure 39, in the form of CDF over probes. *backward extension* produces the same number of IRP changes as *forward inclusion* does. However, IRP changes by *backward extension* are more likely to have a match with RTT changes for 75% probes. These probes are on the right side of the graph where the precision ration is bigger than 1. The significant increase in precision implies that the path changes detected by *backward extension* are more accurate and are thus a necessary improvement to the basic approach. Later on, we refer always to changes detected by *backward extension* when talking about IRP changes.

5.9.2 *Summary of matching between path change and RTT change*

Table 6: Number of RTT changes matched with a path change for the selected 2050 probes.

	`cpt_poisson`	`cpt_np`	# path changes
AS path change	11,794	6,380	51,282
IXP change	9,126	8,341	73,544
IRP change	38,700	36,400	244,713
# RTT changes	481,877	307,312	

Table 6 details the number of matches between path and RTT changes. Each cell tells the number of matches between the corresponding line (path change type) and column (RTT change detection method). The last column contains the total number of path changes of each kind. Similarly, the last line provides the total number of of RTT changes detected by the two methods.

The fraction of AS path changes matched to RTT changes by either detection method is much lower than the reported 72.5% in [94]. It seems that AS path changes have less significant impact on RTT than previous understanding. Is there something particular with our dataset or methods? Moreover, the number of AS path changes matched with `cpt_np` RTT change is only about half of that with `cpt_possion` RTT changes. On the contrary, matching numbers for IXP and IRP changes are quite close across the two RTT change detection methods. Where does this difference come from? All of these are very intriguing phenomena. We try to explore the underlying reasons in the next section with a close-up look.

5.10 CHANGE DETECTION SENSITIVITY AND RELEVANCE

5.10.1 *cpt_poisson matches better with AS path change?*

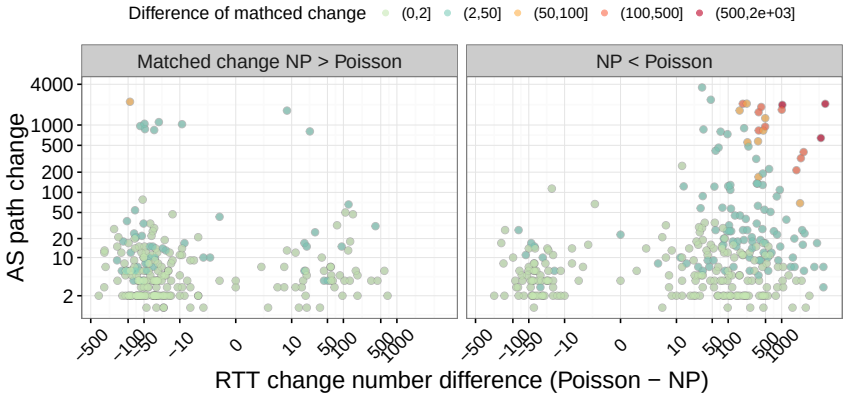


Figure 40: Probe having difference in the number of AS path changes matched to RTT changes detected by `cpt_poisson` and `cpt_np`. Probes are characterized by its AS path change numbers and RTT change number difference between the two methods. The color of each probe indicates the level of difference in matched change. Left panel shows the probes with more AS path changes matched to `cpt_np` RTT changes.

It seems that there are much more `cpt_poisson` RTT changes matched with AS path change, according to Table 6. But really? Among the 880 probes ever experienced AS path changes, 293 probes have more AS path changes matched to `cpt_poisson` RTT changes, 224 have more AS path changes matched to `cpt_np` changes, Among the above mentioned 517 probes, 463 are actually with a difference smaller than 10 AS path changes. The rest 363 (among the 880) probes have no difference across the two methods.

Contrary to what we see in Table 6, the numbers of AS path matched with RTT changes are in fact highly consistent across the two methods for the majority of probes. The difference is caused by a small fraction of probes identified in Figure 40. In the graph, each dot represents one of the 880 probes that ever experienced AS path changes. More reddish the dot (probe) is, larger the difference is between `cpt_poisson` and `cpt_np`. We can tell from the graph that those probes plainly in red all experienced a large number of AS path changes (y-axis). Moreover, when there is more `cp_poisson` RTT changes (the positive side of the X-axis), it's more likely that more AS path changes from that probe are matched to `cp_poisson` RTT changes as well (the left panel). The reverse is as well true when there is more `cp_np` RTT changes. All together, the difference in matched RTT changes with AS path changes fundamentally lies in the difference of detection sensitivity (number of detected RTT changes) across different probe traces. This

difference is manifested through extremely frequent AS Path changes of several specific probes.

5.10.2 *Is `cpt_poisson` more sensitive?*

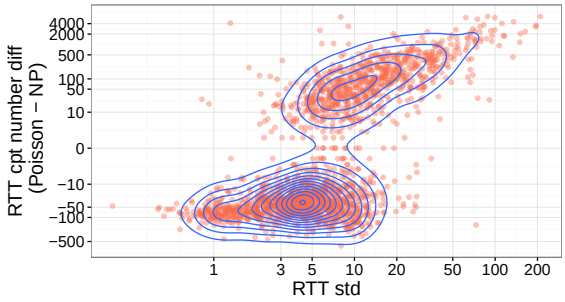


Figure 41: Relation between RTT change number difference by the two changepoint method and the RTT trace `std`.

`cpt_poisson` detects in total more RTT changes than `cpt_np`, according to Figure 36b and Table 6. Therefore, `cpt_poisson` seems to be more sensitive in change detection. But really?

With a per probe breakdown, we discovered that the number of RTT changes detected with the two methods is somehow related to the overall variation level of each probe trace. This relationship is visualized in Figure 41. Each dot represents a probe RTT timeseries. The X-axis indicates the RTT standard deviation in *msec*, while the Y-axis tells the difference in RTT changepoint numbers between the two methods. Again, to facilitate the interpretation, we overlapped the graph with a 2-dimensional density estimation for the datapoints on the surface. Inner contours represent dense area.

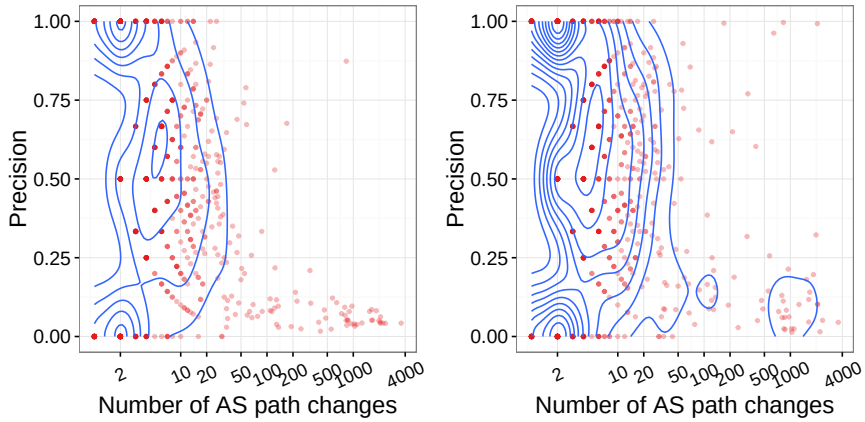
For most probes traces with small overall RTT variation, `cpt_np` is in fact more sensitive and detects more RTT changes, according to the figure. This observation agrees with Figure 36b in the sense that `cpt_np` detects much more changes of small amplitude.

For probes with relatively large overall RTT variation, `cpt_poisson` tends to be more sensitive and the difference in change number increases with the level of RTT variance. With comprehensive manual inspection, we found that those RTT traces with high variance mostly underwent large amplitude RTT oscillations, many of which caused by ping timeouts. As human change detector, we also found very difficult to mark moments of change for these traces.

As a matter of fact, `cpt_poisson` is not very flexible in adjusting detection sensitive according to the variation level of input timeseries. For example, for a RTT timeseries that is full of large amplitude variations, we would expect a detection method to restrain a little bit. This incapability is because the variation tolerance of a Poisson model is coupled with the the timeseries mean. After removing the baseline,

different RTT timeseries are actually of similar mean, thus leading to similar sensitivity. For an overly noisy RTT timeseries, the 'constant' sensitivity turns out to be a large number of changepoints. Meanwhile, `cpt_np` seems to be very elastic in detection sensitivity and less variant in the number of changepoints produced according to Figure 36b.

5.10.3 *How AS path changes match to RTT changes?*



(a) changes by `cpt_np`. (b) changes by `cpt_poisson`.

Figure 42: The relation between precision and AS path change times per probe trace.

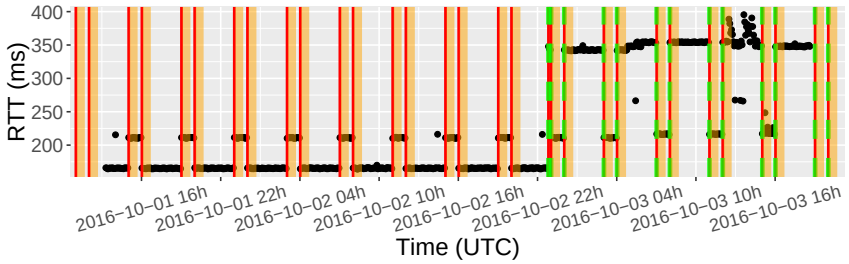


Figure 43: RTT from Probe 12849. Red lines for RTT change detected by `cpt_poisson`; green dotted lines for RTT change by `cpt_np`. Orange strips for AS path changes.

Two 'bizarre' phenomena have been so far observed: 1) the match between AS path change and RTT change are much weaker than what reported in previous studies; 2) Certain ASes experienced extremely frequent AS paths changes. Are these two somehow related?

We investigate this issue through a per probe breakdown between AS path change numbers and path change matching precision in Figure 42. In the graph, each red dot represents a probe. The Y-axis tells the fraction of AS path changes that are matched to an RTT change, i.e. precision. Since many probes (red dots) overlap with each other

on the surface, we estimated the 2-dimensional distribution density and illustrated it with contour lines. Inner contours are area with dense probes.

Figure 42 reveals that probes with extremely frequent AS path changes (right side of the graphs) have in fact very low correlation, in terms of precision, with RTT changes. After extensive manual inspection, those frequent AS path changes appear to be AS-level LB, i.e. the upstream AS employed in reaching the destination are switched very frequently among a few providers. This could be a possible consequence of multipath BGP. Such AS path changes generally don't have an obvious impact on RTT. For probes with fewer AS path changes (the left side of the graphs), the chance of having match between AS path change and RTT change is in fact pretty high.

Not all AS-level load balancing is without consequence. For example, probe 12849 in Figure 43 experienced 170 AS path changes, among which 169 are matched to RTT changes detected by `cpt_poisson` and only 102 are matched to `cpt_np` RTT changes. These AS path changes are highly periodic and coincide with clear cut RTT changes. `cpt_np` failed to detect some of the changes with smaller amplitude. The presence of such case confirms the need for measurement-based TE. Otherwise, unwise routing decisions can be made without realization.

5.10.4 *Pitfalls of IXP and IRP path change detection*

Similar to AS path changes, probes with frequent IXP and IRP changes correlates weakly with RTT changes. For example, in Figure 38b, there is a group of probes that experienced from 100 to 200 IXP changes. Only around 10% of IXP changes on these probes are matched to an RTT change.

We investigate all the 58 probes in the area. These probe passed by AMS-IX to reach b-root most of the time. There were about 147 times, shared by these probes, where AMS-IX hop was replaced by a timeout hop before arriving at AS6939. In such case, no IXP related address appears in the measured path. The presence of IXP is thus uncertain. Still, we count it as an IXP change according to our definition in Section 5.8 [12]. This suggests that the detection of IXP changes are not very accurate, especially for those probes with many IXP changes. This problem is stemmed from the difficulty of IXP inference in path measurements.

The correlation of IRP changes with RTT changes are much weaker than that of AS and IXP path changes. It turned out that most probes

12 The newly released traIXroute v2.1 can detect IXP without the presence of IXP related IP address, if the neighbouring ASes are known to be member of a same IXP. However, it is still possible that two ASes peer at multiple IXPs, where the exact IXP traversed would remain uncertain.

Table 7: Quantiles of unique IP path numbers per probe trace.

5%	10%	25%	50%	75%	95%	100%
20	32	56	91	145	419	4302

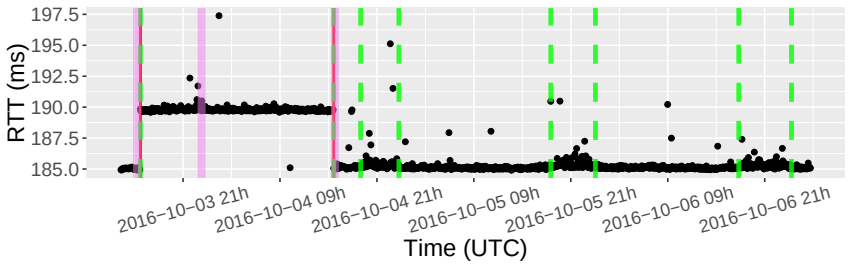
experienced much more than 16 end-to-end IP paths, according to Table. 7. In such case, one Paris ID might have been mapped to more than one IP paths. This might lead to IRP changes without actual routing change. Within in each single AS, the number of different IP paths rarely exceeds 16 toward a destination. However a chain of ASes can produce way much richer combinations of end-to-end IP paths.

Moreover, there is a group of probes having around 250 IRP changes according to Figure 38b. An IRP change takes place roughly every 16 measurements on these probes. These changes are as well poorly correlated to RTT changes. We investigated some probes in the area and found out the frequent changes aren't necessary related to the large amount of end-to-end paths. For some probes, two neighbouring IRPs only differ at one or two Paris IDs. IP paths taken by these Paris IDs oscillates between a few alternatives frequently. For example, the Paris ID 6, 7, 8, 9 of probe 23998 switches a lot among only 2 paths. Such change in general doesn't have obvious consequence on RTT level.

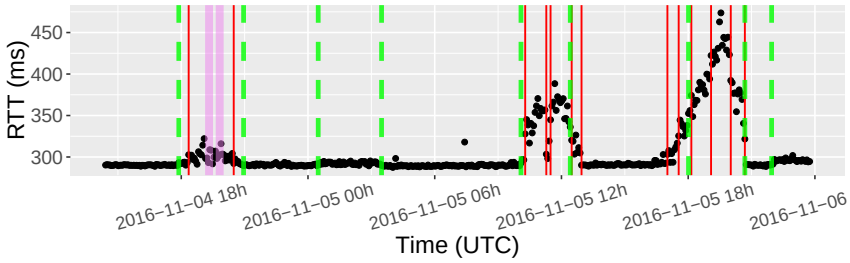
5.10.5 *Unmatched RTT changes*

Several reasons contribute to the large amount and fraction of RTT changes unmatched to any path changes. First, changes on the reverse path are not observed. We were not able to measure the reverse path with RIPE Atlas built-in measurement. Therefore, it is impossible to detect the changes on the reverse path. However, these changes could have contributed to RTT changes. Especially in the context of inter-domain routing where paths are likely to be asymmetric. This implies that the RTT changes caused by the reverse path changes are probably different from those caused by path changes on the forwarding direction.

Second, congestion. Congestion can be independent of path changes and yet is capable of causing significant RTT variations. Figure 44b gives an typical example of RTT changes probably caused by congestion. There are three bumps that can be visually noticed in the plotted RTT timeseries. We say the latter two are probably congestion. First, they do not correspond to any path changes, at least in the forwarding direction. Second, these bumps are probably caused by filling queues along the path. Because, the RTTs within these bumps are not flat. On slightly loaded path, we would expect fairly constant RTT mea-



(a) Probe 28002.



(b) Probe 26328.

Figure 44: RTT trace and change detection example. Red lines for RTT change detected by `cpt_poisson`; green dotted lines for RTT change by `cpt_np`. Violet strips are IRP changes.

surements overtime. It is because the queues are almost empty, thus no room for delay variation. On the other hand, the changing RTTs within the bump are probably a reflection of how the traffic demand on the bottleneck evolves due to end-to-end congestion control mechanism. Both bumps/congestion deviate greatly from the baseline and last for a several hours. They thus have significant impact on transmission performance. We successfully detected them with the studied changepoint detection methods. However, such detection is not possible with the previously proposed method [93]. It performs spectral analysis on RTT timeseries to find out periodically repeated congestion. Such persistent congestion is normally due to lack of network capacity. Meanwhile, transient congestion in Figure 44b is more likely caused by sudden traffic variations. Measurement-based TE aims to avoid both types of congestion when there is alternative paths with available capacity. For that purpose, changepoint detection methods are indeed helpful in notifying the presence of such performance variations.

Third, over-sensitive detection. If we boldly assume that path changes on reverse paths cause a comparable amount of RTT changes as forwarding path changes do, there are still many RTT change unmatched. Some of them might be effectively attributed to congestion, as explained here above. The remaining unmatched RTT changes are plainly the result of over-sensitive detection. We already revealed from a macroscopic view, in Section 5.7 and 5.10.1, that `cpt_poisson` tend to overestimate the number of changepoints when the RTT trace is noisy.

Meanwhile, `cpt_np` is capable of detecting delicate RTT changes. Individual traces are given in Figure 44 to illustrate the sensitivity difference from a microscopic view. In Figure 44a, `cpt_np` detected all the periodic small amplitude congestion. This is actually quite impressive, as these change are only hardly visible for human experts. The changepoints marked by `cpt_np` indeed highlighted their presence, and made them easier to be noticed visually. In Figure 44b, both methods identified the two large bumps near the end of the timeseries. The difference is that `cpt_poisson` marked intermediate level changes as well. These intermediate changepoints are clearly not correlated to any path changes. On top of that, they are as well redundant in informing the congestion that was happening at that moment. The reason for such over-sensitivity was due to its incompetence in adjusting the detection sensitivity according to input variance level. This issue is explored and explained in Section 5.10.2.

CONCLUSION

In this chapter, we proposed an evaluation framework for change detection on RTT time series. The framework is robust with human-labeled dataset and weights RTT changes according to their importance in network operation. We further designed a data transformation adapted to RTT measurements to improve the detection sensitivity of some detection methods. In detecting path changes, we distinguish those caused by routing changes from those due to load balancing. Finally, we correlate the detected RTT and path changes by establishing an one-to-one matching between them. We investigated the sensitivity distinction across different change detection methods. Hidden issues with path changes are as well revealed.

This work is a facilitator for measurement-based TE. Further efforts are required in building a working system.

6

INFERRING THE LOCATION OF RTT CHANGES

ABSTRACT

We set out to infer the network locations of previously detected RTT changes. Knowing which AS or inter-AS link causes an RTT change at a certain moment shed light on TE decisions for destination prefixes to which measurements are not available. We start by grouping RTT time series that share a same RTT change with the help of changepoint analysis. Basing on the assumption that changes are more likely to happen on common parts of these measured paths, we then develop inference procedures for AS and inter-AS link. Finally, we present two visualization tools for the inspection of shared RTT changes across multiple RTT measurements and inferred causes of change on a topology map.

6.1 PERFORM MEASUREMENT-BASED TE WITHOUT DIRECT MEASUREMENTS

Locating the cause of RTT changes is an intriguing research topic in its own right. Meanwhile, it is as well beneficial for measurement-based TE when measurements toward certain destination prefixes are not available.

6.1.1 *Lack of direct measurements*

As explained in Chapter 1, in order to measure the path performance toward a destination prefix, we need to, in first place, identify some hosts in that prefix that respond to our measurements. One possible approach identifying them is to look for hosts listening on some common TCP ports, e.g. 80, 443 etc., in the traffic exchanged with that destination prefix or through port scanning. Then RTT toward the destination prefix is represented by the measurements toward identified hosts [1].

However, not all selected destination prefixes have such 'measurable' hosts [2]. Take client SA appeared in Section 3 as an example. On average, 15% of its outbound traffic involving ~ 330 destination prefixes are without measurements. More than 70% of the 'un-measured' traffic flows toward prefixes owned by mobile operators during peak hours. In the foreseeable future, the proportion of such traffic would be even bigger, given the overall tendency of increasing usage on mobile devices. At this point, we face the problem of *performing measurement-based TE without direct measurement toward the destination*.

6.1.2 *Prefix grouping as a countermeasure*

One possible approximation is to group 'un-measurable' prefixes with those have measurements based on topological/geographical locality, for example those belonging to a same AS or city. The RTT measurements toward some destination prefixes in the group speak for the rest prefix without measurement. Same best route are chosen for prefixes within the group.

However, locality dose not necessarily imply similarity in performance variation. Assuming geographical locality, it at most provides a rough estimation on the baseline RTT toward a destination prefix, not regarding issues with the IP geolocation precision [72]. Since it does not ensure path similarity, prefixes grouped in such way are not

1 Several ways exists to actively measure the RTT through TCP. Methods with few footprint and low measurement cost are employed in port scanning [59], like SYN (also known as half open) or FIN stealth.

2 selected prefixes: destination prefixes with important volume thus selected for TE, more detail in Section 3.

likely to experience most performance variations that are network specific.

Assuming topological locality, say AS-level, is more relevant, but still not good enough. As it has been pointed out, AS is not an atomic point [38]. Prefix-based route announcement in BGP might result different paths for prefixes belonging to a same AS. Moreover, such approximation is helpless when the entire AS is not 'measurable'.

6.1.3 *How RTT change cause inference helps in TE?*

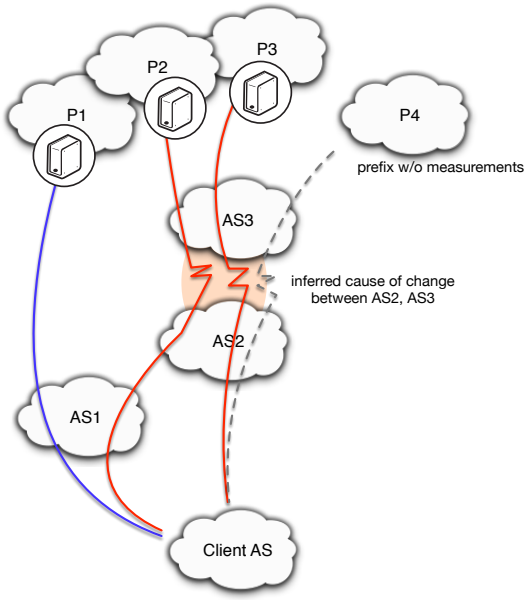


Figure 45: Transmission toward P4 can undergo a potential change as link AS2-AS3 is inferred as cause for RTT measurement changes toward other prefixes.

Without RTT measurements toward P4 in Figure 45, we have no clue on when and which of its available paths might undergo changes and thus are helpless in making TE decisions. However, if we were able to deduce which part of the Internet is responsible for RTT changes using available measurements, we might still stand a chance optimizing for P4. Suppose that the link between AS2 and AS3 is inferred (somehow) as the cause for the RTT change underwent by measurements toward P2 and P3. Since, a path toward P4 (the dashed line) flows as well through AS2 and AS3, we can then reasonably assume that the transmission performance toward P4 on that path shall experience a similar change at the same moment as those toward P2 and P3. Realizing that, we shall switch to alternative paths that remain unaffected to reach P4. That is how RTT change cause inference

shed light on performance variations toward destinations without direct measurements and hence enable TE.

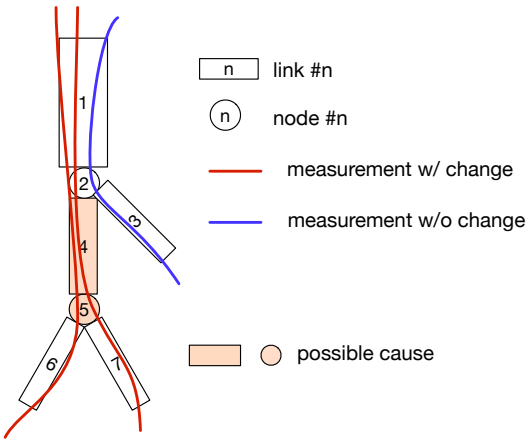


Figure 46: A toy example of RTT change cause inference.

This idea was initially inspired by the case study in Section 4.5 on shared RTT changes by multiple RTT time series from different ASes. We realized that some RTT changes are not exclusive to measurements on one specific Internet path, but rather impact several RTT time series simultaneously, as shown in Figure 29. Optimize the inter-domain routing against the cause of such RTT changes would then be a more fundamental and effective approach to TE than handling each individual prefixes and paths towards them.

In order to infer the location of causes, one reasonable assumption is that such shared RTT changes are more likely caused by the common parts of these paths, instead of being the consequence of perfect synchronization of multiple issues scattered in various places. With that, it is then possible to narrow down the scope of possible causes with measurements having both common parts and divergent parts. A toy example of inference is given in Figure 46. With the assumption we can first scope the cause to link 1 and 4, node 2 and 5. Since there is a measurement free of RTT change traversing link 1 and node 2, link 4 and node 5 are then more likely to be the cause. In Section 6.3, we will more formally describe the assumptions and inference logic for node and link investigation under all possible topology patterns and measurement distributions.

6.2 RELATIONSHIP TO NETWORK DELAY TOMOGRAPHY

6.2.1 *Similarity in assumption and inference logic*

RTT change cause inference aims at identifying internal links or nodes that are responsible for RTT changes using end-to-end measurements. It is not difficult to spot that RTT change cause inference is somewhat similar to the quest of network delay tomography [12], i.e. to infer the internal link delay characteristics using end-to-end measurements. The similarity not only lies in the formulation of the problem, but as well as in the assumptions and general idea of inference.

Many tomography works assume that measurements toward difference destinations experience similar delay on the shared links. As a matter of fact, these works either use multicast [13] or closely time spaced unicast measurements[22, 23] to ensure this assumption. In our case, this intuitive assumption can be naturally extended: multiple RTT measurements shall undergo a same RTT change if caused by the common part on their paths.

The above assumption serves for inference. In delay tomography, since the common links contribute equally to end-to-end delay measurements, then the difference in end-to-end measurements can only come from the divergent part of the path. With carefully designed measurement sets, it is then possible to infer for each internal link its delay properties [37]. The basic inference logic for RTT change cause inference follows the same spirit. It exploits the topological divergence and convergence of a set of measurements to locate possible causes, as illustrated in Figure 46.

6.2.2 *Difference in output*

Despite the similarity between the two problems, the fundamental outcome wanted at the end of inference differs. In delay tomography, one wishes to reconstruct the probability distribution of delay on internal links, where the likelihood of each end-to-end delay measurements is parameterized by a convolution of internal links' delay distribution, from source to destination. The parameters for internal link delay distribution can then be solved through maximum likelihood estimation of the end-to-end path delay distributions given the end-to-end measurements. For that purpose, a series of methods are developed either to accelerate the maximum-likelihood estimation of delay distribution [19, 23], or to capture the time-varying nature of link delays [11, 22, 23]. If we were to apply these same methods to our RTT measurements after changepoint detection (Section 5), we might arrive at a probability distribution on the likelihood of a link causing significant RTT change over the period of measurements. However, it

does not tell at what exact moment the link caused a significant RTT change.

6.2.3 *Methodological compatibility*

Further, one might wonder whether it is possible to apply delay tomography methods directly to Internet RTT measurements over a relatively short time range. And then detect whether the delay distribution experiences obvious change over time, or exhibit multimodality for certain links.

This approach is however not the most appropriate for various reasons. First, the assumption of same delay contribution from same link to different end-to-end measurements no longer holds for Internet RTT measurements for TE uses. It is because the timing of different measurements are not strictly synchronized. Rather, random factors are deliberately added to each individual measurement to avoid creating periodic peaks of measurement traffic. These peaks might introduce interference among measurements and hence harm measurement reliability. Moreover, RIPE Atlas measurements (Section 4) widely employed in this work do not guarantee strict synchronization among different measurements. Those measurements are individually performed by probes loosely coordinated. The timing depends basically on the moment when each probe starts for the first time. Second, delay tomography introduces potentially a scalability issue for Internet measurements. It is first conceived for intradomain uses and are in most cases validated solely on simulated networks way much smaller than the actually part of the Internet that the traffic of a typical content/hosting/service provider could span.

6.3 RTT CHANGE CAUSE INFERENCE

In this section, we describe the input and output of the RTT change cause inference functionality, as well as its internal buildings blocks. Two assumptions to initiate the inference are formulated and justified. We carefully explain our choice of performing inference on AS-level topology, bearing in mind that the true cause of RTT changes could come from sub-AS level structure, like PoP. In order to cope with the discrepancy between inference granularity and actual cause scope, we introduce two quantitative heuristics for node and link liability judgment. Finally, the inference logic is developed based solely on the introduced assumptions for links nodes and links under all possible topology layouts.

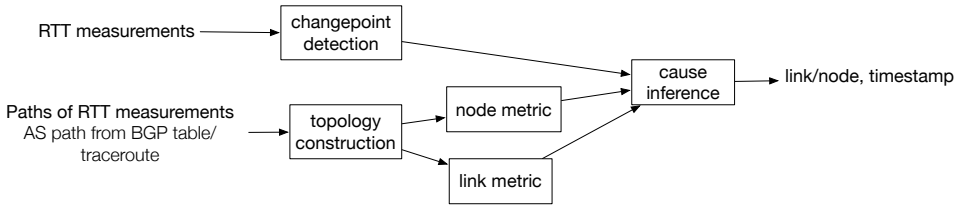


Figure 47: Building blocks of RTT change cause inference.

6.3.1 *The big picture*

With RTT change cause inference, we are interested in identifying the part of the Internet, as precise as possible, that is responsible for RTT changes detected. What we have as input on client TE platforms are 1) RTT measurements from multiple sources to multiple destinations for TE uses; 2) underlying AS paths for these RTT measurements [3].

In this work, we continue to use the same data set collected for RTT change detection in Section 5.3 for the sake of transparency and repoducibility. The major difference with the data available on client TE platform is that, with RIPE Atlas, the underlying path is learnt through traceroute instead of from BGP table [4]. Such difference calls for additional pre-processing, such as IP-to-AS translation, IXP detection etc, so as to be compatible with data from client TE platform. It however should not have impact on the inference logic. On the other hand, given the worldwide distribution of RIPE Atlas probes and divers destinations they probe, the inference result from RIPE Atlas can complement that from client platform measurements, by offering a larger coverage of the Internet, hence bigger possibility of identifying cause of changes having impact on 'un-measurable' destination prefixes.

Figure 47 depicts the logical building blocks required for RTT change cause inference. Changepoint detection (Section 5) symbolizes the RTT timeseries into sequences of RTT change events.

Topology construction builds a graph for hops and links traversed by RTT measurements. This topology graph is an intermediate step in designing a inference metrics for each node and link present in the topology. As seen in Fig 46, whether a node/link is the cause

3 With the real TE system, the sources of measurements are client platforms and destinations are the prefixes clients send traffic to. The source of measurements could be multiple if we merge measurements from multiple client platforms or the client has multiple sites with different provider options.

4 It is as well possible to learn paths of RIPE Atlas RTT measurements via RIPE RIS [145]and Routeviews [146]. However, the coverage of those BGP vintage points are probably not enough to reflect how each hosting AS of Atlas probes (many at the edge) route the traffic, since they mainly locate near the core of Internet.

for RTT change, depends not only on the measurements that traverse it, but as well those flow around. Identifying such measurement sets for each node/link requires knowledge on the topology. Moreover, the topology graph serves as well in visualizing the location of RTT changes.

The exact value of these inference metrics at each moment is calculated from sequences of RTT change events. Then, the cause inference is performed for each node and link based on the value of their inference metrics. The output of the whole system tells the links and nodes that are responsible for RTT changes over time.

6.3.2 *Spatial and temporal granularity of inference*

6.3.2.1 *Spatial granularity*

Spatial granularity defines the finest element to which an RTT change cause can be attributed to. It basically depends on the granularity of the path and topology graph. It is obvious that we can construct the AS-level topology with AS paths from BGP table. With traceroute measurements, IP path to AS path translation is relatively straightforward, except for issues like third-party IP address [18, 67, 92], exact boundaries between ASes [114], the presence of IXP[115], etc. Hence the question here is rather: *do we have the incentive to perform cause inference at granularity finer than AS?*

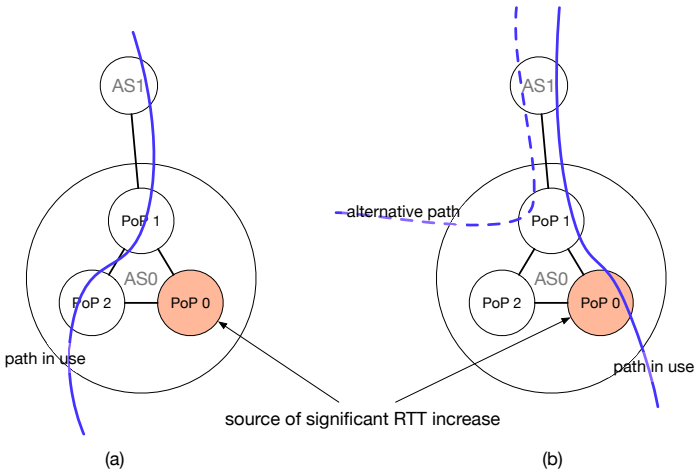


Figure 48: The advantage of performing PoP-level inference granularity.

Theoretically, finer granularity is in general beneficial. Figure 48 illustrates some of these possibilities where a PoP-level cause of RTT change is detected. Figure 48(a) is a case where current in use path is potentially not affected by the potential RTT change, thus no need to change path. While with AS-level inference, one will not be provided

with such distinction. Similarly, in Figure 48(b), with AS-level inference, one could not realize there is actually an alternative path free of potential RTT degradation.

How often such case actually happens is not yet known. However the difficulty in constructing router-level, PoP-level topology and their inaccuracy stemmed from various heuristics used are well studied [42, 87, 111]. On top of that, one other challenge at finer granularity is that the it greatly inflates the number of nodes and links in the topology graph that needs investigation.

Given the above, we decided to build topology and perform inference at AS-level, and leave the study on finer granularity for future work.

6.3.2.2 *Temporal granularity*

The essence of deciding time resolution in cause inference lies in the answers to following questions: *given a set of RTT change event sequences (defined by inference metric), how to detect whether they undergo same changes? If yes, when do changes occur? What is the popularity of these shared RTT changes within the group?*

The main challenge to above questions comes from the fact that a same RTT change event may have shifted time stamps in different event sequences. It is because 1) RTT measurements are asynchronous among RIPE Atlas probes; 2) Detected RTT changes might have time shifts upto several measurement intervals. Several methods are possible to address this issue.

DENSITY ESTIMATION The idea is to project RTT change events from different event sequences onto a same time axis. When the event intensity is high, there then should be an RTT change event shared by many event sequences. The local maximums of the density estimation are regarded as moments of shared RTT changes, the level of intensity as their popularity. The advantage is that the moment of change can be identified with high precision. However, one need to configure the bandwidth of density estimation which impacts greatly the smoothness of resulted density curve as well as the value of intensity. The interpretation of density graph is thus obscure, e.g. what does an intensity of 0.5 mean with bandwidth equaling 0.3 for 30 event sequences? is it a widely shared RTT change event with in the group? Moreover, this approach can not be used in an online fashion. Imagine, now we have RTT change event streams instead of sequences that are with unlimited length and having new events come in continuously. Density estimation can no longer handle such data structure.

BUCKETING BY TIME PERIOD This method has two variations, one with non-overlapping time bins, the other with sliding window. Both

variations group RTT change events in all sequences/streams into time bins/windows, then count the number of events in each bin/window as indicator of the popularity of shared RTT changes. The event count can further be normalized by the number of event sequences/streams. The time bins/windows themselves can be regards as the moments when shared RTT changes happen. Therefore the temporal granularity would be equivalent to the bin size for bucketing by non-overlapping time bins. If by sliding window, the step size at which window glides decides the time resolution.

The bin/window size should be the time range that tolerates the time shift of one shared RTT change. With that, the normalized event count per bin can be interpreted as the percentage of event sequences/streams that undergoes the shared RTT change within each time bin/window. 10 min is a reasonable value, given that 1) the RTT change detection tolerance window in Section 5 is 8 min corresponding to two measurement intervals of RIPE Atlas built-in ping measurement; 2) a shortest RTT segment spans over 3 ping measurements. In terms of sliding step, one can choose from 1 sec, the time resolution of measurement timestamps, up to 10 min which degenerates the sliding window to non-overlapping time bins. It's a trade-off between time resolution and calculation complexity.

How to choose, bin or sliding window? As a matter of fact, bucketing by sliding window can be regarded as an discrete approximating to density estimation with special kernel function (1 within the window, 0 outside). Event count at each time step is just an intermediate result, as one has to search for local maximums to pinpoint the moments of shared RTT changes. In other words, event count at each time step can not be used directly to decide whether at this time step there is a popular enough shared RTT change by comparing the (normalized) event count to a threshold. While with bucketing by non-overlapping time bins, one can.

Given the above, we settled on bucketing by non-overlapping time bins. The resulted time resolution is the same as the bin size, 10 min. We deem it enough in interdomain TE in identifying both transient and long lasting issues. The drawback is as well evident. When an event falls on the boundary of two time bins, the (normalized) event count would be in sufficient in either time bin to qualify it as an wide spread RTT change. Hopefully, the chance of having such case should be relatively low.

6.3.3 *Assumptions*

In order to initiate the inference, we made two assumptions.

Assumption 1-*single cause*. *For each detected RTT change, there is only one single cause, node or link, on the measured path.*

It is a common assumption made in TCP congestion studies [6, 109]. If there are congestion along the path, it will finally stabilize on the link with bottleneck bandwidth. We extend this assumption to inter-AS links and ASes to accommodate to the inference granularity previously discussed.

Assumption 2-*common parts*. *If measurements over multiple paths experience a shared RTT change, the common parts of these measured path are more likely to be the cause.*

It describes one possible way to satisfy Assumption 1-*single cause* when multiple RTT measurements with intersecting paths are considered. It is simply a more probably case than having multiple scattered parts of Internet simultaneously cause a significant change. We use this assumption to design special sets of measurements (inference metric) for each node and link in topology graphs to test their liability at each 10 min time bin for RTT change events.

6.3.4 *Inference for Node*

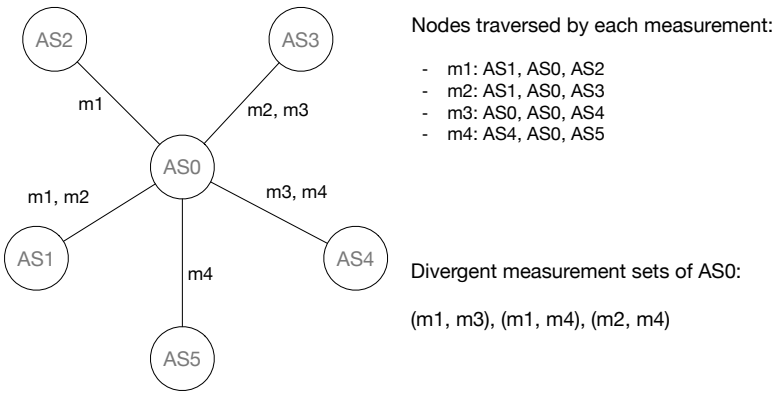


Figure 49: Example of Divergent Measurement (DM) set for AS0.

To verify whether a node (AS) n is the exclusive cause for shared RTT changes, we design for it measurement sets (inference metric), within which the only common element is the node itself. If a majority measurements in such set experience at the same time an RTT change, we can then pinpoint it to the node under investigation, according to Assumption 2-*common parts*.

More formally, we describe a measurement m by a set of nodes it traverses: $m : \{n1, n2, ..\}$. If two measurements $m1, m2$, satisfy $m1 \cap m2 = \{n\}$, we call them Divergent Measurement (DM) of n. A DM set of n, called D, shall have at least two measurements. It contains measurements that are divergent with each other (except for node n),

i.e. $\forall m1, m2 \in D, m1 \cap m2 = \{n\}$. For example, ASo in Figure 49 has three DMs sets. Each covers four divergent links adjacent to ASo.

Heuristic Node-*majority adjacent links*. *If there exists one DM set* D *of node* n *satisfying* $\mathtt{count}(D,t) > 0.5 \times \mathtt{degree}(n)$, *node* n *is then the cause for shared RTT change at time bin* t.

$\mathtt{count}(D,t)$ is the RTT change event count within measurement set D at time bin t. In other words, the heuristic requires a majority of adjacent links of node n (measured in a non-overlapping manner) experience RTT change within a certain 10 min time bin. This heuristic/criterion is a compromise of the following two extreme cases.

THE LOWER BAR According to Assumption 2-*common parts* attributing cause to the common part of the paths, it suffices to have only two measurements solely intersecting at n (which makes the two measurements a DM set) and covering only two adjacent edges to n, to judge n liable for the synchronized the RTT change witnessed by the two measurements. However, this could happen by chance for a high degree node (AS) with several bad quality links.

THE HIGHER BAR If we regard node n as an atomic point, and it causes an RTT change, then measurements following through all of its adjacent edges should be impacted. One thus should expect to find a DM set D at time bin t satisfying $\mathtt{count}(D,t) = \mathtt{degree}(n)$. Such a strict criterion is somehow not realistic from two aspects. First, a node (AS) is not an atomic point. A real issue could be only PoP-wide instead of AS-wide, as showcased in Section 6.3.2.1, where only a fraction of links would be impact. Second, due to topology constraints not all the adjacent edges of node n can be covered by any single DM set, for example ASo in Figure 49.

Heuristic Node-*majority adjacent links* is not perfect. It might still give rise to both false positive and false negative inference results. Hopefully, false positive more likely occurs on low degree nodes, for which the heuristic requirement is relatively easier to meet. Meantime, high degree nodes are prone to false negative inference. Yet, these RTT change events are not ignored, as they will reflect on some of the adjacent links to concerned nodes, and shall eventually be captured in link inference.

6.3.5 *Inference for Link*

6.3.5.1 *Early exemption*

After cause inference for each node, we can exclude all the links with any of its two nodes inferred as cause from further investigation, according to Assumption 1-*single cause*.

Afterwards, if a link causes indeed an RTT change, we shall expect a majority of measurements traversing that link to experience simultaneously (within in a same time bin) that RTT change, which is a necessary but not a sufficient condition for link liability. In other words, violating this condition, the link can be exempted from being cause for the RTT change. More formally, we define the ensemble of measurements that traverse link l as the Feature Measurement (FM) set of link l, denoted as $\mathrm{FMS}(l)$.

Heuristic Link-*majority of FM set*. *Link l is exempted from further cause inference at time bin t, if* $\mathrm{Nc}(\mathrm{FMS}(l), t) = \frac{\mathrm{count}(\mathrm{FMS}(l), t)}{|\mathrm{FMS}(l)|} \leqslant 0.5$, *where* Nc *stands for normalized event count.*

In plain language, if no greater than half of the measurements passing through link l experience an RTT change at time bin t, link l is deemed not likely to be the cause. Ideally, all the measurements in FM set shall be impacted if the link is the cause. However, due to imperfection of RTT change detection and sub-AS level structure, e.g multiple separate links between two ASes, requiring the totality of FM set would be too strict. That is why we lower the bar to half of the measurements and widen the scope for further investigation, in order to mitigate potential false negative, of course, at the cost of more false positive that is less unwanted.

6.3.5.2 *Case by case inference*

For remaining links after early exemption, their liability depends on both adjacent and non-adjacent links. Imagine an incident happens at the core of the Internet or at one large IXP, a wide range of measurements traversing peripheral links, from right next the cause till the edge of the Internet, can be potentially impacted. Yet these peripheral links are not responsible for the RTT change. To traceback the true cause of change, different criteria are needed for links in different topological layouts.

To facilitate the discussion, we denote the two nodes of link l as $n1_l$ and $n2_l$. Following definitions apply to both nodes. We define the Extension Link (EXT) at node $n1_l$ as adjacent links (at $n1_l$) whose FM set has non-empty intersection with $\mathrm{FMS}(l)$. The ensemble of EXT at node $n1_l$ is denoted as $\mathrm{EXT}(n1_l, l)$. We then refer to a link $el \in \mathrm{EXT}(n1_l, l)$ as an Impacted Extension Link (iEXT) of l at time bin t, if $\mathrm{Nc}(\mathrm{FMS}(el) \cap \mathrm{FMS}(l), t) > 0.5$ following the Heuristic Link-*majority of FM set*. The number of iEXTs in $\mathrm{EXT}(n1_l, l)$ at time bin t is denoted as $\langle \mathrm{EXT}(n1_l, l), t \rangle$.

TWO OPEN ENDS **if** $|\mathrm{EXT}(n_1, l)| > 1$ and $|\mathrm{EXT}(n_2, l)| > 1$.

This is the case where link l has multiple extension links at both ends, as shown in Figure 50. If at both ends/nodes, there are more than one iEXT, link l as the only common part besides its two nodes

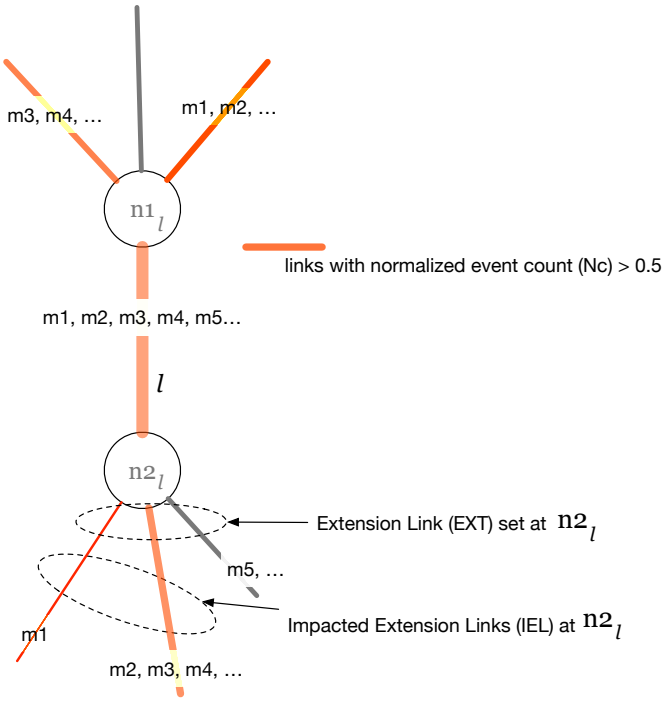


Figure 50: Illustration of link l with two open ends.

(previously not judged liable) can be inferred as the cause of the RTT changes experienced by l itself and all its iEXTs, according to Assumption 2-*common parts*. Otherwise, we don't have enough evidence to do so. Here below the inference logic in pseudo code.

if $\langle \text{EXT}(n1_l, l), t\rangle > 1$ and $\langle \text{EXT}(n2_l, l), t\rangle > 1$ **then**
    l is the cause for RTT change at t ▷ Assumption 2-*common parts*
else
    l is NOT the cause
end if

FORK SHAPE **else if** $|\text{EXT}(n_1, l)| = 1$ and $|\text{EXT}(n_2, l)| > 1$.

This is the case where link l has one EXT at one end, and multiple at the other end, as the link l in Figure 51. If the end with multiple EXTs doesn't have multiple iEXTs at a given moment, there is just no enough evidence to judge the link responsible for the RTT changes. Otherwise, the liability lies on the link l and its single EXT at the other end, since they are the common parts of multiple measurement under RTT change, according to Assumption 2-*common parts*.

If the single EXT is inferred as cause, we then know that link l can not be the cause at the same moment, according to Assumption 1-*single cause*. If we don't have enough evidence to attribute the cause to the single EXT, we are actually not sure whether link l is the cause

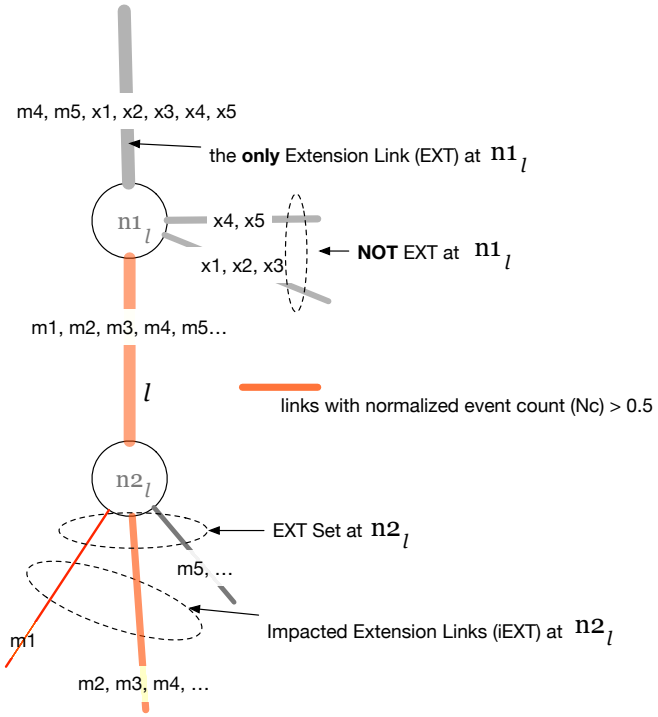


Figure 51: Illustration of link l in fork shape topology.

or not. It is because, the true cause can be up in the further upstream of the single EXT. In order to avoid passing inference results of all the upstream links of the single EXT through stacks of recursive call of inference logic, we flag l as a LIKELY cause. If no other established cause of RTT change sit on the paths traversed by $\text{FMS}(l)$ at time bin t, then link l is the cause. Otherwise, according to Assumption 1-*single cause*, link l is judged free of responsibility.

It is possible that link l and its single EXT run into a loop of inference dependence. That is when the inference result of l's single EXT depends as well on l. The presence of loop indicates that the current measurements are not enough to disentangle l and its single EXT. Therefore, both links are deemed to be LIKELY the cause.

Here below the inference logic in pseudo code for link under fork shape topology, assuming $|\text{EXT}(n1_1, l)| = 1$. The other case with $|\text{EXT}(n2_1, l)| = 1$ and $|\text{EXT}(n1_1, l)| > 1$ is symmetric, thus not repeated.

if $\langle \text{EXT}(n2_l, l), t \rangle > 1$ **then** ▷ the case where $|\text{EXT}(n1_1, l)| = 1$
  if $\text{EXT}(n1_l, l)$ can be early exempted **then**
    l is LIKELY the cause at t ▷ True cause can still be elsewhere
  else ▷ now it depends on inference result of the only link in $\text{EXT}(n1_l, l)$
    if dependence loop between l and $\text{EXT}(n1_l, l)$ **then**

```
            l is LIKELY the cause          ▷ unable to disentangle l and
EXT(n1_l, l)
        else
            if EXT(n1_l, l) is the cause then
                l is NOT the cause          ▷ Assumption 1-single cause
            else
                l is LIKELY the cause       ▷ True cause can still be
elsewhere
            end if
        end if
    end if
else
    l is NOT the cause
end if
```

NOODLE **else if** $|\text{EXT}(n2_1, l)| = 1$ and $|\text{EXT}(n1_1, l)| = 1$.

This is the case where there is only one EXT at both ends of the link, thus the topology is noodle like. Similar to inference under fork shape topology, in this case the result depends on both EXTs and can run in to dependence loop. Here below the inference logic in pseudo code.

```
if both EXT(n2_1, l) and EXT(n1_1, l) can be early exempted then
    l is LIKELY the cause at t      ▷ True cause can still be elsewhere
else▷ it depends on inference result of EXT(n2_1, l) and EXT(n1_1, l)
    if dependence loop between l and EXT(n1_l, l) then      ▷ now
depends on EXT(n2_1, l)
        if EXT(n_2, l) is the cause then
            l is NOT the cause          ▷ Assumption 1-single cause
        else
            l is LIKELY the cause       ▷ unable to disentangle l and
EXT(n1_l, l)
        end if
    else if dependence loop between l and EXT(n2_l, l) then      ▷
symmetric case of the previous if clause
        if EXT(n1_l, l) is the cause then
            l is NOT the cause          ▷ Assumption 1-single cause
        else
            l is LIKELY the cause       ▷ unable to disentangle l and
EXT(n2_l, l)
        end if
    else                                ▷ no dependence loop
        if EXT(n1_l, l) or EXT(n2_l, l) is the cause then
            l is NOT the cause          ▷ Assumption 1-single cause
        else
            l is LIKELY the cause ▷ True cause can still be elsewhere
        end if
```

end if
end if

TWO CLOSED END **else if** $|\text{EXT}(n2_1, l)| = 0$ and $|\text{EXT}(n1_1, l)| = 0$.
This case doesn't necessary mean link l is disconnected from the rest of topology, but rather all the measurements in $\text{FMS}(l)$ are within the two nodes of l. According to Assumption 2-*common parts*, l as common part of $\text{FMS}(l)$ is thus the cause of the RTT changes experienced by measurements in $\text{FMS}(l)$.

CLOSED END FORK **else if** $|\text{EXT}(n1_1, l)| = 0$ and $|\text{EXT}(n2_1, l)| > 1$. This is a simplified case of fork shape illustrated in Figure 51. With one end completely closed, we are free of inference dependency on other links. Here below the inference logic in pseudo code. The case with $|\text{EXT}(n1_2, l)| = 0$ and $|\text{EXT}(n1_1, l)| > 1$ is symmetric to the case under discussion, thus not repeated.

if $\langle \text{EXT}(n2_l, l), t \rangle > 1$ **then**
l is the cause for RTT change at t ▷ Assumption 2-*common parts*
else
l is NOT the cause
end if

CLOSED END NOODLE **else if** $|\text{EXT}(n1_1, l)| = 0$ and $|\text{EXT}(n2_1, l)| = 1$.

This case is a simplified version of noodle topology, where inference dependence can only happen at one end. Again symmetric case, $|\text{EXT}(n2_1, l)| = 0$ and $|\text{EXT}(n1_1, l)| = 1$, will no longer be repeated.

if $\text{EXT}(n2_1, l)$ can be early exempted **then**
l is LIKELY the cause at t ▷ True cause can still be elsewhere
else ▷ it depends on inference result of $\text{EXT}(n2_1, l)$
if dependence loop between l and $\text{EXT}(n2_l, l)$ **then**
l is LIKELY the cause ▷ unable to disentangle l and $\text{EXT}(n2_l, l)$
else
if $\text{EXT}(n1_l, l)$ is the cause **then**
l is NOT the cause ▷ Assumption 1-*single cause*
else
l is LIKELY the cause ▷ True cause can still be elsewhere
end if
end if
end if

Till now, we have discussed all possible topology layouts formed by $\text{FMS}(l)$. We only infer l as causes under Assumption 2-*common parts*. Link l is exempted from liability using Assumption 1-*single cause*, or Assumption 2-*common parts* can not be met when topology actually allows (fork and one closed end fork topology). For rest cases, when

the available measurements does not allow to pinpoint one single link with Assumption 2-*common parts*, a LIKELY cause flag is attributed to multiple links.

6.4 VISUALIZATION TOOLS AND CASE STUDY

6.4.1 *A typical 'fork' illustrated with tools*

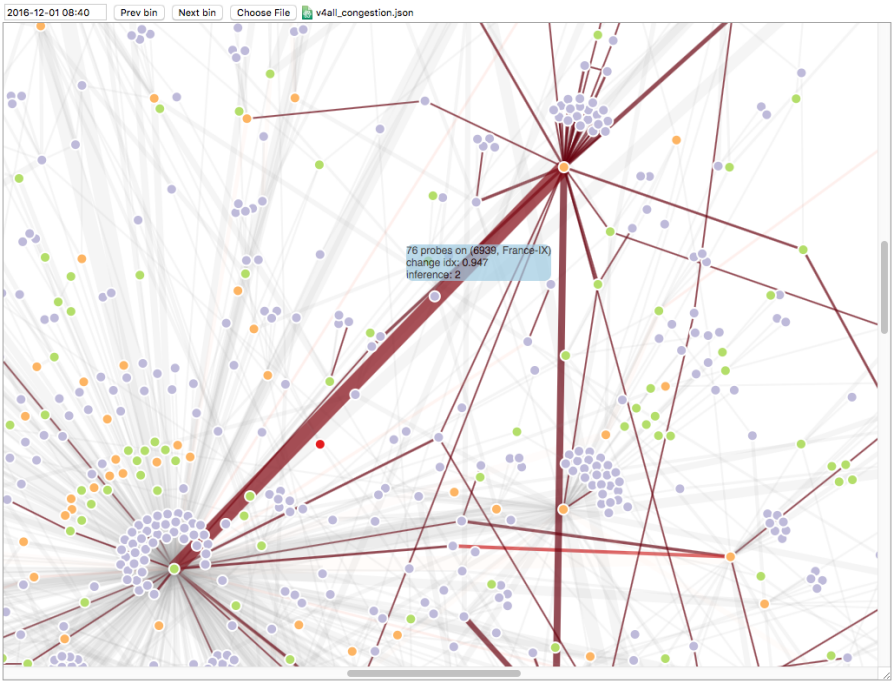


Figure 52: Normalized event count view of the visual inspection tool.

We implemented an interactive visualization tool to inspect the normalized event count and inference result of each link on the AS-level topology revealed by RIPE Atlas traceroute measurements. Figure 52 is a snapshot of the tool under normalized event count view. Each circle in the figure represents an AS or IXP learnt from traceroute measurements. Violet ones host RIPE Atlas probes, orange ones are IXPs, while green ones are transit ASes. More FMs a link has, thicker the corresponding line is. At a given 10 min time bin (indicated in the left top corner of the graph), larger the normalized change event count is, deeper the red color over the link.

In Figure 52, from 2016-12-01 08:40 to 08:50 UTC time, almost all the measurements (72 out of 76) traversing the link between France-IX and Hurricane (AS6939) experienced significant RTT changes. The time series of these RTT measurements are given in Figure 53, visualized with another tool we developed for the inspection of multiple time-series, `https://github.com/WenqinSHAO/rtt_visual_multi.git`. Many EXTs of (France-IX, Hurricane) are as well coated in deep

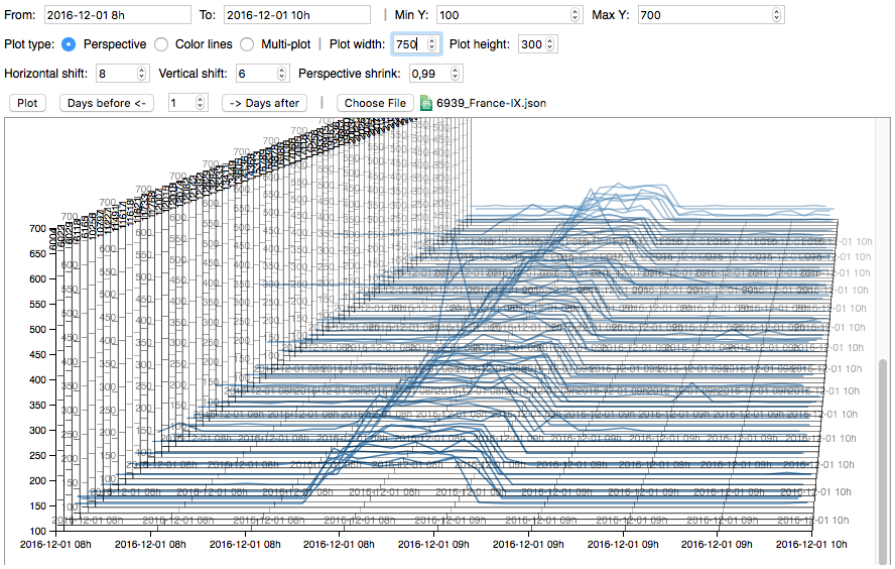


Figure 53: FM RTT measurements of link (France-IX, Hurricane).

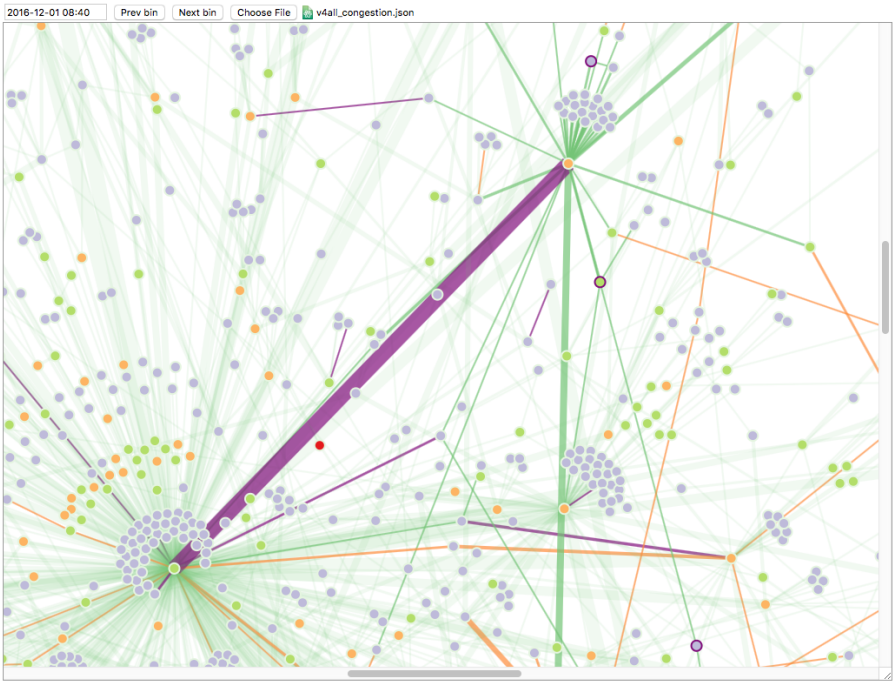


Figure 54: Inference view of the visual inspection tool.

red, sign of commonly shared RTT changes on these links. As all the measurements are toward one single destination DNS b-root, (France-IX, Hurricane) fits in the case of fork shape topology, with it's Hurricane end having only one single EXT. As expected, the proposed inference logic inferred (France-IX, Hurricane) as the single cause for this massively spanned RTT change according to Figure 54. In the figure, links inferred as cause are colored in deep violet, while orange is for LIKELY causes, green for links free of liability within the time bin.

As a matter of fact, node France-IX may as well be the cause, given that all its adjacent links have a near to 1 normalized event count at that moment. However, the topology condition around France-IX, formed by all the measurements passing through it, doesn't allow a DM set covering half of its adjacent link so as to satisfy Heuristic Node-*majority adjacent links*, as all these measurements converges on (France-IX, Hurricane).

6.4.2 *A likely PoP-level issue*

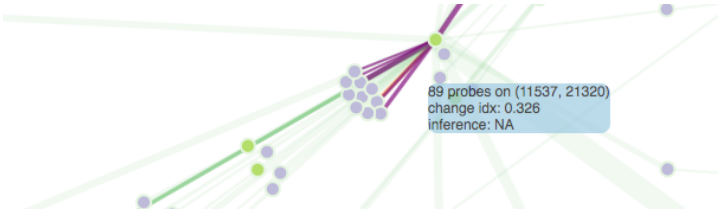


Figure 55: Violation of Assumption 1-*single cause*, a probable PoP-level issue under inference view at 2016-12-01 11:10 UTC time.

Figure 55 offers a probable PoP-level issue. The node surrounded by multiple violet links is Géant (AS21320). Similar to Franc-IX, it is not possible to find a DM set that covers enough of its adjacent links to make it liable. Moreover, the number of these violet links doesn't surpass half of the node degree as required by Heuristic Node-*majority adjacent links*. The node is thus not judged liable for the shared RTT changes. However, link (Géant, Internet2-AS11537), labeled in the figure, is neither the cause of RTT change, since its normalized event count is only around 0.3, far below 0.5 required by Heuristic Link-*majority of FM set*, thus judged clean in early exemption. Since there is no further suspects of cause along the path toward the destination, Géant is the only possible location where the shared RTT change could originate according to Assumption 2-*common parts*. However, with current topology granularity and Heuristic Node-*majority adjacent links* setting, such inference can not be straightforwardly reached. Hopefully, with the help of this visual inspection, such case can be rather easily assessed by a network engineer.

CONCLUSION

In this chapter, we developed a series of inference procedures to identify the location of RTT changes seen in end-to-end measurements. Such inference is made possible by the measurements from widely distributed RIPE Atlas probes. With this visibility, route optimization toward destination prefixes without direct measurements are made possible. To better illustrate the inference process and the identified causes for RTT changes, we developed two interactive vi-

sualization tools to plot inference metrics and results on a topology graph learned through path measurements.

The inference procedure is not perfect as showcased in the case study. Refinements are obviously needed. On top of that, it is extremely hard to validate the output for lack of ground truth. We believe insightful feedback can be received if we publish in quasi-real time inferred causes of RTT change in Internet. This calls for apparently much more efforts on various aspects, such as online change detection, wider measurement coverage (more than b-root), etc. These are all promising directions to pursuit in the future.

7 CONCLUSION

7.1 THESIS SUMMARY

This thesis is developed around a central pursuit of *making better use of various network measurements in outbound interdomain traffic engineering for stub ASes*.

We emphasized the necessity of focusing on the most important destinations. Through investigation over temporal dynamism of per prefix volumes, we came up with simple methods that effectively select destination prefixes with large traffic volumes. The overall scalability of the measurement TE system can hence be improved, without sacrificing much of the traffic coverage.

Later on, we focused on latency measurements. We showcased and diagnosed some data quality issues previously unattended. Guidelines to mitigate their impacts on data processing and route selection were discussed.

To better interpret performance measurements, we introduced changepoint analysis to the processing of RTT time series. These methods detect significant changes in path performance, and serve as trigger for route re-selection. To enable and encourage future efforts, we built an evaluation framework for the change detection on Internet RTT measurements.

Finally, with the help of changepoint analysis, we were able to qualify the percentage of a measurement group that underwent RTT changes at a same moment. We further inferred the network locations that potentially caused these changes. This visibility enables route optimization for prefixes that we were not able to measure directly. To better illustrate the inference process and the identified causes, we devised interactive visualization tools to plot inference metrics and results on a topology graph.

7.2 CONTRIBUTIONS

7.2.1 *Scalable prefix selection*

It is recommended to perform traffic engineering only for destinations with import traffic volume. One key challenge is how to *efficiently* predict the volume importance for a great amount of destination prefixes repeatedly.

We analyzed real traffic measurements from nine different networks located in five different countries. A realistic view has been obtained

on the distribution of traffic volume associated with BGP prefixes, as well as its variation over time. We observed that the prefixes with most important accumulated volume over a week have relatively stable hourly volume. Based on this observation, we proposed three simple metrics (also easy to compute) to proactively select prefixes with important foreseeable traffic volume. We demonstrated that the metrics we proposed led to better volume coverage compared to one existing solution, the Grey model.

7.2.2 *Data quality concerns*

We studies two data quality issues previously unattended. One comes from the RIPE Atlas measurement platform, the other is specific to path performance measurement involving multiple probes.

The problem with Atlas was that some datapoints were missing for measurements meant to be performed regularly at fixed interval. After investigating all the available probes at that time, we found that on a small fraction of probes, measurements were interrupted over a long duration while the probes remain well connected to the central controller. This discovery indicated potential unknown issues with the measurement system. To help advance the investigation, we shared with the RIPE Atlas team all the long missing segments and concerned Atlas probes identified in the study.

When measuring the performance of an AS path, multiple hosts/Atlas probes in a same prefix can be employed. A specific dataset measuring a same AS path revealed that a part of probes experienced additional delay changes originated from access network congestion when measuring. A clear message from this study was to prefer RTT measurement with least additional variations, for a better and purer representation of performance issues from network layer and at AS level.

7.2.3 *Change detection for RTT measurements*

We introduced changepoint analysis to the processing of RTT measurements. Automatically detected changes in path performance can potentially serve as a robust trigger to route re-selection. The main challenge resides in quantifying the detection performance a method over RTT measurements. We henceforth devised an evaluation framework consisting of a carefully labeled 34,008-hour RTT dataset as ground truth and a scoring method tolerates slight time shifts. This framework paved the way for future studies in the domain, such as design online change detection methods for RTT measurements.

To understand the network implication of detected RTT changes, we correlated them with path changes close in time. This allowed

us to investigate the sensitivity distinction across different change detection methods.

7.2.4 *Change location inference*

Detecting the cause of performance issue, such as congestion, in the middle of Internet has always been very challenging. Our approach relied on the massive measurements that are geographically distributed from RIPE Atlas. Basing on the assumption that RTT changes observed on multiple paths tend to occur on the intersecting parts, we developed a series of inference logic to narrow down the scope of potential causes under all possible topology layouts. We built two interactive visualization tools to enable the demonstration of shared RTT changes and inferred causes under a meaningful Internet topology context.

8

FUTURE WORKS

We attacked various topics regarding different measurements and methods in this thesis. What we achieved so far is merely a beginning or an enabling step to those directions. There is still a long journey ahead. We sketch the most important issues left open and discuss possible approaches to them when possible.

8.1 ONE MORE DATA QUALITY CONCERNS

We revealed differences among measurements toward multiple probes within a same BGP prefix in Chapter 4. The adopted clustering approach separated them into two groups, one with 'noisy' time series and the other the with 'smooth' ones. The inherent reason for such difference is the sub-prefix level path difference associated to these probes, e.g. some of these paths are congested while others are not.

Another data quality issue wherefrom rises: *do the measured performance data indeed represent the actual traffic performance toward the same prefix on a given path*? This question can be further split.

First, *are we measuring the 'right' hosts within the destination prefixes?* The concern is already illustrated by the above highlighted sub-prefix level path difference. Even if we measure certain hosts found in real traffic, they don't necessary speak for the rest destinations within the same prefix. Therefore, the question is transformed into: *Do we need to split BGP prefixes into finer pieces according to path and performance homogeneity? And how?* As articulated in Chapter 6.3.2, it is in general beneficial acquiring a visibility of finer granularity, since such information might help optimize certain cases otherwise deemed impossible. However, such benefit comes with a cost. Intuitively, one first has to explore the sub-prefix structure seen from a certain client network, which is apparently not trivial. Lee and Spring [113] explored such sub-prefix route difference. Their work can serve as starting point for further inspection.

Second, *do we measure the traffic in the right way?* ICMP ping is known to have a different queuing priority in forwarding compared to other traffic, thus might not represent the real traffic performance. However, latency measurement via TCP SYN stealth is not perfect either. Increasing web traffic nowadays is transported in QUIC which is UDP based. Its queuing delay could thus differ from competing TCP connections on the same path. The deployment of Active Queue Management (AQM) will signal the load change rather in packet drop instead of latency. Passive measurements are more faithful and could

thus complement active probing. However, how to strike the balance between sampling rating and time resolution is tricky. Application specific instrumentation can as well help. The key challenge consists in well defined and commonly accepted telemetry interface between various applications and the network operation system.

Third, *actually routing traffic on a path might change the the performance previously perceived.* This interaction between traffic and performance is immediate on the transit links. That is why researchers optimized for cost and congestion avoidance on transit links the same time. Recently, this interaction is less likely to happen on well provisioned transit links but is still possible on some bottlenecks in the middle of Internet. Since the capacity of the bottleneck is not explicitly known and shared with other traffic, it becomes much more challenging to predict the change of path performance before actually moving the traffic. One possible countermeasure is move only a part, instead of the totality, of the traffic associated to a prefix. However, that brings challenge to traffic steering under BGP. Source routing might be a cure.

8.2 CHANGE DETECTION FOR STREAMING DATA

In Chapter 5, we examined and discussed the detection sensitivity of several changepoint analysis methods on RTT measurements. However, all of them are offline methods. They consume an entire time series and output at once all the moments of change identified in that time series. Apparently, such methods can only be applied to past measurement records. While in real TE practice, performance measurements are rather endless streams. This requires change detection to be done rather in an online fashion. It is worthy of noting that online detection is as well the basis for other directions that will be discussed later on.

Online change detection updates incrementally moments of change as new data flow in. If we apply repeatedly offline methods every time new measurement data is produced, an 'online'-like behavior can be achieved. However, as the length of data accumulates, the computational cost will become prohibitively high over time. Eventually, most resource is actually spent on detecting again and again changes previously detected. To overcome these issues, we devised an adaptive way of using offline methods to lower down the computation overhead to a practical level: `https://github.com/WenqinSHAO/path_change_alert.git`. In this proposition, we applied a sliding window to measurement streams. Change detection is only performed on the data within the window. Therefore the computational cost doesn't increase with time but rather limited by the window size. To further narrow down the scope, we reinitialize the window once a change-

point is detected. It then suffices to detect at most one single most significant change moment at each arrival of new datapoint.

The disadvantage of the above approach is three-fold. First, the window size needs to be properly tuned. We believe that the evaluation framework introduced in this thesis can help in this process. Second, the method tends to be oversensitive. Since it ignores all the datapoints before previous change, the next potential changepoint could be only locally significant. Finally, the computation of changepoints is not incremental, which leaves room for further performance improvements. Being incremental in calculation means that the detection method can learn from the result of previous run and update it with only the new incoming datapoint to obtain the detection result of current run. It is important to note that the calculation on previous datapoints is not thrown away, which allows incremental update to have the potential to overcome the second issue. Various methods [41, 60, 76, 90, 95, 98] have been proposed. We can again employ the proposed evaluation framework to quantify the detection sensitivity and relevance an on RTT measurements with these methods.

Further, the scoring method used in the evaluation frameworks needs as well adaptation. One fundamental feature of online methods is the delay of detection. It is the time between the actual occurrence of a change and the moment when a method successfully identifies it as a change. Shorter delay suggests that a method is more reactive to changes and thus can leave more time for route optimization. The delay of detection can be seen as a manifestation of sensitivity on time axes. How to strike an appropriate balance between precision and recall on this dimension (i.e. prefer short delay when possible) would be a key issue.

8.3 CONGESTION AND PATH CHANGE

Concerning the temporal correlation between RTT and path change moments, we received a request from one of our paper reviews to classify each RTT change into path change or congestion as its cause. We were willing to but left unable to fulfill the request. It was mainly because path measurements on the reverse direction were not available. Since not all the path changes were identified, RTT changes unmatched to path changes in the forwarding direction were actually a mixture consequence of congestion, path changes in the reserve direction and oversensitive change detection.

Nonetheless, a vague gauge was possible with Table 6. If we boldly assume that path changes on reverse paths cause comparable amount of RTT changes as forwarding paths do, there are still a majority of RTT changes left unmatched to any path changes, thus potentially caused by congestion. This observation highlights the potential importance of congestion in Internet transmission.

Current best practice of Internet congestion detection [93] only identifies persistent congestion that are mainly caused by lack of capacity and demonstrates regular daily pattern. Proper capacity dimensioning is the ultimate cure to it. While in TE, besides persistent congestion, route selection has to as well react to transient congestion. It requires in first place *detecting timely the occurrence of network congestion experienced by end-to-end measurements*.

Through visual inspection, e.g. Figure 44, we noticed that it might be possible to tell apart RTT changes caused by path changes and congestion by merely looking at the shape of RTT time series. The rule of thumb was that if an RTT change was related to congestion, the RTT variation during the congested period was obviously more important. In order to establish this relationship more rigorously, more efforts to understand the RTT time series morphology under various transmission conditions on multiple time resolutions are needed. The emergence of new control mechanism such as BBR [109, 117] and queue scheduling/dropping schemes call for as well such effort. On the other hand, we need online change detection to capture in real time the change of shape, characteristics of RTT measurements.

8.4 RTT CHANGE CAUSE INFERENCE

In Chapter 6, we tried to infer the locations causing RTT changes seen in the end-to-end measurements, through correlation of RTT changes at about same moments. Online change detection will help to enable such functionality in a real time mode. RTT change cause inference is a difficult quest in which we see great utility for interdomain TE and beyond. Both the main inference logic and the demo tools need to be further improved, so that the output is more reliable and hence useful to the community.

VALIDATION For lack of ground truth, it is very challenging to validate the changes, and the causes of change inferred with the proposed methods. The only approach to ground truth on Internet-wide performance events would be trough crowd sourcing from network administrators. Incentivizing the exchange of such information is then the key. One possibility is to persuade network administrators to do so for their own benefit. Our pursuit of inferring RTT change cause provides exactly the visibility that they lack and yet are beneficial for various TE tasks.

Based on real-time RTT change detection (once achieved), we plan to develop an API and a Web-based visualization application that:

- publish in quasi-real time the inferred locations of RTT change;
- allow receiving and processing external measurement streams if certain networks have a specific area they wish to focus on;

- take feedback, event annotation, error report from network administrator users via both programming interfaces and human-friendly approaches.

Each network has only a limited visibility to its surrounding area. Gathering the information from various scattered networks connects these separated shards of visions. More importantly, feed our inference logic with the obtained ground truth can further light up the parts of Internet whose ground truths remain unavailable due to various practical reasons.

To better digest the feedback, we plan to add learning elements to our reasoning procedure. A first step would be revisiting the validity of assumptions and heuristics made previously and evaluate their impact on the precision of inference results. Self-improving inference logical can be achieved through carefully abstracted knowledge representation of available information and automated reasoning.

Since the major measurement sources is RIPE Atlas, we seek for as well a closer cooperation with the team. We envision to integrate our visibility into theirs to better serve the networking community.

HANDLE TOPOLOGY CHANGES During RTT change location inference, a hidden assumption was that the underlying topology remain unchanged, which is not always true. We adopted a simple approximation by constructing topology graph and performing RTT change cause inference day by day, knowing that the AS level path changes are rare on most Internet paths. However, when AS level path changes do happen within a day and cause RTT changes, current inference logic can not properly distinguish the cause being the topology shift or congestion on a same link. Moreover, the presence of AS level load balancing links causes topology changes on an even smaller time scale.

How to correctly attribute RTT changes in these cases is thus essential to the correctness of inference results. It requires first a data structure modification to encode the Internt topology changes in an efficient way. For example, record rather the delta of topology snapshots. Then we need to tie performance measurements flexibly and precisely to different paths. The temporal correlation study between RTT and path change carried out in Chapter 5.9 provides clues. Finally, the inference logic should be aware of topology changes to properly attribute the causes of observed RTT changes.

FINER INFERENCE GRANULARITY Through the case study in Chapter 6, we realized that sub-AS level incident RTT changes can indeed happen and that current inference granularity is not sufficient to pinpoint them. To achieve finer inference granularity, a more precise topology graph, e.g. PoP level, is needed. `http://popmap.io/` is a recent effort to follow in this direction.

MEASUREMENT SCOPING Several improvements regarding measurements scoping are as well possible. All the available Atlas probes were considered in the previous case study. It is not hard to notice that some of them were redundant. The inference efficiency could be improved if we can devise a minimum set of measurements in achieving a specified topology coverage of inference.

Another application of such technique would be the calculation of the additional measurements required for node and links currently that can not be (surely) inferred. Combined with previously mentioned API and Web application, this would allow individual networks to better engineer their own measurements to achieve a better monitoring visibility.

On the other hand, the presence of redundant measurements actually increases the statistical significance of inference results. When redundant measurements do not harm to exist, it would be informative to express as well the statistical confidence nuance across inference results.

8.5 ROUTE SELECTION ALGORITHM

We argued that changepoint analysis offers a robust data representation for RTT measurements by extracting only the moments of significant change. These moments are when route re-selection is potentially needed. It would be better if we actually propose a route selection algorithm basing on change detection, evaluate its performance gain and compare its gain to previous algorithms. It would be even better if we could further incorporate RTT change location inference in route re-selection and demonstrate how much more traffic can hence be optimized. The evaluation of route selection mechanism requires end-to-end measurements via multiple providers, and realistic traffic demand. RIPE Atlas falls short in providing such data. Therefore, we envision closer cooperation with the industry sector to overcome the difficulties concerning data collection from real networks discussed in Chapter 4.

BIBLIOGRAPHY

[1] Hiroaki Sakoe and Seibi Chiba. "Dynamic Programming Algorithm Optimization for Spoken Word Recognition." In: *IEEE transactions on acoustics, speech, and signal processing* ASSP-26.1 (1978), pp. 43–49. ISSN: 00963518. DOI: 10.1109/TASSP.1978.1163055.

[2] Peter J. Rousseeuw. "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis." In: *Journal of Computational and Applied Mathematics* 20 (1987), pp. 53–65. ISSN: 03770427. DOI: 10.1016/0377-0427(87)90125-7. arXiv: z0024.

[3] Van Jacobson. "Congestion avoidance and control." In: *CCR* 18.4 (1988), pp. 314–329. ISSN: 01464833. DOI: 10.1145/52325.52356. arXiv: arXiv:1011.1669v3. URL: http://dl.acm.org/citation.cfm?id=52356.

[4] Deng Julong. "Introduction to Grey System Theory." In: *The Journal of Grey System1* 1 (1989), pp. 1–24. ISSN: 09573720.

[5] Lajos Horvath. "The maximum likelihood method for testing changes in the parameters of normal observations." In: *The Annals of Statistics* 21.2 (1993), pp. 671–680.

[6] Matthew Mathis, Jeffrey Semke, Jamshid Mahdavi, and Teunis Ott. "The macroscopic behavior of the TCP congestion avoidance algorithm." In: *ACM SIGCOMM Computer Communication Review* 27.3 (1997), pp. 67–82.

[7] W. Fang and L. Peterson. "Inter-AS traffic patterns and their implications." In: *GLOBECOM '99*. Vol. 3. IEEE, 1999, pp. 1859–1868. ISBN: 0-7803-5796-5. DOI: 10.1109/GLOCOM.1999.832484.

[8] J S Richman and J R Moorman. "Physiological time-series analysis using approximate entropy and sample entropy." In: *American journal of physiology. Heart and circulatory physiology* 278.6 (2000), H2039–H2049. ISSN: 0363-6135.

[9] Jie Chen and Arjun K. Gupta. "Parametric Statistical Change Point Analysis." In: *Birkhauser* (2001). ISSN: 14337851. DOI: 10.1002/1521-3773(20010316)40:6<9823::AID-ANIE9823>3.3.CO;2-C.

[10] Lixin Gao. "On inferring autonomous system relationships in the Internet." In: *IEEE/ACM Transactions on Networking* 9.6 (2001), pp. 733–745. ISSN: 10636692. DOI: 10.1109/90.974527.

[11] Mark J. Coates and Robert D. Nowak. "Sequential Monte Carlo inference of internal delays in nonstationary data networks." In: *IEEE Transactions on Signal Processing* 50.2 (2002), pp. 366–376. ISSN: 1053587X. DOI: 10.1109/78.978391.

[12] Mark Coates, Alfred O. Hero, Robert Nowak, and Bin Yu. "Internet tomography." In: *IEEE Signal Processing Magazine* 19.3 (2002), pp. 47–65. ISSN: 10535888. DOI: 10.1109/79.998081.

[13] Francesco Lo Presti, N. G. Duffield, Joe Horowitz, and Don Towsley. "Multicast-based inference of network-internal delay distributions." In: *IEEE/ACM Transactions on Networking* 10.6 (2002), pp. 761–775. ISSN: 10636692. DOI: 10.1109/TNET.2002.805026.

[14] Bruno Quoitin, Steve Uhlig, and Olivier Bonaventure. "Using Redistribution Communities for Interdomain Traffic Engineering." In: *From QoS Provisioning to QoS Charging: Third COST 263 International Workshop on Quality of Future Internet Services, QofIS 2002 and Second International Workshop on Internet Charging and QoS Technologies, ICQT 2002 Zurich, Switzerland, October 16–18, 2002 Proceedings*. Ed. by Burkhard Stiller, Michael Smirnow, Martin Karsten, and Peter Reichl. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 125–134. ISBN: 978-3-540-45859-3. DOI: 10.1007/3-540-45859-X_12. URL: https://doi.org/10.1007/3-540-45859-X_12.

[15] Aditya Akella, Srinivasan Seshan, and Anees Shaikh. "An empirical evaluation of wide-area internet bottlenecks." In: *ACM SIGMETRICS Performance Evaluation Review* 31.1 (2003), p. 316. ISSN: 01635999. DOI: 10.1145/885651.781075.

[16] Aditya Akella, Bruce Maggs, Srinivasan Seshan, Anees Shaikh, and Ramesh Sitaraman. "A measurement-based analysis of multihoming." In: *SIGCOMM '03*. New York, New York, USA: ACM Press, 2003, p. 353. ISBN: 1581137354. DOI: 10.1145/863993.863995.

[17] Nick Feamster, Jay Borkenhagen, and Jennifer Rexford. "Guidelines for interdomain traffic engineering." In: *ACM SIGCOMM CCR* (2003).

[18] Young Hyun and Andre Broido. "On third-party addresses in traceroute paths." In: *PAM*. 2003.

[19] Gang Liang and Bin Yu. "Maximum pseudo likelihood estimation in network tomography." In: *IEEE Transactions on Signal Processing* 51.8 (2003), pp. 2043–2053. ISSN: 1053587X. DOI: 10.1109/TSP.2003.814464.

[20] Jessica Lin, Eamonn Keogh, and Wagner Truppel. "Clustering of streaming time series is meaningless." In: *ACM SIGMOD - DMKD* (2003), p. 56. DOI: 10.1145/882095.882096.

[21] B. Quoitin, S. Uhlig, C. Pelsser, L. Swinnen, and O. Bonaventure. "Interdomain traffic engineering with BGP." In: *IEEE Communications Magazine* 41.5 (2003), pp. 122–128. URL: http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=1200112.

[22] Meng Fu Shih and Alfred O. Hero. "Unicast-based inference of network link delay distributions with finite mixture models." In: *IEEE Transactions on Signal Processing* 51.8 (2003), pp. 2219–2228. ISSN: 1053587X. DOI: 10.1109/TSP.2003.814468.

[23] Yolanda Tsang, Mark Coates, and Robert D. Nowak. "Network delay tomography." In: *IEEE Transactions on Signal Processing* 51.8 (2003), pp. 2125–2136. ISSN: 1053587X. DOI: 10.1109/TSP.2003.814520.

[24] David K. Goldenberg, Lili Qiuy, Haiyong Xie, Yang Richard Yang, and Yin Zhang. "Optimizing cost and performance for multihoming." In: *SIGCOMM CCR* 34.4 (Oct. 2004), p. 79. ISSN: 01464833. DOI: 10.1145/1030194.1015478.

[25] Konstantina Papagiannaki, Nina Taft, and Christophe Diot. "Impact of Flow Dynamics on Traffic Engineering Design Principles." In: *INFOCOM '04*. Vol. 4. 2004, pp. 2295–2306. ISBN: 0780383567.

[26] Michal Pióro and Deepankar Medhi. *Routing, flow, and capacity design in communication and computer networks*. Elsevier, 2004.

[27] Ca Ratanamahatana and E Keogh. "Everything you know about dynamic time warping is wrong." In: *Third Workshop on Mining Temporal and Sequential Data* (2004), pp. 22–25. ISSN: 00903493. DOI: 10.1097/01.CCM.0000279204.24648.44. URL: http://spoken-number-recognition.googlecode.com/svn/trunk/docs/Dynamictimewarping/DTW{_}myths.pdf.

[28] Steve Uhlig and Olivier Bonaventure. "Designing BGP-based outbound traffic engineering techniques for stub ASes." In: *ACM SIGCOMM Computer Communication Review* 34.5 (2004), pp. 89–106. ISSN: 01464833. DOI: 10.1145/1039111.1039115. URL: http://dl.acm.org/citation.cfm?id=1039111.1039115.

[29] Qi He, Constantine Dovrolis, and Mostafa Ammar. "On the predictability of large transfer TCP throughput." In: *SIGCOMM '05*. Vol. 35. 4. New York, New York, USA: ACM Press, Aug. 2005, p. 145. ISBN: 1595930094. DOI: 10.1145/1080091.1080110.

[30] Eamonn Keogh and Chotirat Ann Ratanamahatana. "Exact indexing of dynamic time warping." In: *Knowledge and Information Systems* 7.3 (2005), pp. 358–386. ISSN: 0219-1377. DOI: 10.1007/s10115-004-0154-9.

[31] Konstantina Papagiannaki, Nina Taft, Zhi-Li Zhang, and Christophe Diot. "Long-term forecasting of internet backbone traffic." In: *IEEE transactions on neural networks* 16.5 (Sept. 2005), pp. 1110–24. ISSN: 1045-9227. DOI: 10.1109/TNN.2005.853437.

[32] Bruno Quoitin and Cristel Pelsser. "A performance evaluation of BGP-based traffic engineering." In: *International Journal of Network Management* (2005), pp. 1–20. URL: http://onlinelibrary.wiley.com/doi/10.1002/nem.559/abstract.

[33] Chotirat Ann Ratanamahatana and Eamonn Keogh. "Three Myths about Dynamic Time Warping Data Mining." In: *SIAM International Conference on Data Mining* (2005), p. 5. DOI: http://dx.doi.org/10.1137/1.9781611972757.50.

[34] Marcelo Yannuzzi, Xavier Masip-bruin, and Olivier Bonaventure. "Open Issues in Interdomain Routing: A Survey." In: *IEEE Network* 19.6 (2005), pp. 49–56.

[35] Brice Augustin, Xavier Cuvellier, Benjamin Orgogozo, Fabien Viger, Timur Friedman, Matthieu Latapy, Clémence Magnien, and Renata Teixeira. "Avoiding traceroute anomalies with Paris traceroute." In: *IMC* (2006).

[36] P. Cortez, M. Rio, M. Rocha, and P. Sousa. "Internet Traffic Forecasting using Neural Networks." In: *IJCNN '06*. IEEE, 2006, pp. 2635–2642. ISBN: 0-7803-9490-9. DOI: 10.1109/IJCNN.2006.247142.

[37] Earl Lawrence, George Michailidis, Vijayan N. Nair, and Bowei Xi. "Network Tomography: A Review and Recent Developments." In: *Frontiers in Statistics* (2006), pp. 345–366. DOI: 10.1142/9781860948886_0016. URL: http://www.worldscientific.com/doi/abs/10.1142/9781860948886{_}0016.

[38] Wolfgang Mühlbauer, Anja Feldmann, Olaf Maennel, Matthew Roughan, and Steve Uhlig. "Building an AS-topology model that captures route diversity." In: *ACM SIGCOMM Computer Communication Review* 36.4 (2006), p. 195. ISSN: 01464833. DOI: 10.1145/1151659.1159937.

[39] Jorg Jörg Wallerich and Anja Feldmann. "Capturing the variability of internet flows across time." In: *INFOCOM '06* (2006), pp. 1–6. ISSN: 0743166X. DOI: 10.1109/INFOCOM.2006.37.

[40] Wen Xu and Jennifer Rexford. "MIRO: multi-path interdomain routing." In: *Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications - SIGCOMM '06*. Vol. 36. 4. New York, New York, USA: ACM Press, 2006, p. 171. ISBN: 1595933085. DOI: 10.1145/1159913.1159934. URL: http://portal.acm.org/citation.cfm?doid=1159913.1159934.

[41] Ryan Prescott Adams and David J C MacKay. "Bayesian online changepoint detection." In: *University of Cambridge Technical Report, Cambridge, UK* (2007).

[42] Benoit Donnet and Timur Friedman. "INTERNET TOPOLOGY DISCOVERY: A SURVEY." In: *IEEE Communications Surveys & Tutorials* 9.4 (2007), pp. 2–15. ISSN: 1553-877X. DOI: 10.1109/COMST.2007.4444750.

[43] Luigi Iannone and Olivier Bonaventure. "On the cost of caching locator/ID mappings." In: *CoNEXT '07* (2007), p. 1. DOI: 10.1145/1364654.1364663.

[44] Himabindu Pucha, Ying Zhang, Z. Morley Mao, and Y. Charlie Hu. "Understanding network delay changes caused by routing events." In: *SIGMETRICS* (2007). DOI: 10.1145/1269899.1254891.

[45] Bruno Quoitin, Luigi Iannone, Cédric De Launois, and Olivier Bonaventure. "Evaluating the benefits of the locator/identifier separation." In: *Proceedings of 2nd ACM/IEEE international workshop on Mobility in the evolving internet architecture*. ACM. 2007, p. 5.

[46] Jaxk Reeves, Jien Chen, Xiaolan L Wang, Robert Lund, and Qi Qi Lu. "A review and comparison of changepoint detection techniques for climate data." In: *Journal of Applied Meteorology and Climatology* 46.6 (2007), pp. 900–915.

[47] Xiaowei Yang, David Clark, and Arthur W. Berger. "NIRA: A New Inter-Domain Routing Architecture." In: *IEEE/ACM Transactions on Networking* 15.4 (2007), pp. 775–788. ISSN: 1063-6692. DOI: 10.1109/TNET.2007.893888. URL: http://ieeexplore.ieee.org/document/4265613/.

[48] Nancy R Zhang and David O Siegmund. "A modified Bayes information criterion with applications to the analysis of comparative genomic hybridization data." In: *Biometrics* 63.1 (2007), pp. 22–32.

[49] A. Akella, B. Maggs, S. Seshan, and A. Shaikh. "On the Performance Benefits of Multihoming Route Control." In: *IEEE/ACM Transactions on Networking* 16.1 (Feb. 2008), pp. 91–104. ISSN: 1063-6692. DOI: 10.1109/TNET.2007.899068.

[50] Benoit Donnet and Olivier Bonaventure. "On BGP communities." In: *ACM SIGCOMM Computer Communication Review* 38.2 (Mar. 2008), p. 55. ISSN: 01464833. DOI: 10.1145/1355734.1355743. URL: http://portal.acm.org/citation.cfm?doid=1355734.1355743.

[51] Xiaomei Liu and Li Xiao. "Inbound Traffic Load Balancing in BGP Multi-homed Stub Networks." In: *2008 The 28th International Conference on Distributed Computing Systems*. IEEE, 2008, pp. 369–376. DOI: 10.1109/ICDCS.2008.112. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4595905.

[52] Ning Wang, Kin Ho, George Pavlou, and Michael Howarth. "An overview of routing optimization for internet traffic engineering." In: *IEEE Communications Surveys & Tutorials* 10.1 (2008), pp. 36–56. ISSN: 1553-877X. DOI: 10.1109/COMST.2008.4483669. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4483669.

[53] Qingzhao Yu, Richard Scribner, Brad Carlin, Katherine Theall, Neal Simonsen, Bonnie Ghosh-Dastidar, Deborah Cohen, and Karen Mason. "Multilevel spatio-temporal dual changepoint models for relating alcohol outlet destruction and changes in neighbourhood rates of assaultive violence." In: *Geospatial health* 2.2 (2008), p. 161.

[54] Hitesh Ballani, Paul Francis, T. Cao, and Jia Wang. "Making routers last longer with ViAggre." In: *NSDI '09* (2009), pp. 453–466.

[55] T Giorgino. "Computing and Visualizing Dynamic Time Warping." In: *Journal Of Statistical Software* 31.7 (2009), pp. 1–24.

[56] P. Brighten Godfrey, Igor Ganichev, Scott Shenker, and Ion Stoica. "Pathlet Routing." In: *SIGCOMM Comput. Commun. Rev.* 39.4 (Aug. 2009), pp. 111–122. ISSN: 0146-4833. DOI: 10.1145/1594977.1592583. URL: http://doi.acm.org/10.1145/1594977.1592583.

[57] Mehmet H Gunes and Kamil Sarac. "Resolving IP Aliases in Building Traceroute-Based Internet Maps." In: *IEEE/ACM Transactions on Networking* 17.6 (2009), pp. 1738–1751.

[58] Changhoon Kim, Matthew Caesar, Alexandre Gerber, and Jennifer Rexford. "Revisiting route caching: The world should be flat." In: *Lecture Notes in Computer Science* 5448 (2009), pp. 3–12. ISSN: 03029743. DOI: 10.1007/978-3-642-00975-4_1.

[59] Gordon Fyodor Lyon. *Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning*. USA: Insecure, 2009. ISBN: 0979958717, 9780979958717.

[60] Ryan Turner, Yunus Saatci, and Carl Edward Rasmussen. "Adaptive Sequential Bayesian Change Point Detection." In: *In Practice* (2009), pp. 1–4. arXiv: arXiv:0710.3742v1. URL: http://eprints.pascal-network.org/archive/00006628/.

[61] Prabhu Babu and Petre Stoica. "Spectral analysis of nonuniformly sampled data – a review." In: *Digital Signal Processing* 20.2 (2010), pp. 359–378. ISSN: 10512004. DOI: 10.1016/j.dsp.2009.06.019. URL: http://dx.doi.org/10.1016/j.dsp.2009.06.019.

[62] Amanda N. Baraldi and Craig K. Enders. "An introduction to modern missing data analyses." In: *Journal of School Psychology* 48.1 (2010), pp. 5–37. ISSN: 00224405. DOI: 10.1016/j.jsp.2009.10.001. URL: http://dx.doi.org/10.1016/j.jsp.2009.10.001.

[63] Igor Ganichev, Bin Dai, P. Brighten Godfrey, and Scott Shenker. "YAMR: yet another multipath routing protocol." In: *ACM SIGCOMM Computer Communication Review* 40.5 (2010), p. 13. ISSN: 01464833. DOI: 10.1145/1880153.1880156. URL: http://portal.acm.org/citation.cfm?doid=1880153.1880156.

[64] Ken Keys. "Internet-Scale IP Alias Resolution Techniques." In: *SIGCOMM CCR* 40.1 (2010), pp. 50–55.

[65] Craig Labovitz, Scott Iekel-Johnson, Danny McPherson, Jon Oberheide, and Farnam Jahanian. "Internet inter-domain traffic." In: *SIGCOMM Comput. Commun. Rev.* 40.4 (2010), pp. –. ISSN: 1450302017. DOI: 10.1145/1851275.1851194. URL: http://dl.acm.org/citation.cfm?id=1851194http://dl.acm.org/citation.cfm?id=2043164.1851194.

[66] Yaron Schwartz, Yuval Shavitt, and Udi Weinsberg. "A measurement study of the origins of end-to-end delay variations." In: *PAM* (2010).

[67] Y Zhang, Ricardo Oliveira, H Zhang, and Lixia Zhang. "Quantifying the pitfalls of traceroute in AS connectivity inference." In: *PAM* (2010).

[68] Brice Augustin, Timur Friedman, and Renata Teixeira. "Measuring multipath routing in the internet." In: *IEEE/ACM Transactions on Networking* 19.3 (2011), pp. 830–840. ISSN: 10636692. DOI: 10.1109/TNET.2010.2096232.

[69] F S Bao, X Liu, and C Zhang. "PyEEG : An Open Source Python Module for EEG / MEG Feature Extraction." In: *Computational Intelligence and Neuroscience* 406391 (2011), p. 7. ISSN: 1687-5273. DOI: 10.1155/2011/406391.

[70] I.A. Eckley, P. Fearnhead, and R. Killick. "Analysis of Changepoint Models." In: *Bayesian Time Series Models* January (2011), pp. 205–224. DOI: 10.1017/CBO9780511984679.011.

[71] Antonio Molina-Picó, David Cuesta-Frau, Mateo Aboy, Cristina Crespo, Pau Miró-Martínez, and Sandra Oltra-Crespo. "Comparative study of approximate entropy and sample entropy robustness to spikes." In: *Artificial Intelligence in Medicine* 53.2 (2011), pp. 97–106. ISSN: 09333657. DOI: 10.1016/j.artmed.2011.06.007.

[72] Ingmar Poese, Steve Uhlig, Mohamed Ali Kaafar, Benoit Donnet, and Bamba Gueye. "IP geolocation databases." In: *ACM SIGCOMM Computer Communication Review* 41.2 (2011), p. 53. ISSN: 01464833. DOI: 10.1145/1971162.1971171.

[73] Matthew Roughan, Walter Willinger, Olaf Maennel, Debbie Perouli, and Randy Bush. "10 Lessons from 10 Years of Measuring and Modeling the Internet's Autonomous Systems." In: *EEE Journal on Selected Areas in Communications* 29.9 (Oct. 2011), pp. 1810–1821. ISSN: 0733-8716. DOI: 10.1109/JSAC.2011.111006.

[74] Damien Saucez. "Mechanisms for interdomain Traffic Engineering with LISP." PhD thesis. PhD thesis, Université catholique de Louvain, 2011.

[75] Dahai Xu, Mung Chiang, and Jennifer Rexford. "Link-state routing with hop-by-hop forwarding can achieve optimal traffic engineering." In: *IEEE/ACM Transactions on Networking* 19.6 (2011), pp. 1717–1730. ISSN: 10636692. DOI: 10.1109/TNET.2011.2134866.

[76] Fran??ois Caron, Arnaud Doucet, and Raphael Gottardo. "Online changepoint detection and parameter estimation with application to genomic data." In: *Statistics and Computing* 22.2 (2012), pp. 579–595. ISSN: 09603174. DOI: 10.1007/s11222-011-9248-x.

[77] R. Killick, P. Fearnhead, and I. a. Eckley. "Optimal detection of changepoints with a linear computational cost." In: *Journal of the American Statistical Association* 107.500 (2012), pp. 1590–1598. ISSN: 0162-1459. DOI: 10.1080/01621459.2012.737745. arXiv: 1101.1438.

[78] Vasileios Kotronis, Xenofontas Dimitropoulos, and Bernhard Ager. "Outsourcing the routing control logic: better internet routing based on SDN principles." In: *Proceedings of the 11th ACM Workshop on Hot Topics in Networks* (2012), pp. 55–60. DOI: 10.1145/2390231.2390241. URL: http://dl.acm.org/citation.cfm?id=2390241.

[79] Nadi Sarrar, Steve Uhlig, Anja Feldmann, Rob Sherwood, and Xin Huang. "Leveraging Zipf's law for traffic offloading." In: *ACM SIGCOMM CCR* 42.1 (Jan. 2012), p. 16. ISSN: 01464833. DOI: 10.1145/2096149.2096152.

[80] Wei Zhang, Jun Bi, Jianping Wu, and Baobao Zhang. "Catching popular prefixes at AS border routers with a prediction based method." In: *Computer Networks* 56.4 (Mar. 2012), pp. 1486–1502. ISSN: 13891286. DOI: 10.1016/j.comnet.2012.01.003.

[81] Yaping Zhu, Benjamin Helsley, Jennifer Rexford, Aspi Siganporia, and Sridhar Srinivasan. "LatLong: Diagnosing Wide-Area Latency Changes for CDNs." In: *IEEE Transactions on Network and Service Management* 9.3 (2012), pp. 333–345. DOI: 10.1109/TNSM.2012.070412.110180.

[82] Ehsan Elhamifar and René Vidal. "Sparse Subspace Clustering: Algorithm, Theory, and Applications." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 99 (2013), pp. 55–63. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2013.57. arXiv: arXiv:1203.1005v3. URL: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp={\&}arnumber=6482137{\&}contentType=Early+Access+Articles{\&}matchBoolean=true{\&}rowsPerPage=30{\&}searchField=Search{_}All{\&}queryText=({\%}22Sparse+Subspace+Clustering:+Algorithm,+Theory,+and+Applications{\%}22){\%}5Cnpapers2://publi.

[83] Sushant Jain et al. "B4: Experience with a Globally-Deployed Software DefinedWA." In: *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM - SIGCOMM '13* (2013), p. 3. ISSN: 0146-4833. DOI: 10.1145/2486001.2486019. URL: http://dl.acm.org/citation.cfm?id=2486019{\%}5Cnhttp://dl.acm.org/citation.cfm?doid=2486001.2486019{\%}5Cnhttp://dl.acm.org/citation.cfm?doid=2486001.2486019.

[84] Xavier Misseri, Ivan Gojmerac, and Jean-Louis Rougier. "IDRD: Enabling inter-domain route diversity." In: *2013 IEEE International Conference on Communications (ICC)*. IEEE, 2013, pp. 3536–3541. ISBN: 978-1-4673-3122-7. DOI: 10.1109/ICC.2013.6655099. URL: http://ieeexplore.ieee.org/document/6655099/.

[85] Tatsuya Otoshi, Yuichi Ohsita, Masayuki Murata, Yousuke Takahashi, Keisuke Ishibashi, and Kohei Shiomoto. "Traffic Prediction for Dynamic Traffic Engineering Considering Traffic Variation." In: *GLOBECOM '13* (Dec. 2013), pp. 1592–1598. DOI: 10.1109/GLOCOM.2013.6831297.

[86] Cristel Pelsser, Luca Cittadini, Stefano Vissicchio, and Randy Bush. "From Paris to Tokyo : On the Suitability of ping to Measure Latency." In: *IMC* (2013).

[87] Yuval Shavitt and Noa Zilberman. "Internet pop level maps." In: *DataTraffic Monitoring and Analysis*. Springer-Verlag. 2013, pp. 82–103.

[88] Xiaoyue Wang, Abdullah Mueen, Hui Ding, Goce Trajcevski, Peter Scheuermann, and Eamonn Keogh. "Experimental comparison of representation methods and distance measures for time series data." In: *DMKD 2013* 26.2 (2013), pp. 275–309. ISSN: 13845810. DOI: 10.1007/s10618-012-0250-5. arXiv: 1012.2789.

[89] Arpit Gupta et al. "SDX : A Software Defined Internet Exchange." In: *Proceedings of the 2014 ACM conference on SIGCOMM* (2014), pp. 579–580. ISSN: 01464833. DOI: 10.1145/2619239.2626300.

[90] Shen Shyang Ho and Harry Wechsler. "Online Change Detection." In: *Conformal Prediction for Reliable Machine Learning: Theory, Adaptations and Applications* (2014), pp. 99–114. DOI: 10.1016/B978-0-12-398537-8.00005-5.

[91] Rebecca Killick and Ia Eckley. "changepoint: An R Package for changepoint analysis." In: *Journal of Statistical Software* 58.3 (2014), pp. 1–19. ISSN: 1548-7660. DOI: 10.1359/JBMR.0301229. arXiv: arXiv:1501.0228.

[92] Matthew Luckie and kc Claffy. "A Second Look at Detecting Third-Party Addresses in Traceroute Traces with the IP Timestamp Option." In: *PAM* (2014). DOI: 10.1007/978-3-319-04918-2_5.

[93] Matthew Luckie, Amogh Dhamdhere, David Clark, Bradley Huffaker, and Kc Claffy. "Challenges in Inferring Internet Interdomain Congestion." In: *IMC*. 2014.

[94] Massimo Rimondini, Claudio Squarcella, and Giuseppe Di Battista. "Towards an Automated Investigation of the Impact of BGP Routing Changes on Network Delay Variations." In: *PAM* (2014). arXiv: arXiv:1309.0632v1.

[95] Paul Sharkey and Rebecca Killick. "Nonparametric Methods for Online Changepoint Detection." In: (2014).

[96] Ming Zhu, Jun Li, Ying Liu, Dan Li, and Jianping Wu. "TED: Inter-domain traffic engineering via deflection." In: *IEEE International Workshop on Quality of Service, IWQoS*. 61161140454. 2014, pp. 117–122. ISBN: 9781479948529. DOI: 10.1109/IWQoS.2014.6914309.

[97] Saeed Aghabozorgi, Ali Seyed Shirkhorshidi, and Teh Ying Wah. "Time-series clustering – A decade review." In: *Information Systems* 53 (2015), pp. 16–38. ISSN: 03064379. DOI: 10.1016/j.is.2015.04.007. URL: http://linkinghub.elsevier.com/retrieve/pii/S0306437915000733.

[98] Kelsey Anderson. "A Novel Approach to Bayesian Online Changepoint Detection." In: 2010 (2015), pp. 1–15.

[99] Vaibhav Bajpai and Steffie Jacob Eravuchira. "Lessons Learned from using the RIPE Atlas Platform for Measurement Research." In: *SIGCOMM CCR* 45.3 (2015), pp. 35–42.

[100] Balakrishnan Chandrasekaran, Arthur Berger, and Matthew Luckie. "A Server-to-Server View of the Internet." In: *CoNEXT* (2015).

[101] Yanping Chen, Eamonn Keogh, Bing Hu, Nurjahan Begum, Anthony Bagnall, Abdullah Mueen, and Gustavo Batista. *The UCR Time Series Classification Archive*. 2015.

[102] Yi-ching Chiu, Brandon Schlinker, Abhishek Balaji Radhakrishnan, Ethan Katz-bassett, and Ramesh Govindan. "Are We One Hop Away from a Better Internet ?" In: *ACM Internet Measurement Conference* (2015), pp. 523–529. DOI: 10.1145/2815675.2815719.

[103] Thomas Holterbach, Cristel Pelsser, Randy Bush, and Laurent Vanbever. "Quantifying Interference between Measurements on the RIPE Atlas Platform." In: *IMC* (2015). DOI: 10.1145/2815675.2815710.

[104] Yaoqing Liu, Vince Lehman, and Lan Wang. "Efficient FIB caching using minimal non-overlapping prefixes." In: *Computer Networks* 83 (2015), pp. 85–99. ISSN: 13891286. DOI: 10.1016/j.comnet.2015.03.003.

[105] Ramakrishna Padmanabhan, Patrick Owen, Aaron Schulman, and Neil Spring. "Timeouts : Beware Surprisingly High Delay Categories and Subject Descriptors." In: *IMC*. 2015, pp. 303–316. ISBN: 9781450338486.

[106] Wenqin Shao, Francois Devienne, Luigi Iannone, and Jean-Louis Rougier. "On the use of BGP communities for fine-grained inbound traffic engineering." In: (2015). arXiv: 1511.08336. URL: http://arxiv.org/abs/1511.08336.

[107] Peng Sun, Laurent Vanbever, and Jennifer Rexford. "Scalable programmable inbound traffic engineering." In: *Proceedings of the 1st ACM SIGCOMM Symposium on Software Defined Networking Research - SOSR '15* (2015), pp. 1–7. DOI: 10.1145/2774993.2775063. URL: http://dl.acm.org/citation.cfm?doid=2774993.2775063.

[108] Liudmila Ulanova, Nurjahan Begum, and Eamonn Keogh. "Scalable Clustering of Time Series with U-Shapelets." In: *SIAM International Conference on Data Mining (SDM 2015)* (2015).

[109] Neal Cardwell, Yuchung Cheng, C Stephen Gunn, Soheil Hassas Yeganeh, and Van Jacobson. "BBR: Congestion-Based Congestion Control." In: *ACM Queue* 14.5 (2016), 50:20–50:53. ISSN: 1542-7730. DOI: 10.1145/3012426.3022184. URL: http://doi.acm.org/10.1145/3012426.3022184.

[110] Romain Fontugne, Emile Aben, Cristel Pelsser, and Randy Bush. "Pinpointing Delay and Forwarding Anomalies Using Large-Scale Traceroute Measurements." In: (2016). arXiv: 1605.04784. URL: http://arxiv.org/abs/1605.04784.

[111] Antoine Fressancourt. "Improved resiliency for inter-datacenter network connections." PhD thesis. Télécom Paristech, 2016.

[112] Kaylea Haynes, Paul Fearnhead, and Idris A. Eckley. "A computationally efficient nonparametric approach for changepoint detection." In: *Statistics and Computing* (2016). ISSN: 0960-3174. DOI: 10.1007/s11222-016-9687-5. arXiv: 1602.01254.

[113] Youndo Lee and Neil Spring. "Identifying and Aggregating Homogeneous IPv4 / 24 Blocks with Hobbit." In: (2016), pp. 151–165. DOI: 10.1145/2987443.2987448.

[114] Matthew Luckie, Amogh Dhamdhere, Bradley Huffaker, David Clark, and Kc Claffy. "Bdrmap: Inference of Borders Between IP Networks." In: *Proceedings of the 2016 ACM on Internet Measurement Conference - IMC '16* (2016), pp. 381–396. DOI: 10.1145/2987443.2987467. URL: http://dl.acm.org/citation.cfm?doid=2987443.2987467.

[115] George Nomikos and Xenofontas Dimitropoulos. "TraIXroute: Detecting IXps in traceroute paths." In: *PAM* (2016). ISSN: 16113349. DOI: 10.1007/978-3-319-30505-9_26. arXiv: 1611.03895.

[116] Wenqin Shao, Jean-Louis Rougier, Francois Devienne, and Mateusz Viste. "Improve Round-Trip Time Measurement Quality via Clustering in Inter-Domain Traffic Engineering." In: *AnNet* (2016).

[117] *QUIC Loss Detection and Congestion Control draft-ietf-quic-recovery-04*. 2017. URL: https://tools.ietf.org/html/draft-ietf-quic-recovery-04.

[118] Brandon Schlinker, Hyojeong Kim, Timothy Cui, Ethan Katz-Bassett, Harsha V. Madhyastha, Italo Cunha, James Quinn, Saif Hasan, Petr Lapukhov, and Hongyi Zeng. "Engineering Egress with Edge Fabric: Steering Oceans of Content to the World." In: *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*. SIGCOMM '17. Los Angeles, CA, USA: ACM, 2017, pp. 418–431. ISBN: 978-1-4503-4653-5. DOI: 10.1145/3098822.3098853. URL: http://doi.acm.org/10.1145/3098822.3098853.

[119] Kok-Kiong Yap et al. "Taking the Edge off with Espresso: Scale, Reliability and Programmability for Global Internet Peering." In: *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*. SIGCOMM '17. Los Angeles, CA, USA: ACM, 2017, pp. 432–445. ISBN: 978-1-4503-4653-5. DOI: 10.1145/

3098822.3098854. URL: http://doi.acm.org/10.1145/3098822.3098854.

[120] *95th Percentile Internet Billing Method*. URL: http://drpeering.net/white-papers/Ecosystems/95th-percentile-measurement-Internet-Transit.html.

[121] ACM. *Artifact Review and Badging*. URL: https://www.acm.org/publications/policies/artifact-review-badging.

[122] Emile Aben. *Measuring Countries and IXPs with RIPE Atlas*. URL: https://labs.ripe.net/Members/emileaben/measuring-ixps-with-ripe-atlas.

[123] *Advertisement of Multiple Paths in BGP*. URL: https://tools.ietf.org/html/rfc7911.

[124] *Border6 Non-Stop Internet*. URL: http://www.border6.com.

[125] *Built-in Measurements*. URL: https://atlas.ripe.net/docs/built-in/.

[126] Massimo Candela. *LatencyMon*. URL: https://labs.ripe.net/Members/massimo_candela/new-ripe-atlas-tool-latencymon.

[127] *Common IP Filtering Techniques-APNIC*. URL: https://www.apnic.net/manage-ip/apnic-services/registration-services/resource-quality-assurance/filtering/.

[128] *DIMES*. URL: http://www.netdimes.org.

[129] Philip Homburg. *A visual impression of probe lifetime*. URL: https://labs.ripe.net/Members/philip_homburg/a-visual-impression-of-probe-lifetimes.

[130] Philip Homburg. *Further analysis of RIPE Atlas version 3 probe*. URL: https://labs.ripe.net/Members/philip_homburg/further-analysis-of-ripe-atlas-version-3-probe.

[131] Philip Homburg. *Releasing RIPE Atlas Measurements Source Code*. URL: https://labs.ripe.net/Members/philip_homburg/ripe-atlas-measurements-source-code.

[132] Philip Homburg. *Troubleshooting RIPE Atlas probe USB sticks*. URL: https://labs.ripe.net/Members/philip_homburg/troubleshooting-ripe-atlas-probes-usb-sticks.

[133] *IP Transit Prices Continue Falling, Major Discrepancies Remain*. URL: https://www.telegeography.com/press/press-releases/2015/09/09/ip-transit-prices-continue-falling-major-discrepancies-remain/index.html.

[134] *Internet Exchange Point Growth by Country*. URL: https://www.pch.net/ixp/summary_growth_by_country.

[135] Shane Kerr, Daniel Quinn, and Désirée Milosevic. *Halo: Network Outages Dashboard*. URL: https://github.com/RIPE-Atlas-Community/ripe-atlas-halo.

[136] Daniel Kopp. *Investigation of Dependencies between IXPs*. URL: https://youtu.be/9CGQfm_IzvE.

[137] *Locator/ID Separation Protocol*. URL: https://datatracker.ietf.org/wg/lisp/.

[138] Barry O'Donovan, Drew Taylor, and Jacob Drabczyk. *Detecting Asymmetric Routing over IXPs*. URL: https://github.com/inex/ixp-as.

[139] *PlanetLab*. URL: https://www.planet-lab.org.

[140] *RFC4271: A Border Gateway Protocol 4*. URL: https://www.ietf.org/rfc/rfc4271.txt.

[141] *RIPE Atlas API Reference*. URL: https://atlas.ripe.net/docs/api/v2/reference/.

[142] *RIPE Atlas Built-in measurements*. URL: https://atlas.ripe.net/docs/built-in/.

[143] *RIPE Forum [atlas] Actual measurement interval much larger than planned*. URL: https://www.ripe.net/participate/mail/forum/ripe-atlas.

[144] *RIPE Forum [atlas] New on RIPE Labs: Another Look at RIPE Atlas Probe Lifetimes*. URL: https://www.ripe.net/participate/mail/forum/ripe-atlas.

[145] *RIPE RIS*. URL: https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris.

[146] *RouteViews*. URL: http://archive.routeviews.org/bgpdata/.

[147] *Source Code of RIPE Atlas probe firmware*. URL: https://atlas.ripe.net/resources/source-code/.

[148] *Why do Internet Transit Prices Drop?* URL: http://drpeering.net/FAQ/Why-do-Internet-Transit-Prices-Drop.php.

[149] Rene Wilhelm. *Another look at RIPE Atlas probe lifetime*. URL: https://labs.ripe.net/Members/wilhelm/another-look-at-ripe-atlas-probe-lifetimes.

[150] *perfSONAR*. URL: https://www.perfsonar.net.