

Detection and localization of link-level network anomalies using end-to-end path monitoring

Emna Salhi

▶ **To cite this version:**

Emna Salhi. Detection and localization of link-level network anomalies using end-to-end path monitoring. Other [cs.OH]. Université de Rennes, 2013. English. NNT : 2013REN1S021 . tel-00860397

HAL Id: tel-00860397

https://theses.hal.science/tel-00860397

Submitted on 10 Sep 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

ANNÉE 2013





THÈSE / UNIVERSITÉ DE RENNES 1
sous le sceau de l'Université Européenne de Bretagne

pour le grade de

DOCTEUR DE L'UNIVERSITÉ DE RENNES 1

Mention : Informatique

Ecole doctorale MATISSE

présentée par

Emna Salhi

Préparée à l'unité de recherche IRISA
Institut de Recherche en Informatique et Systèmes Aléatoires
Composante Universitaire: ISTIC

Intitulé de la thèse: Detection and Localization of Link-Level Network Anomalies Using End-to-End Path Monitoring

Thèse soutenue à Rennes le 13.02.2013

devant le jury composé de :

Philippe Owezarski
Chargé de recherche – HDR, CNRS / *rapporteur*

Anura P. Jayasumana
Professeur, université de Colorado / *rapporteur*

Sandrine Vaton
Maître de conférence – HDR, Télécom Bretagne / *examinateur*

Miklos Molnar
Professeur – HDR, université Montpellier 2 / *examinateur*

Bernard Cousin
Professeur – HDR, université de Rennes 1/ *directeur de thèse*

Samer Lahoud
Maître de conférence, université de Rennes 1/ *co-directeur de thèse*

Abstract

This thesis investigates the problem of link-level anomaly detection and localization using end-to-end path monitoring. The aim is to come up with cost-efficient, accurate and fast schemes for link-level network anomaly detection and localization. The anomaly detection aims at detecting the occurrence of anomalies in the network (*e.g.*, excessive delays, high loss rate, infrastructure failures, etc.) and identifying a set of links suspect to be the source of the anomaly. The anomaly localization is triggered upon detecting an anomaly. It aims at reducing the set of suspect links identified by the detection process to the anomalous link(s).

It has been established that, for detecting all potential link-level anomalies, a set of paths that cover all links of the network [1] must be monitored, whereas for localizing all potential link-level anomalies, a set of paths that can distinguish between all links of the network pairwise [2] must be monitored. Either end-node of each path monitored must be equipped with a monitoring device.

Most existing link-level anomaly detection and localization schemes are two-step. The first step selects a minimal set of monitor locations that can detect/localize all potential link-level anomalies. The second step selects a minimal set of monitoring paths between the selected monitor locations such that all links of the network are covered/distinguishable pairwise. However, such step-wise schemes do not consider the interplay between the conflicting optimization objectives of the two steps, which results in sub-optimal consumption of the network resources and biased monitoring measurements. One of the objectives of this thesis is to evaluate and reduce this interplay. To this end, one-step anomaly detection and localization schemes that select monitor locations and paths that are to be monitored

1. A link is said to be covered if it is traversed by at least one monitoring path
2. Two links are said to be distinguishable if we are able to decide which one is anomalous when an anomaly occurs on one of them

jointly, thereby achieving a good trade-off between the number and locations of monitoring devices and the quality of monitoring paths, are proposed

Furthermore, we demonstrate that the already established condition for link-level anomaly localization is sufficient but not necessary. A necessary and sufficient condition that minimizes the localization cost drastically is established.

The problems are formulated as integer linear programs and are demonstrated to be $\mathcal{NP}$-Hard. Scalable and near-optimal heuristic algorithms for anomaly detection and anomaly localization are proposed. The effectiveness and the correctness of the proposed schemes and algorithms are verified through theoretical analysis and extensive simulations.

Key Words : Network monitoring, anomaly detection, anomaly localization, end-to-end path monitoring, link-level network anomalies

Résumé en français

Cette thèse étudie le problème de la détection et de localisation des anomalies au niveau des liens en utilisant un monitorage des chemins de bout-en-bout. L'objectif est de trouver des techniques de détection et de localisation des anomalies au niveau des liens qui soient à faible coût, précises et rapides. La détection d'anomalies vise à détecter l'apparition d'anomalies dans le réseau (par exemple des retards excessifs, un taux de perte élevé, des pannes d'infrastructure, etc) et d'identifier un ensemble de liens soupçonnés d'être la source de cette anomalie. La localisation des anomalies est déclenchée en cas de détection d'une anomalie. Elle vise à réduire l'ensemble des liens suspects identifiés par le processus de détection d'anomalies au(x) lien(s) défaillant(s).

Il a été établi que pour détecter toutes les anomalies possibles au niveau des liens d'un réseau, un ensemble de chemins qui couvrent tous les liens du réseau 3 doit être monitoré, alors que pour localiser toutes les anomalies potentielles au niveau des liens d'un réseau, un ensemble de chemins qui peuvent distinguer entre tous les liens du réseau paire par paire 4 doivent être monitorés. Chaque nœud d'extrémité de chaque chemin monitoré doit être équipé d'un dispositif de monitorage.

La plupart des techniques de détection et de localisation des anomalies au niveau des liens qui existent dans la littérature calculent les solutions, c-à-d l'ensemble des chemins à monitorer et les emplacements des dispositifs de monitorage, en deux étapes. La première étape sélectionne un ensemble minimal d'emplacements des dispositifs de monitorage qui permet de détecter/localiser toutes les anomalies possibles. La deuxième étape sélectionne un ensemble minimal de chemins de monitorage entre les emplacements sélectionnés de telle sorte que tous les liens du réseau soient couverts/distinguables paire par paire. Toutefois, ces techniques ignorent l'interaction entre les objectifs d'optimisation contradictoires des deux étapes, ce qui entraîne une utilisation sous-optimale des ressources du réseau et des mesures de monitorage biaisées. L'un des objectifs de cette thèse est d'évaluer et de réduire cette interaction.À cette fin, nous proposons des techniques de détection et de localisation

d'anomalies au niveau des liens qui sélectionnent les emplacements des moniteurs et les chemins qui doivent être monitorés conjointement en une seule étape, ce qui permet de réaliser un bon compromis entre le nombre et l'emplacement des moniteurs et la qualité des chemins de monitorage.

Par ailleurs, nous démontrons que la condition pré-établie pour la localisation des anomalies au niveau des liens est suffisante mais pas nécessaire. Une condition nécessaire et suffisante qui minimise le coût de localisation considérablement est établie.

Les deux problèmes sont formulés sous forme d'un programme linéaire en nombres entiers et il est démontré qu'ils sont $\mathcal{NP}$-durs. Des algorithmes heuristiques scalables et efficaces sont alors proposés. L'efficacité et l'exactitude des technique et des algorithmes proposés sont vérifées par le biais d'une analyse théorique et des simulations.

Mots Clès : Monitorage des réseaux, détection des anomalies, localisation des anomalies, monitorage des chemins de bout-en-bout.

Introduction

L'Internet a connu une transition d'un réseau de transmission des données simples servant un nombre limité d'utilisateurs à un réseau multi-service qui prend en charge diverses applications multimédias aux exigences élevées de qualité de service et servant un nombre fortement croissant d'utilisateurs. Cela est dû à l'évolution rapide des équipements du réseau de plus en plus puissants et accessibles (par exemple, supports de transmission à haute capacité, haute vitesse de commutation, équipements de stockage à grande capacité, etc.). Par conséquence, la nécessité d'outils de surveillance des réseaux efficaces qui garantissent une performance désirées pour les réseaux et fournissent des garanties de qualité de service a augmenté. Un grand nombre de techniques de surveillance et d'outils de mesure des réseaux ont été proposés dans la littérature.

Les plus simples systèmes de surveillance utilisent d'outils réseau existants tels que ping et traceroute [18][23]. Ils sont qualifiés comme simples, car ils ne nécessitent aucune modification dans le réseau. Cependant, leur application est limitée à la détection et la localisation des défaillances d'infrastructure et de l'indisponibilité des chemins [27]. Des système de monitorage qui fournissent une information plus détaillée sur la performance des réseaux ont été proposés. Ils peuvent être classés en deux catégories : des systèmes de surveillance individuelle (par exemple les systèmes de surveillance basés sur le protocol SNMP (Simple Network Management Protocol) [7], RMON [28], Netflow [8]), et des

systèmes de surveillance de bout-en-bout (par exemple [14] [24] [26] [9] [10] [11] [17] [16] [15][21] [20] [19] [29] [1] [6] [22] [2]).

Les systèmes de surveillance individuelle reposent sur l'idée d'équiper chaque équipements réseau par un agent de surveillance qui recueille des statistiques sur les performances du périphérique et de ses liens incidents en observant le trafic réseau qui le traverse. Les statistiques collectées individuellement sont alors exportées vers une entité de gestion et de surveillance du réseau chargé d'analyser les mesures. Les problèmes majeurs de ces systèmes est le coût de l'infrastructure de surveillance qui peut être très élevés quand il s'agit des réseaux de grandes tailles. En outre, l'exportation des statistiques vers l'entité de gestion et de surveillance du réseau peut générer une lourde charge sur le réseau. La surveillance de bout-en-bout est une solution intuitive à ces problèmes. Cela consiste à déduire les performances internes du réseau à partir des mesures de bout-en-bout, ce qui nécessite de déployer moins des dispositifs de surveillance (appelé moniteurs) dans le réseau et aussi réduit la surcharge de la surveillance.

Il existe une autre classification des systèmes de surveillance : les systèmes de surveillance passive et les systèmes de surveillance active. La surveillance passive déduit la performance du réseau par la surveillance du trafic réseau existant. Il existe deux approches pour effectuer ce type de surveillance passive :

- Surveillance à deux points : cette approche déploie deux moniteurs au niveau des nœuds d'entrée et de sortie de chaque flux surveillés. Les mesures de performance sont déduites en comparant les mesures effectuées au niveau des moniteurs d'entrée et de sortie. Ceci nécessite que les moniteurs soient synchronisés et que tous les paquets les traversant puissent être identifiés. Toutefois, le processus d'identification pourrait conduire à un sérieux problème de passage à l'échelle lorsque le volume de trafic traversant les moniteurs est important.
- Surveillance à un point : Cette approche nécessite un seul moniteur pour surveiller un flux. Par exemple, elle exploite les accusés de réception TCP pour déduire des mesures de performance (par exemple le taux de perte, RTT entre le moniteur et le générateur du trafic) entre le point où le moniteur est déployé et le générateur du flux TCP surveillé. Il est clair que l'application de cette approche se limite aux flux échangés au sein des connexions où il y a des messages de contrôle qui circulent en sans inverse des données.

La surveillance active déduit la performance du réseau en effectuant des mesures sur des flux de surveillance spécifiquement générés et injectés dans le réseau par les moniteurs

pour émuler les flux existants. La principale difficulté de la surveillance active est de faire en sorte que, sans provoquer des interférences avec les services du réseau, les flux injectés expérimentent les mêmes conditions que les flux du trafic réel afin d'obtenir des mesures fidéles.

Bien que les deux approches de surveillances ont leurs propres inconvénients, la surveillance active présente deux avantages importants par rapport à la surveillance passive. Le premier est qu'elle préserve la confidentialité pour les services traversant le réseau. En effet, les mesures ne sont pas fait sur des flux réels mais plutôt sur des flux d'émulation. La deuxième est qu'il est possible, en utilisant la surveillance active, d'effectuer des mesures quand il n'y a pas des flux traversant le réseau. Par exemple, un fournisseur de services peut avoir besoin de vérifier la disponibilité et les caractéristiques d'un chemin précédemment non utilisé avant qu'il n'y injecte des services, ce qui n'est pas faisable en utilisant la surveillance passive.

Le problème de surveillance de bout-en-bout, active et passive, a été largement étudié dans la littérature. En dépit de leurs divergences en termes de paramètres mesurés et la méthode d'acquisition des mesures, tous les système de surveillance proposés partagent un objectif commun important : garantir les performances souhaitée, tout en minimisant le coût de surveillance en termes de coûts d'infrastructure et surcharge. L'objectif de cette thèse est de proposer une technique de surveillance des réseaux de bout-en-bout qui permet d'atteindre cet objectif. Nous notons que le problème de surveillance d'anomalies au niveau des nœuds se réduit à un problème de surveillance d'anomalies au niveau des liens. En effet, un nœuds défaillant rend tous les liens qui l'entourent défaillants.

Les techniques de surveillance de bout-en-bout

Les techniques de surveillance de bout-en-bout peuvent être classées en deux catégories : surveillance analogique et surveillance binaire [22].

– Surveillance analogique : elle motivée par l'efficacité des communications multicast en termes d'économie en bande passante, les premières techniques de surveillance de bout-en-bout utilise des sondes d'émulation envoyées en multicast pour inférer les caractéristiques internes du réseau (par exemple, le taux de perte au niveau des liens constituent l'arbre multicast, la distribution de délai, les goulots d'étranglement de la bande passante, etc.) *e.g.*, [3] [14] [26] [24] [4] [25]. Cette technique consiste principalement à corréler les différentes copies des paquets multicast observés au

niveau des récepteurs multicast pour en déduire les performances des liens de l'arbre multicast.

En dépit de ses potentiels avantages, les techniques de surveillance basés sur la communication multicast ne peuvent pas être largement appliquées. En effet, actuellement, le multicast n'est que modestement déployé. Plusieurs travaux de recherche proposent des techniques de surveillance utilisant une communication unicast qui émulent les techniques basées sur le multicast [9], [10] [11], [17], [16], [5]. L'idée consiste à envoyer deux paquets étroitement espacés dans le temps d'un serveur à un ensemble de récepteurs dont les chemins vers le serveur partagent un ensemble des liens. Les paquets sondes issues de la même source et ayant les mêmes caractéristiques sont vraiment susceptibles de subir les mêmes performances sur les liens partagés. Cette corrélation est exploitée de la même manière que les techniques basées sur le multicast pour inférer les performances internes du réseau.

– Surveillance binaire : cette technique de surveillance a été largement largement adoptée. Elle consiste à identifier les déviations de la performance du réseau par rapport à un niveau donné de performance plutôt que d'estimer des mesures de performance exactes. Cette technique repose sur l'hypothèse que les performances au niveau des liens sont séparables, ce qui implique qu'un chemin souffre d'une mauvaise performance si et seulement si au moins un des liens qui le constituent souffre d'une mauvaise performance [15]. Ainsi, l'identification des anomalies de performance peut être fait en identifiant les chemins qui ne respectent pas les seuils de performance. Plus précisément, selon [15], il suffit de surveiller un ensemble de chemins qui couvrent tous les liens du réseau pour détecter toutes les anomalies qui pourraient affecter les liens du réseau. Des chemins additionnels doivent être surveillés afin de localiser la (les) source(s) de l'anomalie.

  De nombreux travaux de recherche ont exploité cette propriété de séparabilité de performance pour mettre au point des technique de détection et de localisation des anomalies au niveau des liens [21][20][19][29], [1][6].

Nous allons donc par la suite décrire les principales techniques utilisées lors de la phase de détection d'anomalie et celles de la phase de localisation d'anomalie.

Détection des anomalies au niveau des liens

Le but de la phase de détection d'anomalies au niveau des liens est de détecter toute dégradation des performances ou défaillances d'infrastructures qui pourraient affecter les liens du réseau. Dans cette thèse, nous considérons des anomalies séparables qui satisfont la propriété de séparabilité des performances développée dans [15]. Comme mentionné précédemment, dans le cas d'anomalies séparables, un chemin souffre d'une anomalie si et seulement si au moins un de liens le constituent souffre d'une anomalie. La conclusion triviale qui peut être tirée de cette propriété est que pour la détection de toutes les anomalies qui pourraient affecter les liens d'un réseau, il suffit de surveiller un ensemble de chemins qui couvrent tous les liens du réseau. Un lien est dit couvert s'il est traversé par au moins un chemin surveillé.

L'information fournie à la fin de la phase de détection est un ensemble de chemins affectés par l'anomalie. Tous les liens du réseau qui sont traversés par seulement des chemins affectés par l'anomalie sont suspects d'être défaillants. Cette information ne permet pas de décider quel(s) lien(s), parmi les liens suspects, est (sont) défaillant(s).

Localisation des anomalies au niveau des liens

La phase de localisation vise à identifier l'origine d'une anomalie détectée. Une condition suffisante pour localiser des anomalies au niveau des liens a été établie dans la littérature [21][6][1]. Elle consiste à déployer un ensemble de moniteurs permettant de *distinguer* entre chaque paire de sous-ensembles de liens du réseau. Ceci implique que, pour chaque paire de sous-ensembles des liens, il existe un chemin entre les moniteurs déployés dont l'intersection avec exactement un de deux sous-ensembles des liens n'est pas vide. Ainsi, si la surveillance du chemin signale une anomalies, alors le sous-ensemble dont l'intersection avec le chemin est vide est défaillant, sinon, l'autre sous-ensemble est défaillant.

En réalité, les anomalies qui affectent plusieurs liens sont des événements rares. Par conséquent, de nombreux travaux de recherche limitent le nombre d'anomalies simultanées dans une tentative de minimiser le coût de localisation. [1] affirme que les anomalies impliquant plus que trois liens sont très peu susceptibles de se produire.

Description de l'infrastructure de détection et de localisation des anomalies au niveau des liens

L'infrastructure de détection (respectivement localisation) d'anomalies est constituée d'un ensemble de moniteurs placés sur un sous-ensemble des nœuds de réseau tel qu'il existe un ensemble des chemins entre les noeuds équipés de moniteurs qui couvrent tous les liens du réseau (respectivement distinguent entre chaque paire de sous-ensembles de liens).

Généralement, l'infrastructure de détection est active en permanence, tandis que l'infrastructure de localisation est activé uniquement suite à la détection d'une anomalie. Ceci est justifié par le fait que les anomalies sont des évènements rares. En outre, en fonction de la topologie du réseau, l'exécution du processus de localisation d'une façon continu peut entraîner une charge lourde sur le réseau sous-jacent.

Par ailleurs, les mesures collectées par les moniteurs sont exportées vers une entité de gestion et de surveillance du réseau. Cette entité analyse et mets en corrélation les mesures collectées individuellement par les moniteurs. Quand une anomalie est détectée, elle déclenche le processus de localisation en activant certains moniteurs permettant de distinguer entre les liens suspects deux à deux.

Les coûts de détection et de localisation des anomalies au niveau des liens

Les coûts de détection et localisation comprennent les coûts suivants :

- Coût d'infrastructure : c'est le coût d'acquisition, de déploiement et de maintenance des équipements et des logiciels de surveillance .
- Coût de la communication : c'est le coût des communications entre l'entité de gestion et de surveillance du réseau et les moniteurs qui sont déployés dans le réseau. L'entité de gestion et de surveillance du réseau collecte les mesures effectués par les moniteurs qui sont activés pour la détection. Lorsqu'une anomalie est détectée, elle déclenche le phase de localisation en activant le processus de localisation sur un sous-ensemble des moniteurs déployés qui sont capables de distinguer entre l'ensemble des liens suspects deux à deux. Il est très important de choisir les endroits où les moniteurs sont déployés judicieusement, afin de réduire la surcharge et les délais de communication.

– Coût des sondes : ce coût exprime la charge de la surveillance des flux de surveillance sur le réseau. Les mesures redondantes et les mesures qui ne fournissent aucune information sur l'état des liens du réseau sont fortement indésirables. En effet, de telles mesures augmentent les délais et la surcharge de détection/localisation.

Sélection des emplacements des moniteurs et des chemins de surveillance pour la détection et la localisation des anomalies au niveau des liens

L'un des problèmes qui ont reçu un grand intérêt au sein de la communauté de la recherche sur la surveillance des réseaux est formulé comme suit : *Comment choisir les emplacements des moniteurs et les chemins de surveillance permettant de détecter/localiser toutes les anomalies qui pourraient se produire, tout en minimisant les coûts et les délais*[6] [1] [20] [21] [29].

Presque tous les systèmes de surveillance de bout-en-bout au niveau des liens existants appliquent une approche en deux étapes pour la sélection des emplacements des moniteurs et des chemins de surveillance. La première étape sélectionne un ensemble minimal d'emplacements des moniteurs permettant de détecter/localiser toutes les anomalies possibles. La deuxième étape sélectionne le plus petit ensemble de chemins entre les emplacements sélectionnés à la première étape qui permettent de detecter/localiser toutes les anomalies possibles [6] [1].

[21] applique une approche en deux étapes inverse. La première étape sélectionne un ensemble minimal de chemins de surveillance qui permettent de détecter/localiser toutes les anomalies possibles, tandis que la seconde étape sélectionne un ensemble minimal d'emplacements de moniteurs qui permettent de surveiller les chemins sélectionnés à la première étape.

[29] propose une techniques de détection multi-round. Cette technique prend en compte la capacité des liens du réseau de supporter les flux de surveillance et la capacité des moniteurs de gérer les flux de surveillance lors de la sélection des emplacements de moniteurs et des chemin de surveillance. Le résultat est un ensemble minimal d'emplacements de moniteurs et des chemins de surveillance qui couvrent les liens du réseau en un certain nombre de rounds.

Comme mentionné précédemment, il a été démontré que la surveillance d'un ensemble de chemins qui couvrent tous les liens du réseau est une condition nécessaire et suffisante pour la détection de toute anomalie qui pourraient se produire dans le réseau. Toutefois, l'ensemble des chemins qui doivent être surveillés pour déterminer la source d'une anomalie détectée a été défini de deux façons. La première consiste à surveiller un ensemble de chemins pré-calculé qui permet de distinguer entre tous les liens du réseau deux à deux quelle que soit l'anomalie détectée [1]. La deuxième consiste à surveiller un ensemble de chemins obtenu suite à la détection d'une anomalie qui permet de distinguer seulement entre les liens suspects [2].

Le problème de sélection des emplacements de moniteurs, ainsi que le problème de sélection des chemins de surveillance sont $\mathcal{NP}$-dur. Par conséquent, plusieurs algorithmes heuristiques ont été proposés.

Les limitations des techniques de détection et de localisation existantes

Les techniques de détection et de localisation des anomalies au niveau des liens présentent les limitations suivantes :

- Les métriques d'optimisation habituellement considérées pour la sélection des emplacements de moniteurs (minimiser le nombre de moniteurs) et pour la sélection des chemin de surveillance (minimiser le nombre de chemins) ne reflètent pas les coûts de surveillance correctement. Par exemple, bien que la minimisation du nombre de chemins de surveillance est fortement désirable afin de réduire le coût de communications due à l'exportation des mesures à l'entité de gestion et de surveillance du réseau, cela pourrait augmenter le coût des sonde en produisant des mesures redondantes.
- Les techniques de sélection des emplacements de moniteurs et des chemins de surveillance en deux étapes ignorent les interactions entre les objectifs d'optimisation de chaque étape, ce qui peut conduire à une utilisation sous-optimale des ressource du réseau. En effet, le nombre et les emplacements des moniteurs ont un grand impact sur la qualité des chemins de surveillance.
- La technique de détection proposée dans [29] étudie les limitations abordées ci-dessus. Elle tient en compte la capacité des liens de supporter les flux de surveillance lors de la sélection des emplacements des moniteurs. Cependant, la principale limite de

cette technique est que les liens sont couverts sur plusieurs rounds, ce qui augmente les délais de détection proportionnellement aux nombre de rounds.

- La sélection des chemins de surveillance suite à la détection d'une anomalie, comme proposé dans [2], induit un délai de localisation non-négligeable.
- La surveillance d'un ensemble de chemins qui distingue entre tous les liens du réseau deux à deux à chaque fois qu'une anomalie est détectée, comme proposé dans [1], génère des mesures inutiles et augmente la surcharge de la surveillance.
- Les heuristiques de détection et de localisation des anomalies sélectionnent les chemins de surveillance parmi un ensemble de chemins candidats. Cet ensemble est décrit dans la littérature comme étant un petit sous-ensemble des chemins du réseau. Cependant, aucune indication sur la façon dont un tel ensemble est calculé est fournie. Il est clair que la réduction de nombre de chemins candidats est fondamentale pour assurer le passage à l'échelle, cependant, la réduction doit se faire de façon judicieuse afin de ne pas dégrader la qualité de la solution de surveillance.

Contribution de la thèse

L'objectif de cette thèse est de mettre au point une technique de surveillance à faible coût, efficace et précise qui surmonte les limitations soulevées dans le paragraphe précédent. Les principales contributions peuvent être résumées comme suit.

- Les objectifs d'optimisation considérés pour la sélection des emplacements des moniteurs et des chemins de surveillance ne sont pas limités à la minimisation du nombre de moniteurs et la minimisation du nombre des chemins de surveillance. Au contraire, les moniteurs sont placés de façon mesurée tel que le coût et les délais de communication avec l'entité de gestion et de surveillance du réseau sont réduits au minimum. En outre, les mesures qui ne fournissent pas d'information supplémentaire sur la performance du réseau sont évitées, ce qui réduit la charge des flux de surveillance sur le réseau.

- Les emplacements des moniteurs et les chemins de surveillance pour la détection, respectivement pour la localisation, d'anomalies sont sélectionnés conjointement en une seule étape. Il sera démontré que cette technique de sélection conjointe réalise

un bon compromis entre le coût de l'infrastructure de surveillance, la surcharge surveillance et les délais.

- Il est démontré dans la thèse que la condition sur l'ensemble des chemins qui doivent être surveillés pour la localisation des anomalies unique au niveau des liens établies dans [1] est suffisante mais n'est pas nécessaire. Une condition nécessaire et suffisante est établie et démontrée.
- Il est démontré que des solutions de localisation complète, les moniteurs qui sont à activer et les chemins qui sont à surveiller suite à la détection d'une anomalie, peuvent être calculées en offline.
- Les problèmes de détection et de localisation des anomalies au niveau des liens sont formulés mathématiquement. Il est démontré que les deux problèmes sont $\mathcal{NP}$-durs.
- Des algorithmes heuristiques pour la détection et la localisation des anomalies au niveau des liens sont développés. Les chemins de surveillance candidats sont sélectionnés de manière prudente, afin de ne pas dégrader la qualité les solutions de détection/localisation, tout en assurant le passage à l'échelle des algorithmes proposés.

La technique de détection proposée est une technique qui sélectionnent les emplacements des moniteurs et les chemins de surveillance conjointement en une seule étape. Une formulation ILP du problème est fournie, et il est démontré que le problème est $\mathcal{NP}$-dur. Deux algorithmes sont, par conséquent, proposés. Le premier algorithme considère l'ensemble de tous les chemins du réseau comme candidats à surveiller. Le second algorithme met en œuvre une procédure de calcul des chemins candidats. Le but de cette procédure est de réduire l'ensemble des chemins candidats afin de garantir le passage à l'échelle de l'heuristique, tout en assurant la qualité de la solution de détection. La technique proposé est comparée aux techniques de détection existantes qui procèdent en deux étapes. Les résultats de comparaison montrent la supériorité la technique proposée, et son efficacité pour réaliser un compromis entre les objectifs d'optimisation considérés.

L'applicabilité de la méthode de détection d'anomalies proposée sur les réseaux multi-domaines est étudié. Un algorithme ILP et un algorithme heuristique qui prennent en compte les propriétés et les limites de ces réseaux sont conçus. Une étude comparative de deux méthodes de détection d'anomalies est effectuée. La première méthode est

une approche globale qui considère le réseau multi-domaine comme étant un domaine unique. Dans un tel cas, le système de détection d'anomalies proposé pour les réseaux mono-domaines peut être appliqué. La deuxième méthode est une approche par domaine qui minimise les interactions entre les domaines pour tenter de surmonter les problèmes de confidentialité. Les résultats la comparaison montrent que la confidentialité est loin d'être la seule limite de la technique globale. En particulier, les résultats montrent que la technique de détection globale donne des solutions avec des chemins de surveillance relativement longs, et ne garantit pas une répartition équitable de la charge de surveillance entre les domaines de détection. En outre, le temps de calcul pour la technique globale est considérablement élevé par rapport au temps de calcul pour la technique par domaine. En revanche, la différence des coûts des solutions fournies par ces deux techniques, en termes de nombre de moniteurs et surcharge de surveillance, est faible.

Bien que la thèse préconise un découplage de la localisation de la détection (le processus de détection d'anomalies est exécuté en continu alors que le processus de localisation est déclenché uniquement en cas de détection d'une anomalie), il exploite le fait que la sortie du processus de détection est une entrée du processus de localisation pour optimiser la solution de localisation. En particulier, il est démontré que, connaissant l'ensemble des chemins surveillés pour détecter une anomalie, tous les scénarios d'anomalies qui pourraient se produire dans le réseau peuvent être déduits en offline [3]. Par la suite, l'ensemble des chemins qui doit être surveillé lors de la détection d'une anomalie est réduite à un petit sous-ensemble de chemins qui peuvent distinguer seulement entre les liens suspects. Cet ensemble est pré-calculé en offline. Tout comme la technique de détection, les emplacements de moniteurs et les chemins de surveillance sont sélectionnés conjointement en une seule étape. Le problème de la localisation est formulé en ILP, et il est démontré que c'est un problème $\mathcal{NP}$-dur. Un algorithme heuristique est donc proposé. La capacité de la technique proposée de localiser toutes les anomalies correctement est vérifiée analytiquement, et sa supériorité sur les techniques de localisation existantes est démontrée par le biais de simulations.

3. Un scénario d'anomalie est caractérisé par un ensemble unique de liens suspects. Des anomalies différentes peuvent provoquer le même scénario d'anomalie.

Contents

Abstract i

Résumé en français iii

List of Figures xix

List of Tables xxi

List of Algorithms xxiii

Glossary xxv

I Background and Technological Context 1

1 Introduction 3

1.1 Overview of End-to-End Monitoring Techniques 5

1.1.1 Analogue Monitoring 5

1.1.2 Binary Monitoring 6

1.2 Link-Level Anomaly Detection 7

1.3 Link-Level Anomaly Localization 7

1.4 Infrastructure Requirements for Link-Level Anomaly Detection and Localization 8

1.5 Link-Level Anomaly Detection and Localization Costs 10

1.6 Monitor Location and Monitoring Path Selection for Link-Level Anomaly Detection and Localization 10

1.7 Limitations of The Existing Link-Level Anomaly Detection and Localization Schemes 11

1.8 Contributions of The Thesis 14

1.9 Outline of The Thesis 15

II Detection of Link-Level Network Anomalies 17

2 Link-level Anomaly Detection in Mono-Domain Networks 19

2.1 Introduction 19

2.2 Network Model 21

2.3 Problem Formulation 21

2.4 Cost Model 23

2.5 Path-based ILP Formulation 24

2.6 Link-Flow-Based ILP Formulation 25

2.7 The Anomaly Detection Problem is $\mathcal{NP}$-Hard 27

2.8 Heuristic Algorithms for joint monitor location and monitoring path selection 30

2.8.1 Exhaustive greedy algorithm 30

2.8.2 Selective Greedy Algorithm 33

2.9 Evaluation 35

2.9.1 Evaluation of The ILP Formulations 37

2.9.2 Evaluation of The Heuristic Algorithms 41

2.10 Conclusion 44

3 Link-Level Anomaly Detection in Multi-Domain Networks 47

3.1 Introduction 47

3.2 Problem Formulation 49

3.2.1 Network Model 49

3.2.2 Problem Definition 49
3.2.3 Architecture and Cost Model of Multi-Domain Anomaly Detection . 51
3.3 ILP formulation 54
3.4 Heuristic Algorithm for Anomaly Detection in Multi-Domain Networks . . . 55
3.4.1 Computation of Candidate Monitoring Paths in Multi-Domain Networks 56
3.4.2 Greedy Monitor Location and Path Selection Algorithm 56
3.5 Performance Evaluation 58
3.5.1 Evaluation Methodology 58
3.5.2 Numerical Results 59
3.6 Conclusion 65

III Localization of Link-Level Network Anomalies 67

4 Localization of Single Link-Level Network Anomalies 69

4.1 Introduction 69
4.2 Network Model and Problem Statement 71
4.3 Not all link pairs need to be distinguishable for localizing any single link-level anomaly 72
4.4 Derivation of potential anomaly scenarios 74
4.5 Anomaly localization cost 77
4.6 ILP Formulation 78
4.7 The Anomaly Localization Problem is $\mathcal{NP}$-Hard 80
4.8 Heuristic solution 83
4.8.1 Monitor location selection 83
4.8.2 Selection of localization paths 84
4.8.3 Candidate path selection algorithm 87
4.9 Performance Evaluation 90

4.9.1 Comparing our Anomaly Localization Scheme with Existing Schemes 91

4.9.2 Evaluating the Scalability and Quality of the Heuristic 95

4.10 Discussion 99

4.11 Conclusion 100

5 Conclusion and Perspectives **103**

Appendix A **107**

Appendix B **109**

Appendix C **111**

Appendix D **113**

Appendix E **115**

Bibliography **117**

Acknowledgment **123**

List of Figures

1.1 A tree-structured topology consisting of one source, one internal node and two receivers 5

1.2 Example of a network topology 8

1.3 Example of a detection infrastructure (gray nodes are monitor locations) and detection paths (thick gray lines) 9

1.4 Example of a single anomaly localization infrastructure and localization paths 9

1.5 Example of an anomaly detection solution with two monitors, three detection paths, and two redundant measurements 12

1.6 Example of an anomaly detection solution with four monitors, seven detection paths, and zero redundant measurements 12

1.7 Example of an anomaly detection solution with four monitors, seven detection paths, and two redundant measurements 13

2.1 Illustrative example of anomaly detection solutions 22

2.2 Example of a graph constructed out of a facility location instance with three facility locations and four clients 28

2.3 Gap-to-Optimality vs. Granted Running Time. (a) Results for the topologies with 8 nodes and 18 links; (b) Results for the topologies with 10 nodes and 31 links. 40

2.4 Performance results for our detection scheme with different values of α, β, and γ ($\beta = \alpha$); and for the existing detection scheme (denoted as EDS). (a) Results for topologies with 8 nodes and 18 links; (b) Results for topologies with 10 nodes and 31 links. 42

3.1 Per-domain detection solution 50
3.2 Global detection solution 50
3.3 Sample Multi-domain Monitoring Architecture 52
3.4 Illustrative multi-domain network 55
3.5 Sample multi-domain topology 59
3.6 Monitoring cost: default setting 60
3.7 Monitoring Cost: doubling inter-domain links 61
3.8 CPU Running Time (s) 62
3.9 Distribution of network links by path length groups 63
3.10 Distribution of monitors and redundant measurements across domains 64

4.1 Illustrative network topology, (a), and an associated detection solution, (b). 75
4.2 Example of a graph constructed out of a facility location instance with four facility locations and four clients 81
4.3 Average number of monitoring paths per anomaly for TOP(8, 18). The first histogram to the left presents results for solutions computed using the hybrid localization scheme (HLS), and the other histograms present results for the solutions computed using our anomaly localization ILP with different values of α ($\beta = 1$). 93
4.4 Localization costs for TOP(8, 18) 94
4.5 Localization cost of the heuristic solutions, $\alpha >> \beta$ 97
4.6 Impact of the number and the quality of candidate monitoring paths on the quality of the localization solution. RProc means random procedure (numerical results for TOP(15, 59)) 98

List of Tables

2.1 Notations used in the pseudo-codes 31

2.2 Summary of the topologies considered in the evaluation 37

2.3 CPU Running Time (CPU) and Gap-To-Optimality (GTO) for TOP(6,10) and TOP(12,41). 39

2.4 CPU running time (OOM means Out Of Memory) 41

2.5 Average number of deployed monitors + average number of redundant measurements (network utilization) 43

2.6 Resource utilization for SGA 44

3.1 Notations used throughout this chapter 53

4.1 Sets of suspect links for all potential anomalies 76

4.2 Anomaly scenarios 76

4.3 Summary of the topologies considered in the evaluation 91

4.4 Average ILP computation time for TOP(8, 18) 92

4.5 Heuristic computation time (all computations are done offline) and percentage of paths explored in one execution of Procedure 2 96

List of Algorithms

1 Exhaustive greedy algorithm for anomaly detection 32

2 Selective Greedy Algorithm 34

- Procedure 1: candidatePathComputation($\mathcal{G}$, n_1, n_2, $\mathcal{CL}$) 36

3 Monitor location and path selection algorithm for anomaly detection in multi-domain networks 57

4 Monitor location and path selection algorithm for single anomaly localization 85

- Procedure 2: candidatePathSelection($m, \mathcal{SM}, \mathcal{G}, \mathcal{S}_a^{(j)}$, $\mathcal{CP}$) 88

Glossary

CPU	Central Processing Unit
EDS	Existing Detection Solution
EGA	Exhaustive Greedy Algorithm
GTO	Gap To Optimality
HLS	Hybrid Localization Scheme
ILP	Integer Linear Programm
LP	Linear Programm
NOC	Network Operations Center
SGA	Selective Greedy Algorithm

Part I

Background and Technological Context

CHAPTER 1

Introduction

The Internet has experienced a transition from being a simple data transmission network serving a few users to becoming a multi-service network that supports various multimedia applications with high QoS requirements (*e.g.*, loss rate, end-to-end delay, jitter, throughput, etc.) and serves a sharply growing number of demanding users. This is due to the rapid development of more and more powerful and affordable network devices (*e.g.*, high-capacity transmission mediums, high-speed switching, high-capacity storage devices, etc.). The need for efficient network monitoring tools that ensure a desired network performance and provide QoS guarantees has subsequently increased. A large number of monitoring schemes and network measurement tools have been proposed in the literature.

The simplest monitoring schemes make use of existing networking tools such as ping and traceroute [18][23]. They are qualified as simple because they do not require any specific feature in the network. However their application is limited to detect and localize infrastructure failures and path outage [27]. Schemes that provide more detailed performance information have been proposed. They can be broadly divided into two categories, individual monitoring schemes (*e.g.*, SNMP(Simple Network Management Protocol)-based schemes [7], RMON [28], Netflow [8]), and end-to-end monitoring schemes (*e.g.*, [14] [24] [26] [9] [10] [11] [17] [16] [15][21] [20] [19] [29] [1] [6] [22] [2]). The basic idea of individual monitoring schemes is to equip every network device with a monitoring agent that collects performance statistics for the device and its incident links by snooping on the network traffic crossing it. Individual statistics are exported to a network operations center for analysis. The major problems of these schemes is that the monitoring infrastructure cost can be very high in large-size network, and the exportation of individual statistics to the operations center may generate a heavy burden on the network. End-to-end monitoring is an intuitive solution to these problems. The idea is to infer internal network performance through end-to-end measurements, which should require much less monitoring devices to be deployed in the network and minimize the monitoring overhead.

There exists another classification of monitoring schemes: passive monitoring schemes and active monitoring schemes. Passive monitoring infers the network performance by snooping on existing network traffic. There are two approaches to perform passive monitoring:

- Two-point monitoring: this monitoring approach deploys two monitoring devices at the ingress and egress nodes of each monitored flow. Performance metrics are inferred by comparing measurements performed at ingress and egress monitors. This requires the timestamps of the monitors to be synchronized and all packets traversing them to be identified. However, the identification process might lead to serious scalability issues when the volume of traffic traversing the monitors is important.
- One-point monitoring: This monitoring approach requires one single monitor for monitoring one flow. It uses TCP acknowledgments to infer performance metrics between the point where the monitor is deployed and the sink of the monitored TCP flow (*e.g.*, loss rate and round trip time on the segment between the monitor location and the sink of the monitored flow). Clearly, the application of this approach is restricted to TCP flows.

Active monitoring infers the performance of the network (*e.g.*, availability, loss rate, delay, etc.) by making measurements on active monitoring flows, called in this context active probes, injected in the network to simulate existing network flows. The main difficulty of active monitoring is to make active probes experience the same conditions as real traffic flows in order to achieve accurate measurements, without interfering with the network services.

Although the two monitoring approaches have their own drawbacks, the active monitoring have two important advantages over passive monitoring. the first is that it preserves privacy and confidentiality of services crossing the network since it does not make measurements on real traffic flows. The second is that it is possible using active monitoring to make measurements when there are no flows traversing the network. For instance, a service provider might need to check the availability and the characteristics of a network path previously not used before it transmits services on it, which is not feasible using passive monitoring.

Both active and passive end-to-end monitoring problems have been widely studied in the literature. Despite their divergence in terms of measured metrics and the approach of measurement acquisition, all the proposed schemes share a common important objective: guarantee a desirable network performance while minimizing the monitoring expense in

terms of infrastructure cost and monitoring overhead. The aim of this thesis is to come up with an end-to-end network monitoring scheme that achieves this objective.

The remainder of this chapter is organized as follows. Section 1.1 provides a classification of existing end-to-end monitoring techniques. Section 1.2 and section 1.3 define the problem of link-level anomaly detection and link-level anomaly localization, respectively. Section 1.4 and section 1.5 describe the infrastructure requirements and the costs incurred for link-level anomaly detection and localization. section 1.6 and section 1.7 presents existing link-level anomaly detection and localization schemes and their limitations, respectively.

1.1 Overview of End-to-End Monitoring Techniques

End-to-end monitoring techniques can be broadly classified into two categories: analogue and binary [22].

1.1.1 Analogue Monitoring

Motivated by the effectiveness of multicast communications in terms of bandwidth saving, the early end-to-end monitoring schemes used end-to-end active multicast probes to infer link-level loss rate, delay distribution, and bottleneck bandwidths (*e.g.*, [3] [14] [26] [24] [4] [25]). The key idea is to correlate the copies of multicast probe packets observed at the multicast receivers to infer the performance of links within the multicast tree.

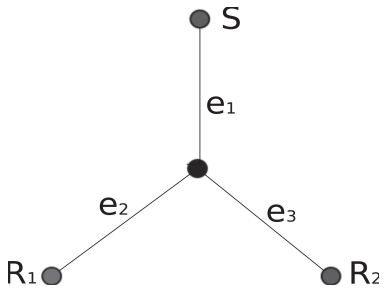


Figure 1.1: A tree-structured topology consisting of one source, one internal node and two receivers

Consider the logical multicast tree depicted in Figure 1.1 to illustrate. The loss events are inferred as follows. If a copy of a multicast probe packet is received by R_1 but not R_2, then a loss has likely occurred on the link e_3. If neither R_1 nor R_2 receive copies of

the probe packet, then losses have likely occurred either on e_1, or on e_2 and on e_3. A probabilistic analysis of repeated multicast probes provides an estimation of the loss rates of the tree links with high probability (textite.g., [14]). Similarly, a probabilistic analysis of the correlations between the delays that make the copies of a probe packet issued by the multicast source to reach the multicast receivers provides an estimation of the link delay distributions (textite.g., [24]). Bottleneck bandwidths can be estimated through correlations of loss statistics across the multicast receivers (textite.g.,[26]).

Despite its potential benefits, multicast-based schemes cannot not be widely applied because multicast is so far only modestly deployed. Several research works proposed to emulate multicast-based monitoring schemes using unicast measurements (*e.g.*, [9], [10] [11], [17], [16], [5]). The idea is to send two closely time-spaced packets, referred to as back-to-back packet pairs, from one server to pairs of receivers whose paths back to the source share a set of common links. The back-to-back packets issued from the same source and having the same characteristics are very likely to experience the same performance on the shared links. This performance correlation is exploited, in the same way as for multicast-based schemes, to infer link-level performance parameters.

1.1.2 Binary Monitoring

A new feature has been widely adopted by the monitoring research community. It consists in identifying the deviations of the network performance from a given performance baseline rather than estimating link-level performance measurements. This feature resets on the assumption that link performance is separable, which implies that a path experiences bad performance if and only if at least one of its constituent links experiences bad performance [15]. Thus, identifying link-level performance violations can be done by identifying paths that violate performance thresholds. More specifically, according to the property of separable performance, it is enough to monitor a set of paths that cover all links of the network for detecting all potential link-level performance violations. Further paths need to be monitored to localize the source(s) of the violation(s). [15] states numerous separable link performance parameters such as connectivity, high-low loss model and delay spike model.

Many research works exploited the property of separable performance to devise link-level anomaly detection and localization schemes (*e.g.*, [21], [20], [19], [29], [1], [6]). we next investigate the problems of link-level anomaly detection and localization.

1.2 Link-Level Anomaly Detection

The goal of link-level anomaly detection is to detect any performance degradation or infrastructure failure that would occur on the network links. In this thesis we consider separable anomalies that satisfy the separable performance property established in [15]. As mentioned previously, a path exhibits a separable anomaly if and only if at least one of its constituent links is anomalous. The trivial conclusion that can be drawn from this property is that for detecting all potential link-level anomalies in a given network, a set of paths that cover all links of the network must be monitored. A link is said to be covered if it is traversed by at least one monitored path. It can be easily shown that this is a necessary and sufficient condition for link-level anomaly detection.

The information delivered by the anomaly detection process is a set of anomalous paths, *i.e.*, monitored paths that exhibit an anomaly. We refer to the set of links that are traversed by only anomalous monitored paths as the set of suspect links. It cannot be decided whether these links are anomalous using only the detection information. Let us consider the network topology depicted in Figure 1.2 to illustrate. Suppose that nodes a and d are equipped with monitoring devices. Consider the bidirectional paths $p_1 = \langle (a,\ b),\ (b,\ c),\ (c,\ d) \rangle$, $p_2 = \langle (a,\ b),\ (b,\ d) \rangle$ and $p_3 = \langle (a,\ d) \rangle$ that cover all links of the network (refer to Figure 1.3 for an illustration). Assume that the detection process which monitors these three paths reports that p_1 is anomalous. According to the separable performance property, all links that are traversed by paths not exhibiting the anomaly are surely not anomalous. We conclude that all links that are not traversed by p_1 as well as the link connecting node a to node b are not anomalous. Thus, the set of suspect links is $\{(b,\ c),\ (c,\ d)\}$. We say that paths p_1, p_2 and p_3 cannot distinguish between the links $(b,\ c)$ and $(c,\ d)$. Further paths must be monitored in order to decide whether $(b,\ c)$, $(c,\ d)$ or both links are anomalous. This operation is called link-level anomaly localization.

1.3 Link-Level Anomaly Localization

Link-level anomaly localization aims at identifying the root cause of a detected anomaly. Let us consider again the anomaly scenario described in the previous section. The set of suspect links constructed out of the detection information when path p_1 exhibits an anomaly is $\{(b,\ c),\ (c,\ d)\}$. To localize the anomalous link(s) among the suspect links,

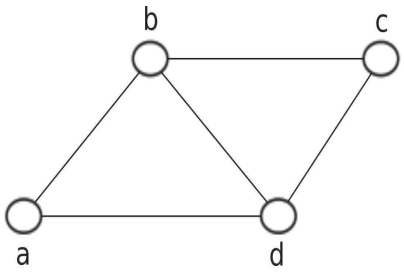


Figure 1.2: Example of a network topology

two additional paths must be monitored. Either path must traverse one of the two suspect links, but not both. Additional monitoring devices may need to be deployed. In this case, one additional monitoring device need to be deployed on node c. The paths monitored during the localization process are $p_4 = \langle (a,\ b),\ (b,\ c) \rangle$ and $p_5 = \langle (c,\ d) \rangle$. If both paths exhibit an anomaly, then both suspect links are anomalous. Otherwise the suspect link traversed by the path that exhibits an anomaly is anomalous.

A sufficient condition for localizing link-level anomalies has been been established in the literature (*e.g.*, [21], [6], [1]). It consists in deploying a set of monitoring devices that can distinguish between every two subsets of the network links. This implies that for each pair of link subsets there exists a path between the deployed monitoring devices whose intersection with exactly one of the two subsets is not empty. For instance, for the sample topology depicted in Figure 1.2, there is only one path, $p_6 = \langle (a,\ b) \rangle$, that can distinguish between the subsets $\{(a,\ b),\ (b,\ c),\ (c,\ d)\}$ and $\{(b,\ c),\ (c,\ d)\}$. Thus, monitoring devices must inevitably be deployed on node a and node b. In practice, multiple link-level anomalies that involve a large number of links are rare events. Therefore, numerous works bound the number of concurrent anomalies in an attempt to minimize the localization cost, *e.g.*, [1] claims that anomalies involving more than three links are very unlikely to occur.

1.4 Infrastructure Requirements for Link-Level Anomaly Detection and Localization

The anomaly detection (respectively localization) infrastructure consists of a set of monitoring devices placed at a subset of the network nodes such that there exists a set of paths between the nodes equipped with monitoring devices that covers all links of the network (respectively distinguish between all subsets of the network links pairwise). The

network nodes that support monitoring devices are referred to as monitor locations. The term monitoring path is used interchangeably with the term detection paths to designate paths that are monitored for anomaly detection, and is used interchangeably with the term localization paths to designate paths that are monitored for anomaly localization.

Figure 1.3 shows an example of a detection infrastructure and detection paths for the sample network topology depicted in Fig 1.2, and Figure 1.4 shows an example of a single link-level localization infrastructure, *i.e.*, simultaneous anomalies involving multiple links are not considered, and localization paths for the same network topology.

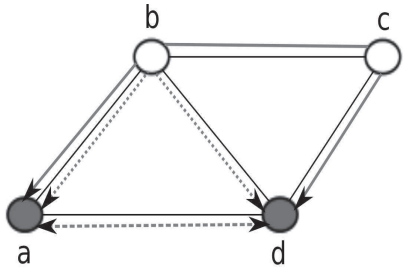


Figure 1.3: Example of a detection infrastructure (gray nodes are monitor locations) and detection paths (thick gray lines)

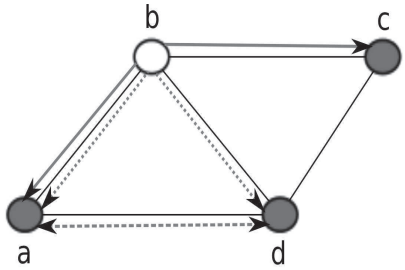


Figure 1.4: Example of a single anomaly localization infrastructure and localization paths

Usually, the anomaly detection infrastructure is continuously active, whereas the anomaly localization infrastructure is activated only upon detecting an anomaly. For instance, if path $\langle(a,\ b),\ (b,\ c),\ (c,\ d)\rangle$ exhibits an anomaly, then, activating only the monitors on node a and node c and monitoring only the localization path $\langle(a,\ b),\ (b,\ c)\rangle$ is sufficient to pinpoint the anomalous link. The rationale behind activating the anomaly localization process only upon detecting an anomaly is that network anomalies are typi-

cally rare events. Moreover, depending on the network topology, running the localization process continuously may incur a heavy burden on the underlying network.

Furthermore, measurements collected by active monitoring devices are exported to a network operations center, referred to as NOC. The NOC analyzes and correlates the measurements collected individually by the monitoring devices. When it detects an anomaly, it triggers the anomaly localization process by activating some monitoring devices that can distinguish between the suspect links.

1.5 Link-Level Anomaly Detection and Localization Costs

The anomaly detection and localization costs consist of the following costs:

- Infrastructure cost: this is the effective cost of acquiring, deploying and maintaining software and hardware monitoring devices.
- Communication cost: this is the cost of communications between the NOC and the monitoring devices that are deployed in the network. The NOC collects monitoring measurements from the monitoring devices that are activated for anomaly detection periodically. When an anomaly is detected, the NOC triggers the localization phase by activating the localization process on a subset of the monitors deployed that can distinguish between the set of suspect links constructed out of the detection measurements. It is of great importance to choose the locations where to deploy monitors carefully, in order to reduce the communication overhead and delays.
- Probe cost: this cost expresses the load of monitoring flows on the network. Measurements of links that do not provide any extra detection/localization information is highly indesirable. Indeed, such measurements increase the detection/localization delays and overhead.

1.6 Monitor Location and Monitoring Path Selection for Link-Level Anomaly Detection and Localization

One of the problems that received great interest within the research community on network monitoring is formulated as follows: *How to choose monitor locations and how to select monitoring paths that can detect/localize all potential anomalies while minimizing the costs incurred and reducing the detection/localization delays* (*e.g.*, [6] [1] [20] [21] [29]).

Almost all existing network monitoring schemes apply a two-step approach for monitor location and monitoring path selection. Usually, the first step selects the smallest set of monitor locations that can detect/localize all potential anomalies. The second step selects the smallest set of paths between the monitor location selected at the first step that cover/distinguish between all potential anomalies (*e.g.*, [6] [1]).

[21] applies an inverse two-step approach of monitor location and monitoring path selection for localizing multiple link failures. The first step selects a set of optimal monitoring paths that can localize all potential multiple failures, whereas the second step selects the smallest set of monitor locations that can monitor paths selected at the first step.

[29] proposes a multi-round link-level anomaly detection schemes. It takes into account the capacity of the network links to support monitoring flows and the capacity of monitoring devices to generate probe messages while selecting monitor locations. The result is a minimal set of monitor locations and monitoring paths that covers all the network links in a certain number of rounds.

As mentioned previously, it is agreed that monitoring a set of paths that covers all network links is necessary for detecting all potential link-level anomalies. However, the set of paths that is to be monitored to pinpoint the source of a detected anomaly has been defined in two ways. The first proposes to monitor a set of paths that can distinguish between every pair of link-level anomalies for any detected anomaly (*e.g.*, [1]), whereas the second monitors a set of paths selected upon detecting an anomaly that can distinguish only between the set of suspect links (*e.g.*, [2]).

Both the problems of monitor location and the problem of path selection are $\mathcal{NP}$-Hard. Therefore, heuristic algorithms, most of them greedy, have been proposed.

1.7 Limitations of The Existing Link-Level Anomaly Detection and Localization Schemes

The existing anomaly detection and localization schemes present the following limitations:

– The optimization metrics usually considered for monitor location selection (minimizing the number of monitors that are to be deployed) and monitoring path selection (minimizing the number of paths that are to be monitored) do not reflect the monitoring costs properly. For instance, although minimizing the number of monitoring

paths is highly desirable to reduce the communication overhead due to exporting the measurements carried out for each monitored path to the NOC at each time interval, this is likely to generate heavy probe overhead. For example, Figure 1.6 and Figure 1.5, each depicting a different anomaly detection solution for the same network topology, illustrate that reducing the number of detection paths from seven paths to three paths generates redundant measurements.

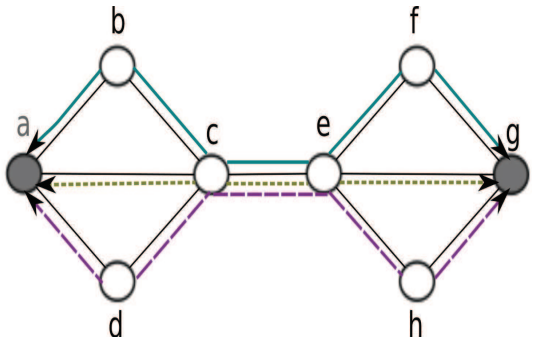


Figure 1.5: Example of an anomaly detection solution with two monitors, three detection paths, and two redundant measurements

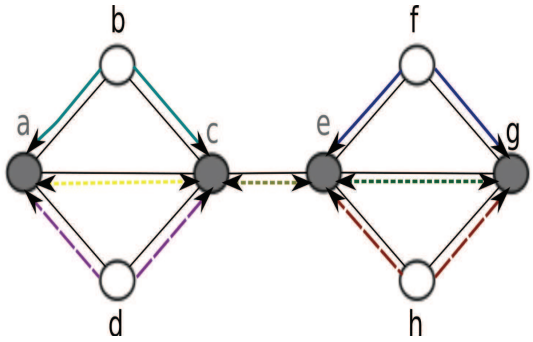


Figure 1.6: Example of an anomaly detection solution with four monitors, seven detection paths, and zero redundant measurements

– The step-wise approaches for monitor location and monitoring path selection ignore the interplay between the optimization objectives of each step, which may lead to sub-optimal consumption of the network resources. We contend that the number and locations of monitoring devices have an impact on the quality of monitoring paths. For instance, Figure 1.6 shows that two monitoring devices are sufficient to detect all potential link level anomalies of the considered network topology, however, as illustrated in Figure 1.5, at least four monitoring devices are required to cover

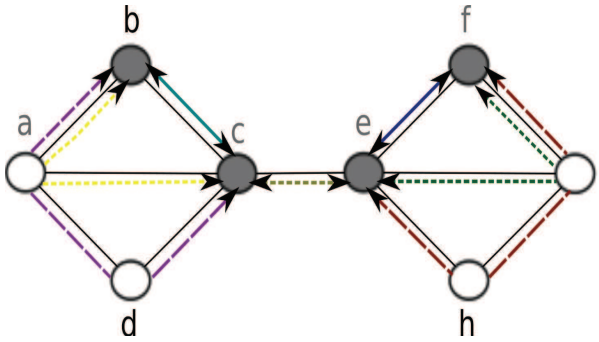


Figure 1.7: Example of an anomaly detection solution with four monitors, seven detection paths, and two redundant measurements

the network links without generating redundant measurements. Figure 1.7 shows that redundant measurements cannot be avoided when changing the locations of two among the four monitoring devices of the solution presented in Figure 1.5, which illustrates the correlation between the locations of monitoring devices and the quality of monitoring paths.

– The anomaly detection scheme proposed in [29] addresses the issues discussed above in that it takes into account the capacity of links to support monitoring flows while selecting monitor locations. However, the major limitation of the proposed scheme is that links are covered over multiple rounds, which increases the detection delays proportionally to the number of rounds.

– Selecting localization paths online, *i.e.*, upon detecting an anomaly, as done in [2], induces non-negligible delay.

– Monitoring a set of localization path that distinguishes between every pair of link-level anomalies whenever an anomaly is detected, as done in [1], incurs unnecessary overhead and delay.

– Heuristic detection and localization algorithms select monitoring paths from a set of candidate paths. This latter is described in the literature as a small subset of the network paths. However, there is any indication on how such a set is computed. Clearly, reducing the number of candidate paths is fundamental for scalability, however, the reduction must be done in a measured way in order not to degrade the quality of the monitoring solution.

1.8 Contributions of The Thesis

The goal of this thesis is to come up with a cost-effective, fast and accurate monitoring scheme. The proposed scheme is to some extent similar to recent monitoring schemes in that it performs anomaly detection and localization in two phases. However, it overcomes the limitations raised in the previous chapter. The main contributions can be summarized as follows.

- The optimization objectives considered for monitor location and monitoring path selection are not limited to minimizing the number of monitoring devices that are to be deployed and the number of paths that are to be monitored. Rather, monitors are placed in a measured way such as the cost and the delays of communications with the NOC are minimized. Moreover, measurements that do not provide extra information are avoided, thereby reducing the monitoring overhead.

- Monitor locations and monitoring paths for anomaly detection, respectively for anomaly localization, are selected in one single step. It will be demonstrated that the joint selection achieves a good trade-off between the monitoring infrastructure cost and the monitoring overhead and delays.

- The condition on the set of paths that need to be monitored for localizing single link-level anomalies established in [1] is proved to be sufficient but not necessary. A necessary and sufficient condition is developed.
- A demonstration that full localization solutions, *i.e.*, monitoring devices that are to be activated and paths that are to be monitored upon detecting a given single link-level anomaly, can be derived offline is provided.

- The anomaly detection and localization problems are formulated as ILPs. Both problems are shown to be $\mathcal{NP}$-hard.

- Heuristic algorithms for anomaly detection and for anomaly localization are devised. Candidate monitoring paths are selected in a careful way, in order not to degrade the quality of the detection/localization solutions, while ensuring the scalability of the heuristic algorithms.

– Operational constraints (*e.g.*, limiting the capacity of monitoring devices to generate and manage monitoring flows, limiting the capacity of links to support monitoring flows, etc.) can be easily introduced into the ILP formulations and the heuristics.

1.9 Outline of The Thesis

The remainder of the thesis is divided into two parts: Detection of Link-Level Network Anomalies and Localization of Link-Level Network Anomalies. The former part is composed of Chapter 2 and Chapter 3. The former chapter addresses the problem of link-level anomaly detection in mono-domain networks, whereas the latter chapter investigates the same problem in multi-domain networks. The latter part addresses the problem of link-level anomaly localization. It is composed of Chapter 4. Chapter 5 concludes the dissertation and presents future perspectives.

Part II

Detection of Link-Level Network Anomalies

CHAPTER 2

Link-level Anomaly Detection in Mono-Domain Networks

2.1 Introduction

Most existing monitoring approaches operate in two phases (*e.g.*, [2], [29], [1], [19], [2]). The first phase is the anomaly detection phase. It consists in deploying as few resources as possible such that all links of the network are covered in order to detect all potential link-level anomalies. The second phase is the anomaly localization phase. It is triggered upon detecting an anomaly in order to identify its root cause.

In this chapter, we focus on the anomaly detection phase. We revisit a widely studied problem that is the placement of monitoring devices and the selection of monitoring paths for anomaly detection (*e.g.*, [29], [2], [1], [19], [6], [21], [32], [37], [34], [36], [35]). The motivation behind our work is that existing solutions suffer from two major shortcomings. The first is that most of them adopt a two-step approach for monitor location and monitoring path selection, and do not address the trade-off between the optimization objectives of each steps. The monitor location step selects locations for a minimal set of monitoring devices such that all links of the network are covered. The monitoring path selection step computes a minimal set of paths between the deployed monitors that cover all links of the network. The second is that existing monitoring cost models do not meet the requirements of the monitored networks. For instance, the number of monitoring paths does not reflect the effective monitoring load. Indeed, minimizing the number of monitoring paths is very likely to produce long monitoring paths that cover some network links multiple times, and thus, generating extra monitoring overhead and extending the detection delays. Furthermore, the monitor locations should be selected carefully with regard to the NOC location

in order to minimize the overhead and the delays of communications between the monitors and the NOC.

We define a monitoring cost model that takes into account realistic constraints and aims at reducing the anomaly detection overhead and delays, and the cost of deploying the monitoring infrastructure. Then, we provide a one-step formulation of the monitor location and the monitoring path selection problems. Our goal is to optimize the associated costs jointly, thereby minimizing the total anomaly detection cost. Two ILP formulations are provided. A path-based ILP that requires high memory capacity and low processing capacity, and a link-flow ILP that requires high processing capacity and low memory capacity. We show that the problem is $\mathcal{NP}$-hard. Commonly, to simplify the problem, the set of candidate paths that are to be monitored is restrained to a small sub-set of the network paths and the set of candidate monitor locations is restrained to a small subset of the network nodes. However, none of existing works on anomaly detection investigated the impact of these restrictions on the quality of the detection solution. Moreover, none of them specified how to choose the set of candidate monitoring paths and the set of candidate monitor locations. We provide a heuristic solution that achieves scalability by reducing the number of candidate monitoring paths in an efficient way, and thus delivers cost-effective detection solutions.

We use extensive simulations to illustrate the interplay between the optimization objectives of the monitor location and the monitoring path selection problems. By way of comparison, we show that our anomaly detection scheme outperforms existing two-step anomaly detection schemes, and we demonstrate the efficiency and the scalability of our heuristic solution.

The remainder of this chapter is organized as follows. Section 2.2 describes the network model, and Section 2.3 states the anomaly detection problem. Section 2.4 introduces the anomaly detection cost model. The Path-based ILP is formulated in section 2.5, whereas the link-flow-based ILP is introduced in Section 2.6. Section 2.7 demonstrates that the anomaly detection problem is $\mathcal{NP}$-Hard. The heuristic algorithms are introduced in Section 2.8. The performance of the proposed anomaly detection scheme is evaluated through simulations in section 2.9. Concluding remarks are provided in Section 2.10.

2.2 Network Model

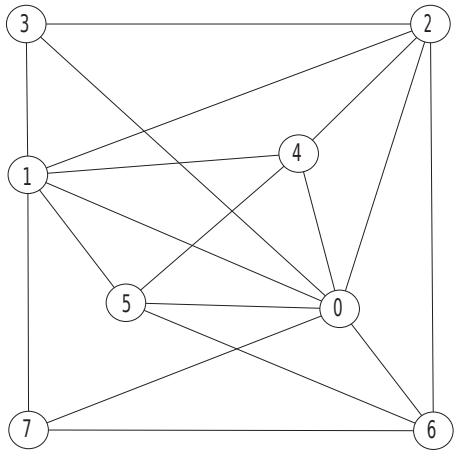
We model the network as an undirected graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{E}$ is the set of links interconnecting them. In some parts of the paper, we express links using the nodes that they connect. For instance, a link $e \in \mathcal{E}$ that connects nodes i and j is denoted as (i, j). Let $\mathcal{P}$ be the set of non-looping paths of the networks, *i.e.* all loop-free paths between every pair of the network nodes, we assume that all the network nodes are candidate to hold monitoring devices and that all the network paths are candidate to be monitored. Thus the set of candidate monitor locations is $\mathcal{N}$ and the set of candidate monitoring paths is $\mathcal{P}$. We assume that the monitoring devices can differently be source or sink of monitoring flows. If a path is selected to be monitored, then its end-nodes must be selected to hold monitoring devices. A monitoring path covers all its constituent links.

The NOC coordinates the monitoring task, collects and processes the monitoring measurements. The anomaly detection phase is run periodically. For active monitoring, this consists in injecting monitoring flows along a set of monitoring paths that cover all the network links. For passive monitoring, it consists in snooping on real traffic flows that cover all the network links. We consider an active monitoring approach. However, the proposed anomaly detection scheme applies for passive monitoring. The particularity of passive monitoring is that the cost associated to injecting monitoring flows in the network in zero.

2.3 Problem Formulation

An anomaly detection solution consists of two parts: A set of locations, *i.e.*, nodes, where to deploy monitoring devices and a set of paths that are to be monitored. There are two satisfactory constraints to be considered while devising the anomaly detection solution. First the selected monitoring paths must start and end at nodes that hold monitoring devices. Second, the union of the monitoring paths must cover all links of the network. Each link must be covered at least by one monitoring path. However, multiple measurements of links do not provide any extra anomaly detection information.

The problem with the existing anomaly detection schemes is that they compute the two parts of the solution in a stepwise fashion without considering the impact of the number and the locations of monitoring devices on the quality of monitoring paths. Consider the network in Fig. 2.1(a) to illustrate the interplay between these metrics. Fig. 2.1(b) depicts



(a) Sample network topology

Solution number	Selected monitor locations	Number of monitoring paths	Number of redundant measurements
1	3, 6	4	7
2	1, 2	6	3
3	1, 3	6	5
4	0, 6	7	5
5	2, 6, 7	5	1

(b) Associated anomaly detection solutions

Figure 2.1: Illustrative example of anomaly detection solutions

some anomaly detection solutions, computed using a stepwise approach, for this network topology. The considered operational constraints are the minimization of the number of monitoring devices that are to be deployed, and the minimization of the number of paths that are to be monitored (Refer to section 2.9.1 for a description of the ILP formulations and the simulation environment used for computing these solutions). It should be noted that Fig. 2.1b does not present the exhaustive list of solutions. On the one hand, Fig. 2.1(b) shows that although the solutions 1, 2, 3 and 4 deploy a small number of monitoring devices, they do not monitor the same number of paths. For instance, solution 1 monitors 43% less paths than solution 4. This illustrates the impact of monitor locations on the number of monitoring paths. On the other hand, we notice that reducing the number of monitoring paths does not necessarily avoid redundant measurements. Indeed, solution 1 incurs four more redundant measurements than solution 2 that monitors three more paths.

Furthermore, adding one monitor in solution 5 reduces drastically the number of redundant measurements.

Based on the above observations, we contend that there is an interplay between the number and the locations of monitors, and the quality of monitoring paths. Moreover, the operational constraints, considered in previous works, are suboptimal. In the remainder of this chapter, we address these two issues. We first introduce a novel anomaly detection cost model that takes into account new operational constraints towards minimizing the anomaly detection overhead and delays. Then we provide ILP formulations that select monitor locations and monitoring paths jointly, thereby achieving a good trade-off between the desired minimization objectives.

2.4 Cost Model

The anomaly detection cost includes three costs:

– Infrastructure cost: This is the effective cost of acquiring, deploying and maintaining software and hardware monitoring devices. Let C_{infra} be the cost of installing and maintaining one monitoring device on a node of the network. Let Y_n be a binary variable that indicates whether node n is selected as a monitor location. The infrastructure cost can be expressed as follows:

$$C_{infra} \sum_{n \in \mathcal{N}} Y_n \tag{2.1}$$

The minimization of (2.1) aims at deploying as few monitoring devices as possible. Note that all or a subset of the network nodes can be candidate to support monitoring devices. We assume, in this work, without loss of generality, that all the network nodes are candidate.

– Communication cost: this is the cost of communications between the NOC and the monitoring devices that are deployed in the network. The NOC collects monitoring measurements from the monitors periodically. When an anomaly is detected, the NOC stops the detection phase and triggers the localization phase. This is done by sending messages to the monitors asking them to switch to the localization phase. The detection phase resumes by sending messages to the monitors when the anomaly is localized and fixed. It is of great importance to choose the locations where to deploy monitors carefully, in order to reduce the communication overhead and delays. Let C_n be the cost of communications between the NOC and a monitor deployed on node

n. For instance, C_n can be proportional to the number of hops that separate node n from the NOC. The total communication cost reads as follows:

$$\sum_{n \in \mathcal{N}} C_n Y_n \tag{2.2}$$

The minimization of (2.2) aims at selecting the monitor locations that incur the lowest communication overhead and delays.

– Probe cost: This cost expresses the load of monitoring flows on the network. Each link of the network must be monitored at least by one monitoring path. However, redundant measurements are highly undesirable. This is because they only increase detection delays and overhead, and do not provide any extra detection information. Let Z_p be a binary variable that indicates whether path p is selected to be monitored. Let δ_{pe} be a binary input parameter that indicates whether path p covers link e. Let C_e be the cost of injecting one detection flow along link e. C_e should be proportional the the load of e [1], in order to avoid redundant measurements of the most loaded links of the network. The number of times a link e is measured equals the number of monitoring paths that cross e, that is $\sum_{p \in \mathcal{P}} \delta_{pe} Z_p$. The probe cost reads as follows:

$$\sum_{e \in \mathcal{E}, p \in \mathcal{P}} C_e \delta_{pe} Z_p \tag{2.3}$$

The objective of our anomaly detection scheme is to find an anomaly detection solution that achieves the best trade-off between these three costs. To this end, we propose two ILP formulations that minimize the three costs jointly. Let α, β and γ be the weights associated to the infrastructure cost, the communication cost, and the probe cost, respectively. The objective functions of the ILPs minimize the total anomaly detection cost that reads as follows:

$$\alpha \, C_{infra} \sum_{n \in \mathcal{N}} Y_n + \beta \sum_{n \in \mathcal{N}} C_n Y_n + \gamma \sum_{e \in \mathcal{E}, p \in \mathcal{P}} C_e \delta_{pe} Z_p \tag{2.4}$$

2.5 Path-based ILP Formulation

This ILP takes as inputs the set of the network links $\mathcal{E}$, a set of links that are to be covered $\mathcal{E}'$ ($\mathcal{E} = \mathcal{E}'$ because we want to cover all the network links), a set of candidate monitor locations $\mathcal{N}$, and a set of candidate monitoring paths $\mathcal{P}$. The problem can be reduced to covering a subset of the network links. It also takes as inputs a set a binary

1. We would suggest that C_e be proportional to the nominal bandwidth of link e, because the load of links can hardly be predicted since it is prone to the variations of the network load.

parameters $\delta_{\mathcal{PE}} = \{\delta_{pe};\ \ \forall p \in \mathcal{P}, e \in \mathcal{E}\}$, where δ_{pe} indicates whether path p covers link e ; a set of binary parameters $\delta_{\mathcal{PN}} = \{\delta_{pn};\ \ p \in \mathcal{P}, n \in \mathcal{N}\}$, where δ_{pn} indicates whether node n is an end node of path p; the link measurement costs $C_e, \forall e \in \mathcal{E}$; the infrastructure cost C_{infra}; and the communication costs C_n, $\forall n \in \mathcal{N}$. For simplicity of notation we define the sets $C_{\mathcal{E}} = \{C_e;\ \ e \in \mathcal{E}\}$ and $C_{\mathcal{N}} = \{C_n;\ \ e \in \mathcal{N}\}$. The input into the ILP can be written as ($\mathcal{E}$, $\mathcal{E}^{'}$, $\mathcal{N}$, $\mathcal{P}$, $\delta_{\mathcal{PE}}$, $\delta_{\mathcal{PN}}$, $C_{\mathcal{E}}$, $C_{\mathcal{N}}$, C_{infra}, α, β, γ).

The objective function minimizes the total detection cost as given by (2.4). The outputs are a set of monitor locations where to deploy monitoring devices and a set of paths that are to be monitored. The ILP is subject to the following constraints:

– Full coverage constraints: these constraints ensure that each link of $\mathcal{E}^{'}$ [2] is covered by at least one monitoring path.

$$\sum_{p \in \mathcal{P}} \delta_{pe} Z_p \geq 1;\ \ \forall e \in \mathcal{E}^{'} \tag{2.5}$$

– Monitor location constraints: These constraints ensure that the either end nodes of each selected monitoring path is selected as a monitor location.

$$Y_n \geq \delta_{np} Zp;\ \ \forall n \in \mathcal{N}, \forall p \in \mathcal{P} \tag{2.6}$$

2.6 Link-Flow-Based ILP Formulation

Clearly, the path-based ILP formulation requires high memory capacity for pre-computing and processing the network paths and the input parameters. In an attempt to overcome this limitation, we propose a link-flow-based ILP formulation. Like the path-based ILP, this ILP minimizes the total anomaly detection cost under the same full coverage and monitor constraints. However, it takes only the network graph as input. A flow is a sequence of directed links that are crossed by a monitoring flow. We use directed links in order to formulate the flow conservation constraints described in the sequel. However, a link needs to be covered only in one direction to enable anomaly detection. Let $\mathcal{A} = \{(i \rightarrow j), (j \rightarrow i);\ \ \forall (i,j) \in \mathcal{E}\}$ be the set of directed links constructed out of $\mathcal{E}$. Let $C_{(i \rightarrow j)}$ denotes the cost of monitoring the directed link $(i \rightarrow j)$. We have $C_{(i \rightarrow j)} = C_{(j \rightarrow i)} = C_{(i,j)}, \forall (i,j) \in \mathcal{E}$. The flows are modeled using a set of binary variables $\{X_{i \rightarrow j}(n, n');\ \ \forall (i \rightarrow j) \in \mathcal{A}; \forall n, n' \in \mathcal{N}\}$, each variable $X_{i \rightarrow j}(n, n')$ expresses whether the flow traveling between the pair of nodes (n, n') and crossing the directed link $(i \rightarrow j)$ is

2. $\mathcal{E}^{'} = \mathcal{E}$ for a full coverage of the network links.

part of the detection solution. The total anomaly detection cost is expressed as follows using the new variables:

$$\alpha\, C_{infra} \sum_{n \in \mathcal{N}} Y_n + \beta \sum_{n \in \mathcal{N}} C_n Y_n + \gamma \sum_{(i,j) \in \mathcal{E}; n,n' \in \mathcal{N}} C_{(i,j)} [X_{i \to j}(n,n') + X_{j \to i}(n,n')] \qquad (2.7)$$

The ILP is subject to the following constraints:

1. Full coverage constraints:

$$\sum_{n,n' \in \mathcal{N}} X_{i \to j}(n,n') + X_{j \to i}(n,n') \geq 1; \quad \forall (i,j) \in \mathcal{E}^{'} \qquad (2.8)$$

2. Flow conservation constraints: multiple monitoring flows might be carried between a pair of nodes [3]. We define a set of integer variables $\{W_{(n,n')}; n, n' \in \mathcal{N}\}$. $W_{(n,n')}$ quantifies the number of monitoring flows that starts from node n and ends at node n'. Let $IN(v)$ and $OUT(v)$ be the set of directed links entering node v and the set of directed links leaving node v, respectively. The flow conservation constraints [4] are, hence, expressed as follows:

$$\sum_{i \to j \in OUT(v)} X_{i \to j}(n,n') - \sum_{i \to j \in IN(v)} X_{i \to j}(n,n') = \begin{cases} W_{(n,n')}, & \text{iff } v = n \\ -W_{(n,n')}, & \text{iff } v = n' \\ 0, & otherwise \end{cases} ; \quad \forall v, n, n' \in \mathcal{N} \qquad (2.9)$$

3. Monitor location constraints:

$$K Y_n \geq \sum_{n' \in N} (W_{(n,n')} + W_{(n',n)}); \quad \forall n \in \mathcal{N}, K > \mid \mathcal{N} \mid ! \qquad (2.10)$$

The above constraints state that Y_n, $\forall n \in \mathcal{N}$, equals 1 iff at least one monitoring flow starts or ends at node n, otherwise Y_n equals 0.

4. Loop-free constraints: toward preventing looping flows, we define a set of integer variables $\{H_{(n,n')}(i); n, n', i \in \mathcal{N}\}$. $H_{(n,n')}(i)$ specifies the number of hops separating node i visited by a flow traveling between the pair of nodes (n, n') from its originating node n. The idea is to force flows to travel through nodes in an ascending order of

3. In this case, the monitoring flows have the same end nodes, but they are carried by different paths

4. The flow that enters a node leaves it except if it is the originating node (in which case the flow only exits), or the terminating node (in which case the flow only enters)

the values of their hop variables, which prevents them from looping. The loop-free constraints can be expressed as follows:

$$H_{(n,n')}(n) = 0; \quad \forall n, n' \in \mathcal{N} \tag{2.11}$$

$$\begin{array}{l} 1 - X_{i\to j}(n,n') + \frac{H_{(n,n')}(j) - 1 - H_{(n,n')}(i)}{K} \geq 0 \\ 1 - X_{j\to i}(n,n') + \frac{H_{(n,n')}(i) - 1 - H_{(n,n')}(j)}{K} \geq 0 \end{array} \quad ; \forall (i,j) \in \mathcal{E}; n, n' \in \mathcal{N}, K > \mid \mathcal{N} \mid ! \tag{2.12}$$

$$H_{(n,n')}(n') \leq |\mathcal{N}| - 1; \quad \forall n, n' \in \mathcal{N} \tag{2.13}$$

Constraints (2.11) assign the value 0 to the hop variable of the originating node of each path, whereas constraints (2.13) set the upper bound of the flow lengths to the number of network nodes. Constraints (2.12) guarantee that flows do not re-visit an already visited node, *i.e.*, a node having a value of hop variable lower than the values of those of visited the nodes.

2.7 The Anomaly Detection Problem is $\mathcal{NP}$-Hard

The anomaly detection problem can be reduced from the $\mathcal{NP}$-Hard facility location problem.

Facility location problem [30]: consider a set of potential facility locations $\mathcal{F}$, and a set of clients $\mathcal{D}$. Opening a facility at location i incurs a non-negative cost that is equal to f_i. The cost of servicing client $j \in \mathcal{D}$ by a facility installed at location $i \in \mathcal{F}$ is d_{ij}. The problem is to find an assignment of each client to exactly one facility such that the sum of the facility opening costs and the service costs is minimized.

We denote by f the set of facility opening costs, $f = \{f_i,\ i \in \mathcal{F}\}$, and we denote by d the set of service costs, $d = \{d_{ij};\ i \in \mathcal{F}, j \in \mathcal{D}\}$. Given an instance $\mathcal{I} = (\mathcal{D}, \mathcal{F}, f, d)$ of the facility location problem, we produce an instance $\mathcal{R}(\mathcal{I}) = (\mathcal{E}, \mathcal{E}', \mathcal{M}, \mathcal{P}', \delta_{\mathcal{PE}}, \delta_{\mathcal{PM}}, C_{\mathcal{E}}, C_{\mathcal{M}}, C_{infra}, \alpha, \beta, \gamma)$ of our path-based formulation of the anomaly detection problem as follows. For each facility location $i \in \mathcal{F}$, we create two nodes labeled by m_{i1} and m_{i2}. For each client $j \in \mathcal{D}$, we create two nodes labeled by n_{j1} and n_{j2} and one undirected link connecting n_{j1} to n_{j2} and labeled by e_j. For each $i \in \mathcal{F}$ and for each $j \in \mathcal{D}$, we create:

- One undirected link connecting m_{i1} to n_{j1}, labeled by e^1_{ij}
- One undirected link connecting m_{i2} to n_{j2}, labeled by e^2_{ij}

We obtain a graph $\mathcal{G} = (\mathcal{E}, \mathcal{N})$, where $\mathcal{N} = \{m_{ik}; i \in \mathcal{F}, k \in [1;2]\} \cup \{n_{jk}; j \in \mathcal{D}, k \in [1;2]\}$, and $\mathcal{E} = \{e^k_{ij}; i,j \in \mathcal{F} \times \mathcal{D}, k \in [1;2]\} \cup \{e_j; j \in \mathcal{F}\}$. An example of a graph constructed out of a facility location instance with three facility locations and four clients is shown in Fig. 4.2.

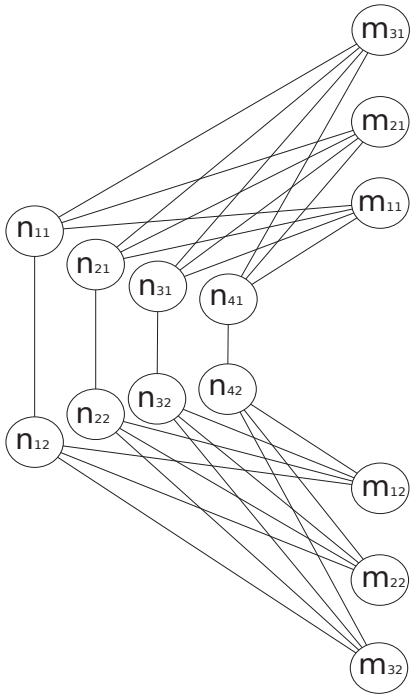


Figure 2.2: Example of a graph constructed out of a facility location instance with three facility locations and four clients

We define the set of candidate monitor locations as $\mathcal{M} = \{m_{ik}; i \in \mathcal{F}, k \in [1;2]\}$, and we define the set of links that are to be covered as $\mathcal{E}' = \{e_j; j \in \mathcal{D}\}$. The set of candidate monitoring paths is defined as $\mathcal{P}' = \{p_{ij}; i \in \mathcal{F}, j \in \mathcal{D}\}$, where p_{ij} is the non-looping path between nodes m_{i1} and m_{i2} that crosses links e_j, e^1_{ij}, and e^2_{ij}. The link measurement costs, the communication costs, and the infrastructure cost are defined are defined as follows:

- $C_{infra} + C_{m_{ik}} = f_i/2$; $\forall i \in \mathcal{F}, k \in [1;2]$
- $C_{e_j} = 0$ and $C_{e^1_{ij}} = C_{e^2_{ij}} = d_{ij}/2$; $\forall i \in \mathcal{F}, \forall j \in \mathcal{D}$

The remaining input parameters to the anomaly detection problem are defined as follows:

- $\delta_{p_{ij}e^k_{i'j'}} = \begin{cases} 1 & \text{if } i = i' \text{ and } j = j' \\ 0 & \text{otherwise} \end{cases}$; $\forall j, j' \in \mathcal{D}, k \in [1;2]$
- $\delta_{p_{ij}e'_j} = 1$ if $j = j'$, 0 otherwise; $\forall i \in \mathcal{F}, \forall j, j' \in \mathcal{D}$

– $\delta_{n_{ik}p_{i'j'}} = \begin{cases} 1 & \text{if } i = i' \\ 0 & \text{otherwise} \end{cases} \quad ; \quad \forall j, j^{'} \in \mathcal{D}, k \in [1;2]$

– $\alpha = \beta = \gamma = 1$

The above reduction has $\mid \mathcal{F} \mid \times \mid \mathcal{D} \mid$ of time complexity, and therefore, it can be carried out in polynomial-time. We now show that there is a solution to the instance $\mathcal{I}$ of the facility location problem if and only if there is a solution to the instance $\mathcal{R}(\mathcal{I})$ of the anomaly detection problem.

We first demonstrate that if there is an optimal solution to $\mathcal{I}$, then, there is a feasible solution to $\mathcal{R}(\mathcal{I})$. Let S_I^* be an optimal solution to $\mathcal{I}$ that assigns each client j to a facility installed at location i. Consider the anomaly detection solution $S_{\mathcal{R}(I)}$ that selects the set of paths $\mathcal{D}_p = \{p_{ij} : S_I^* \text{ assigns } j \text{ to } i; i \in \mathcal{F}, j \in \mathcal{D}\}$, and selects the set of monitor locations $\mathcal{D}_m = \{m_{ik} : \exists j \text{ such that } p_{ij} \in \mathcal{D}_p; i \in \mathcal{F}, i \in \mathcal{D}, k \in [1;2]\}$. Recall that a feasible anomaly detection solution must satisfy the coverage constraint, *i.e.*, selecting a set of monitoring paths that cover all links of the input link set; and must satisfy the monitor location constraint, *i.e.*, selecting the end nodes of each selected monitoring path as monitor locations. Clearly, $\mathcal{D}_p$ covers all links of $\mathcal{E}^{'}$, and $\mathcal{D}_m$ contains the end nodes of all paths of $\mathcal{D}_p$. It follows that $S_{\mathcal{R}(I)}$ is a feasible solution to $\mathcal{R}(\mathcal{I})$.

Conversely, we demonstrate that if there is an optimal solution to $\mathcal{R}(\mathcal{I})$, then, there is a feasible solution to $\mathcal{I}$. Let $S_{\mathcal{R}(I)}^*$ be an optimal solution to $\mathcal{I}$. Let us fix a link e_j of $\mathcal{E}^{'}$. We show by contradiction that any optimal solution to $\mathcal{I}$ selects only one path that crosses e_j. Assume to the contrary that there is an optimal solution whose set of monitoring paths $\mathcal{D}_p^*$ contains two paths p_1 and p_2 each of them crossing e_j. Consider the solution to $\mathcal{I}$ whose set of monitoring paths equals $\mathcal{D}_p^* \setminus \{p_1\}$, and whose set of monitor locations is the same as for the optimal solution. This solution is feasible since it covers all links of $\mathcal{E}^{'}$. Moreover, its cost equals the cost of the optimal solution minus the cost of monitoring p_1. This leads to a contradiction. The facility location solution S_I that assigns each client j to the facility installed at locations i such that $S_{\mathcal{R}(I)}^*$ selects p_{ij} to be monitored is clearly a feasible solution to $\mathcal{I}$.

We now show that the cost of $S_{\mathcal{R}(I)}$ equals the cost of S_I^* (the proof that the cost of S_I equals the cost of $S_{\mathcal{R}(I)}^*$ is similar). Let F_i be a binary variable that indicates whether a facility is installed at location i, and let D_{ij} be a binary variable that indicates whether client j is serviced by a facility installed at location i. As explained above, $S_{\mathcal{R}(I)}$ is constructed such that $D_{ij}^* = Z_{p_{ij}}$ and $F_i^* = Y_{n_{i1}} = Y_{n_{i2}}$, $\forall i \in \mathcal{F}$ and $\forall j \in \mathcal{D}$. Recall that Z_p is a binary variable that indicates whether path p is selected to be monitored, and Y_n

is a binary variable that indicates whether node n is selected as a monitor location. We have:

$$\begin{aligned} Cost(S_{\mathcal{R}(\mathcal{I})}) &= \sum_{n_{i1}\in\mathcal{M}}(C_{infra}+C_{n_{i1}})Y_{n_{i1}} + \sum_{n_{i2}\in\mathcal{M}}(C_{infra}+C_{n_{i2}})Y_{n_{i2}} + \\ &\quad \sum_{p_{ij}\in\mathcal{P}',\ e\in\mathcal{E}} C_e\delta_{p_{ij}e}Z_{p_{ij}} \\ &= \sum_{i\in\mathcal{F}} f_i/2\ F_i^* + \sum_{i\in\mathcal{F}} f_i/2\ F_i^* + \sum_{p_{ij}\in\mathcal{P}',\ e_{ij}^1\in\mathcal{E}'} C_{e_{ij}^1}\delta_{p_{ij}e_{ij}^1}Z_{p_{ij}} + \\ &\quad \sum_{p_{ij}\in\mathcal{P}',\ e_{ij}^2\in\mathcal{E}'} C_{e_{ij}^2}\delta_{p_{ij}e_{ij}^2}Z_{p_{ij}} \\ &= \sum_{i\in\mathcal{F}} f_i\ F_i^* + \sum_{j\in\mathcal{D},\ i\in\mathcal{F}} d_{ij}\ D_{ij}^* \\ &= Cost(S_{\mathcal{I}}^*) \end{aligned}$$

Finally we demonstrate by contradiction that $S_{\mathcal{R}(I)}$ is an optimal solution to $\mathcal{I}$ (the proof that $S_{\mathcal{I}}$ is an optimal solution to $\mathcal{I}$ is similar). Assume to the contrary that $S_{\mathcal{R}(I)}$ is not an optimal solution. Let $S_{\mathcal{R}(\mathcal{I})}^{'*}$ be an optimal solution to $\mathcal{R}(\mathcal{I})$, and let $S_I^{'}$ be a feasible solution constructed out of $S_{\mathcal{R}(\mathcal{I})}^*$. We have $Cost(S_I^*) = Cost(S_{\mathcal{R}(I)}) < Cost(S_{\mathcal{R}(I)}^{'*}) = Cost(S_I^{'})$, leading to a contradiction. Thus, $S_{\mathcal{R}(I)}$ is an optimal solution to $\mathcal{R}(\mathcal{I})$.

2.8 Heuristic Algorithms for joint monitor location and monitoring path selection

In this section, we provide two greedy algorithms using the monitoring cost model introduced in section 2.4. The aim of the algorithms is to find a set of monitor locations and a set of monitoring paths that cover all links of the network, while minimizing jointly the monitor cost, the communication cost, and the probe cost. The first algorithm is based on an exhaustive heuristic that explores all the network paths; whereas the second algorithm is based on selective heuristics that address scalability issues by reducing the number of explored paths. The challenge is to achieve scalability without negatively impacting the quality of the detection solution.

2.8.1 Exhaustive greedy algorithm

Algorithm 1 describes the pseudo-code of the exhaustive algorithm. The algorithm proceeds as follows. Monitor locations and monitoring paths are selected greedily. At each greedy iteration, one monitor location whose communication cost is the smallest is added to the solution (the tie is broken randomly) (line 5). Then, all candidate monitoring paths between the added monitor location and the already selected monitor locations are explored. One path that maximizes the coverage capacity is selected. In case of a tie, a path

Table 2.1: Notations used in the pseudo-codes

Symbol	**Definition**
CP	The set of candidate paths
$\mathcal{CM}$	The set of candidate monitor locations
SP	The set of permanently selected paths
TSP	The set of temporarily selected paths
BP	The set of paths temporarily selected at the previous iteration
SM	The set of selected monitors
$nbCL$	The number of links covered by paths in SP
$nbTCL$	The number of links covered by paths in TSP

that minimizes the incurred probe cost is selected. Further tie is broken randomly (lines 11, 12). This process of selecting one candidate monitoring path is re-iterated until all links are covered or remaining paths cannot cover links still uncovered. Selected monitoring paths that generate redundant measurements, *i.e.*, monitoring paths that cross already covered links, are labeled. They are temporarily stored in $\mathcal{TSP}$. Selected paths that do not generate redundant measurements are permanently stored in $\mathcal{SP}$. By the end of each greedy iteration, the obtained solution is evaluated (line 25). A new greedy iteration is executed if the set of candidate monitor locations $\mathcal{CM}$ is not yet empty, and if one of the following conditions is satisfied:

- The obtained solution cannot cover all links of the network.
- The weighted cost of redundant measurements of the obtained solution is larger then the weighted cost of deploying a new monitoring device. The algorithm attempts to reduce redundant measurements by deploying an additional monitoring device, which is likely to reduce the detection cost. The aim is to achieve a good balance between the detection infrastructure cost and the detection overhead.

Labeled monitoring paths are removed from the solution and stored in $\mathcal{BP}$. They are injected into the set of candidate paths that is examined at the next greedy iteration (line 6). The rational of this operation is to avoid re-exploring all candidate paths between already selected monitor locations. If none of the above conditions is satisfied, the algorithm returns the current solution (line 26).

Note that selecting the monitor location with the smallest communication cost does not necessarily lead to finding a set of monitoring paths that covers the maximum number

Algorithm 1: Exhaustive greedy algorithm for anomaly detection

Input : $\mathcal{G} = (\mathcal{E}, \mathcal{N})$, $\mathcal{P}$, $\mathcal{C}_{\mathcal{N}}$, $\mathcal{C}_{\mathcal{E}}$, C_{infra}, $\mathcal{CM}$

Output : A set of monitor locations and a set of monitoring paths that can cover all links in $\mathcal{E}$

1 $\mathcal{SP} \leftarrow \emptyset$, $\mathcal{BP} \leftarrow \emptyset$, $\mathcal{SM} \leftarrow \emptyset$;
2 $m \leftarrow selectRandomElement(arg \min_{n \in \mathcal{CM}} C_n)$;
3 $\mathcal{SM} \leftarrow \mathcal{SM} \cup \{m\}$, $\mathcal{CM} \leftarrow \mathcal{CM} \setminus \{m\}$;
4 **while** *(true)* **do**
5 $m \leftarrow selectRandomElement(arg \min_{n \in \mathcal{CM}} C_n)$;
6 $\mathcal{CP} \leftarrow \{$all paths of $\mathcal{P}$ whose end nodes are in $\mathcal{SM} \times \{m\}\} \cup \mathcal{BP}$;
7 $\mathcal{SM} \leftarrow \mathcal{SM} \cup \{m\}$, $\mathcal{CM} \leftarrow \mathcal{CM} \setminus \{m\}$;
8 $\mathcal{TSP} \leftarrow \emptyset$;
9 $nbTCL \leftarrow 0$, $ProbeCost \leftarrow 0$;
10 **while** $(nbCL + nbTCL <\mid \mathcal{E} \mid$ or $\mathcal{CP} \neq \emptyset)$ **do**
11 $\mathcal{Q} \leftarrow arg \max_{q \in \mathcal{CP}} coverage_capacity(q, \mathcal{SP} \cup \mathcal{TSP})$;
12 $p \leftarrow selectRandomElement(arg \min_{q \in \mathcal{Q}} probe_cost(q))$;
13 **if** $(coverage_capacity(p, \mathcal{SP} \cup \mathcal{TSP}) = 0)$ **then**
14 $\mathcal{CP} \leftarrow \emptyset$;
15 **else**
16 **if** $(\mid p \mid - coverage_capacity(p, \mathcal{SP} \cup \mathcal{TSP}) = 0)$ **then**
17 $nbCL \mathrel{+}= coverage_capacity(p, \mathcal{SP} \cup \mathcal{TSP})$;
18 $\mathcal{SP} \leftarrow \mathcal{SP} \cup \{p\}$;
19 **else**
20 $nbTCL \mathrel{+}= coverage_capacity(p, \mathcal{SP} \cup \mathcal{TSP})$;
21 $\mathcal{TSP} \leftarrow \mathcal{TSP} \cup \{p\}$;
22 $ProbeCost \mathrel{+}= ProbeCost(p)$;
23 $\mathcal{CP} \leftarrow \mathcal{CP} \setminus \{p\}$;
24 $\mathcal{BP} \leftarrow \mathcal{TSP}$;
25 **if**
$(\mathcal{CM} = \emptyset$ or $(nbCL + nbTCL =\mid \mathcal{E} \mid$ and $\gamma(ProbeCost - \sum_{e \in \mathcal{E}} C_e) \leq \alpha\, C_{infra} + \beta \max_{n \in \mathcal{CM}} C_n))$
then Go to line 26;
26 **return** $(\mathcal{SP} \cup \mathcal{TSP}, \mathcal{SM})$;

of links and minimizes the probe cost. Ideally, all remaining candidate monitor locations should be explored at each greedy iteration. The monitor location whose associated monitoring paths achieve the largest coverage capacity, while achieving the best balance between

the probe cost and the communication cost should be selected. However, this operation is very expensive, mainly because the exhaustive heuristic explores all paths of the network between the currently explored monitor location and the already selected monitor locations. One might suggest not considering all paths of the network as candidate to be monitored. This requires finding an efficient heuristic to compute candidate paths such that the quality of the detection solution is not degraded. This is the aim of the selective greedy algorithm.

2.8.2 Selective Greedy Algorithm

The purpose of this section is to provide a scalable heuristic solution for joint monitor location and monitoring path selection. As discussed above, the quest for scalability should not degrade the quality of the detection solution. Our heuristic is described in Algorithm 2. Similarly to the exhaustive greedy algorithm, this is a greedy algorithm that selects monitor locations and monitoring paths greedily. However, it explores more solutions than the exhaustive algorithm. This is possible due to the use of the candidate monitoring path computation heuristic described in Procedure 1 that reduces drastically the time of exploring candidate monitoring paths.

We now describe the heuristics. Let us fix two candidate monitor locations n_1 and n_2. First, the algorithm computes greedily a set of non-overlapping paths between n_1 and n_2 (lines 4-8 of Algorithm 2). Ideally, one path between n_1 and n_2 that maximizes the coverage capacity, *i.e.*, crosses the maximum number of links still uncovered, should be selected at a time.

As discussed earlier, pre-computing the set of all candidate paths leads to serious scalability issues, and reducing the number of candidate paths arbitrarily leads to quality degradation. To overcome these limitations, we propose to compute a satisfactory path by exploring the network graph selectively as follows. The network graph is explored in an in-depth first order starting from one of the candidate monitor locations, say n_1. If the link connecting the currently explored node to the last explored node is already covered, then the exploration of all the descendants of the currently explored node is abandoned, which means that all the network paths having as prefix the current path will not be explored. If the currently explored node is n_2 and the coverage capacity of the current path dominates the coverage capacity of the best path then the latter path is set equal to the former path. This way we compute a satisfactory path without memorizing any candidate paths.

Algorithm 2: Selective Greedy Algorithm

Input : $\mathcal{G} = (\mathcal{E}, \mathcal{N})$, $\mathcal{P}$, $\mathcal{C}_{\mathcal{N}}$, $\mathcal{C}_{\mathcal{E}}$, C_{infra}, $\mathcal{CM}$ (set of candidate monitor locations)

Output : A set of monitor locations and a set of monitoring paths that can cover all links in $\mathcal{E}$

1 $\mathcal{SM}^* \leftarrow \emptyset, \mathcal{SP}^* \leftarrow \emptyset$;
2 **foreach** $n_1, n_2 \in N$ **do**
3 $\mathcal{SP} \leftarrow \emptyset$, $\mathcal{CL} \leftarrow \emptyset$, $\mathcal{SM} \leftarrow \{n_1, n_2\}$, $ProbeCost \leftarrow 0$;
4 **while** () **do**
5 $p \leftarrow candidatePathComputation(\mathcal{G}, n_1, n_2, \mathcal{CL})$;
6 **if** $(p = Null)$ **then** Go to line 9;
7 $\mathcal{SP} \leftarrow \mathcal{SP} \cup \{p\}$, $\mathcal{CL} \leftarrow \mathcal{CL} \cup \{e \in \mathcal{E} : \delta_{ep} = 1\}$;
8 $ProbeCost \mathrel{+}= probe_cost(p)$;
9 **while** $(\mid \mathcal{CL} \mid < \mid \mathcal{E} \mid)$ **do**
10 $\mathcal{Q} \leftarrow$ {paths composed only of links in $\mathcal{E} \setminus \mathcal{CL}$};
11 $p \leftarrow selectRandomElement(arg \max_{q \in \mathcal{Q}} \mid q \mid)$;
12 $(n_i, n_j) \leftarrow$ the end nodes of p;
13 **if** $(n_i \notin \mathcal{SM})$ **then**
14 $p_1 \leftarrow Dijkstra(\mathcal{G}, n_i, \mathcal{SM}, C_{\mathcal{E}})$;
15 **if** $(\alpha C_{infra} + \beta C_{n_i} > \gamma probe_cost(p_1))$ **then**
16 $p_1 \leftarrow Null$, $\mathcal{SM} \leftarrow \mathcal{SM} \cup \{n_i\}$;
17 **if** $(n_j \notin \mathcal{SM})$ **then**
18 $p_2 \leftarrow Dijkstra(\mathcal{G}, n_j, \mathcal{SM}, C_{\mathcal{E}})$;
19 **if** $(\alpha C_{infra} + \beta C_{n_j} > \gamma probe_cost(p_2))$ **then**
20 $p_2 \leftarrow Null$, $\mathcal{SM} \leftarrow \mathcal{SM} \cup \{n_j\}$;
21 $q \leftarrow concatenate(p_1, p, p_2)$; $ProbeCost \mathrel{+}= probe_cost(q)$;
22 $\mathcal{SP} \leftarrow \mathcal{SP} \cup \{q\}$, $\mathcal{CL} \leftarrow \mathcal{CL} \cup \{e \in \mathcal{E} : \delta_{eq} = 1\}$;
23 **if** $(\beta \sum_{m \in \mathcal{SM}} C_m + \gamma \sum_{e \in \mathcal{E}, p \in \mathcal{SP}} \delta_{ep} C_e < \beta \sum_{m \in \mathcal{SM}^*} C_m + \gamma \sum_{e \in \mathcal{E}, p \in \mathcal{SP}^*} \delta_{ep} C_e)$ **then**
24 $\mathcal{SM}^* \leftarrow \mathcal{SM}, \mathcal{SP}^* \leftarrow \mathcal{SP}$;
25 **return** $(\mathcal{SM}^*, \mathcal{SP}^*)$;

Clearly, the computation becomes faster and easier as the number of covered links increases. However, depending on the network density, the computation can be very complex and expensive in terms of time when the proportion of covered links is still small. Namely, when the set of covered links is empty the problem is reduced to finding the longest path between two nodes which is an $\mathcal{NP}$-Complete problem.

In order to avoid intractable computations, the algorithm proceeds as follows. If the number of covered links is smaller than 50% of the network links, then the network graph is explored randomly 10 times (lines 13-27 of Procedure 1). Each time, the exploration ends up as soon as one path between n_1 and n_2 has been found. Then the path whose coverage capacity is the largest among the 10 paths is selected. If the number of covered links is larger than or equal to 50% of the network links, then the candidate path is computed by exploring the network graph selectively as described above (lines 3-13 of Procedure 1). This configuration parameters have been decided after evaluating many configuration parameters through simulations. We found that it offers a good trade-off between the computation time and the solution quality.

Note that the set of non-overlapping paths minimizes the probe cost since it avoids redundant measurements by avoiding overlaps among its paths. The aim is to cover the largest number of links without generating redundant measurements. Upon selecting the set of non-overlapping paths, the algorithm re-iterates the following operations until all links of the network are covered. It computes the longest path composed of only uncovered links. Let n_i and n_j be the end nodes of that path. Clearly, we have (($n_i \notin \mathcal{SM}$ or $n_j \notin \mathcal{SM}$) or ($n_i \notin \mathcal{SM}$ and $n_j \notin \mathcal{SM}$)). The monitor location constraint requires that the end nodes of any monitoring path must be selected as monitor locations. There are two alternatives to satisfy this constraint. The first is to select each end node that does not satisfy this constraint as a monitor location. The second is to compute for each node not satisfying the constraint one path connecting it to one node in $\mathcal{SM}$. This path must minimize the probe cost. It is computed using the algorithm of Dijkstra. The decision of deploying one of the two alternatives is made by comparing their costs (lines 15, 19 of Algorithm 2).

The above computations are made for all the pairs of candidate monitor locations. Then, the algorithm returns the solution that achieves the smallest detection cost (lines 23, 24, 25 of Algorithm 2).

2.9 Evaluation

We evaluate our ILPs and our heuristics through extensive simulations running on a PC equipped with an Intel Core 2 Duo processor, a clock rate of 2,992.47 MHz, and 3.9 GB of RAM. The ILPs are solved using Cplex11.2 [12], and the heuristics are implemented using C_{++}. All results are the mean over 30 simulations on random topologies generated using

Procedure 1: candidatePathComputation($\mathcal{G}$, n_1, n_2, $\mathcal{CL}$)

```
1  p_c ← newPath(); p_s ← Null;
2  add-node-to-path(m, p_c);
3  if (| CL |≥| E | /2) then
4      depthFirst (m, p_c, G, CL){
5          foreach (n ∈ children(n_1, G) : n ∉ p_c and (m, n) ∉ CL) do
6              add-node-to-path(n, p_c);
7              if (n = n_2) then
8                  if coverage_capacity(p_c, CL) > coverage_capacity(p_s, CL)) then
9                      p_s = p_c;
10                     if (| p_s |=| CL |) then return p_s;
11             else Recursively call depthFirst (n, p_c, G, CL);
12     }
13 else
14     k ← 1;
15     repeat
16         randomDepthFirst (m, p_c, G, CL){
17             n ← randomChild(n_1, G) : n ∉ p_c and (m, n) ∉ CL;
18             if (n = Null) then Go to line 27;
19             else
20                 add-node-to-path(n, p_c);
21                 if (n = n_2) then
22                     if (coverage_capacity(p_c, CL) > coverage_capacity(p_s, CL)) then
23                         p_s = p_c;
24                     k++; Go to line 27;
25                 else Recursively call randomDepthFirst (n, p_c, G, CL);
26         }
27     until k ≤ 10;
28 return p_s;
```

the topology generator BRITE [13] [33] (Waxman model [31]: $\alpha = \beta = 0.4$, random node placement [5]). Our experiments indicate that the results are almost the same for larger number of simulations. Table 4.3 depicts a summary of the topologies considered. We devised an algorithm that computes the set of all non-looping paths in a given network

5. These parameters are not to be confused with the infrastructure cost weight (α) and the communication cost weight (β) introduced in Section 2.4. Their values equal the values used by Waxman to generate network topologies [31].

topology. It was infeasible to store the path sets for topologies with more than 12 nodes and 41 links due to memory insufficiency. In all the simulations, we assume that the NOC is equidistant from all the network nodes. Therefore the communication cost is the same for all the candidate monitor locations. We also assume that $C_e = 1\ \forall e \in \mathcal{E}$, $C_n = 1\ \forall n \in \mathcal{N}$, and $C_{infra} = 1$.

Table 2.2: Summary of the topologies considered in the evaluation

Topology	Number of nodes	Number of links	Average number of paths
TOP(6, 10)	6	10	162
TOP(8, 18)	8	18	3.176
TOP(10, 31)	10	31	209.235
TOP(12, 41)	12	41	3.679.756
TOP(15, 59)	15	58	362.919.718
TOP(20, 80)	20	80	135.604.169.577
TOP(30, 120)	30	120	295.438.105.637
TOP(50, 250)	50	250	536.337.473.112

2.9.1 Evaluation of The ILP Formulations

In this section, we illustrate the trade-off between the number and locations of monitoring devices, and the quality of monitoring paths; and we show how well our one-step detection scheme balances efficiently this trade-off. We compare the performance of the path-based ILP with the link-flow-based ILP, and then, we compare our detection scheme with existing detection schemes. Recall that existing detection schemes start, in the first step, by deploying as few monitoring devices as possible such that all links of the network can be covered. The associated ILP can be expressed as follows:

$$\begin{aligned} &\textit{Minimize:} \quad \sum_{n \in \mathcal{N}} Y_n \\ &\textit{Subject to:} \\ &\qquad \sum_{p \in \mathcal{P}} \delta_{ep} Z_p \geq 1; \quad \forall e \in \mathcal{E} \\ &\qquad Y_n \geq \delta_{np} Z_p; \quad \forall n \in \mathcal{N}, \forall p \in \mathcal{P} \end{aligned} \tag{2.14}$$

In a second step, a minimal set of monitoring paths that cover all the network links is selected. The associated ILP reads as follows:

$$\begin{aligned} &\textit{Minimize:} \quad \sum_{p\in\mathcal{P}} Z_p \\ &\textit{Subject to:} \\ &\qquad \sum_{p\in\mathcal{P}} \delta_{ep} Z_p \geq 1; \quad \forall e \in \mathcal{E} \\ &\qquad Y_n \geq \delta_{np} Z_p; \quad \forall n \in \mathcal{N}, \forall p \in \mathcal{P} \end{aligned} \tag{2.15}$$

In the second ILP, Y_n is constant. It indicates whether node n has been selected by the first ILP as a monitor location.

Path-Based ILP vs. Link-Flow-Based ILP: we compare the two ILPs along two metrics: the CPU running time and the gap-to-optimality, *i.e.*, the worst-case optimality gap between the obtained solution and the optimal solution estimated by the solver. We choose to present the gap-to-optimality instead of the values of the objective functions, because for some topologies we could not obtain optimal solutions in tractable time. In such case, we have granted 1000 s of CPU running time. To simplify the study, we assume that $\alpha = \beta = \gamma = 1$. However, in the next section we will variate the values of α, β, and γ in order to investigate the impact of the number of monitors and their locations on the quality of monitoring paths. For TOP(6, 10)) the solver delivered optimal solutions for the two ILPs, whereas we could not obtain solutions for TOP(12, 41) using the path-based formulation due to memory insufficiency.

Table 2.3 depicts the observed performance for these two topologies. As expected, the path-based ILP is several steps faster than the the link-flow-based ILP (0.03 s against 25.5 s for TOP(6, 10)). However, the link-flow-based ILP scales better for large topologies. Indeed, although the running time required to obtain optimal solutions is in the order of days, we could obtain a solution that is only 25% worse than the optimal solution in 1000 s for TOP(12, 41).

These observations are confirmed for average topologies, *i.e.*, TOP(8, 18) and TOP(10, 31). Fig. 2.3(a) and Fig. 2.3(b) plot the gap-to-optimality versus the granted CPU running time for TOP(8, 18) and TOP(10, 31), respectively.

Two-Step vs. One-Step Detection Scheme: in this part we consider only the path-based ILP because, as shown in the previous section, it delivers good solutions in tractable time (optimal solutions for TOP(8, 18) in 10 s, and solutions with a 15% gap-to-optimality in 250 s for TOP(10, 31)). We compare our one-step detection scheme to the existing two-step detection scheme. The aim is to (i) illustrate the trade-off between

Table 2.3: CPU Running Time (CPU) and Gap-To-Optimality (GTO) for TOP(6,10) and TOP(12,41).

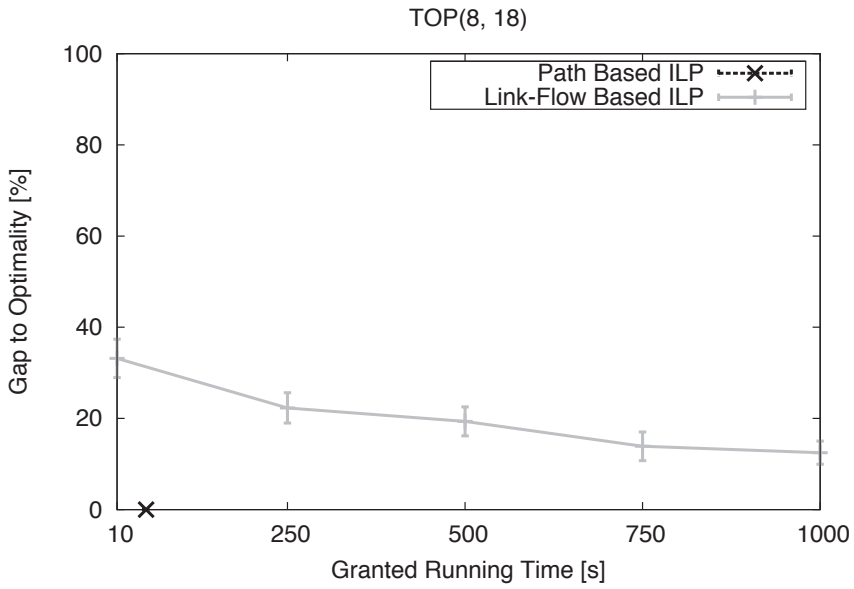
	Path-based ILP		**Link-flow-based ILP**	
Topology	**GTO [%]**	**CPU [s]**	**GTO [%]**	**CPU [s]**
TOP(6, 10)	0	0.03	0	20.5
TOP(12, 41)	Out of Memory		25.01	1000

the optimization objectives of the monitor location step and the optimization objectives of the monitoring path selection step; (ii) demonstrate that the joint optimization of the two objectives balances the trade-off efficiently; (iii) demonstrate that our detection cost model is more appropriate than the existing cost model that expresses the detection cost in terms of the number of deployed monitors and the number of monitoring paths.

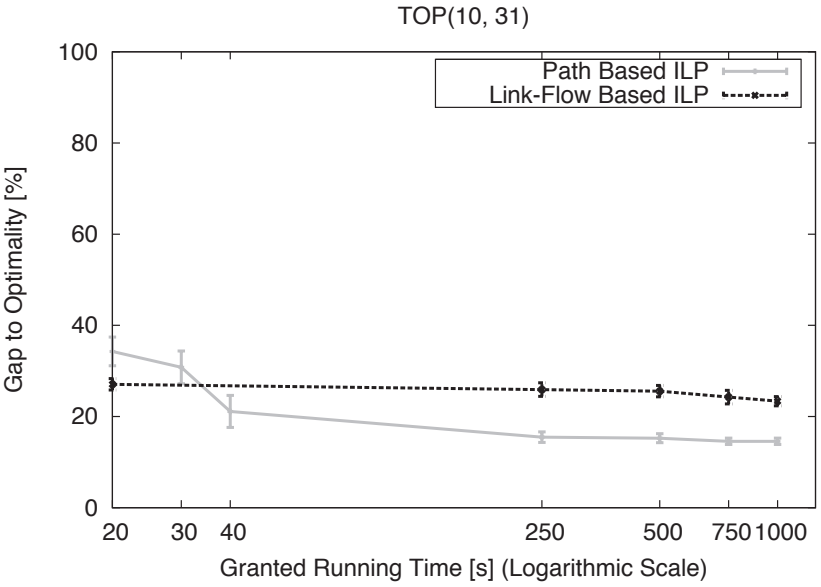
For our detection scheme, we variate the values of α, β and γ. We consider 4 scenarios: $\alpha = \beta = \gamma$, $\alpha = \beta = 2\gamma$, $\alpha = \beta = \gamma/2$ and $\alpha = \beta = \gamma/4$. We report the number of deployed monitors, the number of redundant measurements and the number of monitoring paths for the 4 scenarios and for the existing detection scheme. Fig. 2.4(a) and Fig. 2.4(b) show the results for TOP(8, 18) and TOP(10, 31), respectively. For both topologies, the solutions of the existing detection scheme are optimal solutions that are computed using the ILPs (2.14) and (2.15). Three conclusions can be drawn from Fig. 2.4(a) and Fig. 2.4(b):

1. The number of monitoring paths does not reflect the detection overhead. Consider, for instance, the results for TOP(8, 18) when $\alpha = \gamma/4$ (Fig. 2.4(a)). For this scenario, our detection scheme selects about 43% more paths to be monitored than the existing detection scheme, however, it achieves 100% less redundant measurements. In general, the smaller the set of monitoring paths is, the longer and the more likely to overlap the paths are.

2. The minimization of the number of monitors and the minimization of the detection overhead are two conflicting objectives. Indeed, the figures show that the gap between the number of deployed monitors and the number of redundant measurements increases when the gap between α and γ gets larger. This means that if more importance is given for the minimization of one of the two associated costs, this will intuitively incur the increase of the other cost.



(a)



(b)

Figure 2.3: Gap-to-Optimality vs. Granted Running Time. (a) Results for the topologies with 8 nodes and 18 links; (b) Results for the topologies with 10 nodes and 31 links.

3. The two-step detection scheme delivers sub-optimal solutions with respect to the cost that is minimized in the second step. In effect, although the existing detection scheme minimizes the number of monitoring paths, it could not find the solution delivered by our detection scheme for $\alpha = 2\gamma$ for TOP(10, 31) that monitors about 33% less paths

using the same number of monitors. This demonstrates the impact of the number of monitors and their locations on the quality of the monitoring paths, and validates our assertion that the two-step optimization of conflicting objectives generates sub-optimal solutions. It is worth recalling that for TOP(10, 31) we show optimal solutions for the existing detection scheme and solutions with a 15% gap-to-optimality for our detection scheme.

2.9.2 Evaluation of The Heuristic Algorithms

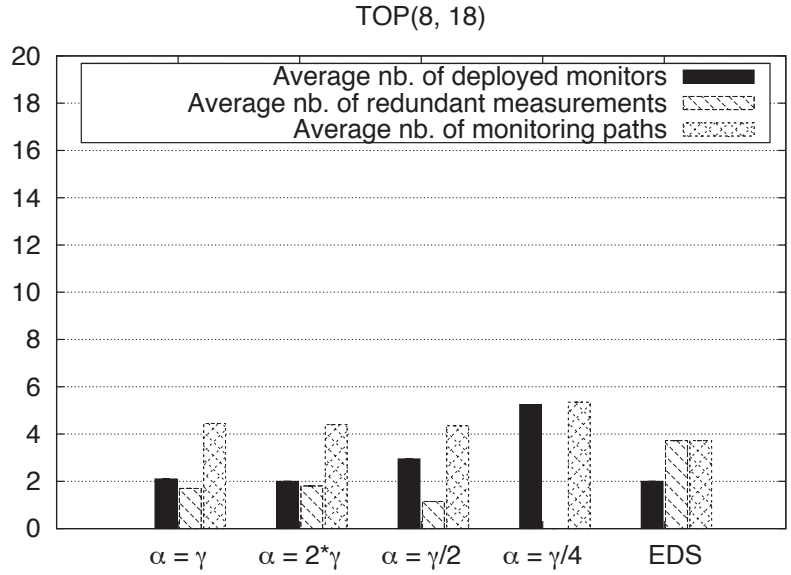
In this section, we investigate the efficiency of our heuristic algorithms. We compare the solutions delivered by the two algorithms with the exact solutions delivered by the path-based ILP, in order to investigate the gap of the greedy solutions to the optimal. Furthermore, we compare the solutions delivered by the two algorithms with the solutions delivered by an LP-assisted exhaustive algorithm. This is an variant of the exhaustive algorithms that takes as input the results of a randomized rounding of the solutions of an LP-relaxation of the path-based ILP. These results constitute a good starting point for the exhaustive greedy algorithm, and reduce the complexity and the computation time of the algorithm since a part of the network links are already covered by the randomized rounding solution.

We refer to the exhaustive greedy algorithm as EGA, and we refer to the selective greedy algorithm as SGA. We assume that $\alpha = \beta = \gamma = 1$.

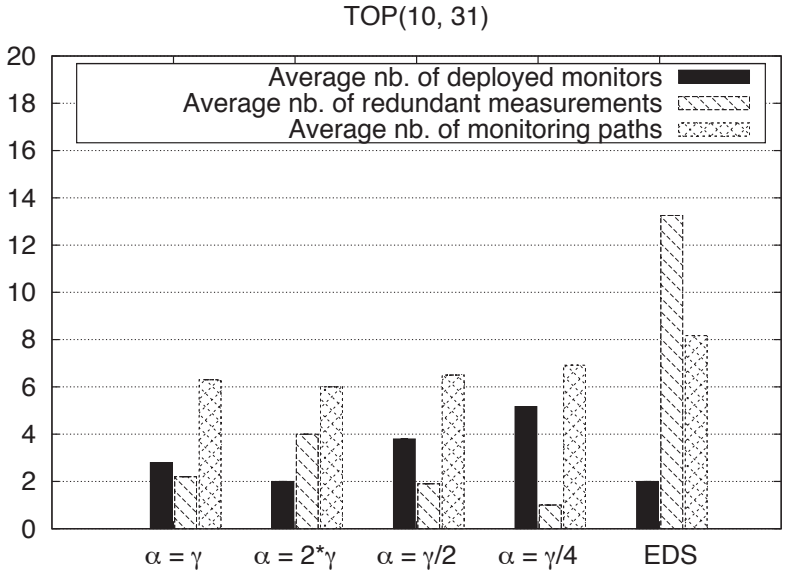
Table 2.4: CPU running time (OOM means Out Of Memory)

Topology	Path-Based ILP	LP-Assisted EGA	EGA	SGA
TOP(6, 10)	0,03 s	0,0035 s	$<$ 1 tic	$<$ 1 tic
TOP(8, 18)	98,3 s	3,75 s	0,02 s	$<$ 1 tic
TOP(10, 31)	-	55242,46 s	3,96 s	0,02 s
TOP(12, 41)	OOM	OOM	OOM	0,02 s
TOP(15, 59)	OOM	OOM	OOM	1,03 s
TOP(20, 80)	OOM	OOM	OOM	4,48 s
TOP(30, 120)	OOM	OOM	OOM	33,11 s
TOP(50, 250)	OOM	OOM	OOM	177, 59 s

TABLE 2.4 depicts the CPU computation time versus the network topology for the five approaches. Results show that the two variants of the exhaustive greedy algorithm



(a)



(b)

Figure 2.4: Performance results for our detection scheme with different values of α, β, and γ ($\beta = \alpha$); and for the existing detection scheme (denoted as EDS). (a) Results for topologies with 8 nodes and 18 links; (b) Results for topologies with 10 nodes and 31 links.

run out of memory for networks with 12 nodes and 41 links and larger. Notice that the computation time of the LP-assisted greedy algorithm increases exponentially. TABLE 2.4 shows that the resolution of exhaustive greedy algorithm takes less than 4 seconds of CPU

time for these topologies. This means that the resolution of the LP takes quite a long time for average topologies and shows serious scalability concerns for large topologies.

As we would expect, the selective greedy algorithm succeeds to overcome the memory limitation, and delivers solutions for all the considered topologies in quite a short time, *e.g.*, 177 seconds for the largest networks. This provides a strong evidence of the scalability of this algorithm.

Table 2.5: Average number of deployed monitors + average number of redundant measurements (network utilization)

Topology	ILP	LP-Assisted EGA	EGA	SGA
TOP(6, 10)	2,7	3,75	3,9	2.8
TOP(8, 18)	3,8	4,35	5,9	4.55
TOP(10, 31)	-	6,11	6,05	4.9
TOP(12, 41)	-	-	7,4	4.9
TOP(15, 59)	-	-	-	5,5
TOP(20, 80)	-	-	-	6.95
TOP(30, 120)	-	-	-	11.95
TOP(50, 250)	-	-	-	20,79

TABLE 2.5 depicts the summation of the number of deployed monitors and the number of redundants measurements of the detection solutions delivered by the four approaches for the eight considered topologies. This metric illustrates the cost gap between the different approaches and shed light on the impact of the heuristics used by the selective algorithm that reduces the number of explored paths.

As expected, the selective algorithm performs better than the exhaustive algorithm, although it does not explore all the network paths. This is because the selective algorithm covers the maximum number of the network links using only two monitors and without generating redundant measurements. Besides, it explores all possible starting points, thereby increasing the number of explored solutions. Furthermore, we observe that the gap between the exact solutions and the solutions of the selective algorithm is quite small for small topologies, which suggests that the heuristics used in the selective algorithm are reasonably accurate.

We define the resource utilization as the proportion of the network nodes and links used for anomaly detection. It reads as follows:

Resource utilization $= 100\ \frac{\sum_{l\in E, p\in P}\delta_{lp}Z_p - |E| + \sum_{n\in N} Y_n}{|N|+|E|}$.

Table 2.6: Resource utilization for SGA

Topology	Resource utilization
TOP(6, 10)	17.50%
TOP(8, 18)	17.50%
TOP(10, 31)	11.95%
TOP(12, 41)	9.24%
TOP(15, 59)	7.43%
TOP(20, 80)	6.95%
TOP(30, 120)	7.96%
TOP(50, 250)	6.93%

Table 2.6 shows the resource utilization values for the selective greedy algorithm. This metric is an alternative representation of the the results shown in table 2.5 that aims at evaluating the performance of the selective greedy algorithm for large topologies. Results show that less than 10% of the network resources are used to detect anomalies in networks with 59 links and larger. This provides a confirmation of the capacity of the selective greedy algorithm to find low-cost detection solutions for large networks.

2.10 Conclusion

In this chapter, we considered the problem of link-level network anomaly detection. We proposed a novel detection cost model, and devised a one-step detection scheme. Unlike existing two-step detection schemes, the one-step detection scheme selects monitor locations and monitoring paths in one step, thereby reducing the trade-off between the number and locations of monitors and the quality of monitoring paths. We provided two ILP formulations for computing a set of monitor locations and a set of monitoring paths that cover all links of the network, while minimizing the associated costs jointly.

We demonstrated that the problem is $\mathcal{NP}$-Hard, and consequently, we proposed two heuristic algorithms, exhaustive and selective greedy algorithms. We verified the effectiveness of our scheme by comparison with the existing two-step detection schemes through extensive simulations. The simulations results illustrate the impact of monitor locations on the quality of monitoring paths. Namely, the results validate our assertion that minimizing

the number of monitoring devices and minimizing the detection overhead are conflicting objectives. Moreover, it is demonstrated that using the same number of monitors, the one-step detection solutions yield much less overhead than the two-step detection solution. This confirms that the existing cost model does not reflect the detection costs properly. Furthermore, results show that the selective greedy algorithm provides near-optimal solutions for small networks, and yields solutions that use less than 10% of the network resources for large-scale networks.

The next chapter investigates the problem of anomaly detection in multi-domain networks. The properties and the limitations of these networks are studied in order to come up with an appropriate anomaly detection scheme.

CHAPTER 3

Link-Level Anomaly Detection in Multi-Domain Networks

3.1 Introduction

Most existing studies on link-level network monitoring have focused on mono-domain networks (*e.g.*, [1] [2] [29] [6]). However, usually, services cross multiple domains that belong to different administrative authorities, and that are likely to have conflict of interests. This raises some confidentiality problems that constrain the monitoring task. Namely, most proposed monitoring schemes, which assume a detailed knowledge of the network topology, cannot be applied on multi-domain networks. This is because domains are usually not willing to disclose detailed information of their network topology and available resources.

In this chapter, we focus on the problem of detecting link-level anomalies in multi-domain networks. This includes deploying monitors and selecting monitoring paths than can cover all the multi-domain network links. Our goal is to come up with an anomaly detection scheme that overcomes the confidentiality limitations. To this end, we investigate the problem along two axes. The first axis ignores confidentiality constraints and considers the multi-domain network as a single domain. This is the global anomaly detection technique. The second axis overcomes the confidentiality issue by minimizing the information that is to be exchanged between domains. Each domain monitors its intra-domain links independently from the other domains, *i.e.*, without disclosing any information of its intra-domain topology. Neighboring domains exchange only the set of their border nodes that are candidate to support monitoring devices, in order to compute monitor locations and paths that can cover the inter-domain links connecting them. This is the per-domain anomaly detection technique. Practically, the global technique might be infeasible due to

confidentiality issues. However, a comparative study of these two anomaly detection techniques aims at finding out and evaluating all the constraints, other than confidentiality, that the multi-domain detection schemes must comply to.

The problem of monitor location and anomaly detection has gained great interest over the few last years. The shared goal of all these works is to minimize the detection cost that includes, usually, the cost of deploying monitoring devices and the detection overhead. The main challenge of this chapter is to extend the anomaly detection scheme proposed in the previous chapter to multi-domain networks with respect to topology characteristics. We provide a mathematical formulation of the problem, and we show that it is $\mathcal{NP}$-Hard. Therefore, we devise a heuristic solution that takes into considerations the characteristics and the limitations of multi-domain network topologies.

Besides the computation time and the detection cost, we consider new criteria that emerge from the characteristics of multi-domain networks to evaluate the two monitoring techniques. First multi-domain networks are large networks. Therefore, the global monitoring technique that considers the multi-domain network as a single domain is likely to monitor long paths that cross multiple domains. This would result in large detection delays. Indeed, the longer the monitored paths are, the larger the anomaly detection delays are. Furthermore, long monitoring paths result in large number of suspect links in case of failure. This is because all the links of a monitoring path that exhibits an anomaly, except those who belong to monitoring paths not exhibiting an anomaly, are suspect to be anomalous. Second, multi-domain networks are composed of domains that belong to different administrative and economic authorities. Therefore, the monitoring solution should distribute the monitoring load among domains fairly, otherwise, the most overloaded domains would not be willing to collaborate.

We show through simulations that confidentiality is so far not the only limitation to global anomaly detection. Indeed, the results show that this anomaly detection technique yields solutions with relatively long monitoring paths, and does not guarantee a fair distribution of monitoring load among domains. Besides, the computation time for the global technique is drastically high compared to the computation time for the per-domain technique. In contrast, the difference of costs of the solutions of the two techniques, in terms of number of monitors and redundant measurements of links, is small. This is due to the characteristics of the multi-domain topology that will be discussed throughout this chapter.

The remainder of this chapter is organized as follows. Section 3.2 states the problem of anomaly detection in multi-domain networks, and describes the network model, the

multi-domain architecture, and the anomaly detection cost model. Section 3.3 provides an ILP formulation of the problem, and Section 3.4 introduces the heuristic algorithm. The comparison results of the two anomaly detection approaches are reported in Section 3.5. Concluding remarks are provided in Section 3.6.

3.2 Problem Formulation

3.2.1 Network Model

We model a multi-domain network composed of M connected domains as a set of undirected graphs $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{L}_i), i = 1, 2, ..., M$. $\mathcal{V}_i$ is the set of nodes of domain i. It is composed of two sets: $\mathcal{V}_i^{inter}$ and $\mathcal{V}_i^{intra}$. $\mathcal{V}_i^{inter}$ represents the set of border nodes that connect domain i to its neighboring domains, and $\mathcal{V}_i^{intra}$ represents the set of core nodes. Similarly, the set of links $\mathcal{L}_i$ is composed of two sets: $\mathcal{L}_i^{intra}$ and $\mathcal{L}_i^{inter}$. $\mathcal{L}_i^{intra}$ represents the set of intra-domain links that connect the core nodes, and $\mathcal{L}_i^{inter}$ represents the set of inter-domain links that connect nodes of $\mathcal{V}_i^{inter}$ to the border nodes of neighboring domains. We denote by $\mathcal{P}_i$, $i = 1, 2, ..., M$ the set of intra-domain paths of domain i. A path $p \in \mathcal{P}_i$ is a set of undirected intra-domain links. We denote by $\mathcal{P}^{inter}$ the set of inter-domain paths of the multi-domain network. A path $p \in \mathcal{P}^{inter}$ includes at least one inter-domain link. We refer to $\mathcal{G}_i^{intra} = (\mathcal{V}_i, \mathcal{L}_i^{intra})$ as the intra-domain graph of domain i. Let $\mathcal{ND} = \{(i, j);$ $i, j = 1, 2, ..., M$; i and j are neighbor domains$\}$ be the set of neighbor domains. We refer to $\mathcal{G}_{(i,j)} = (\mathcal{V}_{i,j}, \mathcal{L}_{i,i})$ as the graph of the inter-domain topology connecting domain i to domain j. $\mathcal{V}_{i,j}$ is the set of border nodes of domains i and j that are connected to each other, and $\mathcal{L}_{i,j}$ is the set of inter-domain links connecting domain i to domain j.

3.2.2 Problem Definition

This work addresses the problem of anomaly detection in multi-domain networks. For mono-domain networks, minimizing the monitor location cost and the probe cost consists in deploying as few monitors as possible in carefully selected locations and avoiding redundant measurements of links, *i.e.*, avoiding overlaps among monitoring paths. These two minimization objectives are conflicting objectives. We have shown in the previous chapter of this thesis that a joint optimization of monitor location and anomaly detection costs balances efficiently the trade-off and reduces the two costs. However the problem is $\mathcal{NP}$-Hard. Heuristics have been proposed for mono-domain networks in chapter 2. For multi-domain

networks, the problem can be formulated as follows. We want to deploy monitors in a multi-domain network and select monitoring paths between the deployed monitors. The aim is to cover all the inter-domain and the intra-domain links, while reducing the number of deployed monitors and avoiding redundant measurements.

The constraints to global anomaly detection in multi-domain networks stem from the characteristics of these networks. The first constraint is related to the structure of multi-domain networks. A multi-domain network is a set of domains that belong to different administrative authorities. Due to economic and security considerations, domains are usually not willing to share detailed information of their network topologies and resources. This is a blocking constraint to the global anomaly detection technique. This technique assumes the existence of a central entity that has a detailed knowledge of the intra-domain topologies of all the domains composing the multi-domain network as well as the inter-domain topologies connecting neighboring domains. An alternative solution would be to let each domain cover its intra-domain links using intra-domain paths only. Neighboring domains collaborate to cover inter-domain links connecting them. This is the per-domain technique.

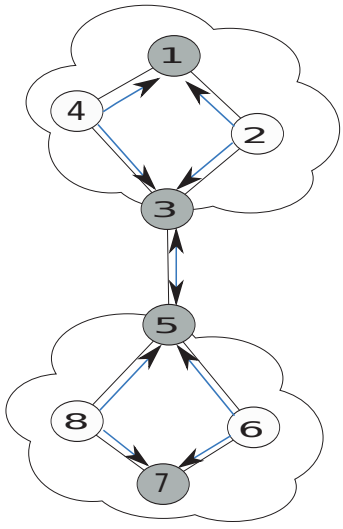


Figure 3.1: Per-domain detection solution

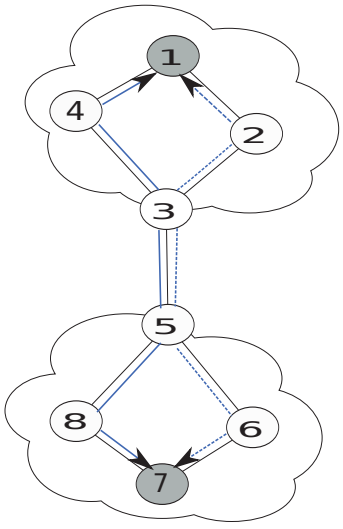


Figure 3.2: Global detection solution

At first glance, when the global topology is known, we tend to assert that the global technique outperforms the per-domain technique. This is because, considering only the metrics of the number of monitors and the number of redundant measurements of links, all the solutions to the per-domain technique are feasible solutions to the global anomaly detection technique. We illustrate our assertion in Figure 3.1. and Figure 3.2. Hereby, we

consider a multi-domain network composed of two domains connected by a single inter-domain link $(3, 5)$. We assume that the cost of deploying a monitor equals the cost of a redundant measurement, *i.e.*, the cost of measuring a link that is already measured. Grey nodes are equipped with monitoring devices. The thick lines draw the monitoring paths. Figure 3.1. depicts a minimal per-domain anomaly detection solution, whereas Figure 3.2. depicts a minimal global anomaly detection solution. We notice that the per-domain solution deploys 4 monitors, against 2 monitors and 1 redundant measurement for the global solution. The global technique succeeded to reduce the detection cost by removing monitors that are deployed on the border nodes of each domain.

The question that arises here is the following: *how worse is the performance of the per-domain technique compared to the global anomaly detection technique ?* To answer this question, we investigate the quality of the global solutions. Reducing the number of monitors results in longer monitoring paths. The figures above validate this claim. Nonetheless, multi-domain networks are usually very large networks. Subsequently, the global technique is likely to select very long monitoring paths. This is the second constraint to global anomaly detection, because the longer the monitored paths are, the larger the anomaly detection delays are and the larger the number of suspect links in case an anomaly occurs is. Furthermore, when domains accept to collaborate to perform global anomaly detection, they expect to achieve individual benefits in return. This means that the monitoring solution should distribute the monitoring load among the participating domains evenly. Therefore, besides the minimization of monitor cost and the probe cost, the quality of monitoring paths and the fairness of monitoring load distribution must be considered in the evaluation of the two anomaly detection techniques.

Based on this discussion, we claim that confidentiality is so far not the only constraint to global anomaly detection, and that the per-domain anomaly detection might turn out to be more efficient with respect to some metrics. We validate our claims in the remainder of this chapter.

3.2.3 Architecture and Cost Model of Multi-Domain Anomaly Detection

Figure 3.3. depicts a sample multi-domain monitoring architecture, only nodes that are equipped with monitoring devices are drawn. In each domain there is a Network Operations Center, denoted by *Domain NOC*, that communicates with the monitors of the domain, in order to collect monitoring information and manage the monitoring task within the domain. A *Domain NOC* has a detailed knowledge of the domain topology and

resources. In addition, there is a central *NOC* that communicates with all the *Domain NOC*s. It collects and analyzes monitoring information collected within the domains. This multi-domain architecture matches the usual architecture proposed in most works on multi-domain monitoring (*e.g.*, [42], [41]). For the global technique, the central *NOC* has a detailed knowledge of the topologies and the resources of all the domains, whereas, for the per-domain technique it does not participate in the detection task.

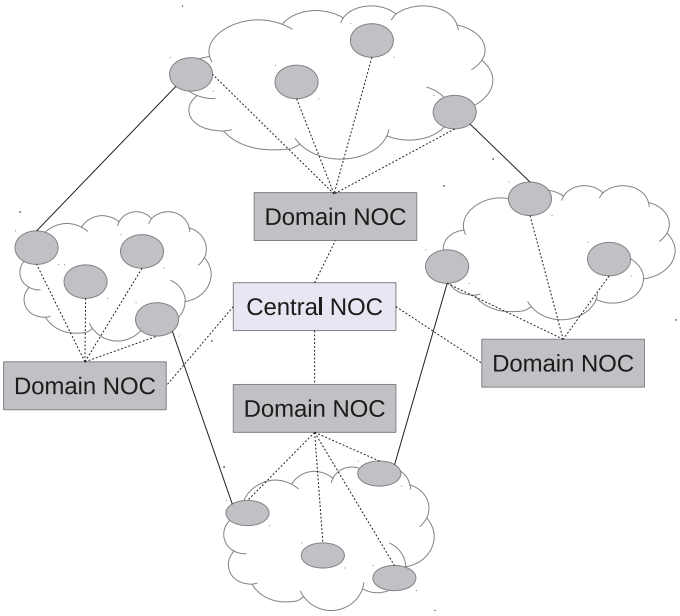


Figure 3.3: Sample Multi-domain Monitoring Architecture

A summary of the symbols used in the remainder of this chapter is depicted in TABLE 3.1.

The multi-domain anomaly detection cost can be expressed as the summation of the following costs:

– Monitor cost: it includes the effective cost of deploying hardware and software monitoring devices and the cost of their maintenance. In addition, it includes the cost of communications between monitors and their corresponding *Domain NOC*. For instance, the cost of communications between a monitor and the *Domain NOC* can be expressed as a function of the number of routing hops separating them. Let us denote by C_n the cost of deploying a monitor on node n, the multi-domain monitor cost can be expressed as follows:

$$\sum_{i=1,..,M,\ n\in\mathcal{N}_i} C_n Y_n \tag{3.1}$$

Table 3.1: Notations used throughout this chapter

Symbol	**Definition**
Z_p	A binary variable that indicates whether path p is selected to be monitored
Y_n	A binary variable that indicates whether node n is selected as a monitor location
C_n	The cost of deploying a monitoring device on node n
C_l	The cost of monitoring the intra-domain link l
$C_{l_{(i,j)}}$	The cost of monitoring the inter-domain link $l_{i,j}$
δ_{lp}	A binary parameter that indicates whether link l belongs to path p
$\delta_{l_{(i,j)}p}$	A binary parameter that indicates whether the inter-domain link $l_{(i,j)}$ belongs to path p
δ_{np}	A binary parameter that indicates whether node n is an end node of path p
$\mathcal{CP}$	The set of candidate monitoring paths
$\mathcal{SP}$	The set of selected monitoring paths
$\mathcal{SM}$	The set of selected monitors
$DR_i(\mathcal{SP})$	the detection ratio of path p_i considering the set of selected monitoring paths $\mathcal{SP}$

– Probe cost: it expresses the overhead of monitoring flows on the underlying network. Each link must be monitored at least once. Redundant measurements of links are considered as monitoring overhead. Let us denote by C_l the cost of measuring link l. C_l must be proportional to the load of link l, in order to avoid multiple measurements of the most overloaded links of the network. The multi-domain probe cost can be expressed as follows:

$$\sum_{(i,l,p)\in\mathcal{S}_1} \delta_{lp} C_l Z_p \; + \sum_{((i,j),l,p)\in\mathcal{S}_2} \delta_{lp} C_l Z_p \tag{3.2}$$

where $\mathcal{S}_1 = \{1,\ldots,M\} \times \mathcal{L}_i \times \mathcal{P}_i \cup \mathcal{P}^{inter}$ and $\mathcal{S}_2 = \mathcal{ND} \times \mathcal{L}_{(i,j)} \times \mathcal{P}^{inter}$.

3.3 ILP formulation

The anomaly detection scheme should minimize the monitor cost (3.1), and the probe cost (3.2). In the previous chapter, we demonstrated that there is an interplay between these two minimization objectives. However, it turned out that the joint minimization of the two objectives balances efficiently this interplay (refer to the previous chapter). Therefore, our ILP formulation minimizes the detection costs defined in the previous section jointly. Let α and β be the weight associated to the minimization of the monitor cost and the weight associated to the minimization of the probe cost, respectively. The objective function reads as follows:

$$\alpha \sum_{i=1,\ldots,M,\ n\in\mathcal{N}_i} C_n Y_n + \beta\Big(\sum_{(i,l,p)\in\mathcal{S}_1} \delta_{lp} C_l Z_p \ + \sum_{((i,j),l,p)} \delta_{lp} C_l Z_p\Big) \tag{3.3}$$

All the links of the multi-domain network, *i.e.*, the intra-domain and the inter-domain links, must be monitored at least once. Practically, this means that each link must belong to at least one monitoring path. These link coverage constraints read as follows:

$$\sum_{i=1,\ldots,M,\ p\in\mathcal{P}_i\cup\mathcal{P}^{inter}} \delta_{lp} Z_p \geq 1; \quad \forall i = 1,\ldots,M, \forall l \in \mathcal{L}_i \tag{3.4}$$

$$\sum_{p\in\mathcal{P}^{inter}} \delta_{l_{(i,j)}p} Z_p \geq 1; \quad \forall (i,j) \in \mathcal{ND}, \forall\ l_{(i,j)} \in \mathcal{L}_{(i,j)} \tag{3.5}$$

Either end node of each monitoring path must be selected as a monitor location. These monitor location constraints read as follows:

$$Y_n \geq \delta_{np} Zp, \quad \forall i = 1,\ldots,M, \forall n \in \mathcal{N}_i, \forall p \in \mathcal{P}_i \cup \mathcal{P}^{inter} \tag{3.6}$$

The equivalent problem for mono-domain networks has been shown to be $\mathcal{NP}$-Hard in chapter 2. The multi-domain monitoring problem is reduced to the mono-domain monitoring problem for $\mathcal{ND} = \emptyset$ and $\mathcal{P}^{inter} = \emptyset$. We conclude that the multi-domain monitoring problem is $\mathcal{NP}$-Hard, and thus, we propose a heuristic solution in the next section.

3.4 Heuristic Algorithm for Anomaly Detection in Multi-Domain Networks

The heuristic algorithm aims at minimizing the monitor cost and the probe cost jointly, thereby balancing the trade-off between these two minimization objectives, while considering the properties and the limitations of inter-domain networks.

Muli-domain networks are, usually, composed of dense domains interconnected by few inter-domain links [38]. Therefore, computing an inter-domain path connecting two nodes each belonging to a different domain is a difficult task. In the previous chapter, we have proposed a heuristic for joint optimization of monitor location and anomaly detection in mono-domain networks. This heuristic performs an in-depth exploration of the network graph, in order to find candidate monitoring paths between two given nodes. It has been shown that this technique delivers good candidate monitoring paths in short time. However, when we ran this heuristic on multi-domain networks and mono-domain networks of the same size (*i.e.*, the same number of links and the same number of nodes), we noted that the computation time of the multi-domain solution is drastically higher than the computation time of the mono-domain solution. As expected, this exponential increase of the computation time is due to the computation time of candidate monitoring paths in multi-domain networks. We consider the multi-domain network depicted in Figure 3.4. to illustrate our assertions.

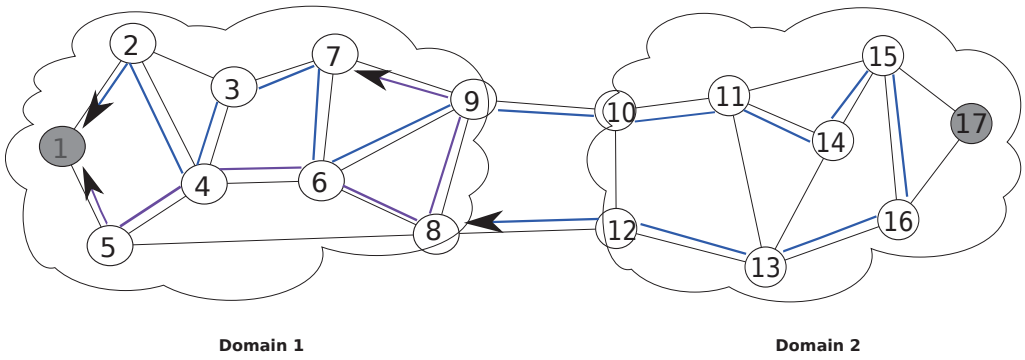


Figure 3.4: Illustrative multi-domain network

The network is composed of two domains denoted by Domain 1 and Domain 2, respectively. The gray nodes are equipped with monitoring devices. Path $\langle(1,2),(2,4),(4,3),(3,7),(7,6),(6,9),(9,10),(10,11),(11,14),(14,15),(15,16),(16,13),(13,12),(12,8)\rangle$ is an inter-domain monitoring path that starts from the monitor deployed in Domain 1, reaches Domain 2 and then returns back to domain 1. Path

$\langle(1,5),(5,4),(4,6),(6,8),(8,9),(9,7)\rangle$ is an intra-domain path that starts from the monitor deployed in Domain 1, crosses the two border nodes of Domain 1 (nodes 9 and 8), but do not reach Domain 2. We note that we avoid looping paths, *i.e.*, paths that cross the same nodes multiple times. These are two examples of excluded paths: long paths that do not end at a monitoring device and whose computation time is long. It is the existence of such inefficient paths that makes the computation time of inter-domain candidate monitoring paths quite long, and therefore, heuristics for mono-domain networks are inappropriate for global anomaly detection in multi-domain networks. Furthermore, existing works on intra-domain anomaly detection have not provided solutions for the problem of candidate monitoring path computation (*e.g.*, [1] [2] [29] [6]).

3.4.1 Computation of Candidate Monitoring Paths in Multi-Domain Networks

The solution that we propose to compute candidate monitoring paths consists in assigning a positive weight to each network link, and exploring the network links with a probability that is proportional to their weights. The underlying idea is to reduce the probability to re-explore bad sequences of links, while increasing the probability to cross inter-domain links. Initially, all the links have an equal weight. This means that links have the same probability to be added to the computed path. The computation ends when the path reaches the target node, this is a good path, or when it reaches a node whose neighboring nodes already belong to the path, this is a bad path. If the computed path is good, the weights of all its links are incremented. Since all the good paths cross inter-domain links, this will increase the probability to use those links. We resume the computation of new paths from the starting node, in order to increase the space of explored paths.

3.4.2 Greedy Monitor Location and Path Selection Algorithm

Here, we give an outline of Algorithm 3. The algorithm starts by selecting two monitor locations with the lowest detection costs (ties are broken randomly). Then, it computes a set of candidate paths between the selected monitor locations as described above. For each candidate path, the algorithm computes a detection ratio that expresses the ratio between the number of links that are covered by the path and the number of redundant measurements, *i.e.*, the number of links that belong to the path and that are already covered by the already selected monitoring paths, *i.e.*, paths in SP. The path that have the highest detection ratio is selected. This is because it achieves the best trade-off between

Algorithm 3: Monitor location and path selection algorithm for anomaly detection in multi-domain networks

1 $\mathcal{SP} = \emptyset$;
2 Select two monitor locations m_1, m_2 that have the lowest monitor locations costs;
3 Add $m1$ and $m2$ to $\mathcal{SM}$;
4 $\mathcal{CP} \leftarrow$ {candidate paths between m1 and m2};
5 $\forall p_i \in \mathcal{CP}, DR_i(\mathcal{SP}) \leftarrow$ (number of links covered by p_i) / (number of links of p_i that are covered by paths in $\mathcal{SP}$);
6 **while** *(not all links are covered)* **do**
7 Find $p_s \in \mathcal{CP}$ such that $\forall p_i \in \mathcal{CP}, DR_s(\mathcal{SP}) \geq DR_i(\mathcal{SP})$;
8 **if** *($DR_s(\mathcal{SP}) == 0$)* **then**
9 Go to line 25
10 /* the deployed monitors cannot cover all the network links*/;
11 **else**
12 Add p_s to $\mathcal{SP}$;
13 Remove p_s from $\mathcal{CP}$;
14 Update $DR_i(\mathcal{SP})$, $\forall p_i \in \mathcal{CP}$;
15 **if** *(Not all links are covered)* **then**
16 Go to line 25;
17 **else**
18 **if** *(the cost of deploying a new monitors $\geq$ redundant measurements incurred by paths in $\mathcal{SP}$)* **then**
19 End of the algorithm;
20 **else**
21 Go to line 25;
22 Select a new monitor that minimize the probe cost;
23 Add the new monitor to $\mathcal{SM}$;
24 Clear $\mathcal{CP}$;
25 $\mathcal{CP} \leftarrow$ candidate paths between the new monitor and the deployed monitors;
26 Remove paths that incur redundant measurements from $\mathcal{SP}$ and add them to $\mathcal{CP}$;
27 Go to line 5;

the number of covered links and the number of redundant measurements. The detection ratios are updated whenever a new path is selected.

Monitoring paths are selected until all the network links are covered, or all the candidate paths have their detection ratios equal to zero. In the latter case, the deployed monitors are not sufficient to cover all the network links, therefore, a new monitor is deployed. In

the first case we get a full monitoring solution, *i.e.*, full coverage of the network links. However, as said earlier, we want to find the best trade-off between the monitor cost and the probe cost. Therefore, when the algorithm gets a full solution, it verifies whether it can diminish the anomaly probe cost by deploying new monitors. It decides to deploy a new monitor if the cost of deploying a new monitor is lower than the probe cost of the current solution .

Now, when a new monitor is deployed, the algorithm removes all the paths that incur redundant measurements from the set of selected paths, and injects them into the set of candidate paths CP. Then it selects monitoring paths with respect to their current detection ratios.

3.5 Performance Evaluation

In this section we first describe the evaluation methodology, and then we present and discuss numerical results.

3.5.1 Evaluation Methodology

The aim of the evaluation is to assess the performance of per-domain anomaly detection versus global anomaly detection in multi-domain networks. To this end, we run the heuristic proposed in the previous section for these two monitoring techniques on several multi-domain network topologies generated randomly using the network generator Brite [13] [33] (Waxman model [31]: $\alpha = \beta = 0.4$, random node placement [1].). Unless mentioned, we consider the following setting to generate multi-domain topologies: the network is composed of three domains; a domain of 10 nodes and 31 links is connected to a domain of 15 nodes and 59 links, which is in turn connected to a domain of 10 nodes and 31 links. The number of border nodes that connect each domain to a neighboring domain ranges from 2 to 3 nodes, and the number of inter-domain links between two neighboring domains ranges from 4 to 6 links. In the remainder of this chapter, we refer to this setting as the default setting. Figure 3.5. depicts a sample multi-domain topology. We assume that all the network nodes are candidate to support monitoring devices and that the cost of deploying monitors is the same for all the nodes; *i.e.*, $C_{n_i} = 1, \forall n_i \in N_i \ \forall i = 1, 2, ..., M$.

1. These parameters are not to be confused with the monitor cost weight (α) and the probe cost weight (β) introduced in Section 4.6. Their values equal the values used by Waxman to generate network topologies [31]

Furthermore, we assume that the link monitoring cost is the same for all the network links; *i.e.*, $C_{l_i} = 1, \forall l_i \in L_i \ \forall i = 1, 2, ..., M$. We assume that $\alpha = \beta = 1$. All simulation measures are the mean over 30 simulations on randomly generated topologies. Our simulation plateform is developed in C++.

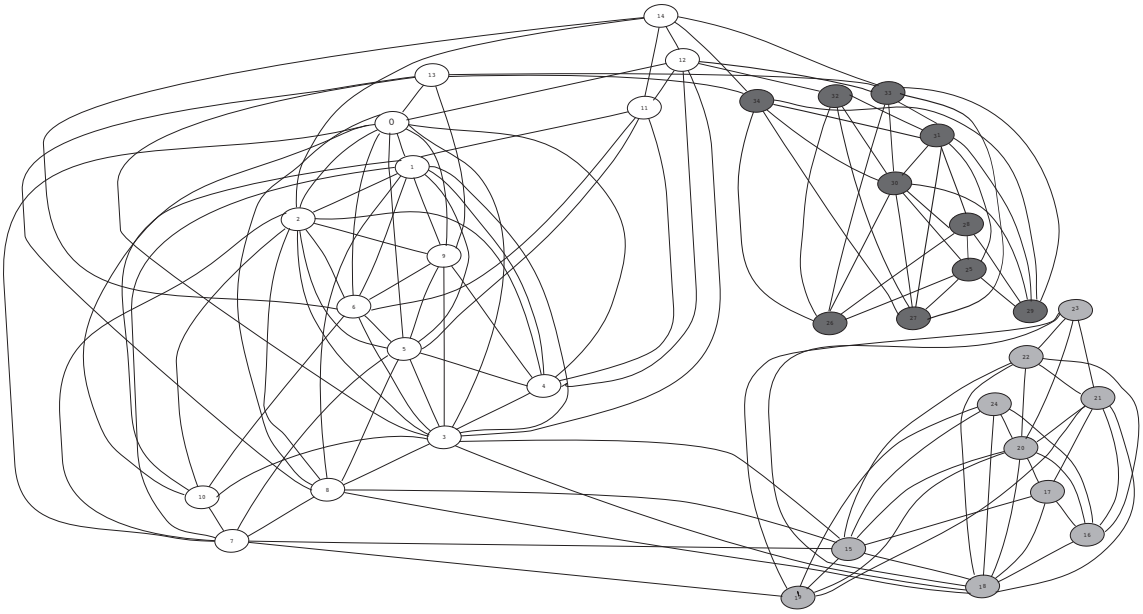
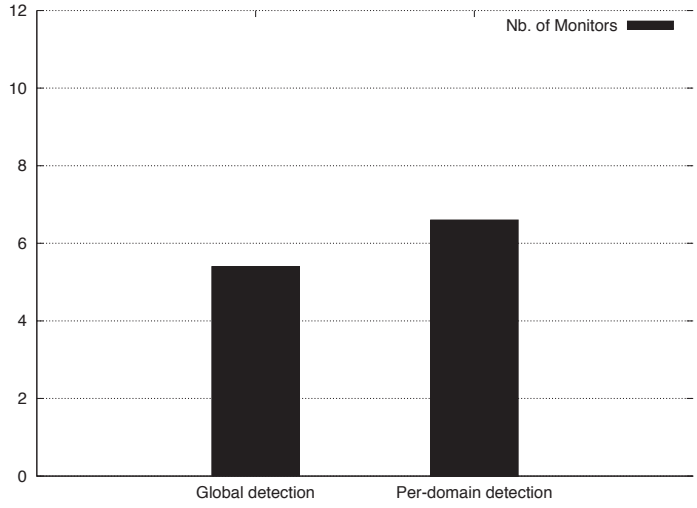


Figure 3.5: Sample multi-domain topology

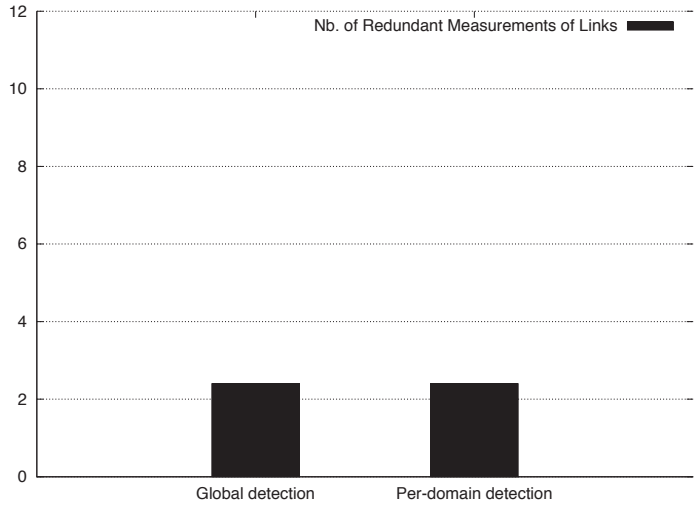
For global anomaly detection, we assume that the central *NOC*, which has a global knowledge of the multi-domain topology, runs the heuristic on the global topology including the three domains and the inter-domain links connecting them. For per-domain anomaly detection, each domain runs the heuristic on its intra-domain topology. Once all the intra-domain links are covered, neighboring domains exchange their set of border nodes that are equipped with monitoring devices, if any, in order to cover the inter-domain links connecting them using the same heuristic on the inter-domain topology. We note that in our simulations, if two intra-domain solutions have the same monitoring cost, we choose the solution that deploys the most monitors on its border nodes so that they can be re-used to cover inter-domain links.

3.5.2 Numerical Results

We evaluate and compare the global monitoring technique and the per-domain monitoring technique along four metrics:



(a)



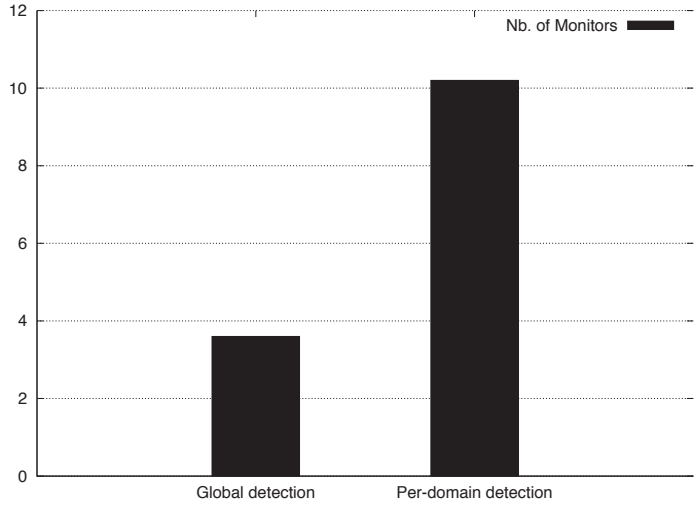
(b)

Figure 3.6: Monitoring cost: default setting

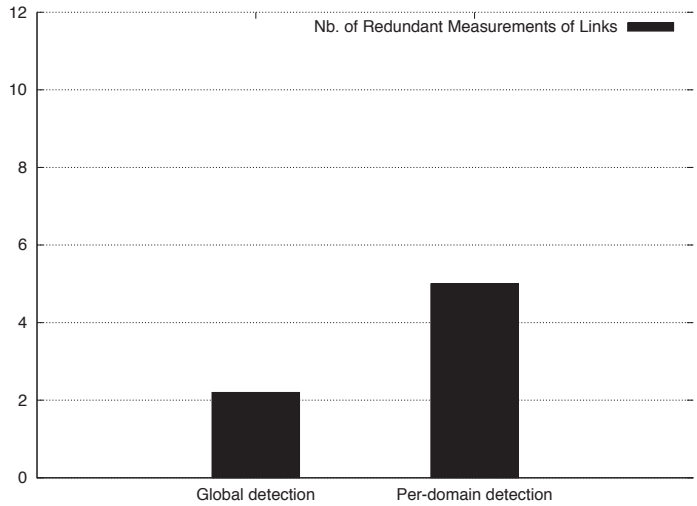
Monitoring Cost

We expect that the fewer are the inter-domain links, the smaller is the difference between the costs of the solutions delivered by each of the two anomaly detection techniques. Indeed, the global detection technique reduces the probe cost by monitoring inter-domain paths, i.e. paths that cross multiple domains. This is because the monitoring of inter-domain paths requires less monitoring devices, and can cover links of crossed domains and also inter-domain links. However, the number of non-overlapping inter-domain monitoring paths is proportional to the number of inter-domain links. Therefore, the global technique

gets blocked by redundant measurements of inter-domain links, and ends by deploying additional monitors to avoid overlaps among inter-domain paths.



(a)



(b)

Figure 3.7: Monitoring Cost: doubling inter-domain links

To validate our expectations, we run the heuristics for global and per-domain anomaly detection on topologies with the default setting, and on topologies for which we doubled the number of inter-domain links. Figure 3.6 plots the number of deployed monitors (a) cost and the number of redundant measurements of links (b) for the two monitoring techniques applied on topologies with the default setting. Fig 3.7 plots the same metrics for the two

monitoring techniques applied on topologies for which we have doubled the number of inter-domain links.

As expected, Figure 3.6. shows that the difference between the monitoring costs of the solutions delivered by the two monitoring techniques is low for the default setting. We notice also that the global monitoring technique deploys few monitors than the per-domain monitoring technique, whereas the number of redundant measurements is slightly larger for global monitoring. Figure 3.7. shows that, compared to the results for the default setting, the global monitoring technique deploys less monitors and achieves almost the same number of redundant measurements. In contrast, the cost of the solutions delivered by the per-domain monitoring technique has almost doubled. Clearly, the per-domain monitoring techniques needs to deploy additional monitors to cover the large number of inter-domain links.

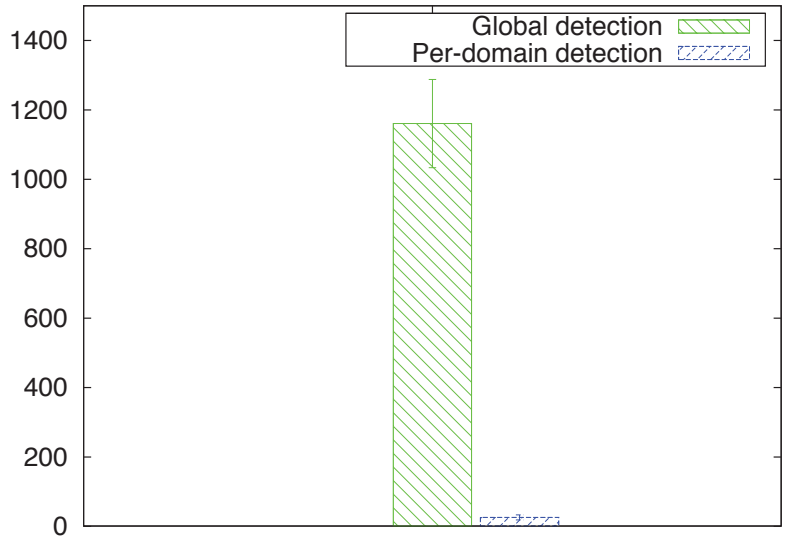


Figure 3.8: CPU Running Time (s)

Computation Time

Figure 3.8. draws the average CPU computation time for global and per-domain monitoring. The figure shows that per-domain monitoring is much more faster than global monitoring. As explained earlier, this is because it takes longer time to compute candidate monitoring paths that cross multiple domains than to compute intra-domain candidate monitoring paths. However, the heuristic succeeds to deliver a solution for global moni-

toring in about 1200 seconds, whereas other heuristics devised for mono-domain networks have stumbled against the topology properties of multi-domain networks.

We note that practically the number of inter-domain connections is generally small in usual networks, and thus, the default setting is more realistic [38].

Quality of paths monitored

We categorize the monitoring paths according to their lengths, in terms of number of links, into five groups: paths of length in [1-5], paths of length in [6-10], paths of length in [11-15], paths of length in [16-20], and paths of length in [21-30]. In Figure 3.9., we show the distribution of network links by path length groups for the two monitoring techniques. A link belongs to a path length group if it is monitored by a path whose length is included in the length range of that group.

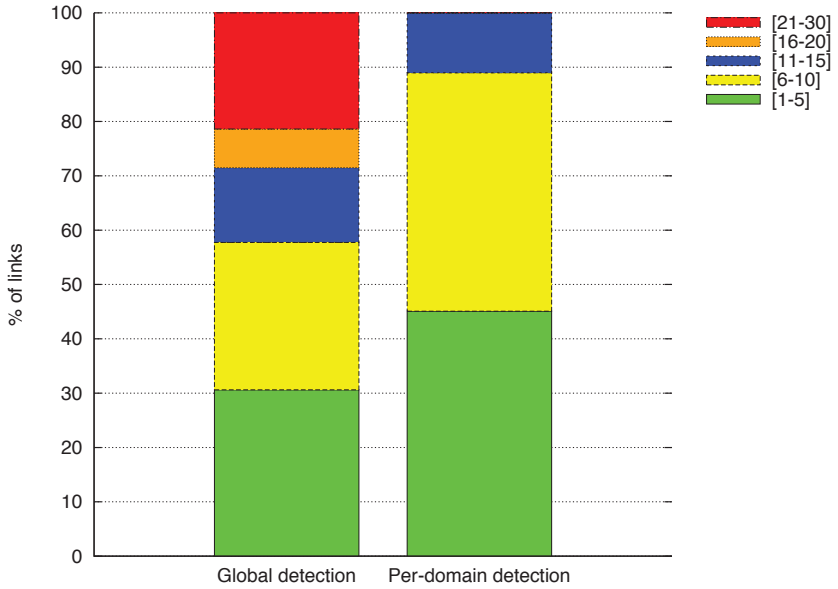


Figure 3.9: Distribution of network links by path length groups

First, we notice that the longest monitoring paths for the per-domain monitoring technique are of length less than or equal to 15 links; whereas for the global monitoring technique, the length of monitoring paths reaches 30 links. This is because the global monitoring technique monitors inter-domain paths that are naturally longer than intra-domain paths. Second, Figure 3.9. shows that more than 40% of the network links are crossed by long monitoring paths, paths whose length exceeds 15. This means that in 40% of cases of link-level anomalies, we get between 15 and 30 suspect links. In contrast, for per-domain

monitoring almost 90% of network links are traversed by short monitoring paths, paths whose length is less than or equal to 10. We conclude that the per-domain monitoring technique reduces the length of monitoring paths, and therefore, reduces anomaly detection delays and the number of suspect links when an anomaly occurs.

Fairness of monitoring solutions

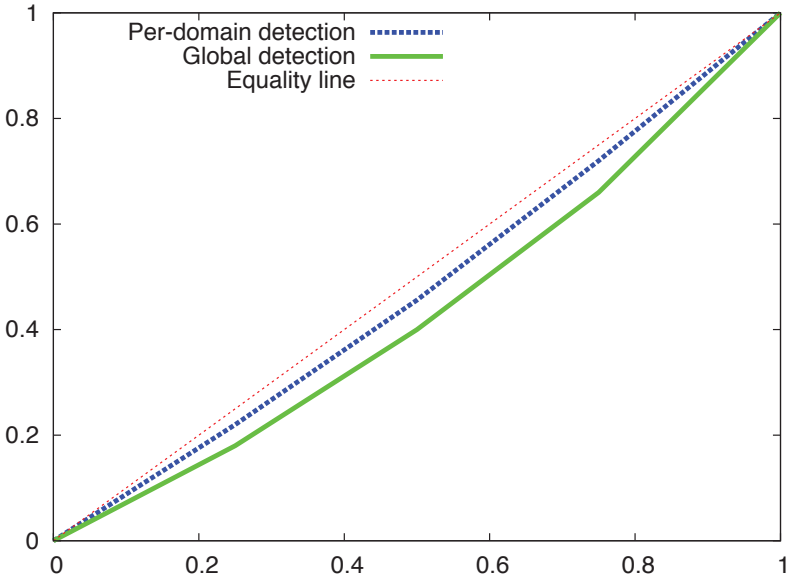


Figure 3.10: Distribution of monitors and redundant measurements across domains

In this section we propose to show the distribution of monitors and redundant measurements across domains. The aim is to evaluate the fairness of the monitoring solutions delivered by the two monitoring techniques in distributing the monitoring load among domains. To this end, we consider in our simulations multi-domain networks composed of four domains having the same number of intra-domain links, 18 links, and the same number of nodes, 8 nodes. Each of the four domains is connected to two other domains. The number of inter-domain connections, *i.e.*, number of inter-domain links and inter-domain nodes connecting two neighboring domains, is the same for each couple of neighboring domains. For such symmetric multi-domain networks, a fair monitoring solution would distribute monitors and redundant measurements among domains evenly.

We use the Gini coefficient to measure the efficiency of the probe cost balancing among domains [39] [40]. Figure 3.10. plots the Lorenz curves for the two monitoring techniques.

Here, the curves are functions of the cumulative percentage of the number of domains ordered by their detection costs, *i.e.*, the number of monitors deployed in the domain and the number of redundant measurements of the domain links, on the x-axis mapped onto the corresponding cumulative percentage of their detection costs on the y-axis. We note that if an inter-domain link is measured multiple times, we add the cost of this redundant measurement to the detection costs of the two domains it connects. If the detection cost is distributed among domains evenly, the Lorenz curve is a diagonal line that we call the line of equality. Uneven distributions generate curves below this line. The larger is the area between the line of equality and the Lorenz curve, the greater is the inequality in the distribution of the detection load among domains.

Figure 3.10. shows that the curve corresponding to the global detection technique falls below the curve corresponding to the per-domain technique. This means that the per-domain technique balances the detection load among domains more efficiently. This is explained by the fact that, in contrast to the per-domain technique, the global technique considers the multi-domain networks as a single domain, which generates uneven distributions of the detection load among domains.

3.6 Conclusion

This chapter investigates the problem of anomaly detection in multi-domain networks. An ILP formulation of the anomaly detection problem is proposed, and a heuristic that takes into account the limitations of multi-domain topologies is devised. This heuristic is used to evaluate and compare two anomaly detection techniques, a global anomaly detection technique and a per-domain anomaly detection technique, with respect to a set of performance metrics that emerge from the properties of multi-domain networks.

Simulation results show that confidentiality is so far not the only constraint to global anomaly detection. Indeed, This monitoring technique yields solutions with relatively long monitoring paths (for the global technique, 40% of the links of the evaluated multi-domain network topologies are covered by paths longer than 15 hops, whereas, for the per-domain technique, 90% of links are covered by paths shorter than 10 hops), and does not guarantee a fair distribution of monitoring load among domains. Besides, the time required for computing a global anomaly detection solution is much larger than the time required for computing a per-domain anomaly detection solution. In contrast, the cost of the per-domain solutions is slightly larger than the cost of global solutions. This makes the

per-domain technique an efficient and secure alternative for anomaly detection in multi-domain networks.

Part III

Localization of Link-Level Network Anomalies

CHAPTER 4

Localization of Single Link-Level Network Anomalies

4.1 Introduction

Upon detecting an anomaly, a set of suspect links is constructed out of the measurements collected during the detection phase. The anomaly localization phase is triggered then. It aims at reducing the set of suspect links to the anomalous link(s). The main challenge of this phase is to pinpoint the root cause of the detected anomaly as fast as possible in order to enable a fast recovery of the network.

Agrawal et al. [1] proposed an accurate link-level anomaly localization scheme that can localize all potential single link-level anomalies in a given network. The key idea is to deploy resources that enable the monitoring of a set of paths that distinguish all links of the network pairwise. Two links are said to be distinguished from each other if we are able to decide which one is anomalous when an anomaly occurs on one of them. Whenever an anomaly is detected, this set of paths is monitored in order to pinpoint the anomalous link. This technique is suboptimal in that it considers all the network links as suspect, ignoring the information provided by the detection process, which generates unnecessary overhead and delays the localization. More recently, Barford et al. [2] proposed another scheme that selects paths that are to be monitored during the localization phase. Although this technique minimizes the localization overhead, because the monitored paths distinguish only between the suspect links pairwise, it suffers from two imperfections. The first is the non-negligible time of computing the set of paths that are to be monitored upon detecting an anomaly, which increases the localization delay (*i.e.*, time elapsed between the moment when an anomaly is detected and the moment when the anomalous link is pinpointed). The

second is that there is no guarantee to localize all potential anomalies, because the deployed monitors ensure only the coverage of links [1]. In this chapter, we demonstrate that 1) not all links of the network need to be distinguishable pairwise for localizing any potential anomaly, 2) all potential anomaly scenarios can be derived offline from any detection solution that covers all the network links. Thus, we compute full and cost-efficient localization solutions, *i.e.*, monitors that are to be activated and paths that are to be monitored upon detecting an anomaly, for all potential anomalies offline. Subsequently, we achieve an important gain in the localization delay and overhead.

Multiple works propose to compute the set of paths that are to be monitored dynamically upon detecting an anomaly (*e.g.*, [43] [46] [44] [45] [47] [48] [49] [50]). Practically, this means that one probe that maximizes the information gain given the previous probe observations is selected and sent in the network at a time. Such an approach is practical for highly dynamic environments. However, it is not practical for networks where anomalies are rare events, especially, because it yields excessive delays.

Furthermore, most existing works consider only one criterion for monitoring path selection that is the minimization of the number of monitored paths, and only one criterion for monitor location selection that is the minimization of the number of deployed monitoring devices (*e.g.*, [2] [1]). However, these criteria do not reflect the localization cost properly. Indeed, to reduce the localization delay and overhead, monitoring of links that do not provide extra localization information during the localization phase must be avoided. Moreover, monitor locations must be selected carefully towards minimizing the delay of communications between the Network Operations Center (NOC) and the deployed monitors. A novel anomaly localization cost model that considers the infrastructure cost, the localization overhead and the localization delay is, therefore, proposed in this chapter. Besides, our anomaly localization scheme selects monitor locations and monitoring paths jointly, thereby enabling a trade-off between the number and locations of deployed monitoring devices and the quality of selected monitoring paths. We formulate the problem as an ILP, and we show that it is $\mathcal{NP}$-hard through a polynomial-time reduction from the facility location problem.

Prior works on anomaly localization propose greedy approaches for computing localization solutions (*e.g.*, [1], [2], [6], [29]). In order to ensure the scalability, the number of candidate monitoring paths should reduced to a small subset of the network paths. Un-

1. The monitors used for anomaly detection are deployed such that all the network links are covered by at least one monitoring paths. They can not necessarily localize all potential anomalies

fortunately, none of these works described how candidate monitoring paths are selected, however, the choice of candidate paths has a great impact on the quality of the localization solution. In this work we propose a heuristic that implements our anomaly localization scheme. We devise an efficient algorithm for candidate path computation that makes the heuristic scalable and near-optimal at a time. The key idea is to use a mathematically proven properties that enable us to find the best candidate monitoring paths between two given monitor locations by exploring a very small proportion of the network paths.

We verify the effectiveness of our anomaly localization scheme through extensive simulations and by comparing it with an hybrid anomaly localization scheme that combines the strengths of the scheme proposed in [1] and the scheme proposed in [2].

The remainder of this chapter is organized as follows. Section 4.2 states the anomaly localization problem and describes the network model. Section 4.3 proves that the condition for single link-level localization established in [1] is sufficient but not necessary. A necessary and sufficient condition is established in the same section. Section 4.4 shows how to derive all potential anomaly scenarios offline. Section 4.5 describes the localization cost model, and section 4.6 introduces the ILP formulation. Section 4.7 demonstrates that the problem is $\mathcal{NP}$-Hard. The heuristic algorithm is introduced in Section 4.8. The performance of the proposed scheme is evaluated through simulations in section 4.9. Section 4.10 discusses the robustness of the proposed schemes. Concluding remarks are provided in Section 4.11.

4.2 Network Model and Problem Statement

We model the network as an undirected graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ comprising a set of nodes $\mathcal{N}$ connected by a set of undirected links [2] in $\mathcal{E}$. Let $\mathcal{P}$ be the set of all non-looping paths of the network. Unless otherwise mentioned, without loss of generality, we assume that all paths in $\mathcal{P}$ are candidate to be monitored and all the network nodes are candidate to support monitoring devices. We use the term monitoring paths to designate paths that are monitored during the detection phase, also referred to as detection paths, or during the localization phase, also referred to as localization paths. We consider that a network path is a set of links, instead of a sequence of links, and therefore, we apply set operations (*e.g.*, $\cap, \cup$) on paths. We denote the anomaly detection solution by $(\mathcal{D}_m, \mathcal{D}_p)$. $\mathcal{D}_m$ is the set of monitor locations where to deploy monitoring devices. $\mathcal{D}_p$ is a set of monitoring paths between the selected monitor locations that covers all the network links, $\cup_{p \in \mathcal{D}_p} p = \mathcal{E}$.

2. This work can be easily applied for directed links. Each directed link is duplicated into

We consider separable anomalies (*e.g.*, connectivity, high-low loss model, delay spike model, etc) that satisfy the following property: *a path experiences an anomaly if and only if at least one of its constituent links is anomalous* [15]. According to this property all links that are traversed by at least one detection path not exhibiting an anomaly are not anomalous, and all paths crossing an anomalous links exhibit the same anomaly. The remaining links constitute the set of suspect links. Anomaly localization aims at reducing the set of suspect links, inferred upon detecting an anomaly from the detection information, to the anomalous link. This requires monitoring additional paths that can distinguish between suspect links pairwise. Two links are said to be distinguishable from each other if we are able to decide which one is anomalous when an anomaly occurs on one of them.

The objective of this work is to come up with a localization scheme that enables the localization of all potential link-level anomalies accurately; while minimizing the cost of acquiring and deploying monitoring devices, the localization overhead and the localization delay. Our localization scheme infers all potential anomaly scenarios from any detection solution that covers all links of the network. This has two major benefits. The first is that we do not need to monitor a set of paths that can distinguish between every single pair of the network links whenever an anomaly is detected. The second is that we pre-compute full localization solutions for all anomaly scenarios offline, thereby accelerating the localization process. The inputs into our localization problem are an instance of the graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ and a set of detection paths $\mathcal{D}_p$ that covers all links in $\mathcal{E}$, and the outputs are a set of monitor locations whose monitors are to be activated and a set of paths that are to be monitored for each potential anomaly. The localization solution must achieve a good trade-off between the monitor deployment cost, the localization overhead and the localization delay. To this end, a novel cost model that measures these three metrics is proposed. Also, our localization scheme selects monitor locations and localization paths jointly; as opposed to existing schemes that apply a two-step selection procedure, therefore omitting the trade-off between the number and locations of monitors and the quality of localization paths.

4.3 Not all link pairs need to be distinguishable for localizing any single link-level anomaly

In this section, we first establish a necessary and sufficient condition to distinguish between two links. Then, we prove that not all link pairs need to be distinguishable for

localizing any potential single link-level anomaly accurately. This excludes an already established condition claiming that it is necessary to monitor a set of paths that can distinguish between all links of the network pairwise whenever an anomaly is detected [1].

Theorem 1. *The necessary and sufficient condition for two links e_1 and e_2 to be distinguishable from each other is the existence of a monitoring path that crosses either e_1 or e_2, but not both.*

Proof. We first demonstrate the sufficiency condition. Assume that either e_1 or e_2 is anomalous. Let p be a path that crosses e_1 (interchangeably e_2) but not e_2 (interchangeably e_1). If p exhibits an anomaly, then the anomalous link must be covered by p. We conclude that e_1 is the anomalous link. If, p does not exhibit an anomaly, then all its constituent links are not anomalous. It follows that the anomalous link is e_2. Thus, p is sufficient to distinguish between e_1 and e_2.

The necessary condition can be proved as follows. Assume that there does not exist any path that crosses only one of the two links. Then, the monitoring path set can be divided into two types of paths: paths that cross both e_1 and e_2, and paths that neither cross e_1 nor e_2. An anomaly on a given link affects all the monitoring paths that cross that link. Therefore, the latter type of paths is not affected by the anomalies that occur on any the two links, whereas the former type of paths is affected by the anomalies that occur on any of the two links. Thus, the set of monitoring paths that are affected by an anomaly on e_1 is exactly the same set of paths that is affected by an anomaly on e_2. This means that e_1 and e_2 cannot be distinguished from each other. □

Existing localization schemes (*e.g.*, [1]) claim that all links of the network must be distinguished pairwise in order to localize any potential anomalies. According to Theorem 1, this means that $\forall e_1, e_2 \in \mathcal{E}$ there exists a localization path that crosses either e_2 or e_2, but not both. However, we will demonstrate that this is a sufficient but not necessary condition, and we show how to infer the minimal set of pair of links that are to be distinguished from a given detection solution that covers all the network links.

Consider a network link $e \in \mathcal{E}$. We denote by D_{e_+} and D_{e_-} the set of detection paths that cross e and the set of detection paths that do not cross e, respectively. The set of suspect links associated to an anomaly on a link e is the set of all links that cannot be distinguished from e using only the detection information.

Theorem 2. *The set of suspect links associated to an anomaly on a given link $e \in \mathcal{E}$ equals $\cap_{p \in D_{e_+}} p - \cup_{p \in D_{e_-}} p$.*

Proof. We prove this theorem by construction. The set of detection paths can be divided into two sets:

- D_{e_+}: paths that cross link e.
- D_{e_-}: paths that do not cross link e.

An anomaly on link e affects only paths that cross this link. Subsequently, paths in D_{e_-} do not exhibit an anomaly. It follows that all the links that are traversed by paths in D_{e_-} are not suspect. Now, let L be the set of links that are traversed by paths in D_{e_+} and that are not traversed by paths in D_{e_-}, $L = \cup_{p \in D_{e_+}} p$ - $\cup_{p \in D_{e_-}} p$. L can be divided into two subsets of links:

- L_1: links that do not belong to $\cap_{p \in D_{e_+}} p - \cup_{p \in D_{e_-}} p$
- L_2: links that belong to $\cap_{p \in D_{e_+}} p - \cup_{p \in D_{e_-}} p$

We prove by contradiction that all links in L_1 are not suspect. Assume to the contrary that a link $l \in L_1$ is suspect. This means that there does not exist any path in D_{e_+} that distinguishes between l and e. It follows that for each $p \in D_{e_+}$, p crosses e and l. Thus $l \in \cap_{p \in D_{e_+}} p - \cup_{p \in D_{e_-}} p$, leading to a contradiction.

Likewise, we prove by contradiction that all links in L_2 are suspect. Assume to the contrary that a link $l \in L_2$ is not suspect, then, there exists at least one path $p \in D_{e_+}$ such that p distinguishes between e and l. Since all paths in D_{e_+} cross e, then p does not cross l. It follows that $l \notin \cap_{p \in D_{e_+}} p - \cup_{p \in D_{e_-}} p$, leading to a contradiction. □

Corollary 1. *A sufficient and necessary condition for localizing any potential link-level anomaly is to distinguish each link $e \in \mathcal{E}$ from links that belong to $\cap_{p \in D_{e_+}} p - \{\cup_{p \in D_{e_-}} p \cup \{e\}\}$.*

Let $\mathcal{S}(e)$ denotes the set of suspect links associated to anomalies on link e, $\mathcal{S}(e) = \cap_{p \in D_{e_+}} p - \{\cup_{p \in D_{e_-}} p \cup \{e\}\}$.

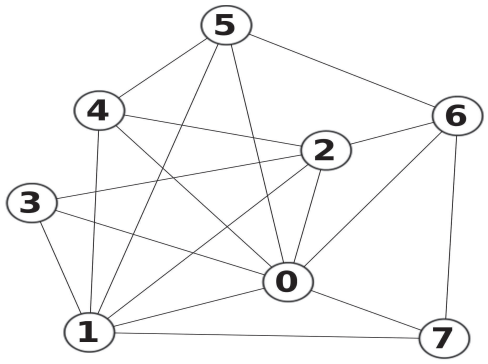
Corollary 2. $e_1 \in \mathcal{S}(e_2) \Leftrightarrow \mathcal{S}(e_1) = \mathcal{S}(e_2)$, $\forall e_1, e_2 \in \mathcal{E}$

Corollary 3. $\mathcal{S}(e_1) \neq \mathcal{S}(e_2) \Leftrightarrow \mathcal{S}(e_1) \cap \mathcal{S}(e_2) = \emptyset$

The properties presented in the above corollaries are demonstrated in Appendix A.

4.4 Derivation of potential anomaly scenarios

Theorem 2 states that the set of suspect links returned at the end of the detection phase whenever an anomaly on link e occurs is $\cap_{p \in D_{e_+}} p - \cup_{p \in D_{e_-}} p$. Therefore, instead of



(a)

Monitor locations	nodes 0, 1 and 7
Detection Paths	$\langle(0,7)\rangle$
	$\langle(0,1)\rangle$
	$\langle(0,4),(4,1)\rangle$
	$\langle(0,2),(2,3),(3,1),(1,7)\rangle$
	$\langle(0,6),(6,5),(5,4),(4,2),(2,1)\rangle$
	$\langle(1,5),(5,0),(0,3),(3,2),(2,6),(6,7)\rangle$

(b)

Figure 4.1: Illustrative network topology, (a), and an associated detection solution, (b).

computing monitors that are to be activated and paths that are to be monitored during the localization phase whenever an anomaly is detected, we propose to perform these computations for all potential anomalies only once offline. Having a set of detection paths that cover all links of the network, we infer the set of suspect links for all potential anomalies as described in Theorem 2. Then, a single anomaly scenario is created for all links that have the same set of suspect links, *i.e.*, an anomaly scenario is created for each distinct set of suspect links. Let us denote by $\mathcal{A}$ the set of all anomaly scenarios, and let $\mathcal{S}_a$ denotes the set of suspect links associated to the anomaly scenario $a \in \mathcal{A}$. Let $dS = \{\mathcal{S}_a, \forall a \in \mathcal{A}\}$. dS have the following properties.

Corollary 4. $\cup_{e\in\mathcal{E}}\mathcal{S}(e) = \cup_{\mathcal{S}(i)\in dS}\mathcal{S}(i) = \mathcal{E}$

Corollary 5. $\sum_{\mathcal{S}(i)\in dS} \mid \mathcal{S}(i) \mid = \mid \mathcal{E} \mid$

Clearly, an upper bound of the number of anomaly scenarios, whatever the topology of network and whatever the detection solution, is the number of the network links. It is easy

Table 4.1: Sets of suspect links for all potential anomalies

Anomalous link	**Set of suspect links**
$(0, 1)$	{(0, 1)}
$(0, 2)$	{(0, 2), (1, 3), (1, 7)}
$(1, 2)$	{(0, 6), (5, 6), (4, 5), (2, 4), (1, 2)}
$(0, 3)$	{(1, 5), (0, 5), (0, 3), (2, 6), (6, 7)}
$(1, 3)$	{(0, 2), (1, 3), (1, 7)}
$(2, 3)$	{(2, 3)}
$(0, 4)$	{(0, 4), (1, 4)}
$(1, 4)$	{(0, 4), (1, 4)}
$(2, 4)$	{(0, 6), (5, 6), (4, 5), (2, 4), (1, 2)}
$(0, 5)$	{(1, 5), (0, 5), (0, 3), (2, 6), (6, 7)}
$(1, 5)$	{(1, 5), (0, 5), (0, 3), (2, 6), (6, 7)}
$(4, 5)$	{(0, 6), (5, 6), (4, 5), (2, 4), (1, 2)}
$(0, 6)$	{(0, 6), (5, 6), (4, 5), (2, 4), (1, 2)}
$(2, 6)$	{(1, 5), (0, 5), (0, 3), (2, 6), (6, 7)}
$(5, 6)$	{(0, 6), (5, 6), (4, 5), (2, 4), (1, 2)}
$(0, 7)$	{(0, 7)}
$(1, 7)$	{(0, 2), (1, 3), (1, 7)}
$(6, 7)$	{(1, 5), (0, 5), (0, 3), (2, 6), (6, 7)}

Table 4.2: Anomaly scenarios

Anomaly scenario	**Set of suspect links**
a_1	S_{a_1} = {(0, 2), (1, 3), (1, 7)}
a_2	S_{a_2} = {(0, 6), (5, 6), (4, 5), (2, 4), (1, 2)}
a_3	S_{a_3} = {(1, 5), (0, 5), (0, 3), (2, 6), (6, 7)}
a_4	S_{a_1} = {(0, 4), (1, 4)}

to show that when this bound is reached, the set of suspect links for an anomaly on link e, $\forall e \in \mathcal{E}$, is reduced to the link e. In such case, the localization of all potential anomalies is immediate from the detection information. According to Corollary 2, we need to deploy

monitors that enable the monitoring of a set of paths distinguishing links of each anomaly scenario pairwise in order to ensure the localization of all potential anomalies.

To illustrate, consider the sample network topology depicted in Figure 4.1(a). An associated anomaly detection solution that covers all links of the network is depicted in Figure 4.1(b). We use Theorem 2 to compute the set of suspect links for all potential anomalies. The result is depicted in Table 4.1. The sets of suspect links associated to link $(2, 3)$ and link $(0, 7)$ are unitary. When an anomaly occurs on one of these two links, there is no need to trigger the localization phase because the anomalous link is immediately pinpointed by intersecting the detection paths that exhibit the anomaly. Furthermore, four non-unitary anomaly scenarios (a_1, a_2, a_3, a_4) are created for this topology (see table 4.2). These are the four distinct non-unitary sets of suspect links.

Let $AllPairs$ denotes the number of all the network link pairs. Clearly, $AllPairs = (\mid \mathcal{E} \mid (\mid \mathcal{E} \mid -1))/2$. Let $dPairs$ denotes the number of pair of links that need be distinguishable for localizing any potential link-level anomaly.

Corollary 6. $dPairs = AllPairs$ - $\sum_{\mathcal{S}(i),\mathcal{S}(j)\in dS:i<j} \mid \mathcal{S}(i) \mid\mid \mathcal{S}(j) \mid$

Corollary 6 confirms that we do not need to distinguish between all the network link pairs unless the number of detection paths equals 1, which is very unlikely.

The proofs of Corollary 4, Corollary 5 and Corollary 6 are described in Appendix A.

4.5 Anomaly localization cost

Consider a set of candidate monitor locations, $\mathcal{M}$, a set of network paths that are candidate to be monitored, $\mathcal{P}$, and a set of anomaly scenarios $\mathcal{A}$. The anomaly localization cost includes two costs:

– *Monitor cost*: it includes the effective cost of acquiring hardware and software monitoring devices and the cost of their maintenance. In addition, it includes the cost of communications between the monitors and the NOC. For instance, the cost of communications between a monitor and the NOC can be expressed as a function of the number of routing hops that separates them. Let us denote by C_n the cost of deploying a monitor on node n. Let Y_n be a binary variable that indicates whether node n is selected to hold a monitoring device. The monitor cost can be expressed

as follows:

$$\sum_{n \in M} C_n Y_n \tag{4.1}$$

– *Probe cost*: it expresses the overhead of monitoring flows on the underlying network. Measurements of links that do not provide localization information should be avoided in order to minimize the monitoring overhead. Clearly, measuring links that do not belong to the set of suspect links of an anomaly scenario does not provide any extra localization information. Furthermore, measurement of links that belong to the set of suspect links might be useless. Revisit Figure 4.1 and table 4.1 to illustrate. Consider an anomaly on link $(6,7)$. The associated set of suspect links is $\mathcal{S}_{a_3} = \{(1,5),(0,5),(0,3),(2,6),(6,7)\}$. Consider now the set of localization paths $\{p_1{:}\langle(1,5)(5,6)(2,6)\rangle;\ p_2{:}\langle(1,5)(0,5)(0,2)\rangle;\ p_3{:}\langle(1,7)(6,7)(2,6)\rangle\}$ that distinguishes between all the links of $\mathcal{S}_{a_3}$ pairwise. Path p_1 divides $\mathcal{S}_{a_3}$ into two subsets: $\mathcal{S}^1_{a_3}\{(1,5),(2,6)\}$ and $\mathcal{S}^2_{a_3}\{(0,5),(0,3),(6,7)\}$. p_1 distinguishes each link of $\mathcal{S}^1_{a_3}$ from each link of $\mathcal{S}^2_{a_3}$. Link $(5,6)$ that is traversed by p_1 does not belong to $\mathcal{S}_{a_3}$, and therefore, it does not provide any localization information. Path p_2 divides $\mathcal{S}^1_{a_3}$ into two subsets: $\mathcal{S}^{11}_{a_3}\{(1,5)\}$ and $\mathcal{S}^{12}_{a_3}\{(2,6)\}$, and divides $\mathcal{S}^2_{a_3}$ into two subsets: $\mathcal{S}^{21}_{a_3}\{(0,5),(6,7)\}$ and $\mathcal{S}^{22}_{a_3}\{(0,3)\}$. Finally, p_3 distinguishes between $(0,5)$ and $(6,7)$. However, it crosses $(2,6)$ that is already distinguished from all the other suspect links. Thus, measuring $(2,6)$ by p_3 does not provide extra localization information, although it belongs to $\mathcal{S}_{a_3}$.

Let us denote by C_e the cost of measuring link e. C_e should be proportional to the load of link e, in order to avoid multiple measurements of the most overloaded links of the network. Consider an anomaly scenario $a \in \mathcal{A}$. Let us denote by $\mathcal{S}_a$ the set of suspect links associated to the anomaly scenario a. Let X_{pa} be a binary variable that specifies whether path p is part of the localization solution of a. Let δ_{pe} be a binary input parameter that indicates whether path p crosses link e. The probe cost of the localization solution of a reads as follows:

$$\sum_{e \in \mathcal{E}, p \in \mathcal{P}'} C_e \delta_{pe} X_{pa} \tag{4.2}$$

4.6 ILP Formulation

The objective of the ILP is to find a localization solution for each anomaly scenario in $\mathcal{A}$ such that the anomaly localization cost is minimized. Let δ_{pn} be a binary parameter

that indicates whether node n is an end-node of path p. For simplicity of notation, we define the following sets:

- $\delta_{\mathcal{PE}} = \{\delta_{pe};\ p \in \mathcal{P}, e \in \mathcal{E}\}$
- $\delta_{\mathcal{PM}} = \{\delta_{pn};\ p \in \mathcal{P}, n \in \mathcal{M}\}$
- $C_{\mathcal{M}} = \{C_n;\ n \in \mathcal{M}\}$
- $C_{\mathcal{E}} = \{C_e;\ e \in \mathcal{E}\}$

Let α be the weight associated to the monitor cost, and let β be the weight associated to the probe cost. $\alpha, \beta \in \mathbb{R}$. The input into the ILP is an instance of the graph $G = (\mathcal{E}, \mathcal{M}, \mathcal{P}, \mathcal{A}, \delta_{\mathcal{PE}}, \delta_{\mathcal{PM}}, C_{\mathcal{E}}, C_{\mathcal{M}}, \alpha, \beta)$. The objective function minimizes the sum of the monitor cost and the probe cost. It reads as follows:

$$\alpha \sum_{n \in M} C_n Y_n + \beta \sum_{a \in \mathcal{A}, e \in \mathcal{E}, p \in \mathcal{P}} C_e \delta_{pe} X_{pa} \tag{4.3}$$

The ILP is subject to two constraints. The first constraint ensures that either end node of each selected monitoring paths is selected as monitor location. It reads as follows:

$$Y_n \geq \delta_{pn} X_{pa};\ \forall n \in \mathcal{M}, \forall p \in \mathcal{P}, \forall a \in \mathcal{A} \tag{4.4}$$

The second constraint ensures that the suspect links associated to each anomaly scenario are distinguishable pairwise. To this end, according to Theorem 2, the constraint ensures that for each anomaly scenario a and for each pair of suspect links $(e_1, e_2) : e_1, e_2 \in S_a$ there exists at least one monitoring path that crosses either e_1 or e_2, but not both. This constraint reads as follows:

$$\sum_{p \in \mathcal{P}} (\delta_{pe_1} + \delta_{pe_2} - 2\delta_{pe_1}\delta_{pe_2}) X_{pa} > 0;\quad \forall a \in \mathcal{A}; \forall e_1, e_2 \in \mathcal{S}_a \tag{4.5}$$

We show that the above inequality is sufficient to distinguish between all the link pairs of each anomaly scenario using the argument of the following theorem.

Theorem 3. *Let P_1 be the subset of paths of $\mathcal{P}$ that cross either e_1 or e_2, but not both. $\sum_{p \in \mathcal{P}} (\delta_{pe_1} + \delta_{pe_2} - 2\delta_{pe_1}\delta_{pe_2}) =\mid P_1 \mid$.*

Proof. Refer to Appendix B. □

Corollary 7. ***If*** *$\sum_{p \in \mathcal{P}} (\delta_{pe_1} + \delta_{pe_2} - 2\delta_{pe_1}\delta_{pe_2}) X_{pa} > 0$,* ***then*** *there exists at least one path in $\mathcal{P}$ that crosses either e_1 or e_2 but not both,* ***then*** *there exists at least one path in $\mathcal{P}$ that distinguishes between e_1 and e_2.*

4.7 The Anomaly Localization Problem is $\mathcal{NP}$-Hard

Theorem 4. *The anomaly localization problem presented in the previous section is $\mathcal{NP}$-Hard.*

Proof. Our formulation of the anomaly localization problem can be reduced from the $\mathcal{NP}$-Hard facility location problem.

Facility location problem [30]: consider a set of potential facility locations $\mathcal{F}$, and a set of clients $\mathcal{D}$. Opening a facility at location i incurs a non-negative cost that is equal to f_i. The cost of servicing client $j \in \mathcal{D}$ by a facility installed at location $i \in \mathcal{F}$ is d_{ij}. The problem is to find an assignment of each client to exactly one facility such that the sum of the facility opening costs and the service costs is minimized.

We denote by f the set of facility opening costs, $f = \{f_i,\ i \in \mathcal{F}\}$, and by d the set of service costs, $d = \{d_{ij};\ i \in \mathcal{F}, j \in \mathcal{D}\}$. Given an instance $\mathcal{I} = (\mathcal{D}, \mathcal{F}, f, d)$ of the facility location problem, we produce an instance $\mathcal{R}(\mathcal{I}) = (\mathcal{E}, \mathcal{M}, \mathcal{P}, \mathcal{A}, \delta_{\mathcal{PE}}, \delta_{\mathcal{PM}}, C_{\mathcal{E}}, C_{\mathcal{M}}, \alpha, \beta)$ of the localization problem as follows. For each client $j \in \mathcal{D}$, we create:

- Three nodes labeled by n_{j1}, n_{j2}, and n_{j3}.
- One link connecting n_{j1} to n_{j2}, labeled by e_{j1}.
- One link connecting n_{j2} to n_{j3}, labeled by e_{j2}.
- An anomaly scenario a_j such that $S_{a_j} = \{e_{j1}, e_{j2}\}$.

For each facility location $i \in \mathcal{F}$, we create two nodes labeled by m_{i1} and m_{i2}. For each $i \in \mathcal{F}$ and for each $j \in \mathcal{D}$, we create one link connecting m_{i1} to n_{j1}, labeled by e^1_{ij}, and one link connecting m_{i2} to n_{j2}, labeled by e^2_{ij}. We obtain a graph $\mathcal{G} = (\mathcal{E}, \mathcal{N})$, where $\mathcal{N} = \{n_{ik}; i \in \mathcal{D}, k \in [1;3]\} \cup \{m_{jk}; i \in \mathcal{F}, k \in [1;2]\}$, and $\mathcal{E} = \{e_{jk}; j \in \mathcal{D}, k \in [1;3]\} \cup \{e^k_{ij}; i \in \mathcal{F}, j \in \mathcal{D}, k \in [1;2]\}$. An example of a graph constructed out of a facility location instance with four facility locations and four clients is shown in Figure 4.2.

The candidate monitor location set is $\mathcal{M} = \{m_{jk}; i \in \mathcal{F}, k \in [1;2]\}$. The set of anomaly scenarios is $\mathcal{A} = \{a_j; j \in \mathcal{D}\}$. The set of candidate localization paths is $\mathcal{P} = \{p_{ij}; i \in \mathcal{F}, j \in \mathcal{D}\}$, where p_{ij} is the non-looping path between m_{i1} and m_{i2} that crosses the links e^1_{ij}, e_{j1} and e^2_{ij}. The monitor deployment costs are defined as follows: $C_{m_{i1}} = C_{m_{i2}} = f_i/2$. The link measurement costs are defined as follows: $C_{e_{i1}} = C_{e_{i2}} = 0$, $C_{e^1_{ij}} = C_{e^2_{ij}} = d_{ij}/2$. The remaining input parameters can be inferred easily from $\mathcal{G}$, $\mathcal{M}$, $\mathcal{A}$ and $\mathcal{P}$ as follows:

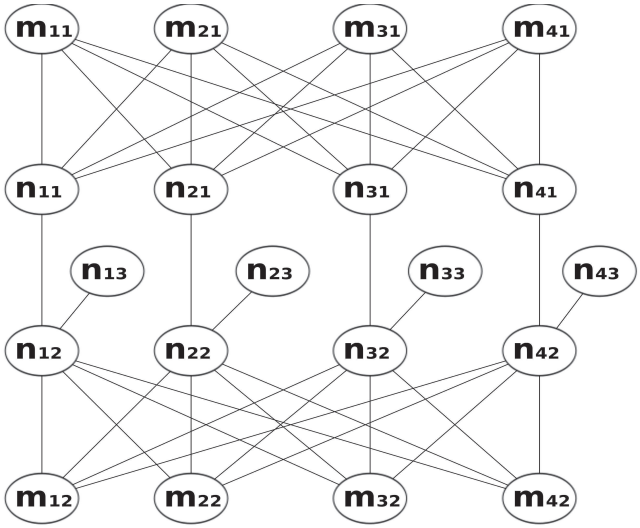


Figure 4.2: Example of a graph constructed out of a facility location instance with four facility locations and four clients

---

- $\delta_{a_j e_{j'k}} = \begin{cases} 1 & \text{if } j = j' \\ 0 & \text{otherwise} \end{cases} \quad ; \quad \forall j, j' \in \mathcal{D}, k \in [1;2]$
- $\delta_{a_j e^k_{ij}} = 0; \quad \forall i \in \mathcal{F}, j \in \mathcal{D}, k \in [1;2]$
- $\delta_{p_{ij} m_{i'k}} = \begin{cases} 1 & \text{if } i = i' \\ 0 & \text{otherwise} \end{cases} \quad ; \quad \forall i, i' \in \mathcal{F}, k \in [1;2]$
- $\delta_{p_{ij} e_{j1}} = \delta_{p_{ij} e^1_{ij}} = \delta_{p_{ij} e^2_{ij}} = 1; \quad \forall i \in \mathcal{F}, j \in \mathcal{D}$
- $\delta_{p_{ij} e_{j2}} = 0; \quad \forall i \in \mathcal{F}, j \in \mathcal{D}$
- $\alpha = \beta = 1$

It can be easily shown that the time complexity of the above reduction is $\mathcal{O}(\mid \mathcal{F} \mid \times \mid \mathcal{D} \mid)$, and therefore, it can be carried out in polynomial-time. In the sequel, we show that there is an optimal solution to the Instance $\mathcal{I}$ of the facility location problem if and only of there is an optimal solution to the instance $\mathcal{R}(\mathcal{I})$ of our anomaly localization problem.

Let us start by demonstrating that if there is an optimal solution to the facility location instance, then there is a feasible solution to the anomaly localization instance. Let the facility location solution assign each client j to a facility installed at location i. Consider the anomaly localization solution that selects for each anomaly scenario a_j the path p_{ij} and the monitor locations m_{i1} and m_{i2}. Then, let us fix an anomaly scenario a_j. By construction, path p_{ij} crosses three links that are e_{j1} and e^1_{ij} and e^2_{ij}. It follows, according

to Theorem 1, that p_{ij} distinguishes between e_{j1} and e_{j2}. Constraint (4.4) states that if p_{ij} is selected to be monitored, then, its end nodes must be selected to hold monitoring devices. Thus, the solution that selects for each anomaly scenario a_j the path p_{ij} to be monitored, and its end nodes, m_{i1} and m_{i2}, as monitor locations is a feasible solution to the anomaly localization instance.

Conversely, we demonstrate that if there is an optimal solution to the anomaly localization instance, then there is a feasible solution to the facility location instance. An optimal solution to the facility location problem selects exactly one path for each anomaly scenario. This is because, by construction, for each anomaly scenario $a_i \in \mathcal{A} \mid \mathcal{S}_{a_i} \mid = 2$. Thus, monitoring one path that crosses exactly one of the two links is sufficient to distinguish between them. Let the optimal anomaly localization solution selects for each anomaly scenario a_j the path p_{ij}, and naturally, the monitor locations m_{i1} and m_{i2}. Trivially, the solution that assigns to each client $j \in \mathcal{D}$ the facility installed at location i is a feasible solution to the facility location instance.

We now prove that the constructed anomaly localization solution has the same cost as its corresponding optimal facility location solution (the proof holds in the converse case). Let W_i be a binary variable that indicates whether a facility is installed at location i, and let Z_{ij} be a binary variable that indicates whether client j is serviced by a facility installed at location i. Using the arguments that $Z_{ij} = X_{p_{ij}a_j}$ and $W_i = Y_{m_{i1}} = Y_{m_{i2}}$ [3], we show that the cost of the localization solution, denoted by $Cost(S_{\mathcal{R}(\mathcal{I})})$, is equal to the cost of its corresponding facility location solution, denoted by $Cost(S_{\mathcal{I}})$, as follows:

$$
\begin{aligned}
Cost(S_{\mathcal{R}(\mathcal{I})}) &= \alpha \sum_{m_{ik} \in \mathcal{M}} C_{m_{ik}} Y_{m_{ik}} + \beta \sum_{a_j \in \mathcal{A}, e \in \mathcal{E}, p_{ij} \in \mathcal{P}} C_e X_{p_{ij}a_j} \\
&= \sum_{m_{ik} \in \mathcal{M}} C_{m_{ik}} Y_{m_{ik}} + \sum_{a_j \in \mathcal{A}, p_{ij} \in \mathcal{P}} (C_{e_{ij}^1} + C_{e_{ij}^2}) X_{p_{ij}a_j} \\
&= \sum_{m_{i1} \in \mathcal{M}} f_i Y_{m_{i1}} + \sum_{a_j \in \mathcal{A}, p_{ij} \in \mathcal{P}} d_{ij} X_{p_{ij}a_j} \\
&= \sum_{i \in \mathcal{F}} f_i W_i + \sum_{j \in \mathcal{D}, i \in \mathcal{F}} d_{ij} Z_{ij} \\
&= Cost(S_{\mathcal{I}})
\end{aligned}
$$

Now, we show that the solution to the anomaly localization instance, denoted by $S_{R(I)}$, that is constructed out of an optimal solution to the facility location instance, denoted by S_I^*, is optimal. Assume to the contrary that $S_{R(I)}$ is not optimal. Let $S_{R(I)}^{\prime *}$ be an

3. Recall that X_{pa} is a binary variable that indicates whether path p is part of the localization solution of the anomaly scenario a, and Y_n is a binary variable that indicates whether node n is selected as a monitor location

optimal solution to the anomaly localization instance, and let $S_I^{'}$ be the facility location solution constructed out of $S_{R(I)}^{'*}$. We have $Cost(S_I^*) = Cost(S_{R(I)}) < Cost(S_{R(I)}^{'*}) = Cost(S_I^{'})$, leading to a contradiction. Using the same arguments, we can show that the solution to the facility location instance constructed out of an optimal solution to the anomaly localization instance is optimal. □

4.8 Heuristic solution

In this section, we provide a monitor location and path selection algorithm for localizing single link-level anomalies. The inputs of the algorithm are a network graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, a set of anomaly scenarios $\mathcal{A}$, a set of candidate monitor locations $\mathcal{M}$, the costs of deploying monitoring devices on the network nodes $C_{\mathcal{M}} = \{C_n;\ n \in \mathcal{M}\}$, and the costs of monitoring the network links $C_{\mathcal{E}} = \{C_e;\ e \in \mathcal{E}\}$. The outputs are a set of monitor locations, $\mathcal{SM}_a$, and a set of monitoring paths, $\mathcal{SP}_a$, that can distinguish between all links of $\mathcal{S}_a$ pairwise, for each $a \in \mathcal{A}$.

Similarly to the ILP, the heuristic solution aims at minimizing the infrastructure cost, the communication cost and the probe cost jointly. To this end, we use a nested greedy approach that selects monitor locations jointly with monitoring paths. Algorithm 4 describes the pseudo-code. ProbeCost$(p, C_{\mathcal{E}})$ is a function that returns the probe cost incurred by monitoring path p. This cost is computed as described in section 4.5. m_s stores the best current candidate monitor location. $\mathcal{SM}$ stores the monitor locations selected at the previous iterations. $minPcost$ stores the current lowest probe cost, and $maxlc$ stores the current largest localization capacity, *i.e.*, the number of link pairs that can be distinguished by monitors in $\mathcal{SM} \cup \{m_s\}$. CP stores paths selected by the current best solution. In the sequel, we define the criteria of monitor location selection and monitoring path selection.

A detailed description of how monitor locations and monitoring paths are selected, and how candidate localization paths are computed is provided in the following subsections.

4.8.1 Monitor location selection

The algorithm starts by selecting one candidate monitor location randomly. Alternatively, the candidate monitor location with the smallest monitor cost (sum of the infrastructure cost and the communication cost) can be selected. However, we advocate random selection for two reasons. The first is that the monitor location with the smallest monitor cost does not necessarily incur the smallest probe cost. The second is that selecting the

starting point randomly enlarges the space of explored solutions over multiple runs of the algorithm. Monitor locations are, then, added to the solution greedily until all link pairs of all the anomaly scenarios are distinguished.

At each greedy iteration (lines 5-26), all the remaining candidate monitor locations are explored. Let us fix a candidate monitor location m. A set of monitoring paths whose end nodes are in $\mathcal{SM} \cup \{m\}$ is selected greedily (lines 7-26). The path selection procedure is described in details in section 4.8.2. The candidate monitor location whose associated monitoring paths can distinguish between the largest number of link pairs over all the anomaly scenarios is selected (line 24). In case of a tie, a monitor location that incurs the smallest localization cost ($\alpha \times$ monitor cost + $\beta \times$ probe cost, where the probe cost is the summation of the probe costs of the associated monitoring paths) is selected.

When a solution that distinguishes between all the link pairs of all the anomaly scenarios is found, the algorithm continues the exploration of the remaining candidate monitor locations, if any, towards reducing the probe cost. However, a filter is applied on these locations before exploring them (line 6). Only candidate locations whose monitor cost is smaller than the probe cost of the current best solution are explored. Clearly, the localization cost of any solution that selects a monitor location not satisfying this filter would be larger than the localization cost of the current best solution. The algorithm ends when the set of candidate monitor locations gets empty, *i.e.*, all candidate monitor locations have been selected, or when remaining candidate monitor locations can neither improve the localization capacity nor the probe cost of the current best solution.

4.8.2 Selection of localization paths

Given a candidate monitor location m and a set of already selected monitor locations $\mathcal{SM}$, the procedure of selecting an associated set of monitoring paths, (lines 7-26), is as follows. Let us fix an anomaly scenario a. A set of monitoring paths that maximizes the number of distinguished pair of links of $\mathcal{S}_a$ while minimizing the probe cost is selected greedily as follows. First, one path that can distinguish between the largest number of link pairs of $\mathcal{S}_a$ is selected. We refer to the number of pair of links of a set of suspect links $\mathcal{S}_a$ that can be distinguished by a path p as the localization capacity of p with respect to $\mathcal{S}_a$, denoted by $lc(p, \mathcal{S}_a)$. It can be easily shown that:

$$lc(p, \mathcal{S}_a) = \mid p \cap \mathcal{S}_a \mid (\mid \mathcal{S}_a \mid - \mid p \cap \mathcal{S}_a \mid) \tag{4.6}$$

Algorithm 4: Monitor location and path selection algorithm for single anomaly localization

1 nbPairs = $\sum_{a\in\mathcal{A}}\sum_{k=1}^{|\mathcal{S}_a|-1} k$; minPcost ← INT_MAX; $maxlc \leftarrow 0$; $\mathcal{CP} \leftarrow \emptyset$;

2 $\mathcal{SM} \leftarrow \{$`selectRandomElement`$(\mathcal{M})\}$; Remove the selected monitor location from $\mathcal{M}$;

3 **while** $(\mathcal{M} \neq \emptyset)$ **do**

4 Reset $m_s \leftarrow Null$;

5 **foreach** $(m \in \mathcal{M})$ **do**

6 **if** $((\ maxlc = nbPairs$ and $\beta minPCost \leq\ \alpha C_m))$ **then** Jump to line 5;

7 Reset $lc \leftarrow 0$; Reset Pcost ← 0;

8 **for** $(a \in \mathcal{A})$ **do**

9 Clear $\mathcal{P}_a$; Clear $\mathcal{M}_a$; $j \leftarrow 1$; $s_{(0)} \leftarrow 1$; $\mathcal{S}_a^{(0)1} \leftarrow \mathcal{S}_a$;

10 **while** $(s_{(j)} > 0)$ **do**

11 $\mathcal{S}_a^{(j)} \leftarrow \{\mathcal{S}_a^{(j)1}, ...\ \mathcal{S}_a^{(j)k}, \mathcal{S}_a^{(j)k+1}\ ..., \mathcal{S}_a^{(j)s_{(j)}}\}$;

12 $p_{a(j)} \leftarrow$ CandidatePathSelection$(m, \mathcal{SM}, \mathcal{G}, \mathcal{S}_a^{(j)}, \mathcal{CP})$;

13 $lc \mathrel{+}= \sum_{1\leq k\leq s_{(j)}}$ `lc`$(p_{a(j)}, \mathcal{S}_a^{(j)k})$;

14 Pcost $\mathrel{+}= ProbeCost(p_{a(j)}, C_{\mathcal{E}})$;

15 $l \leftarrow 1$;

16 **for** $(1 \leq k \leq s_{(j)})$ **do**

17 **if** $(|\ p_{a(j)} \cap \mathcal{S}_a^{(j)k}\ | > 1)$ **then** $\mathcal{S}_a^{(j+1)l} \leftarrow p_{a(j)} \cap \mathcal{S}_a^{(j)k}$; $l \leftarrow l + 1$;

18 **if** $(|\ \mathcal{S}_a^{(j)k} - \{p_{a(j)} \cap \mathcal{S}_a^{(j)k}\}\ | > 1)$ **then**
$\mathcal{S}_a^{(j+1)l+1} = \mathcal{S}_a^{(j)k} - \{p_{a(j)} \cap \mathcal{S}_a^{(j)k}\}$; $l \leftarrow l + 1$;

19 $s_{(j)} \leftarrow l - 1$;

20 **if** $(maxlc = nbPairs$ and $(\alpha C_m + \beta(PCost + \sum_{l=1}^{s_{(j+1)}}$
$ThMinPCost(\mathcal{S}_a^{(j+1)l})\ + \sum_{a'\in\mathcal{A},a'>a} ThMinPCost(\mathcal{S}_{a'}))\ \geq$

21 $\alpha C_{m_s} + \beta minPCost))$ **then**

22 /*Stop the exploration of the current candidate monitor location*/
Jump to line 5;

23 Add the end nodes of $p_{a(j)}$ to $\mathcal{M}_a$; Add $p_{a(j)}$ to $\mathcal{P}_a$; $j \leftarrow j + 1$;

24 **if** $(lc > maxlc$ or $(lc = maxlc$ and $\alpha C_m + \beta Pcost < \alpha C_{m_s} + \beta minPcost))$ **then**

25 $m_s \leftarrow m$; $maxlc \leftarrow lc$; $minPCost \leftarrow PCost$; $\mathcal{SP}_a \leftarrow \mathcal{P}_a$; $\mathcal{SM}_a \leftarrow \mathcal{M}_a$;

26

27 **if** $(m_s = Null)$ **then return** $(\{\mathcal{SP}_a, \mathcal{SM}_a\};\ \ \forall a \in \mathcal{A})$

28 Update $\mathcal{CP} \leftarrow \bigcup_{a\in\mathcal{A}} \mathcal{SP}_a$; Add m_s to $\mathcal{SM}$; Remove m_s from $\mathcal{M}$;

29 **return** $(\{\mathcal{SP}_a, \mathcal{SM}_a;\ \ \forall a \in \mathcal{A}\})$;

In case of a tie, a path that minimizes the probe cost is selected. The algorithm used for computing the candidate monitoring path is described in section 4.8.3. Let $p_{a(1)}$ be the selected path. Two subsets of suspect links are generated: $\mathcal{S}_a^{(1)1} = \mathcal{S}_a \cap p_{a(1)}$ and $\mathcal{S}_a^{(1)2} = \mathcal{S}_a - \{\mathcal{S}_a \cap p_{a(1)}\}$. According to Theorem 1, $p_{a(1)}$ distinguishes between every pair of links (e_1, e_2) such that $e_1 \in \mathcal{S}_a^{(1)1}$ and $e_2 \in \mathcal{S}_a^{(1)2}$. At the next step, we need to distinguish between the links of $\mathcal{S}_a^{(1)1}$ pairwise and between the links of $\mathcal{S}_a^{(1)2}$ pairwise. Hence, a path that maximizes $lc(p, \mathcal{S}_a^{(1)1}) + lc(p, \mathcal{S}_a^{(1)1})$ is selected. Ties are broken by selecting a path that minimizes the probe cost.

Let $p_{a(j)}$ be the monitoring path selected at step (j). Let $s_{(j-1)}$ be the number of non-unitary subsets of suspect links generated at step $(j-1)$. $p_{a(j)}$ is selected such that $\sum_{1 \leq k \leq s_{(j-1)}} lc(p,\ \mathcal{S}_a^{(j-1)k})$ is maximized. In case of a tie, a path that minimizes the probe cost is selected. For each $\mathcal{S}_a^{(j-1)k}$, $1 \leq k \leq s_{(j-1)}$, two subsets of suspect links are generated: $\mathcal{S}_a^{(j-1)k} \cap p_{a(j)}$ and $\mathcal{S}_a^{(j-1)k} - \{\mathcal{S}_a^{(j-1)k} \cap p_{a(j)}\}$. Each link of the former subset is distinguished from each link of the latter subset. Only non-unitary subsets, whose links need to be distinguished from each other, are considered at the next step. This greedy process is re-iterated until all the generated subsets of suspect links are unitary or until no candidate localization path can distinguish between the pair of links of non-unitary subsets. For each selected path $p_{a(j)}$, the localization capacity of m is incremented by $lc(p_{a(j)}, \mathcal{S}_a^{(j)})$ (line 13), and its probe cost is incremented by $probeCost(p_{a(j)}, C_{\mathcal{E}})$ (line 14). The above procedure is applied on the all the anomaly scenarios in $\mathcal{A}$. Then, the localization capacity and the probe cost of m are evaluated (line 24). If the localization capacity of m is greater than $maxlc$, or if the localization capacity of m equals $maxlc$ and its probe cost is less than $minPcost$; then $maxlc$ is set equal to the localization capacity of m, $minPcost$ is set equal to the probe cost of m and m_s is set equal to m.

Furthermore, using the argument of the following theorem, we can compute a lower bound of the probe cost of the explored monitor location at any step in the path selection procedure.

Theorem 5. *The theoretical minimal probe cost relative to a set of suspect links $\mathcal{S}$ denoted by $ThMinPcost(\mathcal{S})$ reads as follows:*

$$ThMinPcost(S) = \sum_{e \in \mathcal{S}} C_e - \max_{e \in \mathcal{S}} C_e \tag{4.7}$$

Proof. Refer to Appendix C. □

The lower bound of the probe cost of a candidate monitor location after path $p_{a(j)}$ is added to the set of its associated monitoring paths reads as follows:

$$Pcost \; + \; \sum_{k=1}^{s_{(j)}} ThMinPcost(\mathcal{S}_a^{(j)k}) \; + \sum_{a' \in \mathcal{A}, a' > a} ThMinPcost(\mathcal{S}_a^{'}), \tag{4.8}$$

where $Pcost$ is the summation of the probe costs of the already selected paths.

When the algorithm finds a solution that can distinguish between all link pairs of all the anomaly scenarios, it continues exploring the remaining candidate monitor locations that satisfy the monitor cost filter (line 6) towards reducing the probe cost. Using (4.8), we propose an optimization of the exploration process of these candidate monitor locations. The idea is to update the lower bound of the probe cost of the explored monitor location whenever a monitoring path is selected, and to infer a lower bound of the localization cost (line 20). The exploration of the considered candidate monitor location is abandoned if, at any step of the path selection procedure, the calculated lower bound of the localization cost dominates the localization cost of the current best solution.

4.8.3 Candidate path selection algorithm

This section describes the procedure *candidatePathSelection* called by Algorithm 4 at line 12. The inputs into this procedure are the network graph, the currently explored monitor location m, the subsets of suspect links generated at the current step of the path selection procedure $\mathcal{S}_a^{(j)} = \{\mathcal{S}_a^{(j)1}, ... \; \mathcal{S}_a^{(j)k}, \mathcal{S}_a^{(j)k+1} \; ..., \mathcal{S}_a^{(j)s_{(j)}}\}$, the set of the already selected monitor locations $\mathcal{SM}$, and the set of monitoring paths selected by the current best solution $\mathcal{CP}$. The output is one monitoring path, whose end nodes are in $\mathcal{SM} \cup \{m\}$, that maximizes the localization capacity while minimizing the probe cost.

The main difficulty of this procedure is the computation of the set of candidate paths. Generally, the smaller the set of candidate paths is, the worst the quality of the heuristic is. This is because good paths might be missed when reducing the number of candidate paths. However, this reduction is imperative to ensure the scalability of the heuristic. The procedure *candidatePathSelection* implements an algorithm for candidate localization path computation. The algorithm considers all the network paths whose end nodes belong to $\{m\} \times \mathcal{SM}$ as candidate to be monitored. However, computing this set of paths is computationally expensive, because it requires exploring all the network graph. Moreover, since Algorithm 4 explores all remaining candidate monitor locations at each iteration, the graph would be explored multiple times; which makes the heuristic non-scalable and

Procedure 2: candidatePathSelection($m, \mathcal{SM}, \mathcal{G}, \mathcal{S}_a^{(j)}, \mathcal{CP}$)

1 $p_c \leftarrow$ newPath();
2 $minli \leftarrow \sum_{\mathcal{S}_a^{(j)k} \in \mathcal{S}_a^{(j)}} \mid \mathcal{S}_a^{(j)k} \mid /2 - 1$; $minPcost \leftarrow \sum_{e \in \mathcal{E}} C_e$;
3 **foreach** $q \in \mathcal{CP}$ **do**
4 $li = \sum_{\mathcal{S}_a^{(j)k} \in \mathcal{S}_a^{(j)}} \mid \mathcal{S}_a^{(j)k} \mid /2 - \mid \mathcal{S}_a^{(j)k} \cap q \mid$;
5 **if** ($li < minli$ or ($li = minli$ and $probeCost(p_c, C_\mathcal{E}) < minPc$)) **then** $minli = li$;
  $minPc = probeCost(p_c, C_\mathcal{E})$; $p_s = q$;
6 add-node-to-path(m, p_c);
7 `depthFirst` (m, p_c)**{**
8 **foreach** ($n \in children(m, \mathcal{G})$ and $(m, n) \notin p_c$) **do**
9 add-node-to-path(n, p_c);
10 $li(p_c, \mathcal{S}_a^{(j)}) = \sum_{\mathcal{S}_a^{(j)k} \in \mathcal{S}_a^{(j)}} absoluteValue(\mid \mathcal{S}_a^{(j)k} \mid /2 - \mid \mathcal{S}_a^{(j)k} \cap p_c \mid)$;
11 **if** ($n \in \mathcal{SM}$) **then**
12 **if** ($li(p_c, \mathcal{S}_a^{(j)}) < minli$ or ($li(p_c, \mathcal{S}_a^{(j)}) = minli$ and $probeCost(p_c, C_\mathcal{E}) <= minPc$)) **then**
13 $p_s \leftarrow p_c$; $minli \leftarrow li(p_c, \mathcal{S}_a^{(j)})$; $minPc = probeCost(p_c, C_\mathcal{E})$;
14 **if** ($minli = 0$ *and* $minPc = 0$) **then**
15 /*end the algorithm*/ Jump to line 23;
16 **else**
17 **if** (($minli = 0$ and ($probeCost(p_c, C_\mathcal{E}) + li(p_c, \mathcal{S}_a^{(j)}) - minli >= minPc$ or $\exists\ \mathcal{S}_a^{(j)k} \in \mathcal{S}_a^{(j)}$ *such that* $\mid \mathcal{S}_a^{(j)k} \cap p_c \mid > \mid \mathcal{S}_a^{(j)k} \mid /2$)) or ($li(p_c, \mathcal{S}_a^{(j)}) > minli$ and $\forall\ \mathcal{S}_a^{(j)k} \in \mathcal{S}_a^{(j)}$ $\mid \mathcal{S}_a^{(j)k} \cap p_c \mid > \mid \mathcal{S}_a^{(j)k} \mid /2$)) **then**
18 do not explore the descendants of n;
19 **else**
20 Recursively call `depthFirst` (n, p_c);
21 **}**
22 return p_s;

non-practical for dense networks. An alternative solution is to compute and store all paths traveling between all candidate monitor locations offline, thereby reducing the number of times the network graph is explored to one. Clearly, this solution is impractical due to memory issues. We conclude, based on the above discussion, that our candidate path computation algorithm must minimize the number of paths that are to be explored, while guaranteeing that good candidate paths are not missed. To this end, we make use of the argument of the following theorem:

Theorem 6.
Let $absval(x)$ be a function that returns the absolute value of the number x, and let $lc(\mathcal{S}_a^{(j)}, p) = \sum_{\mathcal{S}_a^{(j)k} \in \mathcal{S}_a^{(j)}} lc(\mathcal{S}_a^{(j)k}, p)$ be the localization capacity of p with respect to $\mathcal{S}^{(j)}$. We have,

$$\max_{p \in \mathcal{P}} \; lc(\mathcal{S}_a^{(j)}, p) = \min_{p \in \mathcal{P}} \sum_{\mathcal{S}_a^{(j)k} \in \mathcal{S}_a^{(j)}} absval(\mid \mathcal{S}_a^{(j)k} \mid /2 - \mid \mathcal{S}_a^{(j)k} \cap p \mid) \tag{4.9}$$

Proof. Refer to Appendix D. □

We refer to $\sum_{\mathcal{S}_a^{(j)k} \in \mathcal{S}_a^{(j)}} absval(\mid \mathcal{S}_a^{(j)k} \mid /2 - \mid \mathcal{S}_a^{(j)k} \cap p \mid)$ as the localization indicator of path p with respect to $\mathcal{S}_a^{(j)}$, and we denote it by $li(p, \mathcal{S}_a^{(j)})$. According to Theorem 6, the smaller $li(p, \mathcal{S}_a^{(j)})$ is, the higher the localization capacity of p with respect to $\mathcal{S}_a^{(j)}$ is. The localization indicator is used along with the probe cost to avoid exploring all the network graph, while guaranteeing that good candidate paths are not missed. Procedure 2 provides an overview of the pseudo-code. p_s stores the current best candidate path, $minli$ stores the localization indicator of p_s, and $minPc$ stores its probe cost. p_s, $minli$ and $minPc$ are initialized to $Null$, $\sum_{\mathcal{S}_a^{(j)k} \in \mathcal{S}_a^{(j)}} \mid \mathcal{S}_a^{(j)k} \mid /2 - 1$ and $\sum_{e \in \mathcal{E}} C_e$, respectively. Note that the least upper bound of the localization indicator is $\sum_{\mathcal{S}_a^{(j)k} \in \mathcal{S}_a^{(j)}} \mid \mathcal{S}_a^{(j)k} \mid /2$, which corresponds to a path that does not provide any localization information (*i.e.*, $\forall \mathcal{S}_a^{(j)k} \in \mathcal{S}_a^{(j)}, \mathcal{S}_a^{(j)k} \cap p = \emptyset$ or $\exists \mathcal{S}_a^{(j)k} \in \mathcal{S}_a^{(j)}$ such that $p = \mathcal{S}_a^{(j)k}$) [4]; whereas the least upper bound of the probe cost is $\sum_{e \in \mathcal{E}} C_e$, which corresponds to a path that crosses all the network nodes and does not provide any localization information. However, if $\mathcal{CP}$ is not empty, then p_s is set equal to the best path in $\mathcal{CP}$, *i.e.*, the path that maximizes the localization capacity (in case of a tie, a path that minimizes the probe cost); and $minli$ and $minPc$ are initialized to the localization capacity and the probe cost of that path, respectively. The rational behind considering paths in $\mathcal{CP}$ is to avoid re-exploring all candidate paths traveling between the already selected monitors.

The network graph is, then, explored in depth-first order starting from the candidate monitor location m. It is worth noting that we believe that a breadth-first search can find candidate paths faster. However, the depth-first search approach requires much less memory.

We now introduce the optimizations made to avoid exploring all the network graph, which speeds up the search and ensures the scalability of the algorithm. Let n be the

4. By construction, $\bigcap_k \mathcal{S}_a^{(j)k} = \emptyset$

currently explored node and p_c the current path to that node. p_s , $minli$ and $minPc$ are set equal to p_c, $li(p_c, \mathcal{S}_a^{(j)})$, and $probeCost(p_c, C_{\mathcal{E}})$, respectively, if the following condition is true:

$$n \in \mathcal{SM} \text{ and } (li(p_c, \mathcal{S}_a^{(j)}) < minli \text{ or } (li(p_c, \mathcal{S}_a^{(j)}) = minli \text{ and } probeCost(p_c, C_{\mathcal{E}}) < minPc)) \tag{4.10}$$

The above condition implies that the path selection criterion is the minimization of the localization indicator, which is equivalent to the maximization to the localization capacity, and that ties are broken by minimizing the probe cost. Moreover, it ensures that the end nodes of the selected path are in $\mathcal{SM} \cup \{m\}$.

Now, the most important feature of the algorithm is that it is able, using Theorem (6), to decide whether all paths having a given prefix are not good. A good path is a path that dominates the current best path, *i.e.*, a path that satisfies Condition (4.10). In fact, all paths having as prefix the current path p_c are undoubtedly inefficient if one of the following conditions is true:

$$minli = 0 \text{ and } \exists\ \mathcal{S}_a^{(j)k} \in \mathcal{S}_a^{(j)} \text{ such that } \mid \mathcal{S}_a^{(j)k} \cap p_c \mid > \mid \mathcal{S}_a^{(j)k} \mid /2 \tag{4.11}$$

$$minli = 0 \text{ and } \forall \mathcal{S}_a^{(j)k} \in \mathcal{S}_a^{(j)} \mid \mathcal{S}_a^{(j)k} \cap p_c \mid \leq \mid \mathcal{S}_a^{(j)k} \mid /2 \text{ and } probeCost(p_c, C_{\mathcal{E}}) + \min_{e \in \mathcal{E}} C_e li(p_c, \mathcal{S}_a^{(j)}) \geq minPc \tag{4.12}$$

$$li(p_c, \mathcal{S}_a^{(j)}) > minli \text{ and } \forall \mathcal{S}_a^{(j)k} \in \mathcal{S}_a^{(j)} \mid \mathcal{S}_a^{(j)k} \cap p_c \mid \geq \mid \mathcal{S}_a^{(j)k} \mid /2 \tag{4.13}$$

Whenever a node that is not in $\mathcal{SM}$ is explored, the current path to that node is examined. If it satisfies one of the above conditions, then the descendant nodes of the current node will not be explored, *i.e.*, all paths having as prefix the current path will be discarded without exploring their suffixes. This achieves great savings in terms of the number of explored paths and in terms of time. The accuracy of conditions (4.11), (4.12) and (E) is demonstrated in Appendix E.

4.9 Performance Evaluation

Extensive simulations are conducted on network topologies built using the BRITE generator [13] [33] (Waxman model [31]: $\alpha = \beta = 0.4$, random node placement [5]). We use

5. These parameters are not to be confused with the monitor cost weight (α) and the probe cost weight (β) introduced in Section 4.6. Their values equal the values used by Waxman to generate network topologies [31].

Cplex11.2 [12] to solve ILPs and we implement our algorithms using C_{++}. All the numerical results presented in this section are the mean over 30 simulations on random simulations. Our experiments indicate that the results are almost the same for larger number of simulations. Table 4.3 depicts a summary of the topologies considered. Our localization scheme takes as input any detection solution that covers all links of the network. For small topologies, *i.e.*, TOP(8, 18), optimal detection solutions are computed using the ILP proposed in the chapter 2 of this thesis; whereas the anomaly detection heuristic proposed in the same chapter is used to compute detection solutions for larger topologies. Note that the anomaly detection problem is $\mathcal{NP}$-Hard, therefore, optimal detection solutions could not be computed for large topologies.

Table 4.3: Summary of the topologies considered in the evaluation

Topology	**Nb. of nodes**	**Nb. of links**
TOP(8, 18)	8	18
TOP(10, 31)	10	31
TOP(12, 41)	12	41
TOP(15, 59)	15	59
TOP(20, 80)	20	80

The evaluations are performed on a PC equipped with a 2,992.47 MHz Intel(R) Core(TM)2 Duo processor and 3.9 GB of RAM. We assume that every nodes of the network is candidate to support a monitoring device and all paths of the networks are candidate to be monitored. We set $C_n = C_e = 1, \forall n \in \mathcal{N}$ and $\forall e \in \mathcal{E}$.

4.9.1 Comparing our Anomaly Localization Scheme with Existing Schemes

We compare our anomaly localization scheme with an hybrid anomaly localization scheme that combines the strengths of the schemes proposed in [1] and [2]. As proposed in [2], a set of paths that distinguishes only between the suspect links is monitored during the localization phase. However, to guarantee that all potential anomalies can be localized uniquely, a set of monitors that can distinguish between all pairs of the network links is deployed [1]. Such a scheme can be formulated as two ILPs. The first ILP computes a minimal subset of monitor locations that enables the

localization of all potential anomalies. This ILP is run only once offline. It reads as follows:

Minimize $\sum_{n \in \mathcal{M}} Y_n$

Subject to:

$$\sum_{p \in \mathcal{P}} (\delta_{pe_1} + \delta_{pe_2} - 2\delta_{pe_1}\delta_{pe_2}) Z_p > 0; \forall e_1, e_2 \in \mathcal{E}; \quad \forall p \in \mathcal{P}$$

$$\delta_{pn} Y_n \geq Z_p; \quad \forall p \in \mathcal{P}, \forall n \in \mathcal{N}$$

The second ILP is run whenever an anomaly is detected. The input is the set of monitor locations selected by the first ILP, $\mathcal{M}'$, and a set of suspect links $\mathcal{S}$. The output is a minimal set of monitoring paths that can distinguish between the suspect links pairwise. This ILP reads as follows:

Minimize $\sum_{p \in P} Z_p$

Subject to:

$$\sum_{p \in \mathcal{P}} (\delta_{pe_1} + \delta_{pe_2} - 2\delta_{pe_1}\delta_{pe_2}) Z_p > 0; \quad \forall e_1, e_2 \in \mathcal{S}; \forall p \in \mathcal{P}$$

$$Z_p \leq \delta_{pn} Y_n; \quad \forall p \in \mathcal{P}, \forall n \in \mathcal{M}'$$

We refer to this hybrid anomaly localization scheme as HLS.

Only small topologies for which the ILPs can deliver solutions in tractable time are considered. We set the weight associated to the probe cost $\beta = 1$, and we vary the weight associated to the monitor cost, $\alpha \in [1, 2, 4]$ and $\alpha \geq 6$.

We define three metrics for the comparison. The first metric is the time of computing the localization solution, *i.e.*, monitors that are to be activated and paths that are to be monitored when an anomaly is detected. This metric reflects the speed of the localization scheme. The better is to avoid online computations, *i.e.*, computations done upon detecting an anomaly, in order to shorten the localization delay.

Table 4.4: Average ILP computation time for TOP(8, 18)

	Hybrid scheme	**Our scheme**
Offline Computation Time	64.16 s	6.67 s
Online Computation Time	25.7 10^{-3} s	0 s

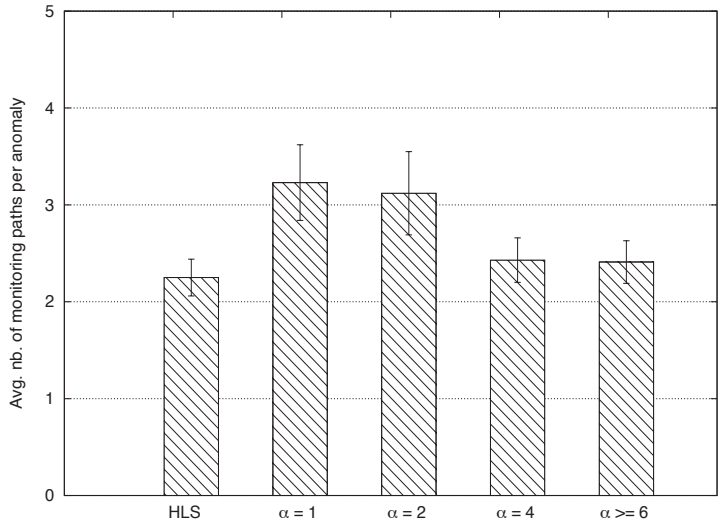
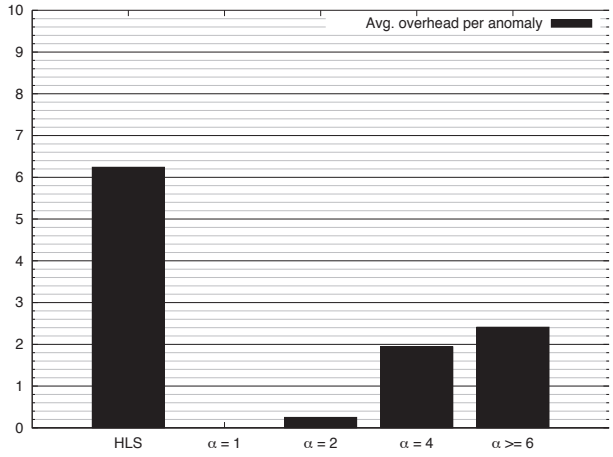


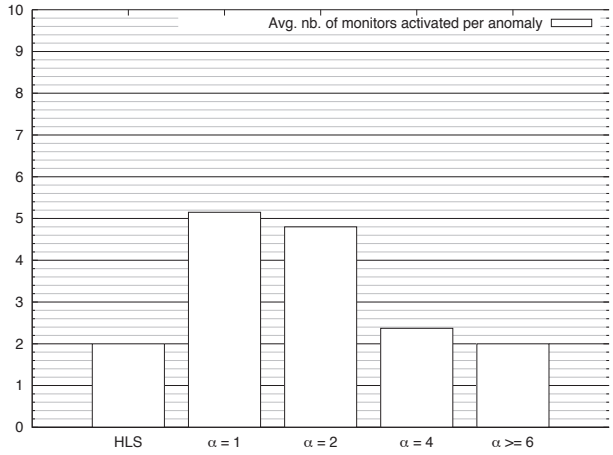
Figure 4.3: Average number of monitoring paths per anomaly for TOP(8, 18). The first histogram to the left presents results for solutions computed using the hybrid localization scheme (HLS), and the other histograms present results for the solutions computed using our anomaly localization ILP with different values of α ($\beta = 1$).

Table 4.4 depicts the online computation time and the offline computation time for the hybrid localization scheme and for our localization scheme. Intuitively, as shown in the table, the online computation time is zero for our localization scheme. This is because we compute full localization solutions for all potential anomalies offline. In contradiction, the hybrid scheme leaves the selection of monitoring paths upon detecting an anomaly, thereby achieving a non-negligible online computation time. This time can be relatively high for large topologies where the number of candidate monitoring paths is large. For the offline computation time, the table shows that our scheme is about 10 times faster than the hybrid scheme, although, it computes full localization solutions for all potential anomalies. We explain this result by the fact that, unlike the hybrid scheme, our scheme does not distinguish between every pair of the network links.

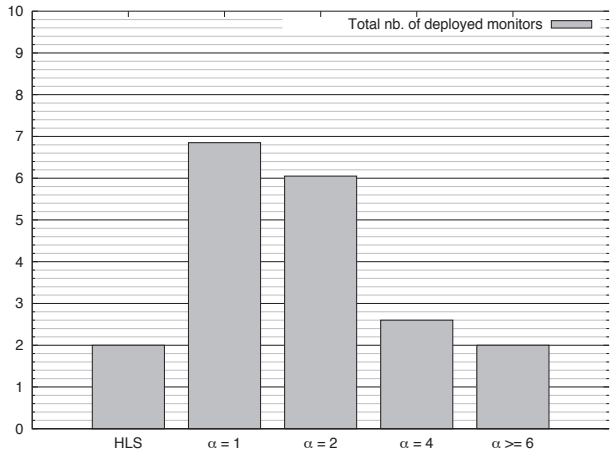
The second metric is the localization cost. Figure 4.4 plots the total number of deployed monitors (Figure 4.4c), the average number of monitors activated per anomaly (Figure 4.4b), and the average overhead (4.4a), *i.e.*, the number of links monitored that provide no localization information, per anomaly for the hybrid localization scheme and for our localization scheme with different values of α. Three conclusions can be drawn from the numerical results. The first is that there is an interplay between the monitor location cost and the probe cost. The different results for the different values of α illustrate this



(a) Average overhead per anomaly



(b) Average nb. of monitors activated per anomaly



(c) Total nb. of deployed monitors

Figure 4.4: Localization costs for TOP(8, 18)

conclusion. Indeed, the larger the value of α is, the fewer the number of monitors is and the larger the localization overhead is. For instance, for $\alpha = 1$, we have localization solutions with zero overhead and 7 monitors, *i.e.*, 7 of the 8 nodes of the network hold monitoring devices. The second is that the existing localization scheme that deploys monitors offline and selects monitoring paths online does not take into consideration this interplay, and therefore, delivers sub-optimal localization solutions. Using the same number of monitors, for $\alpha \geq 6$, our localization scheme can localize any potential anomaly with about 65% less overhead than the existing localization scheme.

The third metric is the number of monitoring paths. Recall that this is the path selection criterion for the existing localization scheme. We do not consider this criterion in our localization scheme for two reasons. The first is that, upon detecting an anomaly, the set of paths that distinguish between the suspect links are monitored simultaneously. Therefore, the minimization of the number of monitoring paths does not reduce the localization delay. The second reason is that this metric is tightly correlated to the number of monitors and the localization overhead. Indeed, if we relax the constraint on the localization overhead, this would allow long monitoring paths that cross a large number of links. Therefore, the number of monitoring paths that can distinguish between the suspect links would decrease. Similarly, if we relax the constraint on the number of monitors, we would deploy more monitors in the network, thus, the monitoring paths would get shorter. Therefore, the number of monitoring paths that can distinguish between the suspect links would increase. Figure 4.3 validates these claims. Hereby, we can observe that the larger α is, the more monitoring paths we have. Not surprisingly, for $\alpha \geq 6$, our localization scheme monitors only 8% more paths than the hybrid localization scheme, while deploying the same number of monitors and incurring 65% less overhead.

4.9.2 Evaluating the Scalability and Quality of the Heuristic

In this section, we evaluate the performance of our anomaly localization heuristic. We set $\alpha >> \beta$. For each network topology, we run the heuristic n times, where n is the number of the network nodes. The first monitor location that is selected randomly must be different for each run. Then, we consider the solution with the smallest localization cost. For TOP(8, 18), we compare the results obtained using the heuristic with the results obtained using our anomaly localization ILP ($\alpha \geq 6$), and the results obtained using the hybrid localization scheme. Furthermore, we evaluate the evolution of resource consumption and computation

time with respect to the network size to evaluate the performance of the heuristic on larger topologies.

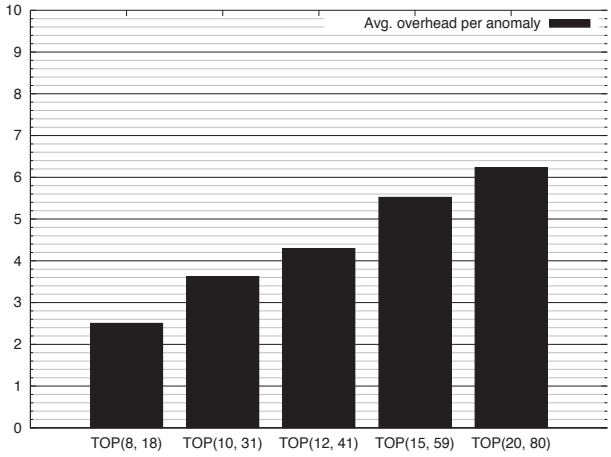
Table 4.5 depicts the heuristic computation time (this is the time of the n runs of the heuristic) and the average percentage of the network paths explored in one execution of Procedure 2 for all the topologies considered. For TOP(8, 18) the heuristic computation time is about 29.10^3 times faster than our ILP, and about 27.10^4 times faster than the hybrid localization scheme. Recall that all computations are done offline. For TOP(10, 31), TOP(12, 41) and TOP(15, 59) the heuristic computation time is in the order of few seconds (< 25 s), while it was infeasible to obtain the ILP results for these topologies in tractable time. For TOP(20, 80), whose number of paths is in the order of hundreds of billions, it was impossible to run the ILPs due to memory insufficiency. However, the heuristic succeeded to compute solutions in less than one hour for these topologies. This confirms the efficiency of our candidate path computation algorithm that minimizes the number of the networks paths that are to be explored. For instance, we found that only 0.007% of the network paths are explored in one execution of Procedure 2 for TOP(20, 80).

Table 4.5: Heuristic computation time (all computations are done offline) and percentage of paths explored in one execution of Procedure 2

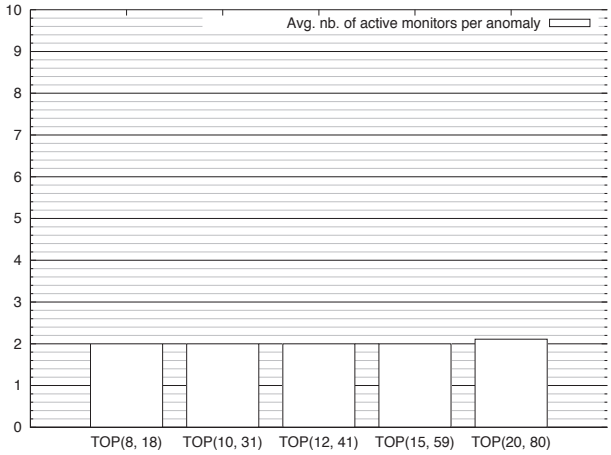
Topology	**Heuristic computation time**	**% of paths explored in one execution of Procedure 2**
TOP(8, 18)	0.00023 s	1.22%
TOP(10, 31)	0.08 s	0.21%
TOP(12, 41)	0.78 s	0.07%
TOP(15, 59)	24.11 s	0.02%
TOP(20, 80)	3525.52 s	0.007%

We now investigate the quality of the solutions delivered by the heuristic. Figure 4.5 plots the total number of monitors deployed (4.5c), the average number of monitors activated per anomaly (4.5b), and the average overhead per anomaly for the topologies considered in the evaluation4.5a.

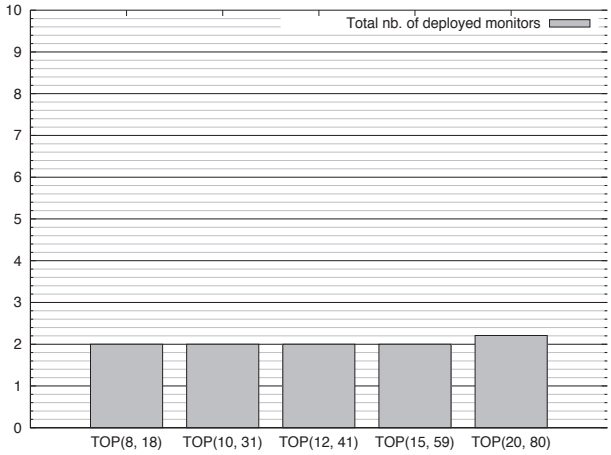
First, we notice that two monitors are sufficient to localize all potential anomalies for all topologies, except TOP(20, 80) for which the average number of monitors deployed and the average number of monitors activated per anomaly are slightly larger than two. This is expected, since we set $\alpha >> \beta$, which means that the heuristic minimizes in priority the



(a) Average overhead per anomaly

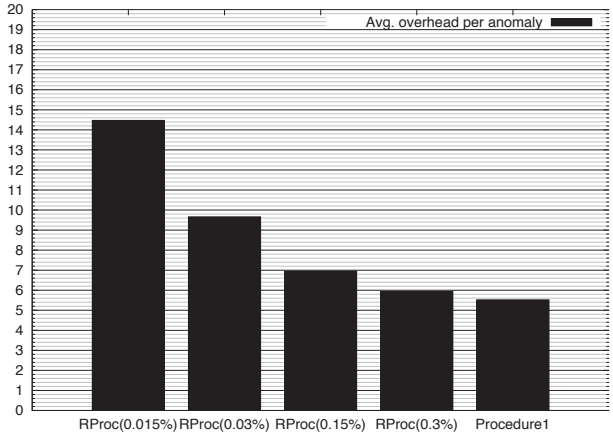


(b) Average nb. of monitors activated per anomaly

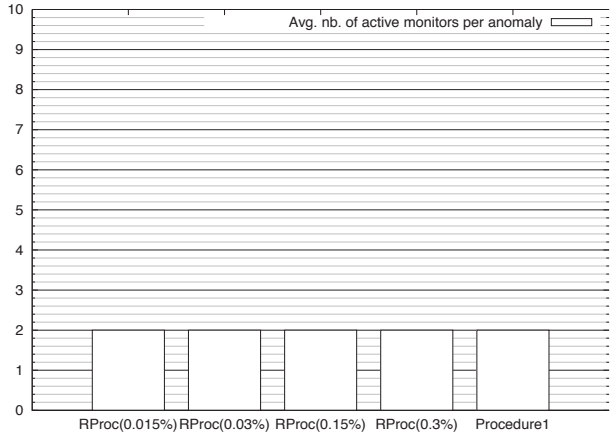


(c) Total nb. of deployed monitors

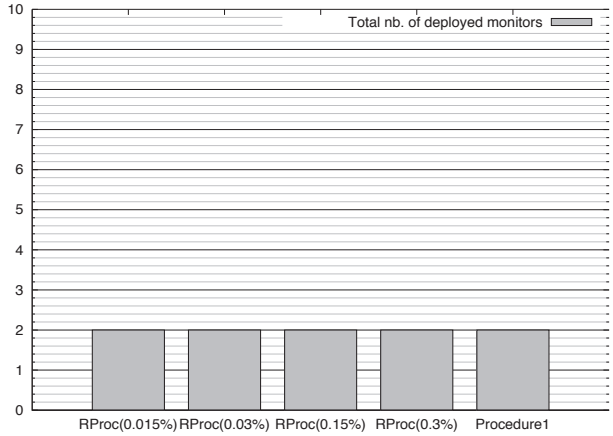
Figure 4.5: Localization cost of the heuristic solutions, $\alpha >> \beta$



(a) Average overhead per anomaly



(b) Avgerage nb. of monitors activated per anomaly



(c) Total nb. of deployed monitors

Figure 4.6: Impact of the number and the quality of candidate monitoring paths on the quality of the localization solution. RProc means random procedure (numerical results for TOP(15, 59))

number of monitors that are to be deployed. A comparison of Figure 4.5 with Figure 4.4 shows that, for TOP(8, 18), the solutions computed using our ILP ($\alpha \geq 6$) is very close to the solutions computed using the heuristic: the heuristic solution gnenerates about 9% more overhead, however, the two solutions deploy the same number of monitors and activate, in average, the same number of monitors when an anomaly occurs. This confirms that the candidate path computation algorithm that avoids exploring all paths of the network does not miss good paths. Moreover, the overhead of the heuristic solutions for TOP(10, 31), TOP(12, 41) and TOP(12, 59) is smaller than the overhead of the hybrid localization scheme solutions for TOP(8, 18). It is worth to recall that the hybrid localization solutions for TOP(8, 18) are exact solutions. This confirms that i) the heuristic succeeds to minimize the localization costs, *i.e.*, the monitor cost and the probe cost, jointly; ii) the heuristic outperforms the hybrid localization scheme, since the former can localize anomalies in large topologies using less resources that those used by the latter to localize anomalies in smaller topologies.

We finally evaluate the impact of the number and the quality of candidate monitoring paths on the quality of the localization solution. To this end, we compare the localization solutions obtained using the proposed heuristic, *i.e.*, Algorithm 4 and Procedure 2, to the localization solutions obtained using Algorithm 4 and a procedure that computes candidate paths randomly (instead of Procedure 2). In the latter case, we variate the number of paths explored per one execution of the random candidate path computation procedure (0.015%, 0.03%, 0.15%, 0.3%). We report the results for TOP(15, 59) when $\alpha >> \beta$ in Figure 4.6 (The results are essentially the same for the other topologies). Not surprisingly, Figure 4.6 shows that, when candidate paths are explored randomly, the larger the number of paths explored is the smaller the localization overhead is. Furthermore, it shows that the proposed heuristic achieves smaller overhead than the random approach, thought it explores more than 15 times less paths as shown in Table 4.5. This validates our claim on the correlation between the number and quality of monitoring paths and the quality of the localization solution.

4.10 Discussion

The anomaly localization solution must be updated whenever the detection solution changes. However, the detection solution changes in rare cases where a persistent anomaly makes a network link unavailable for a long period of time, or where the network topology is modified voluntary (*e.g.*, add and/or removal of network equipments).

Usually, in the first case, the detection solution is updated partially. Only the detection paths that are affected by the anomaly are re-computed. The anomaly scenarios are updated accordingly, and the localization solution is re-computed, partially, for the affected anomaly scenarios. The evaluation results show that, for instance, the average computation time of the localization solution for one anomaly scenario using the heuristic is in the order of 5 minutes for TOP(20, 80). Knowing that anomalies are rare events, we assert that it is rather unlikely that anomalies occur before the localization solution is updated. However, in case an anomaly occurs before the localization solution is updated, the localization process could be executed for the current solution, though, not all anomalies could be localized accurately. The best solution for such situation is to provide backup detection and localization solutions. However, this issue is out of the scope of this thesis.

Furthermore, voluntary network changes are usually planned in advance. Thus, detection and localization updates could be computed offline before voluntary network changes are made.

4.11 Conclusion

This chapter addressed the problem of single link-level anomalies localization. Two findings were demonstrated: 1) Not all pairs of the network links need to be distinguishable for localizing any potential link-level anomaly, 2) All potential anomaly scenarios can be derived offline from any detection solution that covers all the network links. These findings were exploited to develop an anomaly localization scheme that computes full localization solutions offline. In order to achieve a good trade-off between the number and locations of monitoring devices and the quality of monitoring paths, monitor locations and monitoring paths are selected jointly. A novel anomaly localization cost model that expresses the localization overhead and delay besides the localization infrastructure cost was proposed. The problem was formulated as an ILP algorithm and was shown to be $\mathcal{NP}$-hard. Therefore, an efficient heuristic was proposed. The key idea of the heuristic is to reduce the number of candidate paths without missing good paths, thereby achieving scalability and quality.

The proposed anomaly localization scheme was compared with an hybrid anomaly localization scheme that combines the strengths of two existing schemes through extensive simulations. Results demonstrate the superiority of the proposed scheme, in terms of computation time and cost reduction, and its efficiency in balancing the trade-off between

the localization costs. Furthermore, the results confirm that the heuristic algorithm is effective at achieving scalability and quality.

CHAPTER 5

Conclusion and Perspectives

This thesis investigates the problems of anomaly detection and localization in computer networks. Especially, the focus is on the use of end-to-end path measurements for detecting and localizing link-level network anomalies. The aim of the thesis is to answer the following question: *where to place monitoring devices and which paths to monitor towards detecting and localizing all potential link-level anomalies in an accurate, fast and cost-efficient fashion.*

The first step towards answering the above question is the study of existing network anomaly detection and localization schemes. A review of the body of literature relevant to the investigated problems is provided, and the limitations and the strengths of existing anomaly detection and localization schemes are highlighted. The contributions of this thesis consist in coming up with anomaly detection and localization schemes that implement the strengths of existing schemes and overcome their limitations.

The proposed anomaly detection scheme is a one-step scheme that selects paths that are to be monitored and monitor locations jointly. An ILP formulation of the scheme is provided, and the problem is shown to be $\mathcal{NP}$-Hard. Two one-step heuristic algorithms for anomaly detection solution computation are, therefore, devised. The first algorithm considers the total set of the network paths as candidate to be monitored. The second algorithm implements a procedure for computing candidate monitoring paths. The aim of this procedure is to reduce the set of candidate paths in order to achieve the scalability of the heuristic, while ensuring a good quality of the detection solution. The proposed one step scheme is compared to the existing two-step anomaly detection schemes. The

superiority of the one step scheme, and its efficiency to achieve a good trade-off between the optimization objectives of monitor location selection and the optimization objectives of monitoring path selection are demonstrated.

The applicability of the proposed anomaly detection scheme on multi-domain networks is investigated. An ILP algorithm and a heuristic algorithm that take into account the properties and the limitations of such networks are provided. A comparative study of two anomaly detection approaches is conducted. The first approach is a global approach that considers the multi-domain network as a single domain, in which case the anomaly detection scheme proposed for mono-domain networks can be applied. The second approach is a per-domain technique that minimizes the interactions between domains in an attempt to overcome the confidentiality issues. Evaluations results show that confidentiality is so far not the only limitation to the application of the global anomaly detection technique for multi-domain networks. Especially, the results show that the global detection technique yields solutions with relatively long monitoring paths, and does not guarantee a fair distribution of the detection load among domains. Besides, the computation time for the global technique is drastically high compared to the computation time for the per-domain technique. In contrast, the difference of costs of the solutions of the two techniques, in terms of the number of monitors and overhead, is small.

Although the thesis advocates decoupling the anomaly localization from the anomaly detection, *i.e.*, the anomaly detection process is run continuously whereas the localization process is triggered only upon detecting an anomaly as opposed to monitoring a set of paths that can detect and localize anomalies continuously, it exploits the fact that the outputs of the detection process are the inputs to the localization process to optimize the localization solution. Particularly, it has been demonstrated, in the thesis, that, knowing the set of paths that is monitored for anomaly detection, all potential anomaly scenarios can be derived offline [1]. Subsequently, the set of paths that is to be monitored upon detecting an anomaly is reduced to a small subset of paths that can distinguish only between suspect links. Moreover, full localization solutions, *i.e.*, paths that are to be monitored and monitors that are to be activated upon detecting an anomaly, are computed offline for all potential anomaly scenarios. Similarly to the detection schemes, monitor locations and localization paths are selected jointly in one single step. The localization

1. An anomaly scenario is characterized by a unique set of suspect links. Different anomalies can cause the same anomaly scenario.

problem is formulated as an ILP, and is demonstrated to be $\mathcal{NP}$-Hard. A heuristic algorithm is devised. The capacity of the proposed scheme to localize all potential single link-level anomalies accurately is verified analytically, and its superiority over existing anomaly localization schemes is demonstrated through simulations.

Further research need to be performed in order to investigate the problem of localizing concurrent link-level anomalies. Such anomalies are considered as very unlikely, which justifies the scarcity of research on this problem. A solution is proposed in [1]. It requires deploying a set of monitoring devices that can distinguish between every two subsets of the network links. This implies that for each pair of link subsets there exists a monitoring path between the deployed monitoring devices whose intersection with exactly one of the two subsets is not empty. Admitting the complexity of this process and the heavy overhead it yields, the authors propose to limit the number of concurrent anomalies (≤ 3). One of our goals for future work, is to evaluate and optimize this solution. We plan to devise a technique that enables us to decide whether a detected anomaly event is associated to a single anomaly or to multiple concurrent anomalies, and to activate the appropriate localization solution accordingly.

A more frequent type of anomalies that have not been considered enough in the literature dealing with network monitoring is the Shared Risk Link Group (SRLG) anomalies. The particularity of a SRLG anomaly is that it affects a group of links that have a common anomalous resource. A common example of this kind of anomalies is node failures. When a node fails, all its surrounding links fail systematically. A necessary and sufficient condition for localizing accurately any SRLG failure in all-optical networks has been established in [51]. Moreover, [51] provides an interesting scheme for localizing uniquely any SRGL with up to k links in any (k+2)-edge connected all-optical network using one single monitoring device. However, there are key differences between all-optical networks and IP networks, namely the limited capacity of IP links to support traffic flows as opposed to the abundant capacity of fiber-optic links, that constrain the application of this scheme to IP networks. In our future work, we will investigate the problem of SRLG anomaly localization in IP networks. The aim is to establish a necessary and sufficient condition for localizing any SRLG anomaly in IP networks.

Furthermore, when an anomaly occurs in the network, all links that are covered only by paths crossing the anomalous link remain uncovered until the anomaly is fixed. The problem becomes more complicated when an anomaly occurs, while a previously detected anomaly is not yet fixed. This problem of sequential anomalies deserves to be investigated, and a scheme for computing backup detection and localization solutions should be devised.

APPENDIX A

This appendix presents the proofs of corollaries 2, 3, 4, 5 and 6.

Corollary 2. $e_1 \in \mathcal{S}(e_2) \Leftrightarrow \mathcal{S}(e_1) = \mathcal{S}(e_2)$, $\forall e_1, e_2 \in \mathcal{E}$

Proof. $e_1 \in \mathcal{S}(e_2) \Leftrightarrow$ (according to Theorem 1) there does not exist any path that crosses either e_1 or e_2, but not both $\Leftrightarrow$ for each $p \in \mathcal{P}$, p crosses both e_2 and e_1, or p neither crosses e_1 nor $e_2 \Leftrightarrow D_{e_1+} = D_{e_2+}$ and $D_{e_1-} = D_{e_2-} \Leftrightarrow$ (according to Theorem 2) $\mathcal{S}(e_1) = \mathcal{S}(e_2)$ □

Corollary 3. $\mathcal{S}(e_1) \neq \mathcal{S}(e_2) \Leftrightarrow \mathcal{S}(e_1) \cap \mathcal{S}(e_2) = \emptyset$

Proof. We prove the direct implication by contradiction. Assume to the contrary that $\mathcal{S}(e_1) \neq \mathcal{S}(e_2)$ and $\mathcal{S}(e_1) \cap \mathcal{S}(e_2) \neq \emptyset$. Let $e_3 \in \mathcal{S}(e_1) \cap \mathcal{S}(e_2)$. According Corollary 2, $\mathcal{S}(e_3) = \mathcal{S}(e_1)$ and $\mathcal{S}(e_3) = \mathcal{S}(e_2)$. thus, $S(e_1) = \mathcal{S}(e_2)$, leading to a contradiction. The indirect implication is trivially true. □

Corollary 4. $\cup_{e \in \mathcal{E}} \mathcal{S}(e) = \cup_{\mathcal{S}(i) \in dS} \mathcal{S}(i) = \mathcal{E}$

Proof. According to Theorem 2, $e \in \mathcal{S}(e), \forall e \in \mathcal{E}$. Thus, $\cup_{e \in \mathcal{E}} \mathcal{S}(e) = \mathcal{E}$. Obviously, $\cup_{e \in \mathcal{E}} \mathcal{S}(e) = \cup_{S(i) \in dS} S(i)$. □

Corollary 5. $\sum_{\mathcal{S}(i) \in dS} \mid \mathcal{S}(i) \mid = \mid \mathcal{E} \mid$

Proof. According to Corollary 4, $\mid \cup_{\mathcal{S}(i) \in dS} \mathcal{S}(i) \mid = \mid \mathcal{E} \mid$, and according to Corollary 2, $\cap_{\mathcal{S}(i) \in dS} \mathcal{S}(i) = \emptyset$. Thus,

$$\sum_{\mathcal{S}(i) \in dS} \mid \mathcal{S}(i) \mid = \mid \mathcal{E} \mid.$$

□

Corollary 6. $dPairs = AllPairs - \sum_{\mathcal{S}(i), \mathcal{S}(j) \in dS : i < j} \mid \mathcal{S}(i) \mid\mid \mathcal{S}(j) \mid$

Proof. According to Corollary 1, only links that belong to same set of suspect links need to be distinguishable pairwise. Therefore, the set of link pairs that are to be distinguished can be expressed as $\{\{(e_i, e_j); e_i, e_j \in \mathcal{E}\}$ - $\{(e_i, e_j);\ \mathcal{S}(e_i) \neq \mathcal{S}(e_j)\}\}$. We conclude that $dPairs = AllPairs$ - $\sum_{\mathcal{S}(i),\mathcal{S}(j)\in dS:i<j} \mid \mathcal{S}(i) \mid * \mid \mathcal{S}(j) \mid$. Clearly, the number of pair of links that need to be distinguishable equals the number of all link pairs of the network if and only if the number of distinct sets of suspect links equals 1, *i.e.* the number of detection paths equals 1. □

APPENDIX B

This appendix presents the proof of Theorem 3.

Theorem 3. *Let P_1 be the subset of paths of $\mathcal{P}$ that cross either e_1 or e_2, but not both.* $\sum_{p\in\mathcal{P}}(\delta_{pe_1}+\delta_{pe_2}-2\delta_{pe_1}\delta_{pe_2}) =\mid P_1 \mid$.

Proof. Paths in $\mathcal{P}'$ can be divided into three subsets of paths.

- P_1: paths that cross either e_1 or e_2, but not both.
- P_2: paths that cross both e_1 and e_2.
- P_3: paths that neither cross e_1 nor e_2.

On the one hand, we have
$\forall p \in P_2, \quad \delta_{pe_1} = 0$ and $\delta_{pe_2} = 0$.
Thus, $\forall p \ \in P_2, \ (\delta_{pe_1}+\delta_{pe_2}-2\delta_{pe_1}\delta_{pe_2}) = 0$.
Contributing to $\sum_{p\in P_2}(\delta_{pe_1}+\delta_{pe_2}-2\delta_{pe_1}\delta_{pe_2}) > 0$.
On the other hand, we have $\forall p \in P_3, \quad \delta_{pe_1} = 1$ and $\delta_{pe_2} = 1$.
Thus, $\forall p \in P_3, \quad (\delta_{pe_1}+\delta_{pe_2}-2\delta_{pe_1}\delta_{pe_2}) = 0$.
Contributing to $\sum_{p\in P_3}(\delta_{pe_1}+\delta_{pe_2}-2\delta_{pe_1}\delta_{pe_2}) = 0$.
Subsequently, $\sum_{p\in\mathcal{P}'}(\delta_{pe_1}+\delta_{pe_2}-2\delta_{pe_1}\delta_{pe_2}) = \sum_{p\in P_1}(\delta_{pe_1}+\delta_{pe_2}-2\delta_{pe_1}\delta_{pe_2})$.
Now, we have $\forall p \in P_1 \quad \delta_{pe_1}+\delta_{pe_2} = 1$ and $\delta_{pe_1}\delta_{pe_2} = 0$.
Thus, $\delta_{pe_1}+\delta_{pe_2}-2\delta_{pe_1}\delta_{pe_2} = 1$.
Therefore, $\sum_{p\in P_1}(\delta_{pe_1}+\delta_{pe_2}-2\delta_{pe_1}\delta_{pe_2}) =\mid P_1 \mid$.
We conclude that $\sum_{p\in\mathcal{P}'}(\delta_{pe_1}+\delta_{pe_2}-2\delta_{pe_1}\delta_{pe_2}) =\mid P_1 \mid$. □

APPENDIX C

This appendix presents the proof of Theorem 5.

Theorem 5. *The theoretical minimal probe cost relative to a set of suspect links $\mathcal{S}$ denoted by $ThMinPcost(\mathcal{S})$ reads as follows:*

$$ThMinPcost(S) = \sum_{e \in \mathcal{S}} C_e - \max_{e \in \mathcal{S}} C_e$$

Proof. Let $\mathcal{P}^{'}$ be a set of paths that can distinguish between all links of $\mathcal{S}$. According to Theorem 1, for each $e_1, e_2 \in \mathcal{S}$ $\exists p \in \mathcal{P}^{'}$ such that p crosses either e_1 or e_2, but not both. Thus, at most one link of $\mathcal{S}$ is not traversed by paths in $\mathcal{P}^{'}$. We conclude that any localization solution must imperatively monitor $\mid \mathcal{S} \mid -1$ links of $\mathcal{S}$ in order to distinguish between all links. It follows that the localization solution that incurs the minimal probe cost is a solution that monitors exactly $\mid \mathcal{S} \mid -1$ links of $\mathcal{S}$ whose measurement cost if the lowest. Thus, $ThMinPcost(\mathcal{S}) = \sum_{e \in \mathcal{S}} C_e - \max_{e \in \mathcal{S}} C_e$. Note that such a solution is feasible only if each link of the $\mid \mathcal{S} \mid -1$ links is monitored separately, which requires to have monitors deployed on the end nodes of each of these links. □

APPENDIX D

This appendix presents the proof of Theorem 6.

Theorem 6: *Let $absval(x)$ be a function that returns the absolute value of the number x, and let $lc(\mathcal{S}_a^{(j)}, p) = \sum_{\mathcal{S}_a^{(j)k} \in \mathcal{S}_a^{(j)}} lc(\mathcal{S}_a^{(j)k}, p)$ be the localization capacity of p with respect to $\mathcal{S}^{(j)}$. We have,*

$$\max_{p \in \mathcal{P}} \; lc(\mathcal{S}_a^{(j)}, p) = \min_{p \in \mathcal{P}} \sum_{\mathcal{S}_a^{(j)k} \in \mathcal{S}_a^{(j)}} absval(\mid \mathcal{S}_a^{(j)k} \mid /2 - \mid \mathcal{S}_a^{(j)k} \cap p \mid)$$

Proof. We have $\max_{p \in \mathcal{P}} lc(\mathcal{S}_a^{(j)}, p) = \sum_{\mathcal{S}_a^{(j)k} \in \mathcal{S}_a^{(j)}} \max_{p \in \mathcal{P}} lc(\mathcal{S}_a^{(j)k}, p)$, where $lc(\mathcal{S}_a^{(j)k}, p) = \mid p \cap \mathcal{S}_a^{(j)k} \mid * (\mid \mathcal{S}_a^{(j)k} \mid - \mid p \cap \mathcal{S}_a^{(j)k} \mid)$. Consider the variations of $lc(\mathcal{S}_a^{(j)k}, p)$ with respect to the values of $\mid p \cap \mathcal{S}_a^{(j)k} \mid$. It can be easily shown that:

- $lc(\mathcal{S}_a^{(j)k}, p)$ is increasing for $\mid p \cap \mathcal{S}_a^{(j)k} \mid < \mid \mathcal{S}_a^{(j)k} \mid /2$, and decreasing for $\mid p \cap \mathcal{S}_a^{(j)k} \mid > \mid \mathcal{S}_a^{(j)k} \mid /2$
- $\forall\, p_1, p_2 \in \mathcal{P}$, if $absval(\mid \mathcal{S}_a^{(j)k} \mid /2 - \mid p_1 \cap \mathcal{S}_a^{(j)k} \mid) = absval(\mid \mathcal{S}_a^{(j)k} \mid /2 - \mid p_2 \cap \mathcal{S}_a^{(j)k} \mid)$, then, $lc(\mathcal{S}_a^{(j)k}, p_1) = lc(\mathcal{S}_a^{(j)k}, p_2)$
- The global maximum of $lc(\mathcal{S}_a^{(j)k}, p)$ is achieved at $\mid p \cap \mathcal{S}_a^{(j)k} \mid = \mid \mathcal{S}_a^{(j)k} \mid /2$

It follows that $\max_{p \in \mathcal{P}} lc(\mathcal{S}_a^{(j)k}, p) = \min_{p \in \mathcal{P}} absval(\mid \mathcal{S}_a^{(j)k} \mid /2 - \mid p_2 \cap \mathcal{S}_a^{(j)k} \mid)$. Subsequently, $\max_{p \in \mathcal{P}} \; lc(\mathcal{S}_a^{(j)}, p) = \min_{p \in \mathcal{P}} \sum_{\mathcal{S}_a^{(j)k} \in \mathcal{S}_a^{(j)}} absval(\mid \mathcal{S}_a^{(j)k} \mid /2 - \mid \mathcal{S}_a^{(j)k} \cap p \mid)$ □

APPENDIX E

This appendix demonstrates the correctness of conditions (4.11), (4.12) and (E).

Condition 4.11:

$$minli = 0 \text{ and } \exists\ \mathcal{S}_a^{(j)k} \in \mathcal{S}_a^{(j)} \text{ such that } \mid \mathcal{S}_a^{(j)k} \cap p_c \mid > \mid \mathcal{S}_a^{(j)k} \mid /2$$

Condition 4.12:

$$minli = 0 \text{ and } \forall \mathcal{S}_a^{(j)k} \in \mathcal{S}_a^{(j)} \mid \mathcal{S}_a^{(j)k} \cap p_c \mid \leq \mid \mathcal{S}_a^{(j)k} \mid /2 \text{ and } probeCost(p_c, C_{\mathcal{E}}) + \min_{e \in \mathcal{E}} C_e li(p_c, \mathcal{S}_a^{(j)}) \geq minPc$$

Condition E:

$$li(p_c, \mathcal{S}_a^{(j)}) > minli \text{ and } \forall \mathcal{S}_a^{(j)k} \in \mathcal{S}_a^{(j)} \mid \mathcal{S}_a^{(j)k} \cap p_c \mid \geq \mid \mathcal{S}_a^{(j)k} \mid /2$$

Let q be a path, and let p_c be a prefix of q. We have:

(i) $\forall \mathcal{S}_a^{(j)k}, \mid \mathcal{S}_a^{(j)k} \cap q \mid \geq \mid \mathcal{S}_a^{(j)k} \cap p_c \mid$

(ii) $\exists \mathcal{S}_a^{(j)k} \in \mathcal{S}_a^{(j)}$ such that $\mid \mathcal{S}_a^{(j)k} \cap p_c \mid > \mid \mathcal{S}_a^{(j)k} \mid /2 \Rightarrow li(q, \mathcal{S}_a^{(j)}) > 0$

Proof. It is clear that $li(q, \mathcal{S}_a^{(j)}) = 0 \iff \forall \mathcal{S}_a^{(j)k} \in \mathcal{S}_a^{(j)} absval(\mid \mathcal{S}_a^{(j)k} \mid /2 - \mid q \cap \mathcal{S}_a^{(j)k} \mid) = 0 \iff \forall \mathcal{S}_a^{(j)k} \in \mathcal{S}_a^{(j)} \mid q \cap \mathcal{S}_a^{(j)k} \mid = \mid \mathcal{S}_a^{(j)k} \mid /2$. However, according to (i), $\forall \mathcal{S}_a^{(j)k} \mid \mathcal{S}_a^{(j)k} \cap p_c \mid > \mid \mathcal{S}_a^{(j)k} \mid /2 \Rightarrow \mid \mathcal{S}_a^{(j)k} \cap q \mid > \mid \mathcal{S}_a^{(j)k} \mid /2$. Therefore, **(ii)** is true. □

(iii) $\forall \mathcal{S}_a^{(j)k} \mid \mathcal{S}_a^{(j)k} \cap p_c \mid > \mid \mathcal{S}_a^{(j)k} \mid /2 \Rightarrow li(q, \mathcal{S}_a^{(j)}) = li(p_c, \mathcal{S}_a^{(j)}) + \sum_{\mathcal{S}_a^{(j)k} \in \mathcal{S}_a^{(j)}} \mid \mathcal{S}_a^{(j)k} \cap p_c \mid - \mid \mathcal{S}_a^{(j)k} \cap q \mid$

Proof. $\forall \mathcal{S}_a^{(j)k} \mid \mathcal{S}_a^{(j)k} \cap p_c \mid > \mid \mathcal{S}_a^{(j)k} \mid /2 \Rightarrow \forall \mathcal{S}_a^{(j)k}$
$\mid \mathcal{S}_a^{(j)k} \cap q \mid > \mid \mathcal{S}_a^{(j)k} \mid /2 \Rightarrow li(q, \mathcal{S}_a^{(j)}) = \sum_{\mathcal{S}_a^{(j)k} \in \mathcal{S}_a^{(j)}} \mid \mathcal{S}_a^{(j)k} \mid /2 - \mid \mathcal{S}_a^{(j)k} \cap q \mid = li(p_c, \mathcal{S}_a^{(j)}) - \sum_{\mathcal{S}_a^{(j)k} \in \mathcal{S}_a^{(j)}} \mid \mathcal{S}_a^{(j)k} \cap q \mid + \sum_{\mathcal{S}_a^{(j)k} \in \mathcal{S}_a^{(j)}} \mid \mathcal{S}_a^{(j)k} \cap p_c \mid$. □

(iv) $\forall \mathcal{S}_a^{(j)k} \mid \mathcal{S}_a^{(j)k} \cap p_c \mid > \mid \mathcal{S}_a^{(j)k} \mid /2 \Rightarrow ProbeCost(q) \leq probeCost(p_c) + min_{e \in \mathcal{E}} C_e * li(q, \mathcal{S}_a^{(j)}) - li(p_c, \mathcal{S}_a^{(j)})$

Proof. We have $ProbeCost(q) = \sum_{e \in p} C_e = \sum e \in p_c C_e + \sum_{e \in e \in q \setminus p_c} C_e = probeCost(p_c) + \sum_{e \in q \setminus p_c} C_e \leq probeCost(p_c) + min_{e \in \mathcal{E}} C_e * \mid q \mid - \mid p_c \mid$ By construction, $\bigcap_k \mathcal{S}_a^{(j)k} = \emptyset$. Therefore, $\forall p \in \mathcal{P} \mid p \mid \leq \sum_{\mathcal{S}_a^{(j)k} \in \mathcal{S}_a^{(j)}} \mid \mathcal{S}_a^{(j)k} \cap p \mid$. Hence, $ProbeCost(q) \leq probeCost(p_c) + min_{e \in \mathcal{E}} C_e * \sum_{\mathcal{S}_a^{(j)k} \in \mathcal{S}_a^{(j)}} \mid \mathcal{S}_a^{(j)k} \cap q \mid - \mid \mathcal{S}_a^{(j)k} \cap p_c \mid$. Further, $\forall \mathcal{S}_a^{(j)k} \mid \mathcal{S}_a^{(j)k} \cap p_c \mid > \mid \mathcal{S}_a^{(j)k} \mid /2$, thus, according to **(iii)**, $ProbeCost(q) \leq probeCost(p_c) + min_{e \in \mathcal{E}} C_e * li(q, \mathcal{S}_a^{(j)}) - li(p_c, \mathcal{S}_a^{(j)})$ □

(v) $\forall \mathcal{S}_a^{(j)k} \in \mathcal{S}_a^{(j)} \mid \mathcal{S}_a^{(j)k} \cap p_c \mid \geq \mid \mathcal{S}_a^{(j)k} \mid /2 \Rightarrow li(q, \mathcal{S}_a^{(j)}) \geq li(p_c, \mathcal{S}_a^{(j)})$

Proof. According to **(i)**, $\sum_{\mathcal{S}_a^{(j)k} \in \mathcal{S}_a^{(j)}} \mid \mathcal{S}_a^{(j)k} \cap q \mid \geq \sum_{\mathcal{S}_a^{(j)k} \in \mathcal{S}_a^{(j)}} \mid \mathcal{S}_a^{(j)k} \cap p_c \mid$. Therefore, according to **(iii)**, **(v)** is true. □

Bibliography

[1] Agrawal, S., Naidu, K. V. M., and Rastogi, R., Diagnosing link-level anomalies using passive probes, In *Proceedings of INFOCOM,* 2007.

[2] Barford, P., Duffield, N. G., Ron, A., and Sommers, J., Network performance anomaly detection and localization, In *Proceedings of INFOCOM,* 2009.

[3] Nick G. Duffield and Francesco Lo Presti, Multicast Inference of Packet Delay Variance at Interior Network Links, In *Proceedings of INFOCOM,* 2000.

[4] Nick G. Duffield and Francesco Lo Presti, Network tomography from measured end-to-end delay covariance, *IEEE/ACM Transaction on Networking,* Vol. 12, N^o 6, pp. 978-992, 2000.

[5] Nick G. Duffield and Francesco Lo Presti, Inferring Link Loss Using Striped Unicast Probes, In *Proceedings of INFOCOM,* 2001.

[6] Bejerano, Y., and Rastogi, R., Robust monitoring of link delays and faults in IP networks, *IEEE/ACM Transsaction on Networking,* Vol. 14, N^o 5, pp. 1092-1103, 2006.

[7] Case, J., Fedor, M., Schoffstall, M., and Davin, J. RFC 1157 (Historic) 1990

[8] Claise, B., Cisco Systems NetFlow Services Export Version 9,. RFC 3954 (Informational), 2004.

[9] Coates, M., and Nowak, R., Network loss inference using unicast end-to-end measurement, In *Proceedings of ITC Conference IP Traffic, Modeling and Management,* 2000.

[10] Coates, M., and Nowak, R., Network inference from passive unicast measurements, Technical Report TR-0002, Rice University, 2000.

[11] Coates, M., and Nowak, R., Network delay distribution inference from end-to-end unicast measurements, In *Proceedings of the IEEE Conference on Acoustics, Speech, and Signal Processing,* 2001.

[12] CPLEX, 2012, `http://www.ilog.com/products/cplex./`

[13] BRITE, 2012, `http://www.cs.bu.edu/brite./`

[14] Cáceres, R., Duffield, N. G., Horowitz, J., and Towsley, D., Multicast-based inference of network-internal loss characteristics, *IEEE Transactions on Information Theory,* Vol. 45, N^o 7, pp. 2462-2480, 1998.

[15] Duffield, N., Simple network performance tomography, In *Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement,* 2003.

[16] Duffield, N., Horowitz, J., Presti, F. L., and Towsley, D., Network delay tomography from end-to-end unicast measurements, In *Proceedings of the Thyrrhenian International Workshop on Digital Communications: Evolutionary Trends of the Internet,* 2001.

[17] Harfoush, K., Bestavros, A., and Byers, J., Robust identification of shared losses using end-to-end unicast probes, In *Proceedings of the International Conference on Network Protocols,* 2000.

[18] N. Feamster, D. G. Anderson, H. B., and Kaashoek, M., Measuring the effects of internet path faults on reactive routing, In *Proceedings of ACM SIGMERTICS,* 2003.

[19] Naidu, K. V. M., Panigrahi, D., and Rastogi, R., Detecting anomalies using end-to-end path measurements, In *Proceedings of INFOCOM,* 2008.

[20] Nguyen, H. X., Teixeira, R., Thiran, P., and Diot, C., Minimizing probing cost for detecting interface failures: Algorithms, and scalability analysis, In *Proceedings of INFOCOM,* 2009.

[21] Nguyen, H. X., and Thiran, P., Active measurement for multiple link failures diagnosis in IP networks, In *Proceedings of Passive and Active Measurement Workshop,* 2004.

[22] Nguyen, H. X., and Thiran, P., Binary versus analogue path monitoring in ip networks, In *Proceedings of Passive and Active Measurement,* 2005.

[23] Paxson, V., End-to-end routing behavior in the Internet, *ACM SIGCOMM Computer Communication Review,* Vol. 36, N^o 5, pp. 41-56, 2006.

[24] Presti, F. L., Duffield, N. G., Horowitz, J., and Towsley, D., Multicast-based inference of network-internal delay distributions, *IEEE/ACM Transaction on Networking,* Vol. 10, N^o 6, pp. 761-775, 2002.

[25] Duffield, N. G. and Horowitz, J. and Lo Presti, F. and Towsley, D., Multicast topology inference from measured end-to-end loss, *IEEE Transaction on Information Theory,* Vol. 48, N^o 1, pp 761-775, 2006.

[26] Ratnasamy, S., and Mccanne, S., Inference of multicast routing trees and bottleneck bandwidths using end-to-end measurements, In *Proceedings of INFOCOM,* 1999.

[27] Stevens, W. R., *TCP/IP illustrated (vol. 1): the protocols,* Addison-Wesley Longman Publishing Co. Inc, 1993.

[28] Waldbusser, S., RFC 1271: Remote Network Monitoring Management Information Base (Proposed Standard), 1991.

[29] Zhao, Y., Zhu, Z., Chen, Y., Pei, D., and Wang, J., Towards efficient large-scale vpn monitoring and diagnosis under operational constraints, In *Proceedings of INFOCOM,* 2009.

[30] Chudak, F. and Chmyos, D., Improved Approximation Algorithms for the Uncapacitated Facility Location Problem, *ACM SIAM Journal on Computing,* Vol. 33, N^o 1, pp. 1-25, 2004.

[31] Waxman, B., Routing of Multipoint Connections, *IEEE Journal on Selected Areas in Communications,* Vol. 6, N^o 9, pp. 1617-1622, 1988.

[32] Kumar, Ritesh and Kaur, Jasleen, Efficient beacon placement for network tomography, In *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement,* 2004.

[33] Alberto Medina and Anukool Lakhina and Ibrahim Matta and John W. Byers, Efficient beacon placement for network tomography, *9th International Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems,* 2001.

[34] Chen, Yan and Bindel, David and Katz, Randy H., Tomography-based overlay network monitoring, In *Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement,* 2003.

[35] Horton, Joseph D. and López-Ortiz, Alejandro, On the number of distributed measurement points for network tomography, In *Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement,* 2003.

[36] Chen, Yan and Bindel, David and Song, Hanhee and Katz, Randy H., An algebraic approach to practical and scalable overlay network monitoring, In *Proceed-*

ings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications, 2004.

[37] David B. Chua and Eric D. Kolaczyk and Mark Crovella, Efficient Monitoring of End-to-End Network Properties, In *Proceedings of IEEE INFOCOM,* 2005.

[38] Spring, N. and Mahajan, R. and Wtherall, D., and Anderson, T.,, Measuring ISP Topologies with Rocketfuel, *IEEE/ACM Transaction on Networking,* Vol. 12, N^o 1, pp. 2-16, 2004.

[39] Pitoura, T. and Triantafillou, P., Distribution fairness in Internet-scale networks, *ACM Transactions on Internet Technology,* Vol. 9, N^o 4, pp. 16:1-16:36, 2009.

[40] Dagum, C., The generation and distribution of income, the Lorenz curve and the Gini ratio, *Economic Appliquée,* Vol. 33, pp. 327-367, 1980.

[41] C. Schmoll, E. Boschi et al., Final Architecture Specification, *INTERMON Deliverable 15,* 2001.

[42] G. Sadasivan, N. Brownlee, B. Claise, J. Quittek, Adaptive diagnosis in distributed systems, *RFC 5470,* 2009.

[43] I. Rish, M. Brodie, Sheng Ma, N. Odintsova, A. Beygelzimer, G. Grabarnik, K. Hernandez, Adaptive diagnosis in distributed systems, *IEEE Transactions on Neural Networks,* Vol. 16, N^o 4, pp. 1088-1109, 2005.

[44] Cheng, Lu and Qiu, Xuesong and Meng, Luoming and Qiao, Yan and Boutaba, Raouf, Efficient active probing for fault diagnosis in large scale and noisy networks, In *Proceedings of IEEE INFOCOM,* 2010.

[45] Mark Brodie and Irina Rish and Sheng Ma and Natalia Odintsova, Active probing strategies for problem diagnosis in distributed systems, In *Proceedings of the International Joint Conferences on Artificial Intelligence,* 2003.

[46] Irina Rish and Mark Brodie and Natalia Odintsova and Sheng Ma and Genady Grabarnik, Real-time problem determination in distributed systems using active probing, In *Proceedings of the Network Operations and Management Symposium,* 2004.

[47] Natu, Maitreya and Sethi, Adarshpal S., Probabilistic fault diagnosis using adaptive probing, In *Proceedings of the Distributed systems: operations and management 18th IFIP/IEEE international conference on Managing virtualization of networks and services,* 2007.

[48] Natalia Odintsova and Irina Rish and Sheng Ma, Multi-fault Diagnosis in Dynamic Systems, In *Proceedings of Integrated Management,* 2005.

[49] Mark Brodie and Irina Rish and Sheng Ma, Optimizing Probe Selection for Fault Localization, In *Proceedings of Distributed Systems Operation and Management,* 2001.

[50] Maitreya Natu, Active probing approach for fault localization in computer networks, In *Proceedings of End-to-End Monitoring Techniques ans Services,* 2006.

[51] Ahuja, S.S. and Ramasubramanian, S. and Krunz, M., SRLG Failure Localization in Optical Networks, *IEEE/ACM Transactions on Networking,* Vol. 19, N^o 4, pp. 989-999, 2011.

Acknowledgment

I am sincerely grateful for my advisors Prof. Bernard Cousin and Dr. Samer Lahoud for their systematic guidance, consistent encouragement and availability. I extremely appreciate their kindness and willingness to help whenever I was in need.

I would like to mention that the work presented in this dissertation was supported by Alcatel Lucent Bell labs under the grant $n^o 09CT310 - 01$. I especially thank Mr. Nicolas le Sauze and Mr. Richard Douville for their valuable remarks.

I am thankful for all my colleagues at IRISA, especially Alia bellabas and Farah Moety for their continuous support, and nice friendship.

Words cannot express my deep gratitude for my parents, my husband, my brothers, and my mother-in-low. Their support and patience are invaluable !

Above all I owe it all to the One and Only, the Eternal, the Absolute, Who begetteh not, nor is He begotten, and there is none comparable unto Him.