Multi-Constrained Quality of Service Routing in Networks

Walid Khallef

THÈSE POUR OBTENIR LE GRADE DE DOCTEUR DE L'UNIVERSITÉ DE MONTPELLIER

En Informatique

École doctorale I2S*

Unité de recherche LIRMM, UMR 5506

Multi-Constrained Quality of Service Routing in Networks

Présentée par Walid Khallef
Le 24/11/2017

Sous la direction de Miklós Molnár et Sylvain Durand

Devant le jury composé de

Josep Solé Pareta,	Prof, Université Politècnica de Catalunya	Rapporteur
Nicolas Montavont,	Prof, IMT Atlantique	Rapporteur
Abderrahim Benslimane,	Prof, Université d'Avignon et des Payes de Vaucluse	Examinateur
Gilles Trombettoni,	Prof, Université de Montpellier	Examinateur
Miklós Molnár,	Prof, Université de Montpellier	Directeur de thèse
Sylvain Durand,	MCF, Université de Paul Valéry	Co-encadrant

UNIVERSITÉ DE MONTPELLIER

*I2S : ÉCOLE DOCTORAL INFORMATION STRUCTURES SYSTEMES

RÉSUMÉ

Au cours des dernières années, avec la croissance spectaculaire de nombre d'objets connectés sur Internet, les réseaux deviennent de plus en plus complexes et difficiles à gérer. Nous avons maintenant différents types de données échangées entre ces objets, dans des liens d'interconnexions (réseau), comme les données multimédia (voix, images, vidéos,. . . , etc) par exemple.

Les applications multimédia exigent des conditions rigoureuses de Qualité-de-Service (QoS) telles qu'un délai de bout en bout minimal, une gigue bornée, et une utilisation efficace de la bande passante. Les protocoles de routage traditionnels on été conçus pour un trafic de données «*Best-effort* ». Ils construisent des routes principalement basées sur la connectivité et l'optimisation du coût de transmission des données. De telles routes ne peuvent pas satisfaire la QoS à cause du manque de ressources. Récemment, plusieurs algorithmes de routage sensible à la QoS ont été proposés pour trouver des routes réalisables. Cette thèse donne un bref état de l'art pour les problèmes de routages avec QoS dans les réseaux.

En fait, il y a deux aspects impliqués par le routage : les protocoles de routage et les algorithmes de routage. Nous nous intéressons dans cette thèse aux algorithmes de routage qui permettent de respecter une QoS demandée. Il existe dans la littérature des algorithmes exacts, approchés ou heuristiques pour les problèmes unicast / multicast. Dans un premier temps, nous allons faire un aperçu sur les travaux concernant le problème de routage unicast et notre proposition. Ensuite, nous allons expliquer le problème de routage multicast avec QoS et notre contribution en ce qui le concerne. Enfin, nous présentons notre travail lié au problème de routage avec QoS en utilisant le protocole de routage RPL (*Routing Protocol for Low Power and Lossy Networks*), qui est un protocole standardisé, de plus en plus utilisé.

Problématiques étudiées

En raison du trafic important des données multimédia (le trafic vidéo devrait représenter 80% du trafic Internet en 2019 et la vidéo à la demande devrait doubler dans la même année) [2], les applications multimédia sont largement utilisées. En effet, ces applications exigent plusieurs contraintes de QoS afin d'assurer une lecture continue du flux multimédia. Le routage avec des contraintes de QoS multiples est un sujet d'actualité. Il est largement étudié dans la littérature [14, 51]. Le délai, la gigue, la bande passante et le coût sont des paramètres indispensables lors de l'identification de la route unicast / multicast. Dans cette thèse, nous exposons le problème de recherche suivant : Comment trouver la route multicast / unicast qui respecte de multiples contraintes de QoS avec le meilleure utilisation possible des ressources ?

Routage unicast avec contraintes de QoS multiple

Pour calculer des chemins, la plupart des techniques utilisées sont basées sur les algorithmes de Dijkstra [13] ou Bellman-Ford [8] qui ne suivent qu'un seul objectif et ne tolèrent aucune contrainte. Pour concevoir des algorithmes pour le routage avec QoS, l'approche principale est de généraliser ces algorithmes pour qu'ils puissent satisfaire plusieurs contraintes et de développer de nouveaux algorithmes. Pour garantir la qualité de service, il suffit de trouver un chemin réalisable (qui satisfait les contraintes) issu de la source. Ce problème est défini comme problème du chemin multi-contraint (*Multi-Constrained Path problem*, MCP), il est NP-Complet [77]. Si nous cherchons un chemin réalisable avec un coût minimal, le MCP devient le problème du chemin multi-contraint optimal (*Multi-Constrained Optimal Path*, MCOP). Parmi les algorithmes les plus efficaces nous trouvons SAMCRA [54] qui est un algorithme exact. Il se base sur celui de Dijkstra et utilise trois concepts fondamentaux : une définition non linéaire de la longueur d'un chemin [75], le concept de k plus courts chemins [15], et la notion de dominance [28]. La valeur de k est ajustée indépendamment à chaque nœud pendant le calcul des chemins. Sa complexité est de $O(k|V|log(k|V|)+k^2M|E|)$ où V est l'ensemble des nœuds du réseaux et E l'ensemble de ses liens. k est une borne supérieur du nombre de chemins possibles entre deux nœuds et M le nombre de contraintes. SAMCRA peut résoudre les problèmes MCP et MCOP. TAMCRA [57] est la version heuristique de SAMCRA. La seule différence entre les deux c'est que le nombre k est borné par une constante dans TAMCRA.

Notre contribution pour le problème de routage unicast avec des contraintes de QoS multiple

Les problèmes MCP et MCOP sont respectivement NP-Complet et NP-difficile. Jusqu'à maintenant il n'existe pas d'algorithme polynomial pour résoudre ce type des problèmes. Nous avons proposé une version modifiée de l'algorithme le plus connu SAMCRA pour résoudre le problème MCP nommé (*Fast Multiple Constraints Routing Algorithm*, FMCRA) .

Le facteur principal qui influence la complexité de l'algorithme SAMCRA est le nombre k de sous chemins sauvegardés à chaque nœud. La valeur de k étant ajustée indépendamment à chaque nœud pendant le calcul des chemins, il y a un risque de calculer des sous chemins qui ne donnent pas des solutions faisables vers la destination. La version heuristique TAMCRA borne le nombre k dans tout les nœuds par une constante, pour accélérer la recherche et limité la complexité de l'algorithme. Lorsque le $k+1$ expesant ième sous-chemin est le chemin faisable et que les k premiers chemins ne le sont pas, alors l'algorithme va manquer la solution alors qu'elle existe.

Nous avons proposé un compromis avec l'algorithme FMCRA : garder la complétude des calculs en les accélérant. Notre algorithme comporte deux étapes principales : afin de diminuer le nombre de sous-chemin faisable dans chaque nœud, il rend les contraintes le plus strict possible, via le calcul des plus courts chemins pour chaque contrainte. FMCRA cherche alors une solution en utilisant ces nouvelles contraintes. S'il trouve un chemin faisable, il s'arrête sinon il cherche à nouveau la solution avec des contraintes relaxées. Comme FMCRA est un algorithme incrémental, il ne recalcule pas les chemins déjà calculés, mais juste d'autres chemins qui respectent les nouvelles contraintes jusqu'à trouver la solution si elle existe. L'algorithme garantit donc la solution avec un temps d'exécution qui dépend de difficulté du problème (cf. Section 2.3).

Routage multicast avec contraintes de QoS multiple

Le multicast, consiste à envoyer les mêmes données à plusieurs destinataires. Les applications multimédia telles que les vidéoconférences et les applications collaboratives nécessitent une gestion efficace des ressources réseau. Le routage multicast a été introduit dans ce sens pour réduire les coûts de transmission et permettre la gestion des groupes. Le problème de routage multicast avec QoS est lié à plusieurs problèmes de graphes connus pour être NP-Complets. Couvrir la source et un ensemble de destinations avec un coût minimum sans s'occuper des contraintes est le classique problème de Steiner dans les réseaux [80]. Dans la littérature, on divise le problème de routage multicast avec QoS en deux types fondamentaux.

- **Multi-Constrained Multicast (MCM) :** l'objectif est de trouver un arbre qui recouvre l'ensemble des destinations sachant que chaque chemin de la source vers une destination doit respecter les contraintes de QoS.

- **Multi-Constrained Minimum Cost Multicast (MCMCM) :** MCMCM est une combinaison du problème MCM et du problème de Steiner. On cherche donc un arbre de coût minimal qui recouvre la source et l'ensemble de destinations en respectant les contraintes de QoS.

On trouve dans la littérature plusieurs solutions exactes et approchées pour résoudre le problème de Steiner [24, 67]. On s'intéresse dans cette thèse au problème MCMCM.

Jusqu'ici, la plupart des solutions proposées pour le problème MCMCM construisent un arbre de coût minimum enraciné à la source qui couvre les destinations en respectant les contraintes de QoS de bout en bout. Cependant, Molnar et al. [56] ont montré que la solution optimale pour le problème MCMCM n'est pas un arbre mais une hiérarchie. La notion de hiérarchie n'a commencée à être définie et caractérisée qu'à partir de 2010. Pour mieux comprendre et expliquer l'intérêt des hiérarchies, nous utiliserons les notations suivantes.

Homomorphismes et hiérarchies de recouvrement

Un parcours non-élémentaire d'un graphe G n'est pas un sous-graphe, car il existe au moins un sommet de G qui s'y trouve plus d'une fois. En appliquant cette propriété aux arbres de recouvrement, on peut envisager une nouvelle structure, qui serait, dans le cas élémentaire un arbre de recouvrement classique et dans le cas non-élémentaire un arbre qui contiendrait plusieurs fois les sommets et les arêtes du graphe couvert. Dans le cas non-élémentaire, cette structure n'est pas un sous-graphe du graphe couvert. Cette structure, appelée hiérarchie et introduite dans [55], est un homomorphisme d'un arbre dans le graphe.

Un homomorphisme est une application $h : W \longrightarrow V$ entre deux graphes $T = (W,F)$ et $G = (V,E)$ qui associe un sommet de W à chaque sommet de V en préservant l'adjacence. Le triplet (T,h,G) définit une structure de recouvrement dans G. Si T un arbre alors le triplet définit une hiérarchie dans G.

Notre contribution pour le problème de routage multicast avec multiple contraintes de QoS

Comme nous l'avons expliqué précédemment les hiérarchies sont les solutions optimales pour le problème MCMCM. Nous avons proposé dans cette thèse un Programme Linéaire en Nombres Entiers (PLNE) efficace et simple à implémenter pour résoudre le problème MCMCM. Il a été testé

sur des instances de graphes qui simulent les réseaux réels. L'obtention de solutions optimales permet aussi d'analyser l'efficacité des heuristiques proposées pour le problème MCMCM.

Nous modélisons la hiérarchie comme un flot dans un multi-graphe orienté pour les deux raisons suivantes :

- La hiérarchie contient un chemin orienté de la source vers chaque destination.
- La hiérarchie peut utiliser le même arc ou le même nœud plusieurs fois. Cette spécificité est modélisée en dupliquant les liens du réseau initial, pour savoir combien de fois chaque arc a été utilisé.

Notre solution est basée sur les propriétés suivantes. Un arc peut être traversé par d chemins dans le pire des cas où d est le nombre de membres du groupe multicast. Pour cette raison, on duplique chaque arrêt d fois dans chaque sens ce qui conduit à travailler dans un multi-graphe $G' = (V, 2d|E|)$. Plusieurs chemins de la solution peuvent utiliser le même arc dans le multi-graphe s'ils possèdent un préfixe commun dans la solution. Sinon (si les préfixes sont différents) les chemins utilisent des occurrences différentes. Ainsi, lorsque le flot se sépare en un nœud, nous contraignons les chemins correspondants à ne pas réutiliser un arc commun de G' (pour plus de détails, voir Section 3.4 dans la thèse).

La complexité des algorithmes de routage avec QoS est fortement corrélée avec le nombre d'arêtes / arcs. Nous avons proposé également dans cette thèse un algorithme nommé ArceReduce, pour réduire l'espace de recherche dans un graphe pondéré. Le rôle d'ArcReduce est de supprimer toute des arrêtes / arcs qui ne peut pas être utilisé dans la construction des routes et pour résoudre ensuite le problème dans un graphe réduit. Nous avons testé l'algorithme sur différentes topologies, les résultats ont montré qu'ArcReduce est très efficace quand les contraintes sont fortes (cf. Section 3.5).

Routage avec contraintes de QoS multiple dans RPL

RPL est un protocole de routage IPv6 à vecteur de distance qui construit un DODAG (*Destination Oriented Directed Acyclic Graph*, « graphe orienté acyclique »)[81]. Malgré les efforts de RPL pour améliorer le routage, la qualité de service a été quelque peu négligée. RPL est capable de fournir différents niveaux de QoS basés sur les métriques et les contraintes de QoS. En effet, RPL sélectionne les nœuds et optimise les itinéraires en utilisant une Fonction Objective (OF) [70] qui définit comment l'itinéraire sera construit. L'OF est séparée de la spécification du protocole de base ce qui donne à RPL la souplesse nécessaire pour répondre à différents critères d'optimisation requise par différentes applications. Ils existent trois OFs prédéfinies par RPL [70][20] (OF0, MRHOF-ETX et MRHOF-Énergie), toutes les trois essayent de construire un DODAG en respectant des métriques spécifiques. Par exemple, OF0 construit un DODAG avec un nombre de sauts minimum tandis que MRHOF-Énergie minimise la consommation d'énergie.

Certaines applications exigent plusieurs contraintes de QoS simultanément. Pour cette raison, Karkazis et al. [35] ont proposé une méthode pour quantifier les paramètres de routage afin de les combiner d'une manière additive ou lexicale. C'était la première OF permettant la construction du DODAG en prenant en compte plus d'une métrique à la fois. Une autre OF a été proposé par Olfa et al. [18] pour construire un DODGA en prenant en compte des contraintes de QoS multiple. Cette OF se base sur la logique de floue (*Fuzzy logic objective function*, OF-FL). Le principal inconvénient de cette fonction est le fait qu'elle essaye d'optimiser le DODAG de façon générale. Or, les applications multimédia exigent des contraintes de bout-en-bout bien précises,

(par exemple un délai maximum entre source et destination) qui ne sont pas prises en compte avec cette OF.

Notre contribution pour le routage avec contraintes de QoS multiple dans RPL

Afin d'améliorer la QoS dans les réseaux LLNs, nous avons proposé une OF basée sur la longueur non linéaire (*Non-Linear Objective Function, NL-OF*) qui peut à la fois prendre en compte plusieurs métriques de QoS et respecter les contraintes de bout-en-bout. Nous avons montré sur la base d'une évaluation expérimentale en utilisant le simulateur Cooja que notre fonction objective est meilleure de celles qui ont été proposé précédemment par RPL (cf. 4.4). Nous avons aussi proposé un algorithme exact qui garantit la construction des chemins faisable de la source vers chaque destination si ce chemin excite (NL-OF-exacte). Pour éviter le calcul excessif, nous avons proposé un algorithme paramétré et moins complexe (NL-OF-k-limité). Les tests de ces trois algorithme ont été effectués en respectant un ensemble de contraintes de QoS. Les résultats ont montré que l'OF de RPL peut être modifiée en gardant sa performance et sa flexibilité en utilisant la NL-OF-k-limité. A l'inverse la solution exacte n'est pas recommandée pour les LLNs, son temps de calcul étant trop important. De plus, le temps de calcul des algorithmes utilisant les NL-OF et NL-OF-k-limité augmente linéairement avec la taille du réseau ce qui reflète la scalabilité de ces algorithmes.
Cette thèse se divise en cinq chapitres :

- **Le chapitre 1** présente une brève introduction au routage avec contraintes de Qualité de Service multiples dans les réseaux, impliquant l'évolution des réseaux et certains concepts de base de l'architecture du réseau. Ensuite, la motivation de ce travail et le besoin de routage avec QoS sont présentés. Trois problèmes de routage sont ensuite répertories et définis formellement en fonction de la typologie du problème abordé.

- **Le chapitre 2** présente notre contribution au routage unicast avec QoS. Nous proposons un algorithme exact, qui manipule les contraintes pour améliorer de manière significative le temps de calcul. L'algorithme nommé FMCRA est une amélioration de l'algorithme SAMCRA. Nous présentons également des résultats numériques pour montrer les performances de FMCRA en fonction de différents paramètres. Ce travail a été présenté lors de la 27ème édition de la conférence "European Conference on Operation Research", EURO 2O15 [37].

- **Le chapitre 3** présente un Programme Linéaire en Nombres Entiers PLNE efficace pour résoudre optimalement le problème MCMCM. Il a été testé sur des instances de graphes qui simulent les réseaux réels. Comme MCMCM est NP-Complet, un algorithme de prétraitement efficace (ArcReduce) est proposé, pour converger rapidement vers une solution lorsque le problème est très contraint. Dans certains cas, ArcReduce peut même détecter en temps polynomial, si le problème en question a une solution réalisable ou non avant de commencer le processus de résolution.

- **Le chapitre 4** présente la troisième contribution de la thèse, où nous proposons une nouvelle Fonction Objectif, basée sur le concept de la longueur non linéaire NL-OF pour le protocole de routage IPv6 pour les LLNs (de l'anglais *Low Power and Lossy Networks*) nommé RPL. NL-OF est la première OF qui peut assurer un ensemble de contraintes de bout en bout dans les LLNs. Une première partie de ce chapitre vise à présenter NL-OF. Dans la deuxième partie nous présentons l'évaluation de la performance de NL-OF en

utilisant le simulateur Cooja avec différents OFs dans les littératures. Ce chapitre est basé sur des travaux publiés (voir [38] et [39]).

- **Le chapitre 5** conclut la thèse et présente quelques axes de recherche prometteurs.

Abstract

In recent years, the network traffic has been growing explosively. The main reason behind this growth comes from numerous multimedia applications (e.g. Video on Demand, video-conferencing). These applications require several constraints of Quality of Service (QoS) to operate properly. Therefore, delivering this enormous amount of packets to their intended destinations (i.e. routing) with respect of the QoS constraints brought serious challenges for the next generation of networks. In this thesis, we study the multi-constrained QoS routing in network. The objective is to find the route, for instances of wired and wireless networks, to distribute the message to the destination(s) while taking into account both the constraints of QoS and minimizing the total cost.

The first chapter presents a brief introduction to the routing with multiple constraints of QoS in networks, involving the evolution of networks and some basic concepts of network architecture. Then, the role of routing and the need of QoS are presented with the motivation of this work. This chapter presents also the formulation of the problems under consideration.

In order to analyse the Multi-Constrained Path problem (MCP), an efficient algorithms based on an earlier MCP algorithm named SAMCRA is proposed. This algorithm is shown to be able to improve the execution time while maintaining the quality of the solution.

Concerning the Multi-Constrained Multicast Minimum Cost problem (MCMCM), a new Integer Linear Programming (ILP) formulation is proposed to model the hierarchy as an exact solution for MCMCM. It is proven that the optimal structure of MCMCM is neither a tree nor a sub-graph but a hierarchy. An efficient preprocessing-based algorithm (ArcReduce) is also proposed to accelerate the resolution time of the ILP in large size networks.

Regarding the problem of routing with QoS in Low Power and Lossy Networks (LLNs), a new Objective Function (OF) based on Non-Linear Length (NL-OF) is proposed. The standard IPv6 routing protocol for LLNs called (RPL) builds acyclic graphs and applies an OF which is responsible of choosing the best links during the construction of the route map. The NL-OF is the first OF that takes into account any number of metrics and constraints for QoS routing. An exact and some heuristic routing algorithms with QoS constraints for LLNs are also proposed.

Key words: Quality of Service (QoS), Multicast Routing, Unicast Routing, Network, Integer Linear Programming (ILP).

TABLE OF CONTENTS

List of Tables

Table **Page**

List of Figures

CHAPTER 1

INTRODUCTION

This chapter presents a brief introduction to the routing with multiple constraints of Quality of Service (QoS) in networks, involving the evolution of networks and some basic concepts of network architecture. Then, the role of routing and the need of QoS are presented with the motivation of this work. Three routing problems are then classified according to their challenge. The problem formulation and the organisation of the thesis are presented at the end of the chapter.

1.1 Introduction

After several experiments to connect computers together in the late 1950s and early 1960s [49], the computer industries became interested in computer networks. It has been witnessed a large body of work to build computer networks in order to connect and share computers over a long distance. Firstly, the network size was limited in term of geographical area using different technologies such as Ethernet or Token Ring. Because of the massive increase of data flowing through the networks the need of interconnect more computers was a challenging effect. The most important innovation has been the Internet protocol suite (TCP/IP), which allows the combination of different protocols at various layers [16]. The layer concept is a way of sub-dividing a communication system into smaller parts (link layer, Internet layer, transport layer and application layer). This thesis focuses on Internet layer, which is responsible for the transmission of packets across the network (i.e. routing).

Routing is the process of selecting a route to move a packet of data from a source to a destination (or more) through intermediate network nodes by specific packet forwarding mechanisms.

The exchange of information through a network can be made using three main transmission modes:

- Unicast mode forwards a message from a source to one receiver.
- Multicast mode forwards a single copy of the message from a source to N receivers.
- Broadcast mode forwards a message from a source to all nodes in the network.

Each routing protocol uses an appropriate algorithm to deliver the data according the above three basic transmission modes. The aim of the routing algorithms is to find a unicast route, multicast or broadcast route.

In some cases and for several important applications, the routing is nowadays subject to different constraints of Quality of Service (QoS). As defined in [2] *"QoS refers to the capability of a network to provide better service to selected network traffic over various technologies"*. The exigences of QoS are defined by network applications and users. Each application has different network performance requirement. The point-to-point delivery model, which is based on the best-effort service only assures that data will reach their destinations as soon as possible, but with no bandwidth or latency guarantees for example. Standard Internet Protocol (IP) provides a best-effort service by default, this is adequate for typical Internet applications such as email and Web applications, but inadequate for sensitive applications with real-time requirements such as Video on Demand. The IETF proposes many solutions to solve the problem of QoS in IP networks using different approaches. First of all, there exists two states types in network: Hard-state and Soft-state [82]. The IETF proposes the use of two principal approaches to QoS in modern packet-switched IP networks, the Integrated Services (IntServ) which is divided into sub-services with different QoS. Three sub-services of QoS are defined for IntServ [22]:

- Best-effort service, provides basic connectivity with no guarantees.
- Differentiated service, provides an acceptable level of end-to-end QoS for specific classes of traffic.
- Guaranteed service, provides the higher level of end-to-end QoS.

Since IntServ requires per-flow state and per-flow processing, which is not scalable in large networks. An aggregation of flows is needed, which lead to the proposal of the Differentiated Services (DiffServ) architecture [9] as a second QoS approach for IP.

Notice: Delay, delay variation (jitter), bandwidth and packet loss are usually considered, to quantitatively measure the Quality of Service.

With the recent substantial growing of multimedia traffic (video and voice), a large number of clients use the multimedia applications (e.g. Video on Demand, Game on Demand and Internet protocol Television). The video traffic is expected to represent 80% of Internet traffic in 2019

and the Video on Demand is expected to double in the same year [2]. So, it is very important to support the QoS in telecommunication systems, in order to satisfy the desires or the requirements of both users and applications. The QoS requirement differs from an application to another and from a user to another. In other words users can differ in their perception of what is good quality and what is not (e.g. Game on Demand is very sensitive to packet loss and jitter while the Video on Demand is very sensitive to jitter). For this reason, a routing strategy that takes into consideration several constraints of QoS simultaneously is needed. This thesis provides a depth study of routing with multiple constraints of QoS and contributes to solve the most known QoS routing problems in the literatures.

In what follows, we present briefly the routing problems with multiple constraints of QoS and our standard formulation of this problems.

1.2 Routing with multiple constraints of QoS

As mentioned above, various real-time applications are being deployed over the Internet (IPTV, VoD,. . .). They require the routing algorithm to provide routes satisfying the QoS constraints in both wired and wireless networks [14][51]. Routing includes two entities, routing protocol and routing algorithm. A routing protocol specifies how routers communicate with each other, where routing algorithms determine the specific choice of route. In this thesis we focus on the QoS-aware routing algorithms for unicast and multicast in the Internet and the routing strategies for low power and lossy networks in the IPv6 based Internet. The metrics of QoS can be classified into three basic composition rules:

- Additive metric, is a metric that can be calculated by the sum of values of the links constituting the path, (e.g. delay, delay jitter, hopcount and cost).

- Multiplicative metric, is a metric that can be calculated by the multiplication of values of the links constituting the path (e.g. packet loss rate).

- Bottleneck metrics, is the maximum or the minimum of the metric over all links in the path (e.g. bandwidth). This metric can be easily dealt using a preprocessing step called topology filtering.

In our model presented in 1.3, we only consider the additive metrics. Multiplicative metrics can be transformed into additive metrics by using a logarithm function. The bottleneck metrics can be dealt by pruning from the graph all the links that do not satisfy the value of bottleneck. For multicasting, the bottleneck metrics cannot be dealt by a simple preprocessing-based step (we shall see that in the Chapter 3).

1.3 Problem formulation and definitions

1.3.1 Problem formulation

Let $G = (V,E)$ be a weighted connected undirected graph representing the network topology, where V is the set of vertices (nodes) and E is the set of edges representing the possible links. A directed graph will be $G = (V,A)$, where A is the set of arcs. Each edge e is associated with M weights corresponding to the QoS metrics (given by a weight vector $\vec{w}(e) = [w_1(e), w_2(e), ..., w_M(e)]^T$, where $w_i(e) > 0 \ \forall (e) \in E$) and $c(e)$ is the cost of using the link. We will distinguish two main problems:

1. **Multi-Constrained Paths problem (MCP):** given a source node s, a destination d, a set of end-to-end constraints $\vec{L} = [L_1, L_2, ..., L_M]^T$, the objective is to find a feasible path from s to d which satisfies the constraints requirements $\vec{L}$. To simplify, let us notice the path from s to d by p_d. As shown in Figure 1.1.A), the feasible path is $p_d = \{s, b, f, d\}$ where $w_1(p_d) = 5$, $w_2(p_d) = 5$ and $c(p_d) = 4$.

   For additive metrics, the weight of a path p_d corresponding to the metric i is given by:

   $$w_i(p_d) = \sum_{e \in p_d} w_i(e). \tag{1.1}$$

   The problem is to find a path p_d such that:

   $$w_i(p_d) \leq L_i, \forall i = 1, \ldots, M. \tag{1.2}$$

   In some cases, an additional optimisation goal can be considered by trying to minimize $c(P_d) = \sum_{e \in p_d} c(e)$ over all feasible paths. This problem is known in the literature as the Multi-Constrained Optimal Path problem (MCOP) [42].

2. **Multi-Constrained Multicast problem (MCM):** Given a source node s, a set of destinations $D = \{d_1, \ldots, d_d\}$, find a feasible multicast route that connect the source with the destinations set satisfying the QoS constraints for each destinations. For instance in Figure 1.1.B the objective is to connect the source s with the de destinations set $D = \{a, f, d\}$.

   More formally for s, D given, find a multicast route such that for each destination d_j, there exist a path p_{d_j} from s to d_j:

   $$w_i(p_{d_j}) \leq L_i, \forall i = 1, \ldots, M. \tag{1.3}$$

   MCM problem with optimisation of the cost is known as Multi-Constrained Multicast Minimum Cost problem (MCMCM).

Hereafter, we present informally classes P and NP that will be frequently used in this thesis.

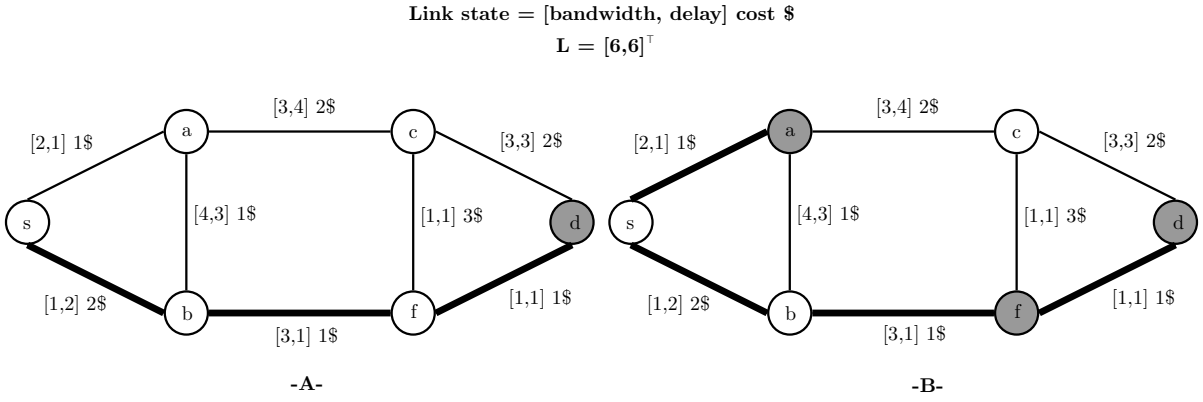

Figure 1.1: Example of multiple constraints problem (Unicast and Multicast) .

1.3.2 Complexity class

In Complexity Theory, problems are classified into classes. The class P refers to those decision problems for which one can devise a polynomial time algorithm to solve them. However, certain decision problems are different in the sense that one can only be able to devise a polynomial time verifier for their solutions. That is to say, if a solution is presented for the problem then it is checkable in a polynomial time. This class is referred to as NP. Obviously $P \subseteq NP$, under the widely held assumption P $\neq$ NP, the decision problems of NP\P have no polynomial time algorithm that can solve them [19]. However, this is not yet proven, since it is one of the open problems in computer science. An interesting class of problems in NP is called NP-complete problems. Those decision problems enjoy the property that they are at least as hard as any problem in NP. This has the implication that if one can provide a polynomial time algorithm that solves an NP-complete problem then we can solve polynomially every problem in NP. This would prove that P = NP which is highly unlikely. To prove that a problem C is NP-complete, one needs to prove that it is in NP and an NP-complete problem is reducible to C in a polynomial time, for more details see [19]. The reduction is to be understood as Many-one Karp reduction [36].

Notice that, MCP, MCOP, MCM and MCMCM are all shown to be NP-complete [77].

In the next sections we provide a state-of-the-art review of the QoS-based routing problems.

1.4 Unicast QoS routing

The state of a path can measured in different ways, it can be determined by a simple metric of QoS such as bandwidth or the combination of two or more QoS metrics such as delay and jitter or delay, jitter and cost, etc. Several problems as well as solutions were analyzed and proposed on this subject. In this section, we present an overview on the unicast QoS routing problems based on their difficulty and the most important algorithms for each problem [64].

1.4.1 Unicast routing with single metric

In [64] the author divided the unicast routing problems with single metric into two sets. The first one is called *link-optimization routing*, the aim of this problem is to achieve a single objective of optimization. An example is the minimum-cost flow problem, which consists in finding the cheapest possible way of sending a certain amount of flow through a flow network [21]. Another example, is the *bandwidth-optimization routing,* which aims to find the path with maximum bottleneck bandwidth (the widest path) [77]. The second set is *link-constrained routing*. The problem is to find a path respecting one end-to-end constraint, path whose bottleneck bandwidth is above a required value for example. The link-optimization routing is solvable by polynomial time algorithms, like Dijkstra's algorithm [13] or Bellman-Ford algorithm [8]. The link-constrained routing problem can be reduced to the link-optimization problem [64].

1.4.2 Unicast routing with multiple metrics

Path computing problem for multiple metrics has been widely investigated in the literature. However the problem is inherently hard and a polynomial time algorithm may not exist. Finding a path with more than one constraint has been proven to be NP-complete [77]. Hence, the algorithms are distinguished by their type of solution, which can be exact, approximate or heuristic. Many works suggest that heuristic solution is the best way to treat them. Jaffe presented two algorithms for the MCP problem with two constraints [31]. The first one is a pseudo-polynomial-time algorithm, the second is a polynomial-time algorithm. It uses an objective function which combines the constraints to create a unique constraint. Then the algorithm finds the shortest path. The closest work is done by Chen and Nahrstedt in [11], who propose a heuristic algorithm for MCP. The idea is to first reduce the NP-complete problem to a simpler one, which can be solved in polynomial time, and then solve the new problem by using one of the following two algorithms. The first one is an extended Dijkstra's shortest path algorithm (EDSP). The second one is an extended Bellman-Ford algorithm (EBF). Another algorithm to solve bandwidth-delay-constrained path was proposed by Wang and Crowcroft [77], which is based on two steps. First, all links that do not respect the bandwidth requirement are eliminated so the output is a feasible graph in term of bandwidth. The second consist in finding the shortest path in term of delay using Dijkstra's algorithm.

Several composite routing problems can be derived from the above problems (e.g. delay-packet-loss-constrained path problem, delay-jitter-constrained path problem, etc.). In this thesis we are interesting on multi-constrained path problem with an arbitrary number of constraints with and without optimization goal which is a generalization of this kind of problems. Korkmaz and Krunz propose two heuristic algorithms [42] [41] for MCP and MCOP respectively. The first algorithm, which is named *"A randomized algorithm"* prunes all the links that cannot be on any feasible path. It then uses a randomized heuristic search to find a feasible path, if one exists. The second one named *"heuristic multi-constrained optimal path" (H-MCOP),* uses a non-linear concept. The

heuristic algorithm H-MCOP minimizes the non-linear length for both the feasibility part and the optimality part. We will develop the application of the non-linear length in the next chapters.

TAMCRA [57] is a heuristic algorithm for MCP, which can find a feasible path with minimum non-linear length. It is based on three concepts: the non-domination of paths, the non-linear length and the k-shortest paths approach. It is an adaptation of the well known Dijkstra's algorithm. To find a solution with multiple constraints, at each node and at each step, the algorithm needs to store more than one path. That is why a simple shortest path algorithms as the label fixation based algorithms do not work satisfactorily. In TAMCRA the number of stored paths is bounded by a constant k. The probability of missing a path is negatively correlated with k while the complexity of the algorithm increases, and thus the execution time increases with k.

In the exact solution review of MCP/MCOP, SAMCRA [54] the exact version of TAMCRA was introduced explaining deeply the three fundamental concepts of these algorithms (non-linear length, k-shortest paths and non-domination of paths). Since the heuristics usually find reasonably good solutions reasonably fast, some works have been done to propose approximation algorithm which is a polynomial-time algorithm that produces a solution whose value is within a factor of $1+\epsilon$ of the value of an optimal solution. In [69] Warburton developed a Fully Polynomial Time Approximation Scheme (FPTAS) [1] for delay-constrained last cost path problem on acyclic graphs. In general graphs, Hassin in [26] proposed an FPTAS with $\theta(mn(\frac{n}{\epsilon})\log(\frac{n}{\epsilon}))$ of time complexity, where n is the number of vertices and $1+\epsilon$ is the approximation factor. Two interesting approximation algorithms were proposed by Xue and all in [83] for MCOP with $\theta(mn\log\log\log n + \frac{mn}{\epsilon})$ time $(1+\epsilon)(k-1)$ approximation algorithm and an $\theta(mn\log\log\log n + m(\frac{n}{\epsilon})^{k-1})$ time $(1+\epsilon)$ approximation algorithm. Here n is the number of vertices and m the number of edges, k the number of additive QoS metrics associated with each edge. Both algorithms provide an $(1+\epsilon)$ approximation when k is reduced to 2. Since $P \neq NP$ is not yet proven, it may be worth to design algorithms that seek an exact solution for NP problem. Sometimes an exponential-time algorithm may be better in practice than a polynomial time algorithm (e.g. with small size networks). Therefore, SAMCRA [54], the exact version of TAMCRA, was proposed. SAMCRA guarantees to find a multi-constrained path with minimum non-linear length if it exists. As TAMCRA, SAMCRA uses the three concepts. The only difference is that SAMCRA allocates queue-space when it needs, although in TAMCRA the allocated queue-space is predefined. SAMCRA can be applied to solve MCP and MCOP. In what follows, we explain of SAMCRA's concepts:

1. As mentioned above, each link e is associated with an M-dimensional weight vector $\vec{w}(e)$ and the QoS requirement is given by $\vec{L}$. The end-to-end weight vector $\vec{w}(p_d) = \sum_{e \in p_d} \vec{w}(e)$ can be associated to path p_d, which is the vector sum of the link weights along this path. The quality of a path p_d from the source s to the destination d can be measured by the

[1] An algorithm A is an FPTAS for an optimization problem P, if given an input I for P and $\epsilon > 0$, A finds in polynomial time in the size of I and in $\frac{1}{\epsilon}$, a solution S for I that satisfies: $|val(I) - val(S)| \leq \epsilon val(I)$, where $val(I)$ is the optimal value of a solution for I.

following non-linear length :

$$l(p_d) = \max_{i=1,\ldots,M} \left(\frac{\sum w_i(e)_{e \in p_d}}{L_i}\right) \tag{1.4}$$

This length function permits to normalize the constraints and take into account the most critical one for a path. This allows to verify the feasibility of path regarding all constraints simultaneously.

2. As it is mentioned in [44] *a sub-path of a shortest path using the non-linear length is not necessarily a shortest paths*. Hence, SAMCRA uses the multiple shortest path concept, which allows it to store all non-dominated paths possible in the queue-space.

3. To avoid storing unnecessary paths in queue-space and reduce the research space of the solution, the Pareto dominance is used. A path p_d dominates another path p'_d if: $w_i(p_d) \leq w_i(p'_d)$ for $i = 1,\ldots,M$. In Figure 1.2, node f has to memorise the two sub-paths $sp = \{s,b,c,f\}$ and $sp' = \{s,b,e,f\}$ because for $i = 1,\ldots,i$ neither $w_i(sp) \leq w_i(sp')$ nor $w_i(sp') \leq w_i(sp)$ for all i.

In order to understand SAMCRA's resolution operation, Figure 1.2 shows an example when the objective is to find a path from the source s to the destination d with respect to the end-to-end constraints $\vec{L} = [8,8]^T$.

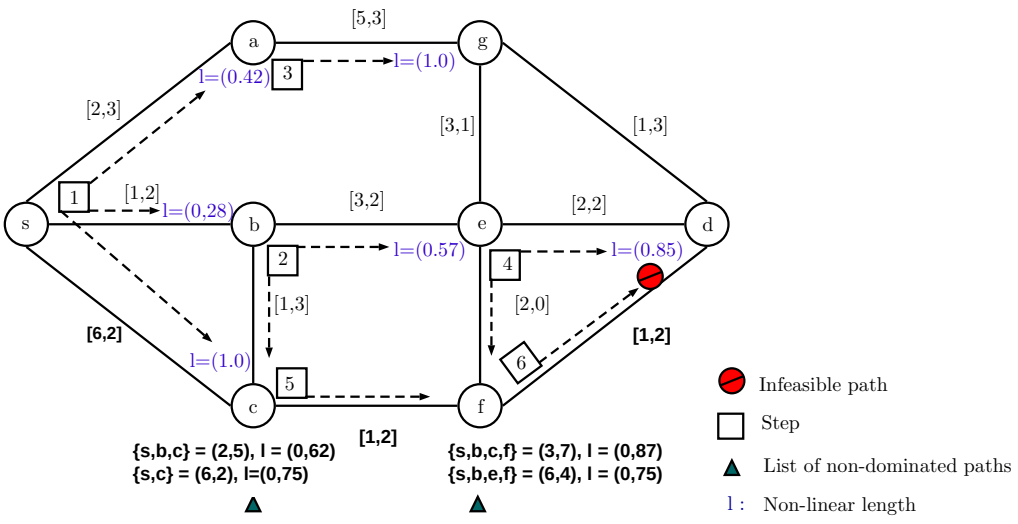

Figure 1.2: Example of the operation of SAMCRA.

To find a feasible path SAMCRA begins at the source node s. At each iteration, the algorithm explores neighbours of a selected current node using the non-linear length function defined in Equation 1.4. The dominated path are regularly dropped, while all non-dominated ones are memorized. In Figure 1.2 for node c, there are two non-dominated sub-paths reaching this node. SAMCRA ends when the destination d is reached and there is no possible improvement. For node

f, the path (s,c,f) is dominated by (s,b,c,f) and thus eliminated. It remains paths (s,b,e,f) and (s,b,c,f).

1.5 Multicast QoS routing

The routing algorithms we have studied so far are designed to provide a constrained path from a single source to a single destination. They are often refereed to as unicast algorithms.

Usually, multicast is used for communications between a single source and multiple destinations in a network. As in the case of the unicast problem, routing algorithms also play a central role in the network layer and multiple variations exist. In this section we present three important variations of the multicast problem and some routing algorithms which have addressed this problem in the literature.

1.5.1 Multicast routing with single metric

The best way to model the minimum cost multicast problem is using the Steiner tree problem in graphs. Given a graph $G = (V,E)$ with non-negative edge weights and a subset of vertices S (source and destinations), find a tree of minimum weight that contains all nodes in S. The decision variant of the Steiner tree problem in graphs is NP-complete and was among Karp's original 21 NP-complete problems. Due to the diversity of applications of Steiner tree problem (e.g. routing problem, design of electronic circuits, etc.), Steiner trees have been extensively studied. Among the proposed solution methods, there are exact, approximation and polynomial algorithms for special instances. Interesting surveys are given in [80] and [29].

Hakimi proposed the Steiner tree enumeration algorithm (STEA) [24] which is an exact algorithm. STEA begins by enumeration all possible subsets SS of nodes that do not belong to S. Then, for all subsets SS, it computes the Minimum Spanning Tree (MST) in the graph generated by $S \cup SS$. The calculation of MST can be fulfilled using any exact algorithm such as Kruskal's algorithm. However, in real applications the exact algorithms may not worth it since even small problem instances take much time to be solved. In certain conditions heuristics have to be applied to produce a solution in a reasonable time. An efficient $\theta(n \log n)$ heuristic was proposed by Smith in [65]. In some cases we need polynomial algorithm with guarantee of performance in different instances. In [67] Tkahashi and Matsuyama proposed an $\theta(|S|n^2)$ time algorithm. The ratio between the cost of the obtained solution and the optimal cost is $2 \times (1 - 1/|S|)$, in the worst case.

1.5.2 Multicast routing with two metrics

The Constrained Steiner Tree (CST) problem is the minimum Steiner tree problem on graph with one additional constraint. For example, the problem of finding a delay-bounded least-cost multicast tree is a CST. Since CST is very expensive to compute (it is also NP-hard), several heuristics were proposed to solve it. Shaikh and Shin proposed the Destination-Driven

MultiCasting (DDMC). The idea of DDMC comes from the well-known algorithms of Prim's minimum spanning tree and Dijkstra's shortest path [13] using a special cost function which gives preference to the paths going through destination nodes seeking a feasible tree in term of delay with lowest cost as possible [63]. In [52] Ibrahim proposes QoS Dependent Multicast Routing algorithm (QDMR) for generating delay-bounded least-cost tree, which is a modification of DDMC. Another heuristic has been proposed by Kompella et al [40], which study the problem as Steiner tree problem constrained by end-to-end delay from the source to each destination. Comparative surveys with performance evaluation on algorithms were done by Tanaka et al [68].

1.5.3 Multicast routing with multiple metrics

The recent applications such as Video on Demand (VoD) and Game on Demand (GoD), etc. need to satisfy several such requirements in term of QoS. Therefore, it is necessary to propose a model that considers an arbitrary number of QoS constraints, as indicated in Section 1.3. We focus on MCMCM problem, which is the hardest one, since it is a combination of two NP-complete problems (MCM and Steiner tree problem). A good algorithm for solving MCMCM with M constraints named MAMCRA was proposed by Kuipers et all in [44]. This algorithm computes a route which is not always a tree. For this, it follows two fundamental steps. In the first step MAMCRA calculates the set of optimal paths, for example with minimum non-linear length, from the source to each destination using the exact multi-constrained path algorithm SAMCRA. Since this set may contain some overlaps, MAMCRA uses in the second step a greedy algorithm to eliminate some of these overlaps without violating the constraints. Later Tasai and Chen [71] also proposed two MCMCM algorithms multi-constrained QoS Dependent Multicast Routing (M-QDMR) and multicast routing with Multi-Constrained Optimal Path selection (M-MCOP). Both algorithms are based on two main phases. The first phase (Constrained-Dijkstra phase) seeks to create a partial multicast tree that covers as many destinations as possible. In the second phase, find a feasible path which connects the source and each uncovered destination by applying a modified "heuristic multi-constrained optimal path" H-MCOP algorithm and then coalesce the path onto the tree based on three procedures.

- Reverse-Dijkstra: It is a slight modification of Dijkstra's algorithm which computes each path weights from every node to the destination.
- Look-Ahead-Dijkstra: It was used firstly in H-MCOP algorithm, Its aim is to find the least cost path with high feasibility using the data collected by Reverse-Dijkstra.
- Merge: It is the procedure responsible for grafting each feasible path discovered by Look-Ahead-Dijkstra onto the partial multicast tree.

An interesting algorithm for multicasting with delay and delay variation constraints has been proposed by Kapoor et al [34] which uses a dynamic programming technique. The characteristic of

this algorithm is that it does not necessarily produce a tree. To sum up, the proposed algorithms for solving multicast routing with M given constraints detected that the solution is not necessarily a tree, but no depth study or rigour formulation was proposed until 2012 when Molnar et al have published the concept of hierarchy as the first exact solution of MCMCM [56]. In this thesis we propose an Integer Linear Program (ILP) to compute a hierarchy which is the exact solution of MCMCM. According to our knowledge, the computation of the exact solution of MCMCM has not yet been published.

As pointed out all the main unicast and multicast routing problems are related. Figure 1.3 shows an outline of these routing problems that we have presented in the previous sections.

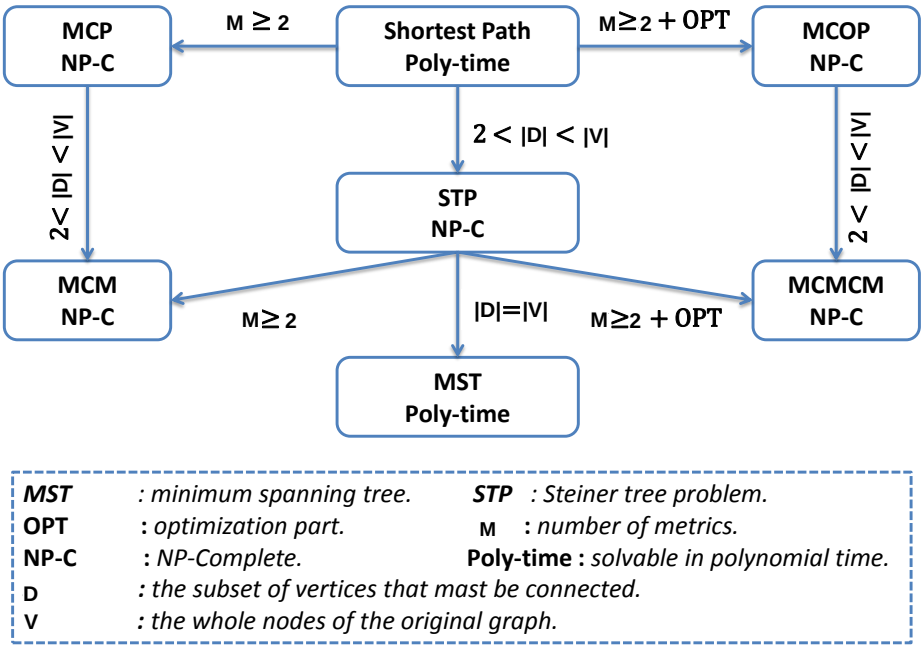

Figure 1.3: Outline of the discussed routing problems in networks.

1.6 QoS-based routing in wireless network

The wireless network is a network that uses electromagnetic waves to connect devices. Thus, mobile devices can move around within the network area freely and have to maintain a connection to the network. But radio wave have some negative properties comparing to Ethernet cable. For example, high loss rates, low data rates, and instability.

Recently, wireless networks are widely used for different purposes such as mobile phone

communication, emergency services, monitoring physical or environmental conditions, etc. As the wired network, wireless network is divided by its geographical area to different types:

- Wireless Personal Area Networks (WPANs), which interconnects devices within a small area.
- Wireless Local Area Network (WLAN), which links two or more devices through an access point.
- Wireless Ad Hoc Network or Mobile Ad Hoc Networks (MANETs), which is a continuously self-configuring, infrastructure-less network of mobile devices connected.
- Wireless Metropolitan Area Networks (WMANs) which connects several WLANs.
- Wireless Wide Area Networks (WWANs), which connect devices within a large geographic area.
- Wireless Global Area Network (WGAN) and Wireless Space Network (WSN), which can be also considered as other types of wireless networks.

The aim of routing in wireless network might differ depending on the application and network architecture. In this thesis, we also analyze a QoS-based routing problem in wireless network, which often balances between energy consumption and data quality. In what follows, we review some interesting works concerning QoS-based routing in Ad Hoc Network, Wireless Sensor Networks and Low Power and Lossy Networks (LLNs).

1.6.1 QoS-based routing in Ad Hoc Networks

The Ad Hoc network is a particular type of wireless network. The network is Ad Hoc because it does not have an infrastructure or access points as other types of wireless networks often have. This kind of networks are multi-hop networks (i.e. use two or more wireless hops to convey information from a source to a destination). Routing with QoS in Ad Hoc network is a very challenging task due to node mobility and lack of central coordination. The Ad Hoc routing protocols can be classified in three groups:

- Table-driven: also called proactive, the routing table of this group maintains all possible destinations and their routes. Therefore, each route is already known and can be used immediately. The main disadvantage of this group is the slow reaction on restriction and failures. Example of proactive protocols are DSDV (Destination Sequence Distance Vector) [59] and (OLSR) (Optimized Link State Routing) [30].
- On-demand: also called reactive, this type of protocols finds the route to a destination address on demand, so the routing table does not memorize all possible destination addresses

route. The main disadvantage of such a protocol is the high latency time in route finding. DSR (Dynamic Source Routing) [33] and AODV (Ad Hoc On-demand Distance Vector) [58] protocols are examples of reactive protocols.

- Hybrid routing (both proactive and reactive): this protocols combine the advantage of proactive and reactive routing. One of main disadvantages of these protocols is the reaction to traffic demand depends on gradient of traffic volume. ZRP (Zone Routing Protocol) [23] is a hybrid protocol.

The design of routing protocols with QoS in Ad Hoc networks is influenced by many challenging factor. One can give as example the unreliable wireless channel, node mobility, lack of centralized control, limited device resource, etc. There is a plethora of protocols which deal the QoS-based routing. Each protocol uses a metric as constraint on route discovery and selection to specify QoS requirements.

C.R Lin et al [50] proposed a bandwidth aware routing protocol for QoS support in multihop mobile network. The protocol contains end-to-end bandwidth calculation and bandwidth allocation. In [12] the authors proposed a distributed QoS routing scheme that selects a path with sufficient resource to satisfy a certain constraint of the delay in multihop mobile network. Other protocols were proposed in [76] and [3] to provide maximum tolerable delay jitter and maximum tolerable packet loss ratio respectively. An excellent survey about QoS-based routing in Ad Hoc network was presented by L. Hanzo and R. Tafazolli in [25].

1.6.2 QoS-based routing in Wireless Sensor Networks

Wireless Sensor Networks (WSNs), consist of few to hundreds or even thousand of sensors with wireless communications capabilities. Unlike ad-hoc networks, WSNs have additional requirement that were not specifically addressed. The development of WSNs was motivated by military applications. Today WSNs are used in various spheres of life. We can find sensors in our cars, in our electronic objects, etc. But recently with the appearance of real time applications, QoS is becoming a real challenge facing the research community.

The design of WSNs routing protocols is influenced by many factors such as node deployment, scalability, fault tolerant and the QoS is one of them. Therefore, achieving an efficient communication needs routing protocols to overcome these factors. Routing in WSNs can be divided into flat-based routing, hierarchical-based routing, and location-based routing depending on homogeneity or functionality of the nodes [4]. The main objective of these protocols is balancing between energy consumption and data quality. Especially, the routing map has to satisfy certain QoS metrics (i.e. delay, packet loss, jitter, etc.).

In [66] the authors proposed a protocol for WSNs named Sequential Assignment Routing (SAR) which introduces for the first time the notation of QoS into routing decision. SAR is table-driven multipath protocol, the construction of multipath from a source to each destination

is dependent on energy resource, QoS metrics and the priority level of each packet. Another QoS-based routing protocol for WSNs was proposed in [27] to provide a real-time communication called SPEED. It provides three types of real time communication services, real-time unicast, real-time area-multicast and real-time area-anycast. The design of SPEED is inspired by the observation that some constraints of QoS in wireless network are different in wired networks. SPEED claims that the delay can be estimated, so that it ensures a certain speed for each packet in the network, which allows to satisfy some QoS constraints. Compared to other protocols such as Dynamic Source Routing (DSR) [32] and Ad Hoc On-demand Distance Vector routing (AODV) [60], SPEED has a better performance in term of end-to-end delay.

1.6.3 Routing with multiple constraints of QoS in Low Power and Lossy networks

Low Power and Lossy Networks (LLNs) are composed of large populations of constrained devices (things and their interconnects). LLN routers are usually constrained by limited processing capacity, memory and energy [81]. The interconnects are characterized by their type of links, such as IEEE 802.15.4, Bluetooth, Low Power WiFi links, etc. Generally, LLN links suffer from high loss rates, low data rates, and instability. In recent years, there have been significant efforts to standardize a routing protocol for LLNs. This effort has culminated in standard IPv6 routing protocol for LLNs (RPL), by the ROLL Working Group in the Internet Engineering Task Force (IETF) [81]. Since LLNs consist of few dozen to thousands of routers, simple unicast flows are not enough to satisfy the specific routing requirements. In [73] the ROLL Working Group categorizes the traffic pattern into, upward traffic from nodes to a Border Router (i.e. Multipoint-to-point), downward traffic from Border Router to nodes (i.e. multipoint-to-point) and traffic from node to node (i.e. point-to-point traffic). The main interest of RPL is to improve routing in LLN minimizing the usage of network resources. Since LLNs have no predefined topology, routers run the protocol to discover links and then select peers sparingly in such a way that no cycles are present. To meet that, a Directed Acyclic Graph (DAG) is built that is partitioned into one or more Destination Oriented DAGs (DODAGs), one DODAG per root, using an Objective Function (OF) [2] to reach specific objectives. Since we formulate propositions for the QoS routing in LLNs, we present the outline of RPL.

1.6.3.1 Constructing Topologies

During the construction of the DODAG, Destination Information Object messages (DIO) and three types of nodes are participating. The first type is the root node, often corresponding to a Border Router, which is frequently the destination of communications. It starts the DODAG construction by sending DIO messages. Each node that receives the message from several potential parents runs an algorithm to choose an appropriate parent. The choice is based on a metric and constraints defined by the Objective Function OF. The second type are routers, that repeat the same steps

and the process terminates when a DIO message hits the third type of nodes (the leaves) or when no more nodes are left in range.

Figure 1.4 depicts an example of a DODAG construction, the dashed lines depict the possible links between devices. In this example the OF optimizes the number of hops during the construction of the DODAG. As shows in Figure 1.4.B each node chooses the parent that offers the minimum number of hops from the Border Router. When the density of the topology is higher, the probability of forming rooting loops is greater. This is a common problem in all kind of networks. RPL tries to avoid creating loops by using different avoidance mechanisms [72]. More details about the construction of the DODAG will be present in Chapter 4.

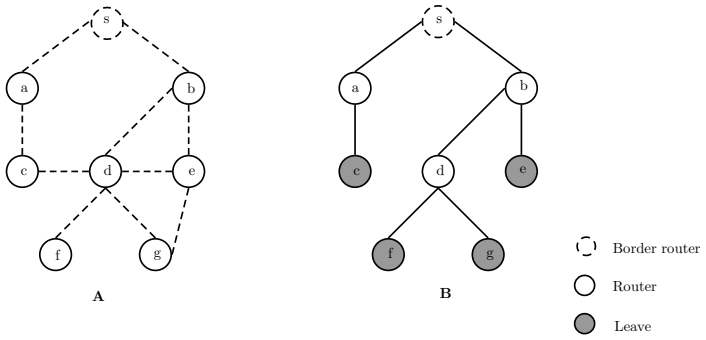

Figure 1.4: Example of RPL's constructing topology.

1.6.3.2 QoS-aware routing in RPL

Despite RPL's efforts to improve routing, the Quality of Service (QoS) has been somewhat overlooked. RPL is able to provide different levels of QoS based on QoS metrics and QoS constraints. In this thesis, we investigate to construct a routing scheme for QoS in RPL during the construction of the routes.

QoS metrics and QoS constraints may or not be of the same type. For example, the number of hops can be considered as a constraint if a node does not select parents advertising a ceiling number of hops, but it can also be considered as a metric if a node select a parent minimising the number of hops without constraint. A plenty of constraints/metrics were defined in [74]. Most of them are QoS measures.

RPL selects nodes and optimizes routes within an RPL instance using an Objective Function (OF) [70] that defines how the route will be constructed. The OFs are separated from the core protocol specification which gives RPL the flexibility for meeting different optimization criteria required by different applications. There are three predefined OFs in RPL [70] [20]. All of them try to construct a DODAG with respect to specific metrics. OF0 constructs a DODAG with the lowest number of hops. MRHOF-ETX constructs a DODAG with the lowest Expected Transmission

Count (ETX)[2] along the path. MRHOF-energy minimizes the energy consumption. However, some applications require several constraints of QoS to operate correctly.

Karkazis et al [35] proposed a method to quantify the routing metrics in order to combine them in an additive or lexical manner. It is the first work providing a combination of metrics that allows the construction of DODAG taking into account more than one metric. This is not the case for predefined OFs that consider only one constraint during the construction process.

The existing objective functions take into account only one metric [70] [20] or combine two metrics [35]. As a result, the DAGs cannot fully satisfy some recent applications, which require to respect several QoS constraints at once (errors, loss and delay, etc.). For instance, the hop count routing metric allows choosing the shortest path, but it does not necessarily ensure the end-to-end delay requirement, which is an important constraint for interactive applications.

An interesting OF was proposed in [18] to combine a set of metrics using Fuzzy Logic, and an Objective Function for RPL based on Fuzzy Logic (OF-FL). The OF-FL uses linguistic variables to describe the state of each metric separately. The linguistic variables are between true and false and each QoS metric is associated to some variables ("near", "vicinity" and "far" for hop count for example). Then the membership function collects the information to quantify the linguistic terms which is largely used in Fuzzy Logic system. The proposed OF-FL is very interesting when the goal is to optimize globally the network. In other words, when there are no defined constraints to respect. Unfortunately, it is not the case with some recent applications having strict requirements (e.g. the end-to-end delay must be less than 30 ms, etc.). Therefore, using the combination of metrics does not guarantee that each path respects the end-to-end constraints from the root to any node.

In order to construct a constrained DODAG (a DODAG that meets the requirements of applications relying on QoS), we discuss later in Chapter 4 a new objective function.

1.7 Thesis structure

In addition to this chapter which provides the context of our research, the thesis contains four chapters.

- Chapter 2 introduces our contribution in QoS-based unicast routing. We propose an exact algorithm, that manipulates the constraints to improve significantly the computation time. The algorithm named FMCRA is an improvement of SAMCRA algorithm. We also present numerical results to show the performance of FMCRA under different circumstances. This work was presented in 27th European Conference on Operation Research, EURO 2O15 [37].

[2] ETX: is the number of transmissions a node expects to make to a destination in order to successfully deliver a packet [74].

- Chapter 3 presents an Integer Linear Program (ILP) that finds the optimal hierarchies for MCMCM and gives the optimal solution in reasonable time, with instances of data that simulate real networks. As MCMCM is NP-hard an efficient pretreatment algorithm (ArcReduce) is proposed to rapidly converge to a solution when the problem is very hard. ArcReduce can even detect in polynomial time some instance for which the problem has a solution or not before starting the resolution process.
- Chapter 4 is about the third contribution of the thesis where we propose a new Objective Function based on a Non-Linear Length concept NL-OF for RPL the standard IPv6 routing protocol for Low-power and Lossy Networks LLNs. NL-OF is the first OF that can ensure a set of end-to-end constraints in LLNs. A significant part of this chapter aims at presenting the NL-OF. The second part presents the performance evaluation of NL-OF using Cooja simulator with different OFs in the literatures. Moreover, we introduce some important research track that can be pursued further. This chapter is built upon the work published in [38] and [39].
- Chapter 5 concludes the thesis and present a number of interesting future research problems.

CHAPTER 2

UNICAST ROUTING WITH MULTIPLE CONSTRAINTS OF QOS

Routing consists in finding a route from sending node to receiving node in the physical topology. For this reason, unicast route subject to multiple constraints of QoS should be clearly identified for some recent applications (e.g. multimedia applications). The main objective of this chapter is to improve the performance of the exact QoS unicast route based on the currently proposed algorithm called SAMCRA. In Section 2.1, we introduce the problem with brief review and its formal definition. Next in Section 2.2, we describe SAMCRA and its heuristic version TAMCRA presenting the main lines of each algorithm. In Section 2.3, we present our proposed algorithm and the limitation of using SAMCRA or TAMCRA. We also compare our algorithm with these algorithms. Finally, we present the numerical results of the evaluation of our algorithm and the conclusion of this chapter.

2.1 Introduction

The current Internet has been designed to support connectivity based routing of best effort traffic which assures only that data will reach their destination as soon as possible. The classical algorithms such as Dijkstra and Bellman-Ford algorithm [8, 13] can match perfectly the needs of routing with best effort traffic. However, the continuous growth in network traffic with Quality of Service requirement is calling for new QoS-oriented protocols. One of the most challenging task in the design of such protocols is the identification of the route satisfying multiple constraints of QoS (e.g., delay, jitter) with the best utilisation of resource.

In this chapter, we focus on searching for a feasible, possibly optimal path when each link in the network is associated with multiple parameters which can roughly be classified into additive

and non-additive. Usually bottleneck metrics can be easily dealt with a preprocessing step by pruning all links that do not satisfy the constraints associated with this metrics. Multiplicative metrics can be transformed into additive metrics. The formulation of the underlying unicast routing problem under additive QoS constraints is stated in Section 1.3.

Due to its importance, the MCP problem has been extensively studied. Since this problem is NP-hard, the solution proposed in the literature are either time-consuming, or do not guarantee to find a solution even if the problem is feasible. To the best of our knowledge, some of the most efficient algorithms are SAMCRA for the exact solution and TAMCRA for the heuristic. Based on the same ideas, in this chapter we propose an exact algorithm, that manipulates the constraints to improve significantly the computation time. We will see that for hard instances, the results are similar to SAMCRA, but for easier instances, reinforcing the constraints leads to reduce the search space and helps finding a solution.

2.2 TAMCRA and SAMCRA

2.2.1 TAMCRA

Tunable Accuracy Multiple Constraints Routing Algorithm (TAMCRA) is a heuristic based algorithm which can solve the MCP problem with possibility of missing a feasible solution. It is an adaptation of the well known Dijkstra's shortest path algorithm. To find a solution for the MCP problem, the proposed algorithm must store more than one path at each node. Therefore, a simple shortest path algorithms like the label fixation based algorithms do not work satisfactorily. In TAMCRA the number of stored paths is bounded by a constant k. The probability of missing a path is inversely correlated with k where the execution time increases with incrementing k. The three principles of TAMCRA and SAMCRA are presented in Section 1.4.2.

2.2.2 SAMCRA

SAMCRA is an exact version of TAMCRA. Unlike TAMCRA, SAMCRA does not limit the number of stored paths in the queue to k. For a node pair (s,d), SAMCRA returns the shortest path regarding the non-linear length. Hence, this path satisfies the constraint vector if the problem has a feasible solution. We illustrate the algorithms by following a brief outline of SAMCRA (cf. also in Section 1.4.2).

- SAMCRA selects at first the source node s.
- At each iteration, the algorithm explores the neighbors of a selected current node and chooses the closest node as the next current node using the non-linear length function defined in Equation 1.4. All non dominated paths to reach this node are memorized.
- SAMCRA ends when the destination d is reached and there is no more improvement for it.

SAMCRA has a worst case complexity of [54]
$O(k|V|\log(k|V|)+k^2M|E|)$

2.3 Fast Multiple Constraints Routing Algorithm (FMCRA)

Obviously the complexity of TAMCRA and SAMCRA depends on the number of memorized paths. In [47] the authors analyzed the complexity of SAMCRA, which can be expensive and expressed by a combinatorial complexity of: $O(k|V|\log(k|V|)+k^2M|E|)$, where k is the number of non-dominated paths at each node queue. *"The queue in SAMCRA can never contain more than $K|V|$ path lengths. Moreover, SAMCRA self-adaptively allocates the queue-size at each node, k may differ per node [47]."* As a result, proposing an algorithm for MCP that memorizes fewer number of paths is desired. Since MCP is NP-hard, the key question is always which is the most efficient paradigm to follow (i.e. exact or approximate solution) ? In what follows we present the advantage and the disadvantage of each paradigm and the reason leading to FMCRA proposal.

2.3.1 Limitations of TAMCRA and SAMCRA algorithms

TAMCRA is an efficient algorithm when the problem is easy, because it gives the solution faster than SAMCRA. Nevertheless, the probability of missing some feasible paths is high when the problem is hard. In contrast, SAMCRA guarantees to find a feasible path if it exists but the execution time increases exponentially.

As a trade-off, we propose an algorithm that guarantees finding a solution while generally keeping the execution time reasonable. Our proposed algorithm FMCRA represents an alternative to accelerate the search for a feasible path by handling the end-to-end constraints which influence the number of stored paths and eventually the required time to find the feasible path. Notice that the worst case execution time is exponential, because the problem is NP-hard. The algorithm consists of two main parts.

In the first part, FMCRA calculates the shortest path for all metrics separately and replaces the original end-to-end constraint by the lengths of these shortest paths. The objective is to make constraints more strict. At a first time, the constraints are the strictest possible.

In the second part, the algorithm seeks the solution using the modified constraints. With such strict constraints, the number of non-dominated paths is very small and the number of feasible paths as well. The algorithm find a feasible path rapidly if it exists. If no feasible path is found FMCRA will continue the search by using less strict constraints, without recomputing the already calculated sub paths. At the worst case, FMCRA finds the solution consuming the same time as SAMCRA.

2.3.2 Search space illustration

Figure 2.1 represents the original search space of an instance, where the black points are the non-dominated sub-paths. TAMCRA and SAMCRA consider all the sub-paths located in this area from the beginning. In that case, all (or at most k) non-dominated sub-paths in the intermediate nodes are memorized, even though some of these sub-paths do not lead to a feasible path to the destination. On the contrary, FMCRA begins by a small search area as shown in Figure 2.2.A, to save the most important sub-paths in each intermediate node. FMCRA stops with a smaller number of computed paths if it finds the solution in this area. Else, the search area become larger (cf. Figure 2.2.B) and so on until reaching the feasible path.

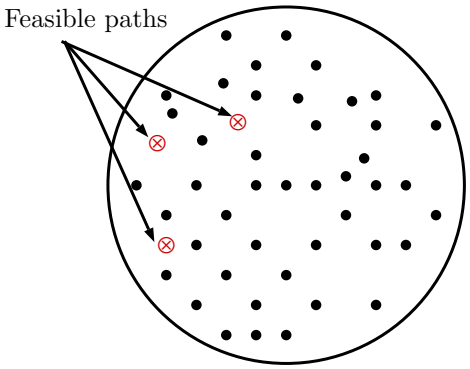

Figure 2.1: The original search area of a given MCP problem.

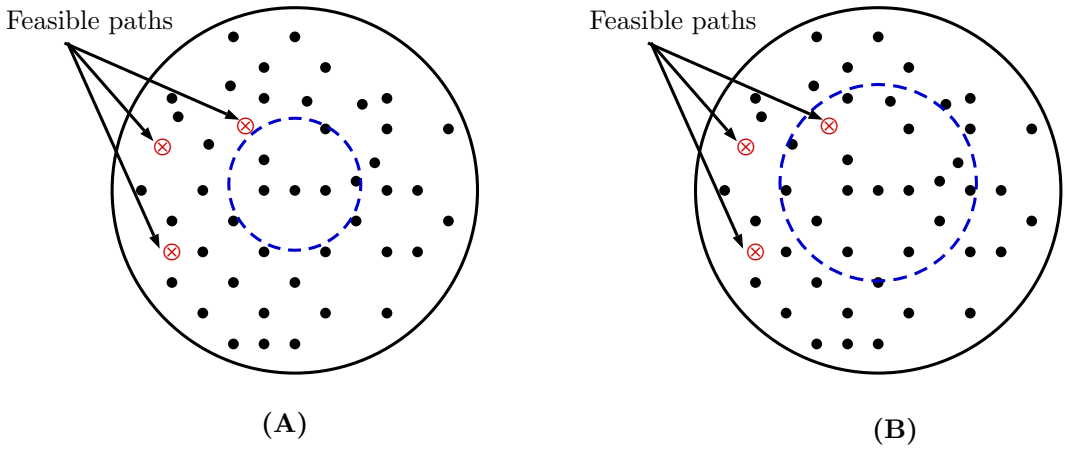

Figure 2.2: The search method of FMCRA.

The meta-code of FMCRA is listed below.

$\overrightarrow{L_n}$ is the modified constraint. In line 2, each constraint is changed by the strictest value. Then the algorithm starts the search with this modified constraints. If the destination is not

Algorithm 1 The meta-code of FMCRA.

Require: G: graph, s: source node, d: destination node, $\overrightarrow{L}$: $constraints$
Ensure: Path $p(s,d)$, if it is possible

for i=1....M **do**
  L'_i := Dijkstra(s,d,i); / / *Where Dijkstra(s,d,i) returns the length of the shortest path between s and d regarding the metrics i.*
end for
while ($\overrightarrow{L'} \leq \overrightarrow{L}$ and ! FIND-PATH (G,s,d,$\overrightarrow{L'}$)) **do**
  for i=1...M **do**
    $L'_i := L'_i + AVG$;
  end for
  if $\overrightarrow{L'} > \overrightarrow{L}$ **then**
    $\overrightarrow{L'} := \overrightarrow{L}$;
  end if
  FIND-PATH (G,s,d,$\overrightarrow{L'}$);
end while

Algorithm 2 FIND-PATH.

Require: FIND-PATH (G,s,d,$\overrightarrow{L'}$)
Ensure: Feasible path

path(s) := null and length(s) := 0;
put s in queue;
FIND-PATH := false ;
while (queue≠ empty & ! FIND-PATH) **do**
  u := EXTRACTED-MIN(queue); / /*The node with minimum length in queue*
  if (u == d) **then**
    FIND-PATH := true;
  end if
  for all neighbors v of u **do**
    if v ≠ previous node of u **then**
      PATH := path(u) + (u,v); / /*path(u): is the path from s to u*
      LENGTH := l(PATH); / /*l: is the non linear length*
      if LENGTH < 1 **and** PATH is not dominated **then**
        path(v) := PATH;
        length(v) := LENGTH;
        put v in queue;
        if v == d **then**
          FIND-PATH := true;
        end if
      end if
    end if
  end for
end while

reached, the algorithm broadens the constraints in line 9 by adding AVG, which is the average of the MAX and the MIN value of the metrics in the graph: $MAX = \max_{i\in 1,\ldots,m}(w_i(e)), \forall e \in E$ and $MIN = \min_{i\in 1,\ldots,m}(w_i(e)), \forall e \in E$.

$$AVG = \frac{MAX + MIN}{2}. \tag{2.1}$$

2.3.3 An example of the operation of TAMCRA, FMCRA and SAMCRA

The following example illustrates the operation mechanism of each algorithm for the same topology where the objective is to find a path from the source node s to the destination node d with respect to the constrained vector $\overrightarrow{L} = [11,11]^T$.

Figure 2.3 shows an example of operation of TAMCRA with $k = 2$. As we can see TAMCRA misses the feasible paths because it can store at most two sub-paths in each node while the third sub-path $p_d\{s,a,b,c,g,\}$ is the only sub-path that leads to the destination respecting the constraints.

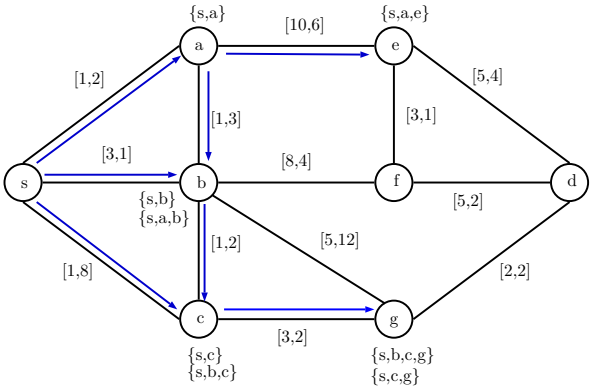

Figure 2.3: Example of the operation of TAMCRA.

In Figure 2.4, SAMCRA returns the path that satisfies the constraint vector $\overrightarrow{L}$. As SAMCRA uses only the non-domination of paths to reduce the search space, all non-dominated sub-paths subject to the constraint vector $\overrightarrow{L} = [11,11]^T$ are memorized.

As shown in Figure 2.5 FMCRA returns the path satisfying the constrained vector $\overrightarrow{L}$ in two trials. First, FMCRA changes the original constraints by $\overrightarrow{L}$=[6,7]. This step leads to reduce the search space because some possible paths to the intermediate nodes becomes infeasible with the new constraints. Thus, if the feasible path to the destination exists, in this area FMCRA finds the solution faster. The algorithm will continue searching without recalculating the already calculated paths using more loose constraints if the feasible path does not exist. As shows, FMCRA does not find a feasible solution with $\overrightarrow{L}$=[6,7], so it uses more loose constraints $\overrightarrow{L'} = [9,11]$. The feasible path is $p_d = \{s,a,b,c,g,d\}$ where $w_1(p_d) = 8$ and $w_2(p_d) = 11$.

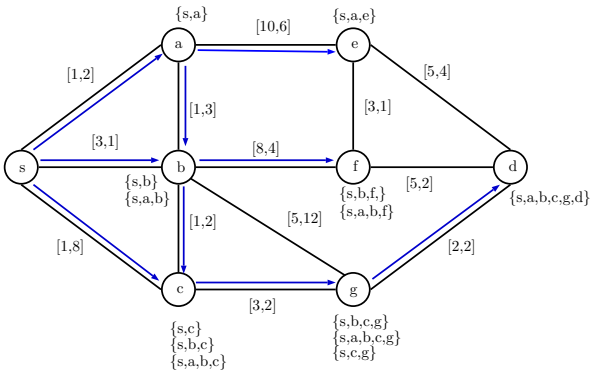

Figure 2.4: Example of the operation of SAMCRA.

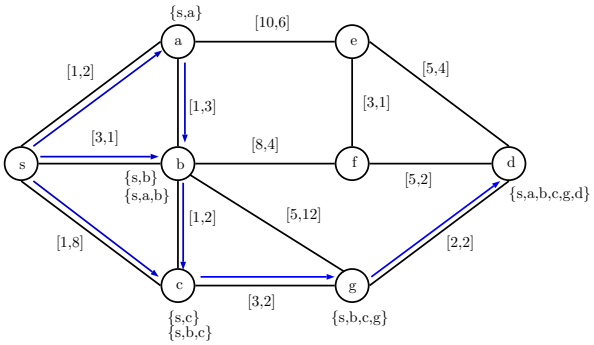

Figure 2.5: Example of the operation of FMCRA.

In this example FMCRA memorizes only six non-dominated sub-paths before finding the solution, where SAMCRA memorizes twelve.
Notice that when the topology becomes more complex the number of non-dominated paths will be more influential on execution time.

2.4 Simulation results

To compare the performance of the algorithms in typical cases, we present a study based on simulations. We randomly generate 1000 graphs of 100 nodes using the random graph generator of the class WAXMAN [79]. *"The Waxman's generator is a geographic model for the growth of the network. In this model nodes are uniformly distributed in a given area, and links are added according to probabilities that depend on the distances between the nodes. The probability to have a link between nodes i and j is given by: P(i,j) =* $\alpha * e^{-d/\beta * d_{\max}}$ *. Where* $0 < \alpha, \beta \leq 1$*, d is the distance from i to j, and* $d_{\max}$ *is the maximum distance between any node pair".*
Each edge of the graphs is weighted by integer metrics, representing the QoS metrics. These values of edge are randomly chosen in {1,..,10}.

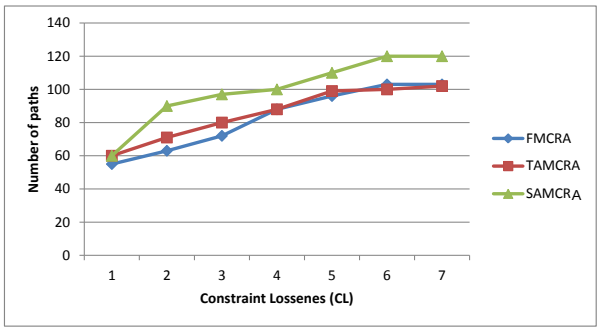

Figure 2.6: Number of computed sub-paths by FMCRA, TAMCRA and SAMCRA.

2.4.1 Performance comparison with SAMCRA and TAMCRA

The three algorithms SAMCRA, TAMCRA and FMCRA are executed independently to find a solution. Three performance measures are computed.

- *Number of computed paths*: the number of stored paths before finding the feasible one.
- *Execution time*: the required time to find a feasible path if it exists.
- *Quality of paths*: the average of the non-linear length from the source to the destination.

These measures are used to evaluate the performance of each algorithm.

2.4.1.1 Parameters

The algorithms were tested under different settings issued by modifying the following parameters:

- *Number of Constraints* (NC): number of metrics associated to each edge.
- *Constraints Looseness* (CL): the CL is the average value of $\frac{L_i}{\max(value_i)}$, $i \in \{1,\ldots,M\}$, where $\max(value_i)$ is the maximum possible value of metric of index i. Since the values of the QoS metrics for each link are in $\{1,\ldots,10\}$, $\max(value_i) = 10$ for all metrics. In order to generate instances with a specific CL, all L_i are equal to $10 \times CL$.

Notice that the number k of stored sub-paths defined in TAMCRA is variable in order to keep the erroneous decision rate (EDR) [1] below 15%.

2.4.1.2 Number of computed sub-paths

In Figure 2.6 we notice that SAMCRA computes more paths than FMCRA before finding the solution, with different degree of CL. It is the same case with TAMCRA, except when the

[1] EDR is the percentage of instance for which TAMCRA does not find a feasible solution despite its existence on the total number of routing request.

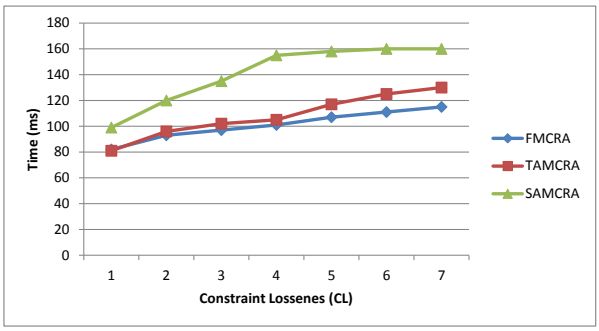

Figure 2.7: Execution time of FMCRA, TAMCRA and SAMCRA.

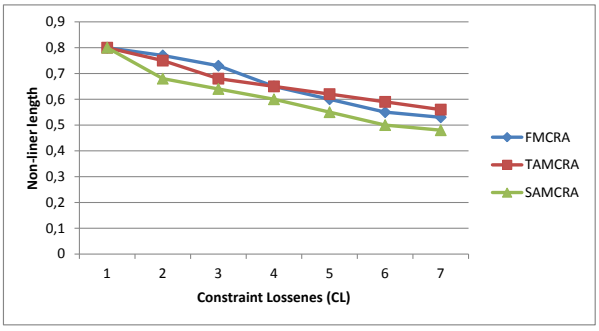

Figure 2.8: Quality of paths provided by FMCRA,TAMCRA and SAMCRA.

constraints are more loose, because the number of feasible sub-paths increases and the non-dominated sub-paths as well. As we previously explained, the number of computed paths is the most influential factor in the complexity of the solution.

2.4.1.3 Execution time

Figure 2.7 represents the average execution time to find a feasible path. We notice that the execution time of FMCRA is lower than the execution of TAMCRA for strict constraints, and gradually increases when the constraints become less strict. For easy instances FMCRA and TAMCRA are more efficient than SAMCRA.

2.4.1.4 Quality of computed paths

In Figure 2.8, we see that FMCRA and SAMCRA find a solution with a slight difference of non-linear length, both found better results than TAMCRA except when the constraints are very strict. This is because the number of feasible paths is very small, so the feasible path found by TAMCRA has a high probability of being the optimal one.

2.4.1.5 Performance of FMCRA

In this part we investigate the performance of FMCRA in term of number of computed paths. We varied the Number of Constraints NC between 2, 4 and 6 in random graphs with 50 and 100 nodes. The following figures show the obtained results.

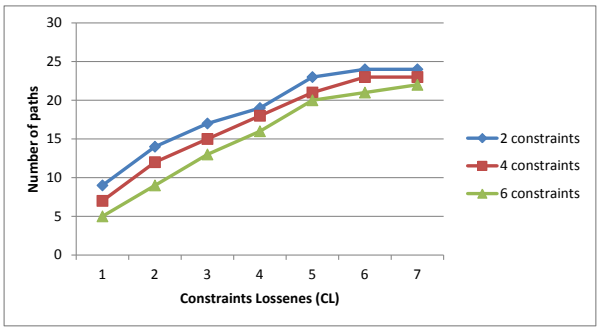

Figure 2.9: Number of computed sub-paths in graphs of 50 nodes by FMCRA.

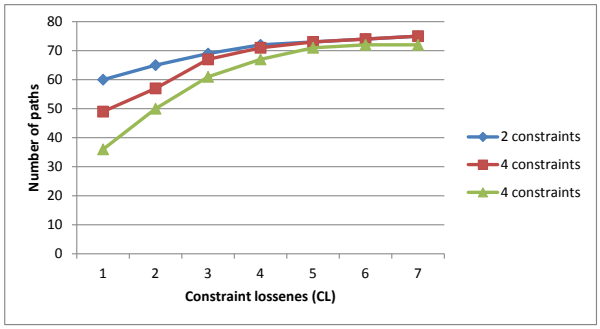

Figure 2.10: Number of computed sub-paths in graphs of 100 nodes by FMCRA.

We can sum up the results shown in the graphics above in two main points:

- The number of computed sub-paths is not much influenced by the modification of the constraints.
- With loose constraints, FMCRA calculates an almost constant number of paths even though there is a large number of feasible sub-paths. This is because FMCRA uses more strict constraints which make some feasible sub-paths in the original constraints infeasible.

2.5 Conclusion

In this chapter we have analysed the unicast routing with multiple constraints of QoS. We have also presented the formal definition of the problem. Hence, we have proposed FMCRA which

is an exact algorithm for obtaining a feasible path with a new technique to reduce the size of the search space. Its main property is to adapt itself to the difficulty of the instance. Through this technique, the algorithm will use only a reduced search area. As a result, the number of computed paths is decreased significantly as well as time. The main results:

- Execution time: the results show that the execution time of FMCRA can compete TAMCRA which provides a heuristic based solution. SAMCRA takes more time compared to FMCRA and TAMCRA.
- Number of computed paths: FMCRA finds the solution computing the lowest number of paths comparing to other algorithms.
- Quality of paths: FMCRA provides a good quality of solution in term of the non-linear length.

CHAPTER 3

MULTICAST ROUTING WITH MULTIPLE CONSTRAINTS OF QOS

In this chapter we provide our contribution to multicast QoS routing, in particular to the MCMCM problem. After a short introduction, in section 3.2 we recall the formal definition of the MCMCM, the hierarchy concept and their main properties. In section 3.3 we present an Integer Linear Programming (ILP) formulation for the MCMCM problem. Section 3.5 presents an efficient pretreatment algorithm (ArcReduce) to accelerate the resolution time of the ILP. Section 3.6 shows numerical results to evaluate the efficiency of our ILP and the ArcReduce. The tests were conducted with various sizes of input data regarding the execution time, the success rate and the quality of multicast route generated. Finally, section 3.8 concludes this chapter.

3.1 Introduction

As we have explained in the previous chapter, the simplest unicast QoS routing consists in finding a feasible path between a source and one destination. However in many real-time applications like videoconferencing, Video on Demand, Game on Demand, and Internet Protocol Television the objective is to route from a single source node to a set D of destination nodes, subject to multiple constraints. This routing, involving an important part of next-generation multimedia Internet applications is based on multicasting, is called multicast QoS routing. These applications involve accessing large volume of multimedia information which lead up to seeking new methods providing a certain level of QoS during the data forwarding in network. Several researches have treated the QoS-aware multicast routing problem in different manners, such as individual node resource management [84], or QoS routing without resource consideration [75]. An interesting method was proposed by Ahmed et al in [6] to manage the problem of QoS routing with a general

framework with resource allocation. The crux of the method is the formulation of QoS routing, which incorporates the hardware/software implementation and its relation to the allocated resources into a single framework. Our biggest concerns in this chapter is that of finding a multicast QoS route that satisfies the end-to-end constraints without resource allocation.

3.2 Formal definition and hierarchies

3.2.1 Model and notation

In short, our problem aims to find the route $H^* = (V^*, E^*)$ rooted at the source vertex s and spanning a non-empty destination set $D = \{d_1, d_2, ..., d_d\}$, respecting the end-to-end requirement from the source to each destination (i.e. containing a feasible path from the source to each destination). The cost of the route $c(H^*)$ is the sum of the cost of all edges in H^*. As we will see later in this Chapter, the cost of an edge $e \in H^*$ can multiply several times to this sum if it is crossed by several paths. This is for example the case of edge (e, f) in Figure 3.1. For more details about the notations and the formal definition of MCMCM see Section 1.3.

Definition 3.1. (MCMCM problem) For s, D given, find $H^* = (V^*, E^*)$ containing for each destination $d_j \in D$ a path $p(s, d_j)$ between s and d_j, such that

$$l_i(p(s, d_j)) \leq L_i, i = 1, \ldots, M, j = 1, \ldots, d \tag{3.1}$$

and the cost $c(H^*)$ is minimum.

Remark: In $H^* = (V^*, E^*)$, V^* is a set of vertex occurrences and E^* the set of edge occurrences. The route may refer to vertices/edges several times (it is not a sub-graph but a spanning hierarchy cf. in the next subsection). Consequently, one vertex or one edge may be present several times in V^* and E^* respectively.

Property 3.1. *MCMCM is NP-hard for $M \geq 2$.*

Proof. Obvious: The special case $M \geq 2$ and $|D| = 1$ is the multi-constrained optimal path problem (MCOP), which is proved to be NP-hard [78]. ■

3.2.2 Hierarchies in graphs

In order to define exactly the optimal solution, we recall in this section the concept of the hierarchies, with an illustrative example.

Definition 3.2. Let $T = (W, F)$ be a tree (i.e. acyclic connected graph) and $G = (V, E)$ a connected graph. A homomorphism $h : W \longrightarrow V$ maps each vertex in W to a vertex in V such that the mapping preserves the adjacency, i.e., $(u, v) \in F \Longrightarrow (h(u), h(v)) \in E$. The triple (T, h, G) defines a

hierarchy H in G. This hierarchy can also be represented as $H = (V', E')$, where V' and E' are multi-sets.

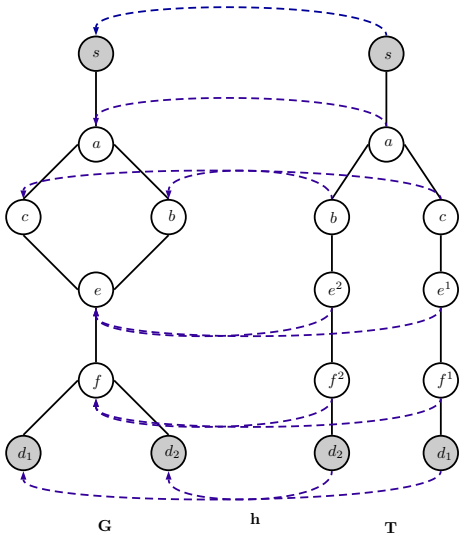

Figure 3.1: Mapping of vertices for a hierarchy.

Figure 3.1 gives an example applying a mapping h from a tree T to a target graph G for a hierarchy H. Each vertex of T is associated with a unique vertex of G. In the reverse direction, some vertices of G are mapped more than once in T, (e.g., vertices e and f). As we can see, the set of vertices and edges induced by T in G is not a subgraph of G. H remains connected and can be used as a spanning structure for multicast routing. Especially, when the mapping h is injective, the hierarchy corresponds to a tree. Thus, a tree is a special case of hierarchy. To distinguish the occurrences in hierarchy H associated with the same vertex v in G, we label them $v^1, v^2, \ldots, v^k$ in T. The concept of hierarchy can naturally be extended for oriented graphs.

By relaxing the tree structure restriction, spanning hierarchies provides more solutions in comparison with spanning trees, thereby resulting in more flexible solution for the routing.

3.2.3 Properties of the solution of the MCMCM

Property 3.2. *The solution H^* of the MCMCM is not necessarily a tree.*

Proof. As we show in Figure 3.2.A, in the optimal spanning solution satisfying multiple QoS constraints a vertex or an edge can be crossed several times.
In this example, there are two possible paths to reach each destination in $\{d_1, d_2\}$. $(s-a-b-e-f-d_1)$ and $(s-a-c-e-f-d_1)$ to reach d_1 with a total weight of $[9,9]^T$ and $[10,8]^T$ respectively, $(s-a-b-e-f-d_2)$ and $(s-a-c-e-f-d_2)$ to reach d_2 with a total weight of $[10,8]^T$ and $[9,9]^T$ respectively. If the end-to-end QoS constraints are given by $\vec{L} = [9,9]^T$, the only feasible paths for d_1 and d_2 are $(s-a-c-e-f-d_1)$ and $(s-a-b-e-f-d_2)$ respectively. The cost optimal

solution is not the set of the two paths. Messages from s may be sent only once, the edge (s,a) can be shared for the two destinations and the messages should be duplicated in vertex a. After the duplication, the messages follow the remaining route to the destinations. It is clear that the route generated by these two paths is not a tree in G. ■

Property 3.3. *Even when there is a feasible tree, a hierarchy satisfying the constraints may be cheaper.*

Proof. As shown in Figure 3.2.B, we can reach each destination in $\{d_1,d_2\}$ using the previous hierarchy or also by a tree $((s,(a,(b,(e,(f,d_2))),(c,d_1))))$. However, in terms of cost, the hierarchy cost is 9$ (the cost of (e,f) in the hierarchy solution is counted twice since the two paths do not have a common prefix to reach vertex e), while the tree cost is 12$. ■

Theorem 3.1. *The optimal multicast route H^* with respect to multiple constraints on positive additive metrics is always a directed partial spanning hierarchy, if it exists [56].*

Property 3.4. *In the optimal hierarchy H^*, two paths or more can share the same arc if they have a common prefix*[1]*. On the contrary, an arc must be used several times if it is crossed by two or more paths and no common prefix exist between these paths.*

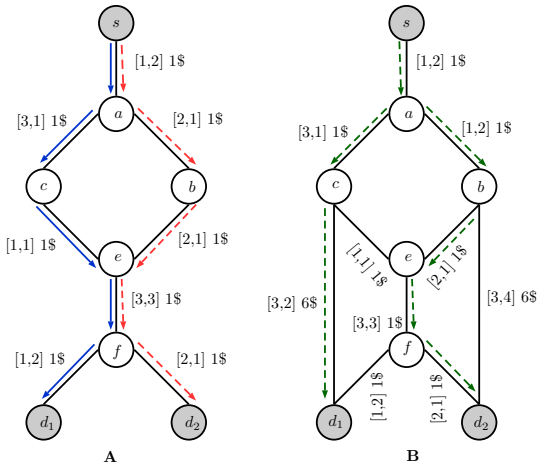

Figure 3.2: Example of optimal hierarchies for $\vec{L} = [9,9]^T$.

3.3 ILP Formulation

In this section we present an Integer Linear Program formulation of the hierarchies solving the MCMCM problem. After defining briefly the Linear and Integer Programming (LP and ILP

[1]The common prefix between two paths P and P' corresponds to the set of links from the source node to the node preceding on of the first node not belonging to both paths.

respectively), we show the limitations of existing methods in the literature explaining the main reason behind using the flow based method in a particular undirected multigraph to formulate the problem.

3.3.1 Linear and Integer Programming

Linear Programming (LP) is a method of optimization which is a special case of mathematical programming. In this mathematical model, the constraints are represented by linear relationships (i.e. linear equality and linear inequality) in order to optimise a linear objective function. To solve Linear Programming several methods have been developed over the years. The number of applications of LP is very large including industrial applications, engineering applications and so on.

More formally, the general linear programming problem can be stated in the following standard form:

$$\text{Maximize} f(x_1, x_2, \ldots, x_n) = c_1x_1 + c_2x_2 + \cdots + c_nx_n \tag{3.2}$$

Subject to the constraints

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &= b_m \end{aligned} \tag{3.3}$$

$$\begin{aligned} x_1 &\geq 0 \\ x_2 &\geq 0 \\ &\vdots \\ x_n &\geq 0 \end{aligned} \tag{3.4}$$

Where x_j are the decision variables, and c_j, b_j, and a_{ij} are known constants.

The (Mixed) Integer Linear Programming (M)ILP is the name given to LP problems which have the additional constraint that (some of) the variables have to be integer.

3.4 Limitation of standard ILP formulations

In the classical formulation of the MCMCM problem [44] the goal is to find a subgraph G' of G having a feasible path from the source s to each destination $d_j \in D$, and the length of the sub-graph is minimal. This definition can be used to formulate the MCMCM problem by an integer program which introduces some flow variables and special constraints to ensure the feasibility of each path [62]. Using the same definition, the classical cut formulation of spanning trees can also be used. However, in [56], Molnar et al. proved that the optimal solution of MCMCM are hierarchies which are not necessarily sub-graphs. It is true that the solution generates a subgraph but inversely, the subgraph (the image of the solution) is not sufficient to exactly design the solution. Since a hierarchy can return to vertices and cross arcs several times, simple flow and cut-based formulations in the topology graph are not any more possible (usual flow conservation conditions and cut formulations cannot be expressed). Thus, the classical formulations are no longer adequate. We choose a special flow-based method in a particular undirected multigraph to solve the problem for the following reasons:

- An edge can be used at most $|D| = d$ times in a given directions, if all paths of the multicast route use it and no common prefix exists between them. An example is the edge (e,f) in Figure 3.3 which can easily be generalized. To solve the problem, we duplicate each initial edge d times in both directions to build a directed multigraph (digraph). This separation of the edge/arc occurrences permits the coding of the flows traversing the original edges several times and in both directions.
- Vertices are not duplicated because the flow conditions can be expressed even if several flows traverse some vertices.
- Two or more flows can share the same arc e in the digraph if, and only if, the prefix of these paths to the arc e is the same. Let p be a path from a vertex s to an vertex t and $e = (x, y)$ be an edge of p we note $pref(p, e)$ the sub-path p from s to x. In Figure 3.3, p_1 (s, a, c, a, f, d_1) and p_2 (s, a, b, e, f, d_2) share the arc (s, a) because $pref(p_1, (s, a)) = pref(p_2, (s, a))$, but they do not share the arc (e, f) because $pref(p_1, (e, f)) \neq pref(p_2, (e, f))$. A simple flow method cannot ensure this property.

Existing ILP for the hierarchy: in [7] the author proposes an ILP to model the hierarchy. This ILP is based on a simple flow method, it works satisfactorily if no edge is traversed several times. However, as shown the hierarchy may traverse an edge several times in both directions.

3.4.1 ILP formulation of the exact solution of MCMCM

We define a multigraph $G = (V, 2dE)$ obtained by the duplication of each edge of the topology network $d = |D|$ times in both directions. We explain below the notation for the network parameters in Table 3.1 and the variables used to realize this ILP in Table 3.2.

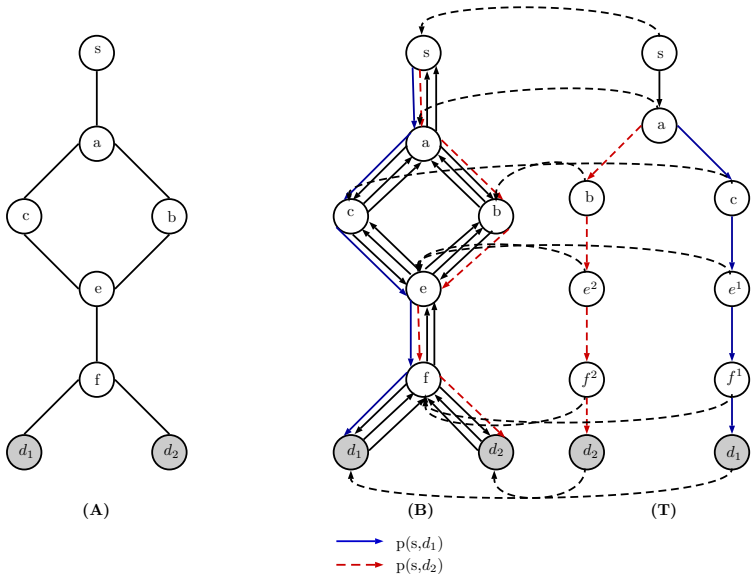

Figure 3.3: (A) initial graph; (B) multi-graph; (T) routing solution.

Table 3.1: Network parameters

Parameter	Description
$G = (V, 2dE)$	The digraph obtained by the duplication of edges from the network topology (each edges of the original network is present $d = \|D\|$ times in both directions).
s	Source.
$D = \{d_1, d_2, ..., d_d\}$	Set of destinations.
$d = \|D\|$	Number of destinations.
$In(m)$	Set of vertices leading an arc to vertex m.
$Out(m)$	Set of vertices forwarded by arcs from m.
$\vec{w}(m,n) =$ $[w_1(m,n), ..., w_M(m,n)]^T$	Weight vector of arc(m,n).
$c(m,n)$	Cost of arc(m,n).
$\vec{L} = [L_1, L_2, ..., L_M]^T$	Constraint vector for each destination.

Objective

The objective function of minimizing the solution's total cost can be expressed as follows:

$$\text{Minimize} \sum_{m \in V} \sum_{n \in Out(m)} \sum_{i \in \{1,...,d\}} U_i(m,n) \cdot c(m,n). \tag{3.5}$$

Constraints

-Paths constraints:

Table 3.2: ILP Variables for the ILP formulation

Variable	Description
$U_i(m,n)$	Binary variable. Equals to 1 if multicast route uses arc of indice i of arc (m,n), equals to 0 otherwise ($2d \cdot \|E\|$ variables).
$B_i(k,m,n)$	Binary variable. Equals to 1 if path $p(s,d_k)$ in the solution uses arc of indice i of arc (m,n), equals to 0 otherwise ($2d^2 \cdot \|E\|$ variables).
$R_i(k,l,m,n)$	Equals to 1 if paths $p(s,d_k)$ and $p(s,d_l)$ in the solution share the same arc of indice i of arc (m,n), equals to 0 otherwise ($2d^3 \cdot \|E\|$ variables).

For the source

$$\sum_{n \in Out(s)} \sum_{i \in \{1,...,d\}} \sum_{k \in \{1,...,d\}} B_i(k,s,n) = d. \tag{3.6}$$

For the destinations

$$\sum_{m \in In(d_k)} \sum_{i \in \{1,...,d\}} B_i(k,m,d_k) - \sum_{n \in Out(d_k)} \sum_{i \in \{1,...,d\}} B_i(k,d_k,n) = 1, \forall d_k \in D. \tag{3.7}$$

For the other vertices

$$\sum_{q \in In(m)} \sum_{i \in \{1,...,d\}} B_i(k,q,m) = \sum_{n \in Out(m)} \sum_{i \in \{1,...,d\}} B_i(k,m,n), \forall m \in V \setminus \{D \cup \{s\}\}, \forall k \in \{1,...,d\}. \tag{3.8}$$

The constraints 3.6, 3.7 and 3.8 guarantee (in this order) that the number of outgoing paths from the source equals the number of destinations, each destination vertex has exactly one path from the source. The incoming flow in the intermediate vertices equals the outgoing (conservation of flow).

-QoS constraints:

$$\sum_{m \in V} \sum_{n \in Out(m)} \sum_{i \in \{1,...,d\}} B_i(k,m,n) \cdot w_h(m,n) \leq L_h, \forall k \in \{1,...,d\}, \forall h \in \{1,...,M\}. \tag{3.9}$$

The constraint 3.9 guarantees that each path respect the given QoS constraints.

-Linking paths and usage of arcs constraints:

$$U_i(m,n) \leq \sum_{k \in \{1,...,d\}} B_i(k,m,n), \forall (m,n) \in A, \forall i \in \{1,...,d\}\ . \tag{3.10}$$

$$U_i(m,n) \geq B_i(k,m,n), \forall (m,n) \in A, \forall i,k \in \{1,...,d\} \times \{1,...,d\} \; . \tag{3.11}$$

Setting the values of $R_i(k,l,m,n)$ constraints:

$$R_i(k,l,m,n)\, R_i(k,l,m,n) = B_i(k,m,n) \cdot B_i(l,m,n)$$

$$\begin{aligned} R_i(k,l,m,n) &\leq B_i(k,m,n) \\ R_i(k,l,m,n) &\leq B_i(l,m,n) \\ R_i(k,l,m,n) &\geq B_i(k,m,n) + B_i(l,m,n) - 1, \\ \forall \{k,l\} \in \{1,...,d\}^2, &\forall (m,n) \in A, \forall i \in \{1,...,d\}. \end{aligned} \tag{3.12}$$

Constraints 3.11 and 3.12 ensure that each arc in the multicast route is traversed by one path at least or by all paths at most, i.e. the multicast route contains only the set of arcs used in the solution.

$$\begin{aligned} \sum_{j \in \{1,...,d\}} \sum_{q \in In(m)} R_j(k,l,q,n) \geq \sum_{j \in \{1,...,d\}} \sum_{n \in Out(m)} R_j(k,l,m,n), \\ \forall \{k,l\} \in \{1,...,d\}^2, \forall m \in V, if (m \neq s). \end{aligned} \tag{3.13}$$

The constraints 3.13 ensure that an arc (m,n) is shared by two paths p_1 and p_2, if and only if it is already the case in a previous arcs (q,m).

Property 3.5. *The solution of the ILP in the multigraph corresponds to an optimal hierarchy in the original graph.*

Proof. Let $G = (V,E)$ be the original graph of an instance of MCMCM with a source s and a destination set $D = \{d_1,\ldots,d_d\}$. Let $Gd = (V,A)$ the multigraph obtained by the duplication of each edge of the original graph in both directions. The graph G and Gd are defined on the same vertex set. Each edge in E corresponds to $2d$ arcs. Let us suppose that $H1 = (T,h1,Gd)$ is an optimal solution in Gd computed by the ILP. Using the correspondence existing between the arcs of Gd and edges of G, we obtain a hierarchy $H2 = (T,h2,G)$ in G. Each path $p(s,d_i)$ in G satisfies the QoS constraints (by construction of $H1$). We now prove that H_2 have an optimal cost.
Proof by contradiction: Suppose that $H2$ is not optimal and there is a hierarchy $H3$ s.t. $c(H3) < c(H2)$. Let $H4$ be the hierarchy projected in Gd following $H3$. the hierarchy $H4$ satisfies the QoS constraints. $c(H4) < c(H1)$ which is in contradiction with the fact that $H1$ have minimum cost. ■

Example: Figure 3.4.A presents the original graph G when the objective is finding the optimal solution of MCMCM from the source s to the destination set $D = \{d_1, d_2\}$. Figure 3.4.B presents the multigraph $Gd = (V, A)$ obtained by the duplication two times of each edge in the original graph (i.e. $d = 2$) in both directions. G and Gd are defined on the same vertex set. Each edge in E corresponds to $2d.|E|$ arcs in A in which $d.|E|$ arcs in each direction.

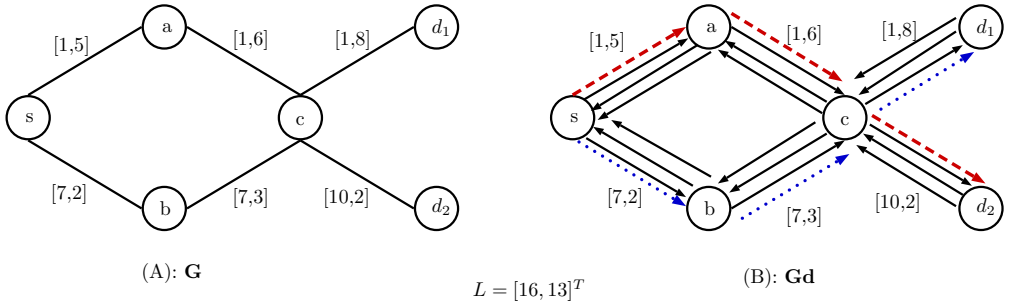

Figure 3.4: Projection of the optimal solution in the original graph.

3.4.2 Experiments

The objective of our experimentation is to analyse the optimal route based on the hierarchies in the solution and to compare the solution of MAMCRA to the optimum. Remember that MAMCRA is a well known and efficient heuristic.

Datasets. We evaluate the practical potential of our ILP on random graphs generated by the WAXMAN model [79], already presented in Section 2.4, and also in NSF topology: Network of the National Science Foundation (United States), and in NTT topology [5]: Network of the Nippon Telephone Telegraph (Japan).

Each arc of the graphs is weighted by integer metrics, representing the QoS metrics and cost. These values of arc are randomly chosen in {1,..,10}.
We also analyse the efficiency of the MAMCRA algorithm using the same topologies. Due to the large resolution time of the ILP, *each experimentation in this section was conducted on 50 instances*. The type of networks, the number of vertices and arcs are presented in Table 3.3.

Since each link in real networks is bidirectional, we organise the run of experiments in the following manner. Each link is replaced by two opposite arcs. The values of the QoS metrics and the cost may be different in each direction to simulate as possible the real networks.

Performances. We study the behaviour of the ILP and MAMCRA algorithm regarding two main parameters.
- *Destination vertices Density (DD):* Percentage of multicast members regarding the total number of vertices $|V|$.
- *Constraints Looseness (CL):* The CL is the average value of $\frac{L_i}{max-value_i}, i \in \{1, \ldots, M\}$, see Section 2.4.1.1 for the definition of the Constraints Looseness.
We fixed the number of destinations to 5 and the CL to 6 when the DD (or the number of destinations)

Table 3.3: Dataset information of the ILP tests

Networks	Information			
	$\|V\|$	$\|A\|$	α	β
NSF topology	14	21	-	-
NTT topology	55	144	-	-
WAXMAN 50	50	300	1	0.11
WAXMAN 100	100	600	1	0.073
WAXMAN 200	200	1200	1	0.048

and CL are not mentioned. The number of constraints is set to 2 if it is not mentioned.

Environment. The experiments are made on a 2 cores PC (Intel i3-3110M) having 4GB of memory and running under Windows 7 (64bit). The ILP is solved with IBM ILOG CPLEX 12.6.2[2].

Table 3.4: CPU time to solve the ILP

Datasets	CPU time				
Number of destinations	2	4	6	8	10
NSF topology	0.07s	0.31s	1.05s	4.14s	7.45s
NTT network	0.09s	3.10s	5.02s	19.40s	28.80s
WAXMAN 50	0.11s	5.18s	8.67s	28.09s	47.01s
WAXMAN 100	0.66s	20.18s	125.90s	380.30s	680.40s
WAXMAN 200	1.95s	90.18s	522.90s	725.30s	DNF

3.4.3 Execution time

First, we investigate the CPU time to solve our ILP, since the execution time plays an important role in network routing. We run our ILP with different sizes of network topologies, Table 3.4 presents the average CPU time with different destination sets. DNF means it Did Not Finish within one hour. In small networks and for small group sizes (e.g. 2 destinations and 50 nodes), the execution time is found to be reasonable. These values also show the possibility of the application of the exact computation in Software Defined Network SDN [43] for off-line route computation. We see in the table that, the larger the number of destinations $d = |D|$ is, the more complex the solution will become. It is due to the fact that the edges of the original graph are duplicated. Hence the algorithm will compare more combinations to find the best one which influences considerably the CPU time.

3.4.4 Proportion of optimal hierarchies

We investigate the proportion of instances that do not admit a tree as a solution, but a hierarchy. We conduct this test on 500 instances in the NTT topology. The set of destinations and the source are randomly

[2]http://www-03.ibm.com/software/products/en/ibmilogcpleoptistud/

Table 3.5: Proportion of hierarchies in feasible instances

Number of destinations	5 destinations				8 destinations			
Constraints looseness	CL = 6		CL = 7		CL = 6		CL = 7	
Number of constraints	FI	POH	FI	POH	FI	POH	FI	POH
2	60.60%	29.83%	61.00%	27.72%	20.40%	64.70%	20.40%	55.88%
4	58.40%	28.23%	58.80%	25.34%	16.00%	58.75%	18.00%	46.66%
6	57.20%	34.61%	57.40%	34.49%	15.60%	62.82%	17.40%	51.72%
8	54.80%	33.94%	56.00%	32.14%	14.60%	63.01%	16.40%	51.21%

chosen. Table 3.5 presents the proportion of Feasible Instances (FI) and among them, the percentage of the optimal solutions which are hierarchies and not tree (POH).

This test demonstrates the interest of the hierarchy and shows the frequency of this phenomenon in real networks. As shown in Table 3.5, the rate of instances for which the optimal solution is a hierarchy and not a tree is strongly correlated with:
Number of destinations: When the number of destinations increases the number of paths increases as well, which causes more crosses between paths to obtain the exact solution.
Number of constraints: With a large number of constraints the matching between paths will be less probable and the generation of overlaps between paths increases which explains the obtained results.
Looseness of constraints: The numerical results in Table 3.5 show that the proportion of hierarchies decreases when the constraints are more loose ($CL = 7$). However, the theoretical part proves that even when the constraints are loose the optimal solution may be a hierarchy (Property 3.3). With very loose constraints the solution converges to the Steiner tree.

3.4.5 Evaluation of MAMCRA algorithm

In the following tests, we implement two different versions of MAMCRA. MAMCRA-NLF with the Non-linear Length Function (NLF) is proposed in [53]. It minimises this non-linear length function without taking into account the optimisation of the cost, while MAMCRA-cost [46] tries to optimise the cost.
We investigate in each experimentation the cost of the multicast route.
CL tests: Cost of the multicast structure generated regarding the constraint's looseness. Figure 3.5, shows that the solutions given by MAMCRA-NLF become worse when the constraints are very loose. The quality improves when the constraints are tightened. The main reason for this may be that when the number of feasible paths is small, MAMCRA-NLF and the ILP frequently choose the same paths. On the contrary, the quality of the solution returned by MAMCRA-cost is very good, especially when the constraints are very loose or very tight.
DD tests: Cost of the multicast structure generated regarding destinations vertices density. For NSF topology Figure 3.6, shows that the solutions given by MAMCRA-NLF are further from the optimum when the DD increases, because the number of edges that construct the multicast route is increasing to cover all destinations. For the others (WAXMAN and NTT) MAMCRA-NLF gives a closer solution to the optimum. Meanwhile, MAMCRA-NLF chooses the feasible edges without optimizing the cost.
Here again, the solution returned by MAMCRA-cost are very close to the optimum. MAMCRA-cost gives

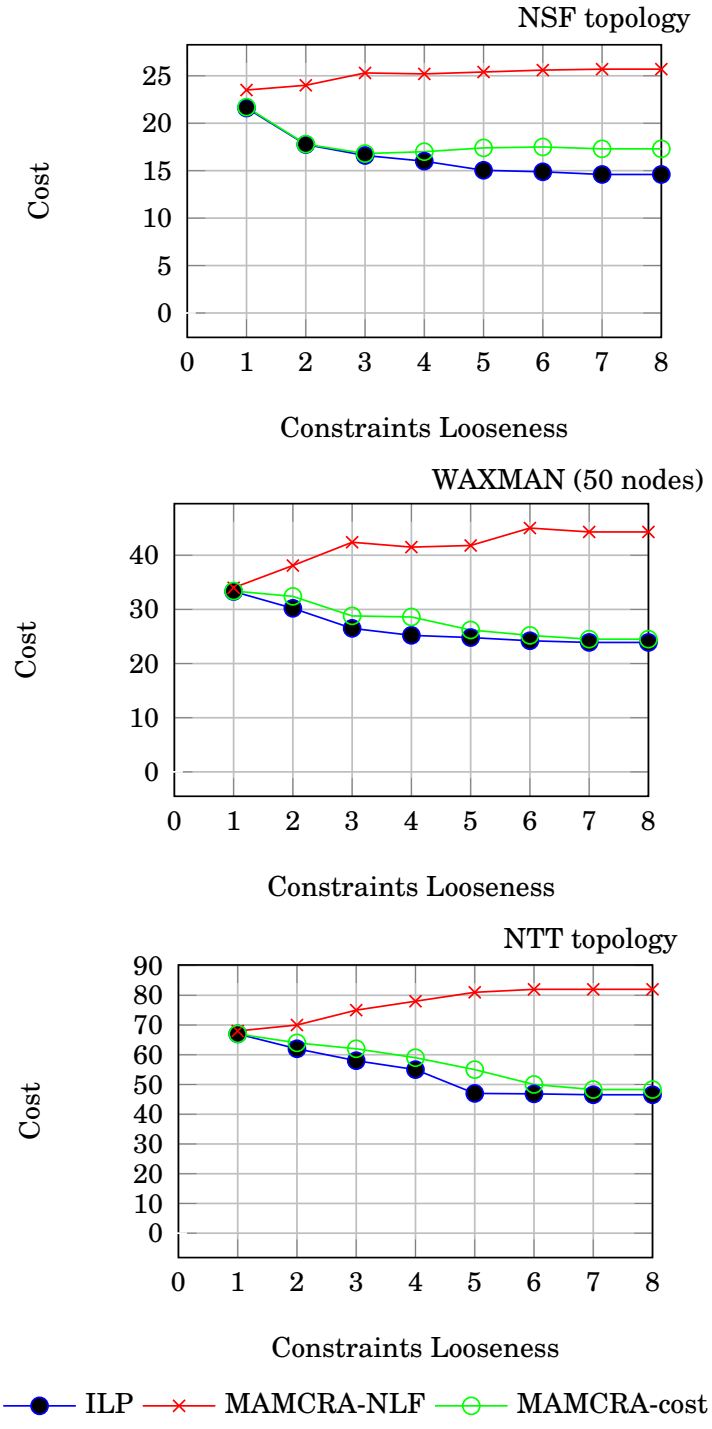

Figure 3.5: CL tests: Cost regarding to the Constraints Looseness.

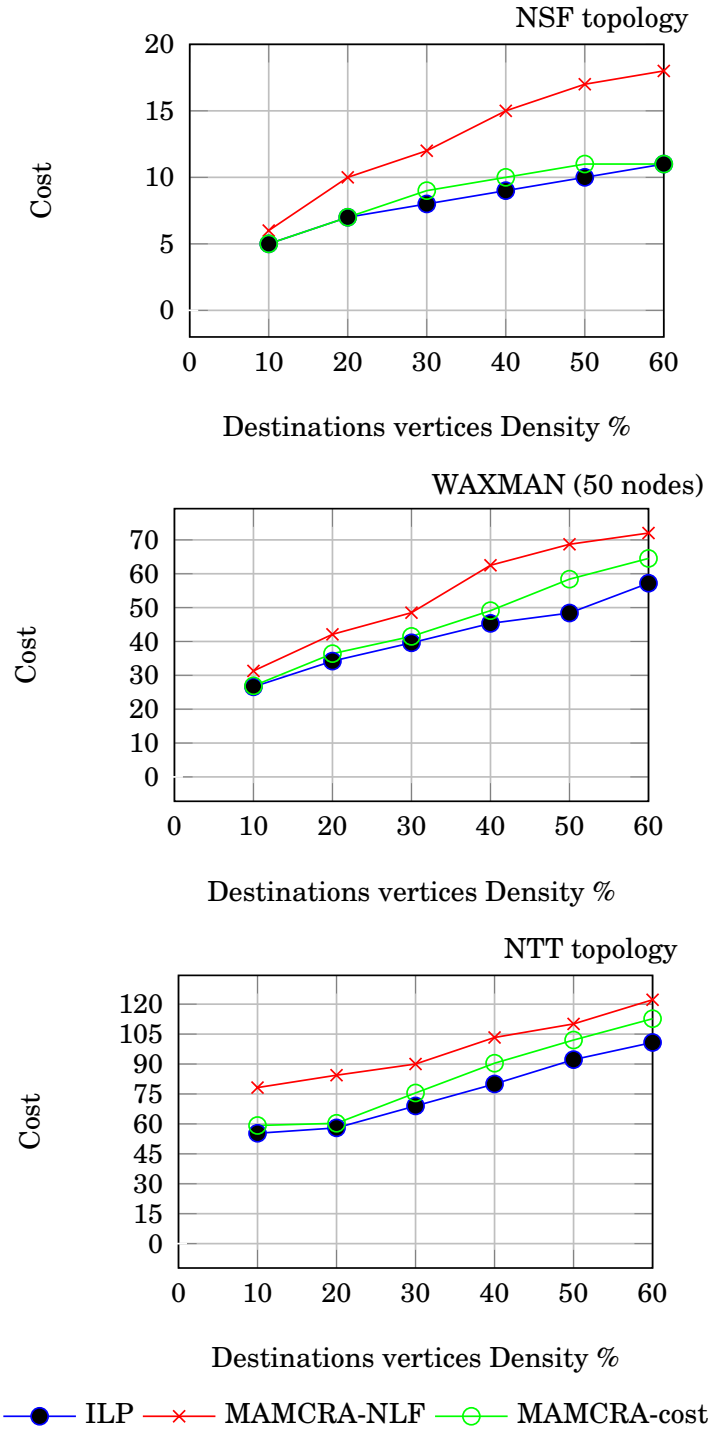

Figure 3.6: DD tests: Cost regarding the Destinations vertices Density.

an efficient (even if not always optimal) solution.

3.5 Pretreatment ArcReduce

To solve the QoS routing more efficiently we propose a new algorithm named ArcReduce which prunes the weighted topology graph to produce a reduced graph with fewer arcs. ArcReduce can even answer in polynomial time, if the problem has a solution or not before starting the resolution process.

3.5.1 Motivation

Usually the complexity of all proposed QoS routing algorithms including our ILP is strongly correlated with the number of arcs/vertices. Therefore, the size of the input data (i.e. graph) is fundamental. The role of ArcReduce consists to eliminate the arcs that cannot be part of the feasible QoS multicast routes. ArcReduce uses Dijkstra's algorithm to compute the shortest paths from the source and the destination to an arc's source and target respectively for each metric separately. An arc can be eliminated if it is infeasible for all destinations. In other words, if the arc cannot be part of any feasible path to any destination which makes it useless. In this case, naturally it will not be considered in the resolution process. We will call these arcs as useless arcs.

Let us take the graph in Figure 3.7 as an illustrative example, when the vertex s is the source node and $\{c,e\}$ are the destinations nodes. As we see, the arc (a,d) cannot be a part of a feasible multicast route. Since for w_1, the distance between s and a is 3, the distance between d and e is 3 and for w_1(a,d) = 5, so the shortest path for w_1 from s to e = 11 which is $\geq L_1 = 9$. The same for the destination c where the shortest path for w_1 from s to c is 15. Thus, ArcReduce can eliminate (a,d) before starting the process resolution without influence the final solution as illustrated in Figure 3.7.B.

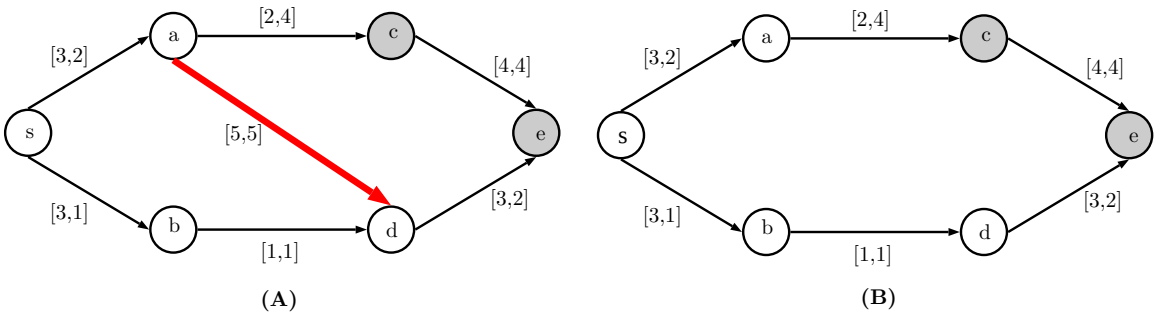

Figure 3.7: Example of an infeasible arc.

Theorem 3.2. *Let (x,y) be useless arc for G. An arc (x',y') is useless for G if and only if it is useless for $G\backslash\{(x,y)\}$.*

Proof. Let (x,y) be useless arc for G. $l_i^*(a,b)$ is the length of a path $p(a,b)$ using the weight of index i So:

$$\forall j \in \{1...d\}, \exists i \in \{1...M\} \mid l_i^*(x,y) + w_i(x,y) + l_i^*(y,d_j) > L_i \qquad \text{(A)}$$

- If (x',y') is useless for G, it is obviously useless for $G\backslash\{(x,y)\}$.
- Suppose that (x',y') is useless for $G\backslash\{(x,y)\}$ and not for G.

$$\text{Then } \exists j \in \{1...d\} \mid \forall i \in \{1...M\},\ l_i^*(s,y') + w_i(x',y') + l_i^*(y',d_j) \leq L_i \qquad \text{(B)}$$

Since this property is false in $G\backslash\{(x,y)\}$, it means that:
$\exists i \in \{1...M\} \mid (x,y) \in l_i^*(s,x')$ or $(x,y) \in l_i^*(y',d_j)$
If $(x,y) \in l_i^*(s,x')$ then (B) implies that, $l_i^*(s,x) + w_i(x,y) + l_i^*(y,x') + w_i(x',y') + l_i^*(y',d_j) \leq L_i$.
But $l_i^*(y,x') + w_i(x',y') + l_i^*(y',d_j) \geq l_i^*(y,d_j)$ so $l_i^*(s,x) + w_i(x,y) + l_i^*(y,d_j) \leq L_i$ which contradicts (A).
The argumentation is the same if $(x,y) \in l_i^*(y',d_j)$.

■

Definition 3.3. Let $(u,v) \in A$ be an arc. Let $l_i^*(s,u)$ and $l_i^*(v,d)$ be the length of the shortest path from the source s to u and from v to the destination d respectively concerning the metric $i \in \{1,...,M\}$.
The arc (u,v) is infeasible for d (i.e. cannot be used in a route from s to d respecting the QoS constraints):

$$l_i^*(u,v) + w_i(u,v) + l_i^*(v,d) > L_i \qquad (3.14)$$

Property 3.6. *An arc (u,v) is useless and can be eliminated from the graph G if it is infeasible for all destinations in D.*

Proof. Obvious ■

3.5.2 The ArcReduce algorithm

The meta-code of the ArcReduce algorithm is listed in Algorithm 3. ArcReduce checks an arc that is not checked yet (line 2). Notice that Theorem 3.2 indicates, the order of arcs for checking can be arbitrary. Then, the algorithm calculates for each metric $i \in \{1,\ldots,M\}$, the shortest path l_i^* from the source to (u) and from (v) to the destination, using Dijkstra's algorithm and check the feasibility (lines 6-9) (cf. Definition 3.3). If the arc (u,v) is infeasible for one constraint the algorithm does not have to check the other constraints and repeat the same calculation with all $d \in D$. If the arc (u,v) is feasible for one destination the algorithm does not have to check the other destinations and goes directly to (line 2). Otherwise, if the arc is infeasible for all destinations, the arc will be dropped from A and an other arc will be tested.

3.5.3 Complexity

ArcReduce calculates the shortest path using Dijkstra's algorithm. The complexity of Dijkstra's algorithm using a Fibonacci heap is $O(|A| + |V|\log|V|)$. For all arcs the complexity is $O(|A|(|A| + |V|\log|V|)$. The for loop starting on line 4 is invoked at most $O(|D|)$ times. Calculating the feasibility of the arc with all constraints line 6 takes at most $O(M)$ times. The worst-case time-complexity of ArcReduce is:

$$O((|A||D|M)(|A| + |V|\log|V|)). \qquad (3.15)$$

where M is the number of QoS metrics and $|D|$ is the number of destinations.

Algorithm 3 ArcReduce

Require: $G(V,A)$: digraph, s : source, $D = \{d_1,\ldots,d_k\}$: destinations set, $\overrightarrow{L} = [L_1,\ldots,L_M]^T$: constraints.
Ensure: $G'(V',A')$: reduced graph.

```
ArcEliminated := ∅;
for all (u,v) ∈ A do
    InfeasibleDestination:= 0;
    for all d ∈ D do
        InfeasibleConstraints:= 0;
        i = 0;
        while (i ≤ M and InfeasibleConstraints := 0) do
            i = i + 1;
            if l_i^*(u,v) + w_i(u,v) + l_i^*(v,d) ≥ L_i then
                InfeasibleConstraints := InfeasibleConstraints+1;
            end if
        end while
        if FeasibleConstraints ≠ 0 then
            InfeasibleDestination := InfeasibleDestination+1;
        end if
    end for
    if InfeasibleDestination == |D| then
        Add (u,v) to ArcEliminated ;
    end if
end for
G^*(V^*,A^*) == G(V, A \ ArcEliminated) ;
```

3.6 Experiments for the pre-treatment ArcReduce

3.6.1 Percentage of eliminated arcs

In this part we investigate the percentage of eliminated arcs regarding the three performance parameters used in the experiments of the ILP: *Number of Destinations ND, Constraints Looseness CL* and *Number of Constraints NC*.

Table 3.6: Percentage of eliminated arcs with ArcReduce (100 nodes)

ND	2 destinations			4 destinations		
NC	2	4	6	2	4	6
CL= 2	90.82%	91.12%	91.41%	87.69%	94.03%	94.93%
CL= 3	53.61%	81.96%	86.24%	36.44%	64.67%	68.52%
CL= 4	5.29%	16.34%	17.82%	1.82%	5.29%	6.40%

3.6.2 Execution time of the ILP with and without using ArcReduce

Figure 3.8 represents the comparison between the runtime of the three algorithms: ArcReduce, ILP without ArcReduce and ILP with ArcReduce . The CL and NC are fixed in 3 and 2 respectively. As shown by the results above, we can gain important time using the ArcReduce. In turn, the ArcReduce filters the graph in a negligible amount of time compared the execution time of the ILP (cf. Figure 3.8).

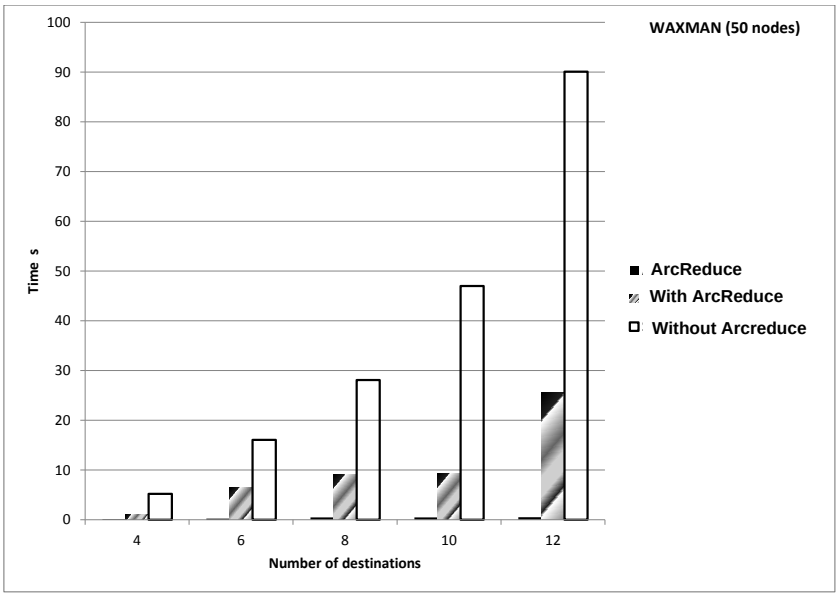

Figure 3.8: Execution time with/without using ArcReduce.

3.6.3 Discussion

It is clear that the percentage of eliminated arcs is directly correlated with the CL, as illustrated in Table 3.6. ArcReduce is effective when the problems become harder (i.e. the constraints are very strict) because the number of eliminated arcs is very important, which significantly accelerates the resolution process (cf. Figure 3.8). When the problem is easier, with relaxed constraints, most of the arcs are feasible, so the algorithm cannot significantly reduce the search area. However, with relaxed constraints feasible solution becomes easier to find and even a naive heuristic can find a good one.

Figure 3.9 represents the rate at which exact solutions are found by an algorithm designed to find the optimal Steiner tree (we have used an existing ILP formulation of Steiner tree) [65] without considering the QoS constraints. As shown, the success rate is 100% when the Constraints Looseness are greater than 6. This is due to the fact that when the constraints are very loose almost all paths are feasible. Therefore, even if the ILP tries to optimize only the cost, the generated solution respect the constraints too.

3.6.4 Connectivity

We highlight in this part the connectivity of the graph obtained after the filtering step, especially the connectivity of the multicast members. For this purpose we extend the ArcReduce algorithm to check

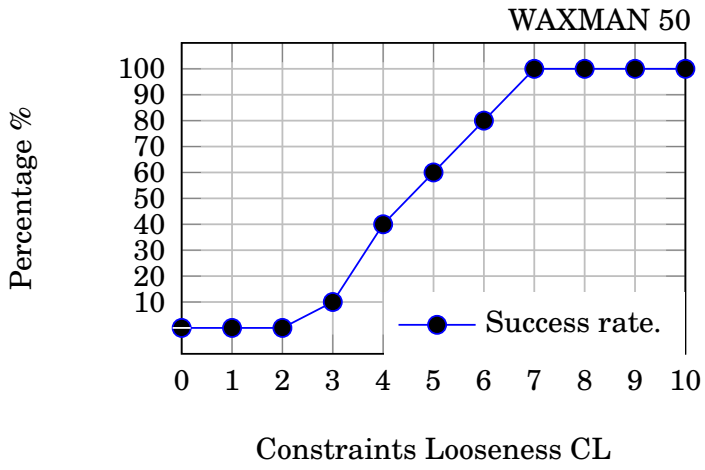

Figure 3.9: The success rate using an exact Steiner algorithm.

whether there is at least a path from the source to each destination using breadth-first search strategy. The connectivity test allows us to know if a given problem has a solution or not before searching for the solution.

With simple modification of the connectivity test, we can save the set or the partial set of destinations which are reachable from the source, to decide later if we want to solve the problem partially or not.

3.7 ILP extension for bandwidth metric

The bandwidth is one of the most important bottleneck metrics, which refers to the volume of information per unit of time that the transmission medium can handle. Bandwidth is typically expressed in bits per second (e.g. 50 Mbps or 50 Mb/s which means a data transfer rate of 50 million bits every second). For example, each multimedia application has a different bandwidth which provides the bandwidth constraints as an input. Then, it is worth eliminating all links with less bandwidth before starting the routing process (i.e. preprocessing step or topology filtering). The topology filtering cannot be used in our model because a link can be crossed several time in the hierarchy. As mentioned above, the previously proposed ILP model does not take into account the bottleneck metrics. We present in this section a modified version of our ILP to manage the bottleneck constraints. For that, we add a new constraint to handle the bottleneck. Its role consists in ignoring all infeasible links in term of bandwidth in the initial graph during the construction of the route. Presenting hierarchies as solution, we have to verify the feasibility of the link during the calculation of the route.

Let $Bb(m,n)$ be the bandwidth capacity of arc (m,n). We suppose that this capacity may be different in the two directions.

Let B be the bandwidth needed to sent the message.

The bandwidth constraint can be expressed as follow:

$$\sum_{i\in\{1,\dots,d\}} U_i(m,n).B \leq Bb(m,n), \forall (m,n) \in E. \tag{3.16}$$

3.8 Conclusion

This chapter has shown that a multicast tree is not always the exact solution of the MCMCM. A more complex structure called hierarchy is sometimes the only optimal/feasible solution. In response to this property we propose an ILP to solve optimally this NP-hard problem. We also propose a new pretreatment algorithm, which is efficient to reduce the search space.

The Integer Linear Program solving this problem is not trivial, because the solution is different from a sub-graph. The proposed ILP finds the optimal solution when it exists. This allows us to evaluate the proportion of instances in which the solution is not a tree (which is often considered to be the only possible solution). Moreover, we proposed an extension to take into account bottleneck constraints in the computation of hierarchies. Even if the computation time is reasonable in small networks and independent of the number of constraints, it is of course not tractable when the size of the instances or the number of destinations grows. A pre-treatment called ArcReduce has been proposed to accelerate and increase the resolution capacity of our ILP in large datasets. One of the main interest of our exact solution is that it allows us to know the real performances of the heuristics proposed to solve this problem. In the examined benchmark networks, the evaluation tests using the well known heuristic MAMCRA show that the solution returned by the heuristic is at most 20.6% larger than the optimal solution in the worst case (and most often around 8.5% larger).

CHAPTER 4

ROUTING WITH MULTIPLE CONSTRAINTS OF QOS IN LOW POWER AND LOSSY NETWORKS

In this chapter we are interested in the problem of routing with QoS in Low Power and Lossy Networks (LLNs). For some applications in these networks, Quality of Service requirement may be imposed. For this, some propositions have been formulated to satisfy QoS requirements in LLNs using the IPv6 routing protocol (RPL) even if there are several QoS constraints. We start in Section 4.1 by an introduction. Then, in Section 4.1.1 we give a short description of the role of the Objective Function (OF). In Section 4.2 we present the problem of the DODAG computation with QoS. Section 4.3 presents our Non-Linear Length OF (NL-OF) which take into account any number of additive constraints for QoS routing. Section 4.4 presents numerical studies to compare our proposition with three OFs predefined in RPL. In Section 4.5, we present the exact and some heuristic routing algorithms with QoS constraints. One can find the numerical results in Section 4.6 and Section 4.7 concludes this study.

4.1 Introduction

More and more new applications are developed based on the Internet of Things and for some applications, a Quality of Service (QoS) should be respected. However, the devices connected in this kind of networks are often vulnerable and dotted by limited energy (they are considered as Low-power and Lossy Networks, LLNs). These networks need special solutions for routing and RPL [81] is a standard IPv6 routing protocol to respond to these needs. As it is presented, RPL build Destination Oriented Directed Acyclic Graphs (DODAGs) following an Objective Function (OF) for routing [70]. In its basic version, RPL uses simple OFs, for example the hop-count to minimize the energy consumption or the expected transmission count [70] [20]. Propositions have been formulated to combine two or several metrics [35][18] in an additive or lexical manner.

In this chapter, we propose a greedy DODAG construction based on a Non-Linear Length which takes

into account any number of constraints for QoS routing. The results are partially presented in [38]. Let us suppose that the QoS requirement is given as a set constraints on additive end-to-end metrics. In order to construct the constrained routing DODAGs, multi-constrained paths (sub-paths) are computed based on the Non-Linear Length proposed for exact multi-constrained QoS path computation [54]. This solution takes into account an arbitrary number of metrics and tries to provide paths corresponding to the input constraints. However, it is based on a greedy path computation and can not always guarantee a solution for all of the nodes which can have a solution.

It is known that the multi-constrained QoS routing is an NP-hard problem and its solution is not always a simple acyclic graph (not a tree) [56]. In the second part of this chapter, we investigate to construct and analyze the routing scheme which is needed to respect QoS constraints using the basic idea (the computation of destination oriented directed routes) of RPL and the related route construction mechanisms and messages. We present the computation of the exact solution in LLNs between a Border Router and the nodes respecting a set of QoS constraints. The solution can not always be computed with polynomial time algorithms. Nevertheless, this computation permits to evaluate the performance of the earlier proposed greedy algorithm, and find a trade-off between efficiency and computation time.

4.1.1 The Objective Function

As explained in Chapter 1, RPL uses an Objective Function (OF) during the construction of the desired DODAG. To do so, the OF takes into account the routing metrics and constraints. RPL comes with three predefined OFs, i.e. OF0, MRHOF with ETX and MRHOF with energy.[1] These OFs focus on only one routing metrics: number of hops, Expected Transmission Count and energy consumption. However, some applications such as video and collaborative applications, require several constraints of QoS to operate correctly. RPL implementations can choose to adopt a simple approach based on the use of a single metric with no constraint (the optimization of the cost for example). RPL can separate the OFs from the core of the protocol which allows it to meet the different optimization criteria required. Each DODAG instance in RPL's DAG is associated with a particular OF. Whereas other implementations can use a larger set of link (and/or node) routing metrics and constraints.

The result of the route construction is a Destination Oriented Directed Acyclic Graph (DODAG), which respects the specific metrics as the hop-count, the energy consumption or the expected transmission count [70] [20]. We are interested in the optimization of the cost respecting several Quality of Service (QoS) constraints.

4.1.2 RPL routing attributes (Metrics and Constraints)

Unlike traditional wired networks, LLNs have unique characteristics that require the specification of new routing metrics and constraints [74]. There are several metrics and constraints based on them, which can be used in path computation by RPL and can be categorized into two basic types:

- Node metrics and related constraints (e.g., node state and attribute and node energy.).
- Link metrics and related constraints (e.g., link throughput and link latency.).

[1] See chapter 1, page 15 for a detailed explanation.

4.1.3 QoS metrics in RPL

Before presenting our proposition, let us give a brief overview about the most interesting node and link metrics for our study on QoS routing in RPL [74].

- Hop count: metric to report the number of traversed nodes before reaching a given destination.
- Node energy: an important metric which permits to avoid selecting a node with low residual energy.
- Latency: this metrics allows to adjust globally the end-to-end delay on the subnet, on a link-by-link basis, or not at all. The minimization of this metric is critical fo real time applications.
- Link reliability: as the change in link quality can affect network connectivity, the reliability of links is crucial. Different link reliability metrics can be defined in order to reflect several reliability aspects (link quality level, ETX metric, etc.) can be defined.
- Moreover, additional metrics can be proposed for QoS aware applications such as the jitter and the probability of packet losses.

Our work is based on additive link metrics. In the following section we formulate the multi-constrained routing problem in RPL.

4.2 Problem Formulation for the DODAG Computation with QoS

Let $G = (V, E)$ be the weighted connected graph representing the topology, where V is the set of nodes (devices) and E is the set of edges representing the possible links. Each edge e is associated with M weights corresponding to the QoS metrics (given by a weight vector $\vec{w}(e) = [w_1(e), w_2(e), ..., w_M(e)]^T$) and $c(e)$ is the cost of using the link. We suppose that the link values are the same in both directions.

The objective is to build a DODAG that connects the root node s (a target Border Router for example) to the others nodes (routers and leafs) with the guarantee that each path from the root to any other node respects the end-to-end constraints given by the constraint vector $\vec{L} = [L_1, L_2, ..., L_M]^T$. These paths can be used in both directions. We note p_{d_j} the path from s to d_j (all examined paths have s as origin).

In LLNs the link and node metrics are usually (but not always) additive. In this work we only consider additive metrics.

The final DODAG (cf. Fig. 4.1.B) has to contain a feasible path from the root to each leaf such that:

$$w_i(p_{d_j}) \leq L_i, i = 1, \ldots, M, j = 1, \ldots, |D| \tag{4.1}$$

Remark: the DODAG computation with QoS is a particular case of MCMCM, the only difference is that MCMCM aims to find a feasible route from the source to a partial set of nodes while the DODAG computation with QoS aims to find a feasible route from the source to all other nodes.

We can consider two problems:

- Multi-Constrained DODAG problem, where we are interested by feasible paths satisfying the constraints

- Multi-Constrained Minimum Cost DODAG problem, if we also take into account the cost optimization. This latter can be the cost of paths or the cost of the DODAG. The importance of using a cost function during the construction of the DODAG is that the provided DODAG can be handled according the applications setting and the user's requirements. For instance, a cost function which is defined for a setting where energy is more critical than delay is more suitable for that specific context than a general-purpose cost function. This makes our approach custom-tailored to the context in hand.

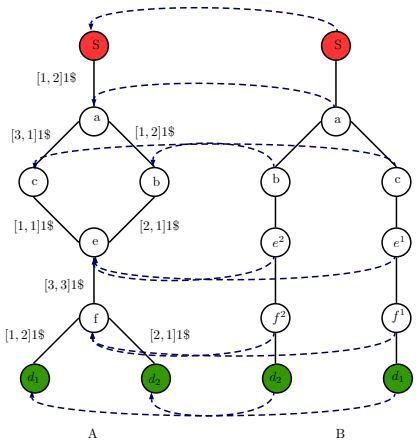

Figure 4.1: Example of LLNs topology and its DODAG.

Note that, unlike the previous works in the literature, we propose to handle an arbitrary number of metrics and constraints to satisfy QoS routing requirements. Our objective is the respect of the QoS constraints. One of the most challenging research tasks in the domain is the routing with multiple constraints (i.e., finding a unicast route or multicast route or even broadcast that respects a set of QoS constraints). A route (even if it is a simple path) can be good for a given metric but bad for another. The different constraints on the different metrics can leave to severe contradictions.

4.3 The proposed objective function

To select routes, RPL uses an Objective Function (for instance OF0) which takes into account only one metric. OF0 remains insufficient to build a DAG (or DODAGs) that can satisfy more precise QoS requirements. Some works have been done to ensure the QoS in RPL, they try to combine two metrics or more using different methods. We propose a solution for the QoS routing with RPL by extending its mechanism to support several metrics and QoS constraints. Our DODAG construction is based on the non-linear length of paths using multiple metrics [54].

4.3.1 Non-Linear Length based Objective Function

For a given path p_d, the non-linear length is defined by the formula 1.4 given in Section 1.4.2.

The non-linear length of paths is used in SAMCRA to quantify the quality of the path p_d for MCP problem.[2]

Our objective (NL-OF) is to construct DODAGs from roots to nodes such that the non linear length is the smallest possible. Trivially, a constructed instance will depend on the set of roots, the set of metrics and the QoS requirement $\vec{L}$.

4.3.2 Greedy construction of DODAGs with the NL-OF

Let us assume that there is a single root node and that each other node can store the weight vector computed from the root to this node.

As already pointed out, the construction of the DODAG starts from the root which sends a DIO message to its neighbors. We propose that the DIO message contains the values of the cumulated weight vector from the root (the root has a null vector for the metrics). After receiving all possible DIO messages in RPL, a node v can compute the cumulated weight vector for each path concerned by the messages. Let $\vec{w}(p_u)$ be the weight vector for the node u, which send a DIO message to node v and let $\vec{w}(e)$ the weight on the link e between u to v. Trivially, the weight vector $\vec{w}(p_v^u)$ associated with the path from the source to v passing by u is:

$$\vec{w}(p_v^u) = \vec{w}(p_u) + \vec{w}(e) \tag{4.2}$$

v can select all parent nodes, which are in feasible paths (with non-linear length less then 1) and choose its preferred parent (Figure 4.2). The preferred parent is the node which insure the minimal non-linear length for v. If a node cannot compute all metrics for the path through a neighbor node u, the non-linear length of this path will be set to infinity. Subsequently, the node v sends a DIO message to its neighbors if its non-linear length on the preferred path is less than 1, and so forth until an other node is reached. The first range of nodes (root's neighbors) will choose the root as a preferred parent and send the DIO message to their neighbors, etc.

Our algorithm select parent nodes and construct acyclic graphs in a greedy manner (Figure 4.2.B). However, the optimal paths with minimal non-linear length are not guaranteed by this procedure. In other words, the constructed DODAG does not always contain the paths with minimum length, and some feasible paths may be lost (e.g. d_2 in Figure 4.2.B).

4.3.3 Analysis of the solution

To illustrate the weakness of the greedy strategy, let us analyze the exact solution of our optimal QoS routing problem.

4.3.3.1 Exact solution

If our objective is the minimization of the non-linear length from the source to each node, the exact solution (set of routes, which contains the best paths) is not a tree we identified, it is a hierarchy. Seeking an exact solution does not produce necessarily a DODAG as a route. In addition, even if the preferred parent has the shortest sub-path from the root, it does not guarantee that using this sub-path will lead to the path with the best non-linear length, to select the path with minimal non-linear length. In the worst case, all

[2]See chapter 1, page 7 for a detailed explanation of non-linear length used in SAMCRA.

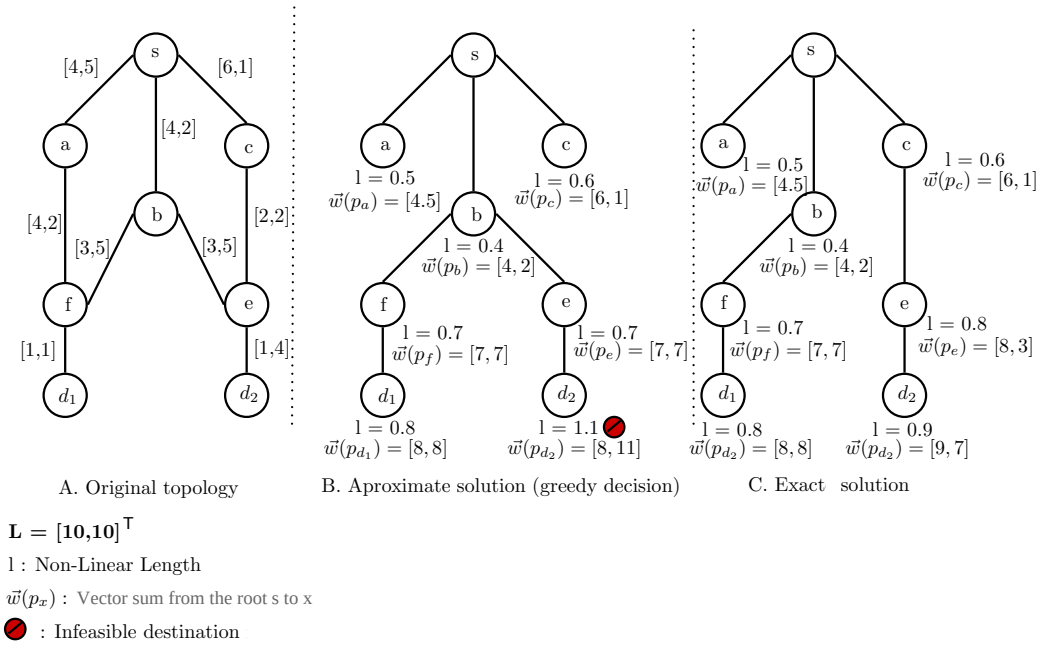

Figure 4.2: Construction of the DODAG

paths from the source to the node should be compared. (As an illustration, there is two feasible paths to the node e in Figure. 4.2). The number of paths can grow in an exponential manner even if we can reduce the research space using the Pareto dominance [28]. For example the path $\{s, b, f\}$ dominates the path $\{s, a, f\}$ in Figure 4.2. An exact computation is presented in 4.5.3.

4.3.3.2 Greedy solution

Since RPL is designed for LLNs in which the devices are constrained with memory, computation capacity and energy, the storage of a large number of paths and the computation of the exact solution is very costly and not always feasible. So we use a simple greedy method for the construction of a DODAG (as we explained above). In 4.3.2 we propose a DODAG which is an approximate solution since all the possible and feasible paths are not stored and transmitted for further computation.

To summarize: Adding all non-dominated paths to our OF-NL mechanism, one can ensure that all possible sub-path are preserved. This gives to each successor the opportunity of choosing the best path according to the objective. Notice that when the exact solution is computed, some overlap between paths may be produce (cf. Figure 4.1). The exact solution will be presented later in this chapter (Section 4.5.3).

4.3.4 Complexity

As mentioned in [81], the rate of DIO transmission is controlled by Trickle algorithm[3], so the number of DIO messages does not change when we use NL-OF.

Let $|V|$ be the number of nodes and $|E|$ the number of edges, each node saves one DIO message containing the vector sum from the border router. Selecting the minimum non-linear path length among

[3]The Trickle algorithm controls the amount of routing traffic in the form of DIO's that enter the network.

$|V|$ different path lengths takes at most $\theta(\log(|V|))$. Calculating the length from a given node to the border router takes $\theta(M)$ when there are M metrics while verifying the feasibility condition takes $\theta(M)$ at most. The worst-case time-complexity of the greedy solution is

$$\theta(|V|\log(|V|)+M|E|) \tag{4.3}$$

4.4 Performance Evaluation

In this section, we conduct numerical studies to compare our greedy algorithm with three OFs predefined in RPL. *OF0* uses only the hop count, while *MRHOF with ETX* minimizes the Expected Transmission Count along the path and *MRHOF with energy* chooses the DAG's nodes with minimum energy consumption. The first part is dedicated to evaluate the performance of NL-OF, MRHOF-ETX, MRHOF-energy and OF0 in term of two QoS metrics of interest: End-to-End Delay and Packet Loss Ratio. The second part will evaluate RPL in term of Average Power Consumption. In order to better understand our simulation, we present in what follows the explanation of each performance metric and the simulation environment.

4.4.1 Performance metrics

- **QoS performance metrics.** The first performance metric is the *End-to-End Delay*. It refers to the time taken for a packet to be transmitted across a network from a node to the DODAG root. The second performance metric is the *Packet Loss Ratio*, which measures the ratio of packets lost to the total packets sent.

- **RPL performance metric.** The key metric for RPL evaluation is the *Average Power Consumption*, which is the average percentage of power consumed by all the nodes in the network.

4.4.2 Network setup

To analyse the results with high level of accuracy, we use Cooja simulator under Contiki operating system [1]. We design in all simulations a network of 39 routers and 1 root of sky node type. The nodes are randomly distributed over a square space (100m × 100m). For the control traffic, we define the DIO Interval Minimum which is used as an initial interval for the control packet transmission and DIO Interval Doubling[4] to place an upper limit on the rate of this transmission. The start delay defines when the nodes start transmitting their messages to the sink node. To reach a stable state we run each simulation 1800s. Table 4.1 presents the simulation parameters.

Notices: we fixed the end-to-end constraints as follows: MAX-END-TO-END-DELAY = 700 ms and MAX-PACKET-LOSS = 7 pkt/minute. So, $L = [700, 7]^T$. All information about Delay and Packet Loss are gathered using data-collected-view option in Cooja, which presents the simulation results in graphical manner and statistic manner.

We only consider the communication between nodes and the root, local communication is outside of scope of these tests.

[4]DIO Interval Minimum and DIO Interval Doubling present the Trickle's parameters [48]. In RPL each node use Trickle's parameters to control the rate of DIO transmission.

Table 4.1: Parameters used to evaluate different RPL's OFs.

Parameters	Values
Square Space	(100m × 100m)
RPL Nodes	40
DIO Min	10
DIO Doubling	70
Type of Nodes	Sky
Radio Environment	Unit Disk Graph Model (UDGM)
Start Delay	60 s
Simulation Time	1800s

4.4.3 Simulation and discussion

The aim of the first part of the simulation is to analyse the response of OFs to the requirements of susceptible applications, such as control systems [61]. As explained above, each OF has a special routing strategy.

- End-to-End Delay: Figure 4.3 shows the average latency from nodes to the root according to the number of hops between the nodes and the root. The latency is the amount of time it takes a frame to be transmitted from source to destination. As shown the NL-OF provides less latency than the OF0 and MRHOF-energy all the time while the MRHOF-ETX provides the almost the same average latency of NL-OF at 1 and 3 hops. OF0 provides more latency which confirms that the shortest path in term of number of hop does not necessarily minimize the delay.

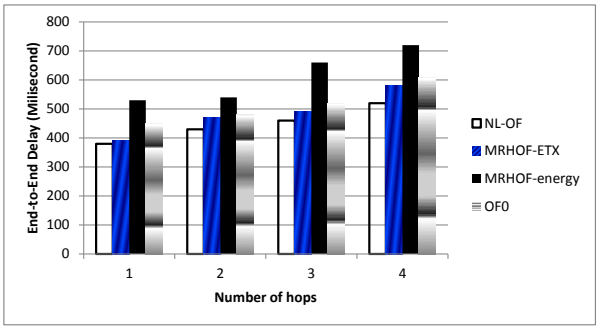

Figure 4.3: The average End-to-End Delay as a function of the number of hops.

- Packet Loss: Figure 4.4 shows one of the main advantages of NL-OF. NL-OF guarantees a less amount of packet loss regarding to other OFs, which augments the reliability of the network. This result demonstrates the efficiency of NL-OF routing strategy for packet loss which severely degrades some applications. As the results show, NL-OF is always better than the other objective functions over time of the test. The network loses less number of packets when the OF is based on ETX because MRHOF-ETX chooses the next hop with minimum ETX which provides paths with low Packet Loss ratio. For the others OFs (OF0 and MRHOF-energy) the Packet Loss is higher due to the routing mechanism of these OF which

optimize only the number of hops and the energy consumption respectively.

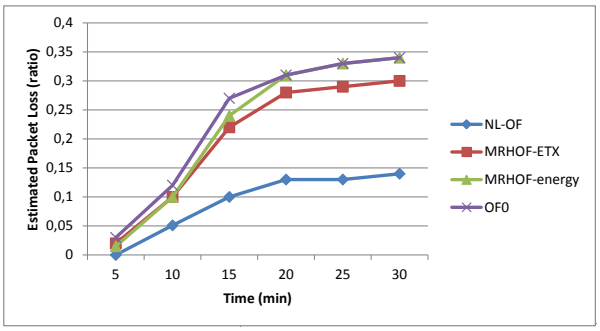

Figure 4.4: Lost of packets over time.

- Power Consumption: To estimate the network lifetime with NL-OF, we have compared NL-OF with OF0, MRHOF-ETX and MRHOF-energy in term of Power Consumption over time. Figure 4.5 presents the average power consumption. Trivially MRHOF-energy is the best in this challenge: 1.20 MW of energy was consumed through the 30 minutes of the test. MRHOF-ETX and OF0 consumed 1.5 MW when NL-OF consumed 1.4 MW. The consumption of energy was higher at the start of the experiment, because the DAG was not in a stable state (i.e. some nodes have not yet joined the DAG). Consequently, the Average Power Consumption is high. After the 25th minute, the network is more stable and the consumption of energy is significantly lower.

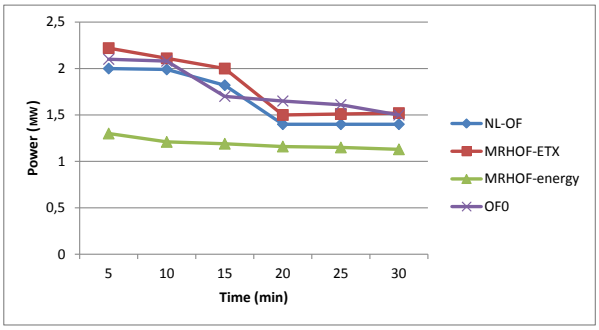

Figure 4.5: Average Power Consumption over time.

4.5 The analyzed QoS aware route computations

All the algorithms are based on the framework of RPL. Starting by the root node (usually a Border Router), DIO messages are sent to the neighbor nodes and to the neighbor of neighbors, etc. The DOI messages contain the essential QoS related information: the path from the root to the sender node with the cumulated values of the QoS metrics. Moreover, we suppose that each node knows the QoS values on the adjacent links. In this manner, when a DIO message arrives, the receiver can compute immediately the QoS weight vector for the received path. If this path is not feasible or dominated (in the sense of Pareto

dominance) by an other received path, it is neglected (deleted). Otherwise the node decides the treatment following the policy of the route construction.

1. Reception of a DIO for a new construction. Initialize the timer l_1 and l_2
2. Collect DIOs until l_1. Delete dominated prefixes
3. Send the remained prefixes to neighbors
4. Collect responses from potential leaves, sub-trees until l_2
5. Select preferred parents depending on the different sources
6. Send the parents to the neighbors

Notice: steps 3) and 4) are useless in the first greedy algorithm selecting only one preferred parent. The three policies are enumerated hereafter.

4.5.1 Greedy algorithm [38]

After receiving DIO messages during a fixed delay, a node u can compute the cumulated weight vector for each path concerned by the messages. u can select all parent nodes, which are in feasible paths and choose its preferred parent. The preferred parent in the DODAG is the node, which corresponds to the minimal non-linear length for u. Subsequently, the node u sends a DIO message to its neighbors if its non-linear length on the preferred path is less than 1 and so forth until reaching the other nodes. The path with the minimal non-linear length and the corresponding cumulated weight vector is sent to the neighbors. This algorithm selects only one parent node for each node and connect all nodes to the root with feasible paths if it is possible. This version construct an acyclic graph (a tree) in a greedy manner. Unfortunately, the optimal paths with minimal non-linear length are not guaranteed by this procedure. Let u and v be two nodes s.t. p_s^v is the prefix (a sub-path) of p_s^u. Let us suppose that p_s^v is the path with minimal non-linear length between s and v. Another path p_s^v may exist which can result a better non-linear length for u than the path p_s^v. Using the greedy selection, some feasible paths may be lost to compute good paths for the neighbors.

4.5.2 k-limited algorithm

The efficiency of the greedy algorithm can be improved: the number of reachable nodes respecting the constraints can be increased if the number of selected paths to parents for a node may be more than 1. Consequently, the route does not correspond to a tree based DODAG but to a Destination Oriented Directed Hierarchy. In order to avoid expensive and very large solutions, the number of possible parents may be limited by an integer value k.

In this version of the computation, a node receiving the non dominated and feasible path propositions can forward at most k paths to its neighbors. In this manner, critical neighbors can find more probably a path corresponding their needs, but it is always possible that there are not reached nodes because all of the possible paths are not enumerated. Regarding the example of Figure 4.6, if k is equal to two, the node vd can transmit only two paths to its neighbors and one of the nodes from $\{v1, v2, v3\}$ can not find a feasible path to s.

4.5.3 Exact algorithm

To avoid the loss of theoretically reachable nodes, the algorithm can not limit the number of transmitted and stored feasible and non dominated paths. In this case, we have an exact algorithm. To diminish the costs, the algorithm preserve only non reducible path. That is, redundant parents are eliminated beginning from the leaves going to the root. This algorithm insures that a node can have a feasible path to the root, if such a path exists. The counterpart of this property is the excessive storage, computation and communication need.

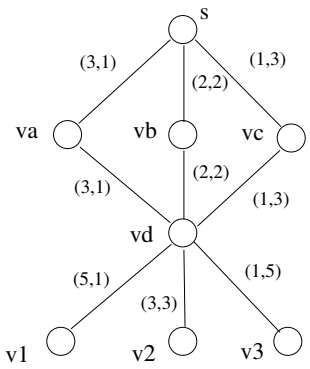

Figure 4.6: Example of a non reducible hierarchy for $L = [7,7]^T$

4.6 Performance evaluation of the algorithms

In this section, we evaluate each algorithm in terms of three performance metrics:
Execution time. The required time to construct the DODAG.
Missing nodes. The percentage of not reachable nodes by the constructed DODAG.
Quality of paths. The average of the non-linear length from the source to each node in the DODAG.

To evaluate the practical potential of each algorithm we have used random graphs generated by the WAXMAN model [79]. Each edge of the graphs is weighted by integer metrics, representing the QoS metrics and cost. These values are randomly chosen in {1,...,10}

Table 4.2: Simulation parameters of the used network.

	Information			
	$\|V\|$	$\|E\|$	α	β
	50	200	1	0.11

We analyse the performance of each algorithm regarding two parameters.
- *Constraints Looseness (CL)*, see Section 2.4.1.1 for the definition of CL.
- *Number of constraints (M).*

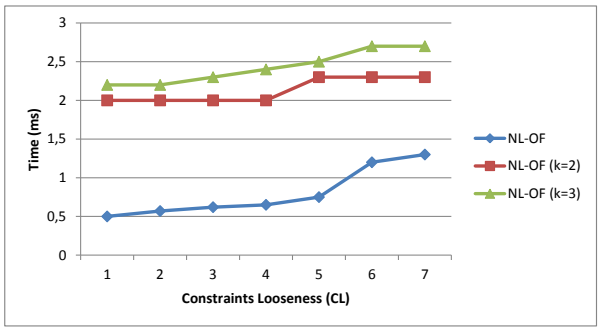

Figure 4.7: CPU time regarding different constraints looseness.

4.6.1 Execution time

We first illustrate the execution time of each algorithm which is primordial for LLNs, because the objects connected with this kind of network are suffering from low-power capacity. Therefore, the smaller the execution time is, the less energy it will consume.

In Figure 4.7 we observe that all algorithms take more time when the constraints are more loose. On another hand, since this problem is NP-hard the exact solution takes several minutes to construct the DODAG which is not reasonable in a realistic networks (cf. Table 4.3).

The explication of the illustrated results, is that each algorithm seeks the preferred parent for each node during the construction of the DODAG, so if the constraints are loose the number of feasible paths increases and the number of comparisons to choose the best parent in term of QoS increases too. As a results, the construction of feasible DODAGs takes much time as shown in Figure 4.7.

Table 4.3: Execution time of the exact algorithm

	CPU time						
CL	1	2	3	4	5	6	7
Time(min)	1.30	2.30	3.00	3.40	3.44	3.45	3.45

Figure 4.8 presents the scalability of the proposed algorithms with different sizes of networks. As we can see the time is increases linearly with the size of the network.

4.6.2 Missing nodes

This test presents the main part of the evaluation processes, since RPL tries to ensure that each network node is joining the DODAG. The first question we examine is how many nodes are missing during the construction of the DAG using greedy algorithm with the Non-Linear Objective function (NL-OF) [38], k-limited algorithm (NL-OF(k)) and the exact algorithm.

As shown in Figure 4.9, 4.10 and 4.11, the number of missing nodes decreases when the constraints are more loose. This is because more paths are feasible which creates more opportunities for the nodes to join the DODAG. However, the DODAG misses more nodes when the number of constraints increases.

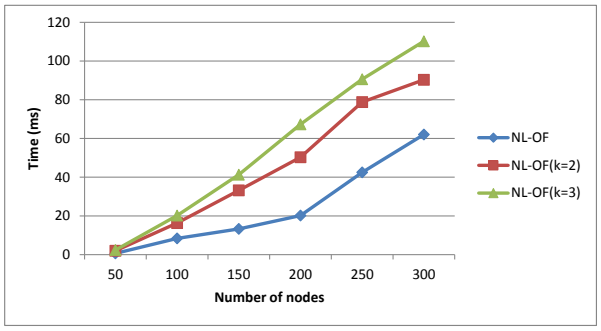

Figure 4.8: CPU time regarding different number of nodes.

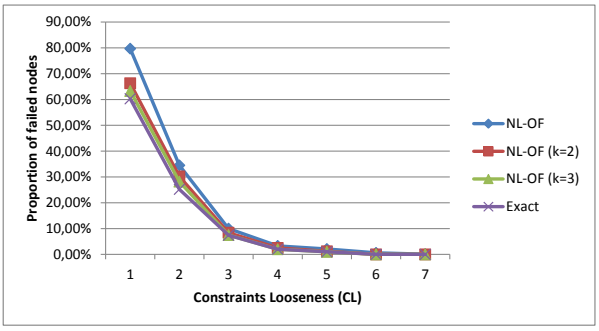

Figure 4.9: Number of failed nodes with M = 2.

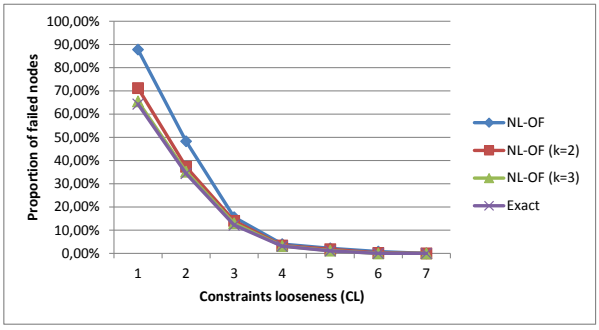

Figure 4.10: Number of failed nodes with M = 4.

To make the results of the missing nodes more visible we present Figure 4.12, which is the ratio between the three algorithms and the exact solution. As shown, the NL-OF (k=3) is very close to the exact solution where NL-OF (k=2) loses fewer nodes compared to NL-OF over time.

4.6.3 Quality of the DODAG

In this test, we evaluate the produced DODAGs by calculating the average of the non-linear length from the source to each node. This test allows us to measure the quality of the solution offered by the algorithms

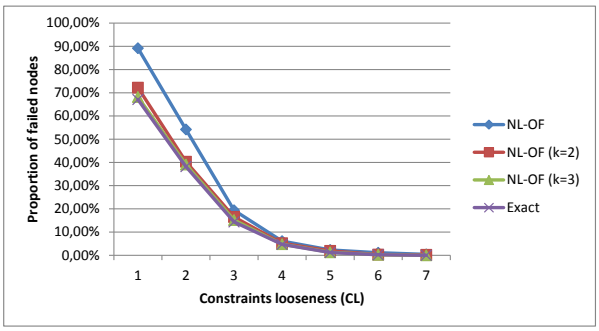

Figure 4.11: Number of failed nodes with M = 6.

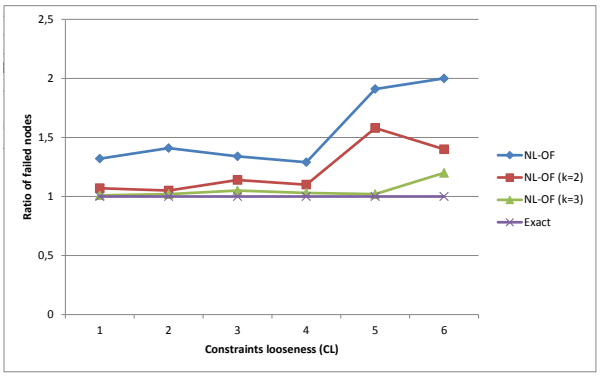

Figure 4.12: Ratio of failed nodes between the three algorithm and the exact solution with M = 6.

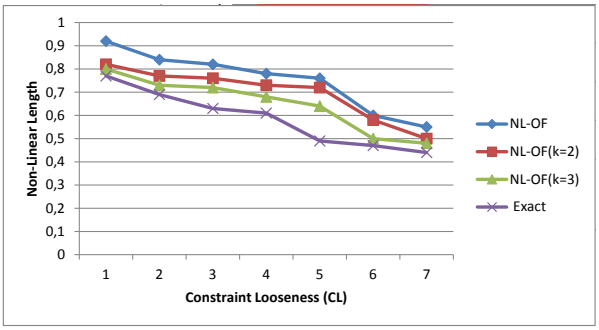

Figure 4.13: Quality of the DODAG.

and choose the best quality / time ratio algorithm.

4.7 Conclusion

We presented in this chapter a new objective function, the Non-Linear Length OF (NL-OF) for improving the QoS in RPL. Our proposed OF performs with any number of additive metrics and constrains. On the theoretical part, we have argued that our greedy approach, although being approximate, fits better our

setting while considering the memory constraints of LLNs devices. We have shown, based on experimental evaluation using Cooja, that our approach outperforms other OFs with respect to End-to-End Delay, Packet Loss and Power Consumption. We observed that our approach performs better than others for Packet Loss. The construction of routes respecting a set of QoS constraints in LLNs is a challenging task. The exact computation is NP-hard. For this, we present an exact algorithm that guarantees the construction of feasible paths from a root node to the other nodes if theses paths exist. To avoid the excessive computation, we propose a parameterized, less complex algorithm. We presents also the performance analyze of the three algorithms that can be used to construct DODAGs respecting a set of QoS constraints. The illustrated results show that RPL can be modified for the construction of DODAGs keeping its performance and flexibility and by using the greedy and the parameterized k-limited computation, while the exact solution is not recommended for LLNs. Moreover, the polynomial algorithms insure a good accessibility of nodes (the number of failed nodes regarding the exact solution is low). Another important result of the analysis is that the computation time of the algorithms using the objective functions NL-OF and NL-OF(k) increases linearly with the size of the network which reflects the scalability of these algorithms.

CHAPTER 5

CONCLUSION AND PERSPECTIVES

This chapter concludes the thesis and presents several possible directions for future work.

5.1 Conclusion

At the beginning of the thesis we set out the following research problem.

Research problem

How do we find the multicast/unicast route subject to multiple constraints of QoS with the best use of the resources ?

This thesis deals with the QoS routing problems, which determine the QoS routes from the source to the destination(s). The examined optimization objective is to minimize the network's resources often expressed by a cost function.

To validate the thesis we provided three main contributions within the framework of QoS routing.

- Fast Multiple Constraints Routing Algorithm (FMCRA) (Chapter 2).
- ILP Formulation of the Exact Solution of Multi-Constrained Minimum Cost Multicast and an efficient pretreatment algorithm (Chapter 3).
- Multiple Constrained QoS Routing with RPL (Chapter 4).

In the first contribution, we first provided a study of Multi-Constrained Path problem (MCP) along side with the the widely-used algorithms in the literature on this subject, i.e. TAMCRA and SAMCRA. We also showed the main reason behind the big computation time of these algorithm. It appeared to be the number of calculated paths at each step before reaching the solution. Therefore, we developed an efficient algorithm called FMCRA that reduces the number of computed paths if possible while finding the exact solution.

FMCRA is validated with numerous experiments and is compared to the most known MCP algorithms in the literature (TAMCRA and SAMCRA). The performance of FMCRA was measured in term of: number of computed paths, execution time and the quality of paths. The experiments have shown that FMCRA outperforms SAMCRA on number of computed paths. When it comes to quality of paths, FMCRA outperforms TAMCRA, however it is less slightly efficient than SAMCRA. For the execution time TAMCRA is the best in this challenge due to the fact that TAMCRA is a heuristic algorithm with lower complexity.

In the second contribution, we have shown that the exact solution of the Multi-Constraints Multicast Minimum Cost problem (MCMCM) is not always a tree but a hierarchy. We have proposed an ILP as the first valid ILP formulation for the exact solution of MCMCM. Since MCMCM is NP-hard, we proposed the ArcReduce algorithm which provides a pretreatment step before the process of finding a solution by the ILP. The pretreatment step consists of eliminating the arcs that cannot be part of the feasible QoS multicast route. Therefore, ArcReduce accelerates the process and improve significantly the scalability of the ILP.

To evaluate our proposed ILP we have analyzed the optimal route based on the hierarchies in the solution in terms of:

- **Execution time:** as MCMCM is NP-hard we looked at the execution time of our ILP with different sizes of input data to show when the execution time is reasonable and when is not.
- **Proportion of optimal hierarchies:** an important advantage of our ILP is that it can provide the exact MCMCM solution whether it is tree or not. For this reason, we computed the proportion of instances that do not admit a tree as an optimal solution but a hierarchy to show the frequency of this phenomenon in real networks.

Evaluation of MAMCRA algorithm: since our ILP gives the exact solution, it can be used to evaluate the performance of the proposed heuristic algorithms.

In the third contribution we provided a new approach with regard to routing with QoS in Low-power and Lossy Networks (LLNs). The most known routing protocol of LLNs named RPL was proposed without a working mechanism that permits to take into account several constraints of QoS simultaneously. In the first part of the contribution, we have proposed a non-linear objective function (NL-OF) that can provide a routing map subject to an arbitrary number of constraints. In the second part of this contribution, we presented a theoretical study to analyze the exact and heuristic routing algorithms with QoS constraints in RPL. We have found that the proposed structure by RPL which is the Destination Oriented Directed Acyclic Graph (DODAG) is limited and not able to guarantee the satisfaction of multiple QoS constraint at once.

To investigate the efficiency of our NL-OF we have conducted an experimental evaluation comparing with three Objective Functions predefined in RPL in terms of:

- **End-to-End Delay:** we looked at the time taken for a packet to be transmitted across a network between two nodes.
- **Packet loss:** we looked at the ratio of lost packets over the total sent packets.
- **Jitter:** we looked at the delay variation over time.

We have shown, based on experimental evaluation using Cooja simulator under Contiki operating system that our OF has achieved an important improvement in the RPL-based network especially in term of QoS metrics.

In the second part of the contribution, we have evaluated each algorithm (i.e. exact and heuristic algorithm) in terms of: execution time, missing nodes and quality of paths.

5.2 Perspectives

Many challenging issues on QoS routing are not covered in the thesis. Several potential perspectives are suggested bellow for future researches.

5.2.1 Short term perspectives

Multicast problems with multiple sources (MMS) are worth looking into. In such problem, the bandwidth in a link will be shared between messages from different sources. Therefore, the final solution need not be a tree, and the hierarchies may be considered as a potential solution.

As proven in our thesis the optimal solution of Multi-Constrained Minimum Cost Multicast (MCMCM) problem is always a partial spanning hierarchies. Based on this concept, MMS can be thoroughly studied using our proposed algorithms with slight modifications. For instance, our ILP can be adapted to support MMS if we modify and add some constraints.

We can dived MMS into two sub-problems:

- Multiple sources for a single multicast group.
- Multiple sources for multiple multicast groups.

5.2.2 Mid-term perspectives

In this thesis, we have studied the QoS routing problems assuming that each link in the network is associated with static metrics. However, dynamic metric values are interesting for routing because the route can be recomputed based on prevailing traffic pattern to ensure certain QoS criteria (e.g. packet delay and packet loss) [10]. Our work can be adapted to such situation taking into account the state of each link over a period of time. This change would not effect the main core of the algorithm but rather how one would deal with dynamic metrics. In order to recompute the route constantly when the value of links are dynamic.

To propose more realistic and accurate algorithms, QoS routing should take into account the QoS constraints with resource considerations. The objective is to propose algorithms that find an optimal path between a source and destination node and computes the amount of resources needed at each node.

5.2.3 Long term perspectives

Since the metrics value of each link in the network can be variable and changing, links may only be valid for a certain period of time. In other words, a change on a specific QoS metric on a particular link can result in invalidating the solution (i.e. the feasibility of the route). In such context, the definition of the solution can become probabilistic in the sense that it is more interesting to find those routes that may stay valid as long as possible. Hence, a probability can be attributed to routes to indicate how likely the route will be valid in presence of such changes. Such probabilistic estimation would be mainly based on how likely a link can be subject to a change, (i.e. probability distribution).

BIBLIOGRAPHY

[1] *Contiki operating system. http://www.contiki-os.org/.*

[2] *Quality of Service Networking*, Cisco Systems. Cisco IOS 12.0 Quality of Service. Indianapolis: Cisco Press, 1998.

[3] A. ABDRABOU AND W. ZHUANG, *A position-based QoS routing scheme for UWB mobile ad hoc networks*, IEEE Journal on Selected Areas in Communications, 24 (2006), pp. 850–856.

[4] J. N. AL-KARAKI AND A. E. KAMAL, *Routing techniques in wireless sensor networks: a survey*, IEEE wireless communications, 11 (2004), pp. 6–28.

[5] B. BARAN AND R. SOSA, *A new approach for antnet routing*, in Computer Communications and Networks, 2000. Proceedings. Ninth International Conference on, IEEE, 2000, pp. 303–308.

[6] A. R. BASHANDY, E. K. CHONG, AND A. GHAFOOR, *Generalized quality-of-service routing with resource allocation*, IEEE Journal on Selected Areas in Communications, 23 (2005), pp. 450–463.

[7] A. BELLABAS, *Quality of Service Multicast Routing Subject to Multiple Constraints*, PhD thesis, INSA Rennes, 2011.

[8] R. BELLMAN, *On a routing problem*, Quarterly of Applied Mathematics, 16 (1958), pp. 87–90.

[9] S. BLAKE, D. BLACK, AND M. CARLSON, *An architecture for differentiated services*, IETF RFC 2475, (1998).

[10] J. CHEN, P. DRUSCHEL, AND D. SUBRAMANIAN, *A new approach to routing with dynamic metrics*, in INFOCOM'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, vol. 2, IEEE, 1999, pp. 661–670.

[11] S. CHEN AND K. NAHRSTEDT, *On finding multi-constrained paths*, in IEEE International Conference on Communications, 1998. ICC 98., vol. 2, Jun 1998, pp. 874–879.

[12] S. CHEN AND K. NAHRSTEDT, *Distributed quality-of-service routing in ad hoc networks*, IEEE Journal on Selected areas in Communications, 17 (1999), pp. 1488–1505.

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

[14] S. EHSAN AND B. HAMDAOUI, *A surevey on energy-efficient routing technique with QoS assurances for wireless multimedia sensor networks*, IEEE Communication Survey Tutorials, 14 (2012), pp. 265–278.

[15] D. EPPSTEIN, *Finding the k shortest paths*, SIAM Journal on Computing, 28 (1998), pp. 652–673.

[16] K. R. FALL AND W. R. STEVENS, *TCP/IP illustrated, volume 1: The protocols*, Addison-Wesley, 2011.

[17] G. FENG, *A multi-constrained multicast QoS routing algorithm*, Computer Communications, 29 (2006), pp. 1811–1822.

[18] O. GADDOUR, A. KOUBA, N. BACCOUR, AND M. ABID, *Qos-aware fuzzy logic objective function for the RPL routing protocol*, Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), (2014), pp. 365–372.

[19] M. R. GAREY AND D. S. JOHNSON, *Computers and intractability*, vol. 29, wh Freeman New York, 2002.

[20] O. GNAWALI AND P. LEVIS, *The ETX objective function for RPL*, Internet Draft, (2010).

[21] A. V. GOLDBERG, *An efficient implementation of a scaling minimum-cost flow algorithm*, Journal of Algorithms, 22 (1997), pp. 1–29.

[22] J. GOZDECKI, A. JAJSZCZYK, AND R. STANKIEWICZ, *Quality of service terminology in IP networks*, IEEE Communications Magazine, 41 (2003), pp. 153–159.

[23] Z. J. HAAS, M. R. PEARLMAN, AND P. SAMAR, *The zone routing protocol (ZRP) for ad hoc networks*, in IETF Internet Draft, 2002.

[24] S. HAKIMI, *Steiner's problem in graphs and its implications*, Networks, 1 (1971), pp. 113–133.

[25] L. HANZO AND R. TAFAZOLLI, *A survey of QoS routing solutions for mobile ad hoc networks*, IEEE Communications Surveys & Tutorials, 9 (2007), pp. 50–70.

[26] R. HASSIN, *Approximation schemes for the restricted shortest path problem*, Mathematics of Operations Research, 17 (1992), pp. 36–42.

[27] T. HE, J. A. STANKOVIC, C. LU, AND T. ABDELZAHER, *Speed: A stateless protocol for real-time communication in sensor networks*, in Distributed Computing Systems, 2003. Proceedings. 23rd International Conference on, IEEE, 2003, pp. 46–55.

[28] M. I. HENIG, *The shortest path problem with two objective functions*, European Journal of Operational Research, 25 (1986), pp. 281–291.

[29] R. D. S. HWANG, F. K. AND P. WINTER, *The Steiner tree problem*, Annals of Discrete Mathematics, 1992.

[30] P. JACQUET, P. MUHLETHALER, T. CLAUSEN, A. LAOUITI, A. QAYYUM, AND L. VIENNOT, *Optimized link state routing protocol for ad hoc networks*, in Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century. Proceedings. IEEE International, IEEE, 2001, pp. 62–68.

[31] J. M. JAFFE, *Algorithms for finding paths with multiple constraints*, Networks, 14 (1984), pp. 95–114.

[32] D. B. JOHNSON AND D. A. MALTZ, *Dynamic source routing in Ad Hoc wireless networks*, Mobile computing, (1996), pp. 153–181.

[33] D. B. JOHNSON, D. A. MALTZ, J. BROCH, ET AL., *Dsr: The dynamic source routing protocol for multi-hop wireless Ad Hoc Networks*, Ad hoc networking, 5 (2001), pp. 139–172.

[34] S. KAPOOR AND S. RAGHAVAN, *Improved multicast routing with delay and delay variation constraints*, in Global Telecommunications Conference, 2000. GLOBECOM '00. IEEE, vol. 1, 2000, pp. 476–480 vol.1.

[35] P. KARKAZIS, H. C. LELIGOU, L. SARAKIS, T. ZAHARIADIS, P. TRAKADAS, T. H. VELIVASSAKI, AND C. CAPSALIS, *Design of primary and composite routing metrics for RPL-compliant wireless sensor networks*, in 2012 International Conference on Telecommunications and Multimedia (TEMU), July 2012, pp. 13–18.

[36] R. M. KARP, *Reducibility among combinatorial problems*, in Complexity of Computer Computations, Springer, 1972, pp. 85–103.

[37] W. KHALLEF, S. DURAND, AND M. MOLNÁR, *Improved exact resolution of multi-constrained path problem*, in EURO: European Conference on Operational Research, 2015.

[38] W. KHALLEF, M. MOLNAR, A. BENSLIMANE, AND S. DURAND, *Multiple constrained QoS routing with RPL*, in 2017 IEEE International Conference on Communications (ICC), May 2017, pp. 1–6.

[39] W. KHALLEF, M. MOLNAR, A. BENSLIMANE, AND S. DURAND, *On the QoS Routing with RPL*, 2017 International Conference on Performance Evaluation and Modeling in Wired and Wireless Networks (PEMWN), (2017).

[40] V. P. KOMPELLA, J. C. PASQUALE, AND G. C. POLYZOS, *Multicast routing for multimedia communication*, IEEE/ACM Trans. Netw., 1 (1993), pp. 286–292.

[41] T. KORKMAZ AND M. KRUNZ, *Multi-constrained optimal path selection*, in Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No.01CH37213), vol. 2, 2001, pp. 834–843 vol.2.

[42] T. KORKMAZ AND M. KRUNZ, *A randomized algorithm for finding a path subject to multiple QoS requirements*, Computer Networks, 36 (2001), pp. 251–268.
Theme issue: Overlay Networks.

[43] D. KREUTZ, F. M. RAMOS, P. E. VERISSIMO, C. E. ROTHENBERG, S. AZODOLMOLKY, AND S. UHLIG, *Software-defined networking: A comprehensive survey*, Proceedings of the IEEE, 103 (2015), pp. 14–76.

[44] F. KUIPERS AND P. V. MIEGHEM, *Mamcra: a constrained-based multicast routing algorithm*, Computer Communications, 25 (2002), pp. 802 – 811.

[45] F. KUIPERS, P. VAN MIEGHEM, T. KORKMAZ, AND M. KRUNZ, *An overview of constraint-based path selection algorithms for QoS routing*, IEEE Communications Magazine, 40 (2002), pp. 50–55.

[46] F. A. KUIPERS, *Quality of Service Routing in the Internet: Theory, Complexity and Algorithms*, DUP Science Delft, 2004.

[47] F. A. KUIPERS AND P. VAN MIEGHEM, *QoS routing: Average complexity and hopcount in m dimensions*, in QofIS, Springer, 2001, pp. 110–126.

[48] P. LEVIS, N. PATEL, D. CULLER, AND S. SHENKER, *Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks*, in Proc. of the 1st USENIX/ACM Symp. on Networked Systems Design and Implementation, 2004.

[49] J. LIKLIDER, *Memorandum For Members and Affiliates of Intergalactic Computer Network*, 1963.

[50] C. R. LIN AND J.-S. LIU, *Qos routing in ad hoc wireless networks*, IEEE Journal on Selected Areas in Communications, 17 (1999), pp. 1426–1438.

[51] X. MASIP-BRUIN, M. YANNUZZI, J. DOMINGO-PASCUAL, A. FONTE, M. CURADO, E. MONTEIRO, F. KUIPERS, P. V. MIEGHEM, S. AVALLONE, G. VENTRE, P. ARANDA-GUTIERREZ, M. HOLLICK, R. STEINMETZ, L. IANNONE, AND K. SALAMATIAN, *Research challenges in QoS routing*, Computer Communications, 29 (2006), pp. 563–581.
Networks of Excellence.

[52] I. MATTA AND L. GUO, *QDMR: An efficient QoS dependent multicast routing algorithm*, Journal of Communications and Networks, 2 (2000), pp. 168–176.

[53] I. MATTA AND L. GUO, *Qdmr: An efficient qos dependent multicast routing algorithm*, Journal of Communications and Networks, 2 (2000), pp. 168–176.

[54] P. V. MIEGHEM, H. D. NEVE, AND F. KUIPERS, *Hop-by-hop quality of service routing*, Computer Networks, 37 (2001), pp. 407–423.

[55] M. MOLNAR, *hierarchies to solve constrained spanning problems*, tech.rep, LIRMM, University Montpellier 2, France, https://hal-lirmm.ccsd.cnrs.fr/lirmm-00619806, 2011.

[56] M. MOLNAR, A. BELLABAS, AND S. LAHOUD, *The cost optimal solution of the multi-constrained multicast routing problem*, Computer Networks, 56 (2012), pp. 3136 – 3149.
Challenges in High-Performance Switching and Routing in the Future Internet.

[57] H. D. NEVE AND P. V. MIEGHEM, *Tamcra: a tunable accuracy multiple constraints routing algorithm*, Computer Communications, 23 (2000), pp. 667–679.

[58] C. PERKINS, *Ad hoc on-demand distance vector (AODV) routing*, RFC 3561), (2003).

[59] C. E. PERKINS AND P. BHAGWAT, *Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers*, in ACM SIGCOMM Computer Communication Review, vol. 24, ACM, 1994, pp. 234–244.

[60] C. E. PERKINS AND E. M. ROYER, *Ad Hoc on-demand distance vector routing*, in Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA '99. Second IEEE Workshop on, Feb 1999, pp. 90–100.

[61] K. PISTER, P. THUBERT, S. DWARS, AND T. PHINNEY, *Industrial routing requirements in low-power and lossy networks*, tech. report, 2009.

[62] M. A. RAAYATPANAH, *Multicast routing based on data envelopment analysis with multiple quality of service parameters*, International Journal of Communication Systems, (2015).

[63] A. SHAI AND K. SHIN, *Destination-driven routing for low-cost multicast*, IEEE Journal on Selected Areas in Communications, 15 (1997), pp. 373–381.

[64] K. N. SHIGANG CHEN, *An overview of quality of service routing for next-generation high-speed networks: problems and solutions*, IEEE Network, 12 (1998), pp. 64–79.

[65] J. M. SMITH, D. T. LEE, AND J. S. LIEBMAN, *An o(n log n) heuristic for Steiner minimal tree problems on the euclidean metric*, Networks, 11 (1981), pp. 23–39.

[66] K. SOHRABI, J. GAO, V. AILAWADHI, AND G. J. POTTIE, *Protocols for self-organization of a wireless sensor network*, IEEE Personal Communications, 7 (2000), pp. 16–27.

[67] H. TAKAHASHI AND A. MATSUYAMA, *An approximate solution for the Steiner problem in graphs.*, Japonica, 24 (1980), pp. 573–577.

[68] Y. TANAKA AND P. HUANG, *Multiple destination routing algorithms*, IEICE Transaction on Communication, 1993.

[69] R. R. T.H. CORMEN, C.E. LEISERSON AND C. STEIN, *Introduction to Algorithms*, McGraw, 2001.

[70] P. THUBERT, *Objective function zero for the routing protocol for low power and lossy networks*, RFC 6552 and (Cisco Systems)), (2012).

[71] K. TSAI AND C. CHEN, *Two algorithms for multi constrained optimal multicast routing*, International Journal of Communication Systems, 16 (2003), pp. 951–973.

[72] T. TSVETKOV AND A. KLEIN, *Rpl: Ipv6 routing protocol for low power and lossy networks*, Network, 59 (2011).

[73] J. VASSEUR, *Terminology in Low power And Lossy Networks.*, draft-ietf-roll-terminology-11, (2013).

[74] J. VASSEUR, M. KIM, K. PISTER, N. DEJEAN, AND D. BARTHEL, *Routing metric used for path calculation in lowpower and lossy networks*, RFC 6551 and (Cisco Systems), (2012).

[75] R. VOGEL, R. G. HERRTWICH, W. KALFA, H. WITTIG, AND L. C. WOLF, *Qos-based routing of multimedia streams in computer networks*, IEEE Journal on Selected Areas in Communications, 14 (1996), pp. 1235–1244.

[76] M. WANG AND G.-S. KUO, *An application-aware QoS routing scheme with improved stability for multimedia applications in mobile ad hoc networks*, in VTC-2005-Fall. 2005 IEEE 62nd Vehicular Technology Conference, 2005., vol. 3, Sept 2005, pp. 1901–1905.

[77] Z. WANG AND J. CROWCROFT, *Quality-of-service routing for supporting multimedia applications*, IEEE Journal on Selected Areas in Communications, 14 (1996), pp. 1228–1234.

[78] Z. WANG AND J. CROWCROFT, *Quality-of-service routing for supporting multimedia applications*, IEEE Journal on selected Areas in Communications, 14 (1996), pp. 1228–1234.

[79] B. M. WAXMAN, *Routing of multipoint connections*, IEEE journal on Selected Areas in Communications, 6 (1988), pp. 1617–1622.

[80] P. WINTER, *Steiner problems in networks: a survey*, Networks, 17 (1987), pp. 129–167.

[81] T. WINTER, P. THUBERT, A. BRANDT, J. HUI, R. KELSEY, P. LEVIS, K. PISTER, R. STRUIK, J. VASSEUR, AND R. ALEXANDER, *Rpl: Ipv6 routing protocol for low-power and lossy networks*, RFC 6550 (Proposed standard), (2012).

[82] X. XIAO, *Technical, commercial and regulatory challenges of QoS: An internet service model perspective*, Morgan Kaufmann, 2008.

[83] G. XUE, W. ZHANG, J. TANG, AND K. THULASIRAMAN, *Polynomial time approximation algorithms for multi-constrained QoS routing*, IEEE/ACM Trans. Netw., 16 (2008), pp. 656–669.

[84] L. ZHANG, S. BERSON, S. HERZOG, AND S. JAMIN, *Resource reservation protocol (RSVP)–version 1 functional specification*, Resource, (1997).