



**HAL**  
open science

## mQMA multi-constrained QoS Multicast Aggregation

Naouel Ben Ali, Joanna Moulhierac, Abdelfattah Belghith, Miklos Molnar

► **To cite this version:**

Naouel Ben Ali, Joanna Moulhierac, Abdelfattah Belghith, Miklos Molnar. mQMA multi-constrained QoS Multicast Aggregation. [Research Report] PI 1881, 2008. inria-00258515v1

**HAL Id: inria-00258515**

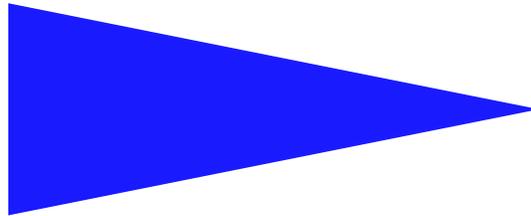
**<https://inria.hal.science/inria-00258515v1>**

Submitted on 25 Feb 2008 (v1), last revised 6 Mar 2008 (v2)

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

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

PUBLICATION  
INTERNE  
N° 1881



MQMA: MULTI-CONSTRAINED QOS MULTICAST  
AGGREGATION

NAOUEL BEN ALI, JOANNA MOULIERAC ,  
ABDELFAH BELLGHITH AND MIKLÓS MOLNÁR



## mQMA: multi-constrained QoS Multicast Aggregation

Naouel Ben Ali<sup>\*</sup>, Joanna Moulrierac<sup>\*\*</sup>, Abdelfattah Belghith<sup>\*\*\*</sup> and  
Miklós Molnár<sup>\*\*\*\*</sup>

Systèmes communicants  
Projet Armor2

Publication interne n° 1881 — Février 2008 — 32 pages

### Abstract:

Traditional IP Multicast has been proposed in order to manage group communications over the Internet in a bandwidth efficient manner. Although this proposition has been well studied during the last decade, there are still some problems for its deployment. In this paper, we propose a new algorithm mQMA that deals with two important problems of traditional IP multicast, *i.e.*, multicast forwarding state scalability and Quality of Service routing. The algorithm mQMA deals with these two main problems while utilizing the technique of multicast aggregation in case of multi-constrained QoS. With this technique of aggregation, several groups can share the same tree respecting the same QoS requirements. The algorithm mQMA builds trees satisfying multiple QoS constraints. We show, during the simulations, that mQMA reaches the same QoS performances as Mamcra which is the main multi-constrained multicast routing algorithm. Moreover, mQMA reduces dramatically the number of trees to be maintained and the number of multicast forwarding states.

**Key-words:** Network, multicast routing, QoS, multi-constrained optimization, aggregation

(Résumé : *tsvp*)

Ce travail a été réalisé dans le cadre d'une collaboration entre l'Irisa de Rennes et l'ENSI de Tunis et a été financé par le programme INRIA/STIC - Universités Tunisiennes.

\* [naouel.benali@ensi.rnu.tn](mailto:naouel.benali@ensi.rnu.tn)  
\*\* [joanna.moulrierac@sophia.inria.fr](mailto:joanna.moulrierac@sophia.inria.fr)  
\*\*\* [abdelfattah.belghith@ensi.rnu.tn](mailto:abdelfattah.belghith@ensi.rnu.tn)  
\*\*\*\* [miklos.molnar@irisa.fr](mailto:miklos.molnar@irisa.fr)

## Agrégation multicast avec plusieurs contraintes de Qualité de service

**Résumé :** IP multicast tel qu'il a été proposé a pour objectif d'offrir un moyen pour le déploiement communications de groupes tout en gérant efficacement les ressources du réseau. Malgré que cette proposition a été longtemps étudiée, son déploiement est encore à l'état embryonnaire. Dans ce rapport, nous proposons un nouveau algorithme mQMA qui s'intéresse à deux parmi les plus importants problèmes de l'IP multicast qui sont le problème du routage multicast multicritère et le problème du scalabilité induit par le maintien des états d'acheminement. mQMA permet de mettre en place peu d'arbres et nécessite le maintien de peu d'état grâce à la technique d'agrégation d'arbres. Ceci permet à plusieurs groupes multicast de partager un même arbre d'acheminement. Par ailleurs, mQMA construit des arbres respectant plusieurs contraintes de QoS exigés par les destinations. Les simulations montrent que mQMA est proche de l'optimalité en terme de contraintes QoS. mQMA permet aussi de réduire considérablement le nombre d'arbres à mettre en place au niveau du réseau et l'utilisation des ressources réseaux.

**Mots clés :** Réseaux, routage multicast, QoS, optimisation multicritère, agrégation

## 1 Introduction

Recently, the Internet has shown a tremendous growth. Emergent multimedia applications like audio/video conferencing, video on demand, IP-telephony usually have other requirements than traditional Internet services such as e-mail and file transfer services. Moreover, these applications may involve a set of users and not only a sender and a receiver.

IP multicasting techniques have been proposed to support group communications over the Internet with the aim of reducing the network resource consumption. Although IP multicast [9] has been studied for a long time, it is not totally well deployed over the Internet. The multicast forwarding state scalability problem and the lack of QoS support can be considered among reasons for IP multicast not to be widely deployed. In traditional IP multicast, data for the group is forwarded through a tree structure covering the members of the group. The on-tree routers must maintain a per-group forwarding state. With the evolution of the Internet applications, the number of groups considerably grows, increasing the number of forwarding states. Large forwarding tables induce large memory requirements and slow the address look-up process. In unicast, the aggregation of the IP addresses has been possible as the addresses are given in a hierarchic manner. In IP multicast, this aggregation is much more difficult as multicast addresses do not convey any information of members location. Moreover, maintaining multicast trees requires an exchange of control messages causing an important overhead as the number of multicast groups grows. This represents the problem of forwarding state scalability problem [10, 2]. This problem may be solve before a large number of groups can communicate.

Nowadays, Internet applications requires QoS aware environment. The best-effort functioning provided by IP based network does not satisfy application needs. In addition to traffic engineering mechanisms, QoS oriented routing algorithms are needed to meet QoS sensitive applications requirements.

Multicast tree aggregation [6, 13] is a recent approach that deals with the problem of multicast forwarding state scalability. In this approach, multiple groups share the same delivery tree within a domain whereas in traditional IP multicast, a tree is built and maintained for each group. With the multicast tree aggregation, fewer trees are maintained and then this approach reduces both the number of forwarding states in routers and the overhead induced by control messages. Some information has to be added in the border routers of the domain in order to multiplex the data for the groups onto an aggregated tree. Several algorithms have been proposed to perform tree aggregation: AM [6] (Aggregated Multicast) and STA [13] (Scalable Tree Aggregation) are the more simple BEAM [8] (Bi-dirEctional Aggregated Multicast) deals with distributed multicast tree aggregation, AMBTS [18] (Aggregated Multicast Based on Tree Splitting) and TALD [21] (Tree Aggregation in Large Domains) achieve good aggregation ratio in large domains where traditional multicast aggregation algorithms fail.

However, these proposed algorithms aggregate multicast groups without considering any QoS requirements. Aggregation is based on trees computed by IP multicast routing protocols which use the shortest path tree algorithm optimized on one single metric, typically the hop count. Two solutions have been proposed in the literature to deal with QoS multicast

aggregation: AQoSM [7] (Aggregated QoS Multicast) and Q-STA [20] (QoS Multicast Tree Aggregation). The goal of these two algorithms is to aggregate groups to trees while respecting bandwidth requirements of groups depending on the multimedia application. However, multicast applications today need to satisfy more than one or two QoS criteria that's why multiple QoS multicast aggregation is needed. Tree shared by multicast members must fulfill flow requirements such as delay, bandwidth, variation in the delay experienced by the receiver (also known as jitter), packet loss that can be tolerated and/or number of hops... To achieve QoS multicast aggregation under several QoS constraints, the forwarding structure must be computed by multi-constrained multicast routing algorithms. In some cases, this structure is not a tree and it may contain some cycles. Most proposed algorithms perform single or dual QoS multicast routing [4]. For the multi-constrained multicast routing problem which involves multiple QoS metrics, few algorithms has been proposed due to the complexity nature of the problem.

In this paper, we propose a new algorithm called mQMA (multi-constrained QoS Multicast Aggregation) which performs multicast aggregation taking into account multiple QoS requirements. To the best of our knowledge, it is the first proposed protocol that deals with these two main problems of multicast forwarding state scalability and of multi-constrained multicast routing. Our protocol mQMA is based on two principles. Firstly, the technique of multicast tree aggregation reduces the number of trees to be maintained and secondly mQMA builds trees satisfying several additive constraints.

To achieve multi-constrained QoS multicast aggregation, it is mandatory to present previous work that dealt with multicast tree aggregation and multi-constrained multicast routing.

This paper is organized as follows: section 2.2 presents previous work that dealt with these two fields. Section 3 details our proposed algorithm mQMA in three main steps. Section 4 shows how mQMA reduces the forwarding structure by eliminating as more redundancies as possible. Section 5 presents and analyzes the simulation results. Finally, we conclude our work in section 6.

## 2 Related work

This section describes the related work for two important domains related to multicast. Firstly, multicast aggregation is detailed and then multi-constrained multicast QoS routing.

### 2.1 Multicast Tree Aggregation

#### 2.1.1 Multicast Tree Aggregation Principles

Multicast tree aggregation is proposed as a solution to the multicast forwarding state scalability problem. It is a multicast scheme which forces multicast groups to share the same delivery tree within a domain. Data packets for different groups are delivered via the same distribution tree called aggregated tree. This induces less tree construction within domains and so less forwarding states to be stored. This way, routers within a domain need to keep state only per aggregated tree and not per multicast group. This protocol can replace an intra-domain multicast routing protocol.

Routing the messages on an aggregated tree cannot be no longer performed using the original group multicast IP address. Each aggregated tree is assigned a label and the edge routers maintain the group-label matching in a group-label table. This label represents either a MPLS label or a multicast address of a group which is not active in the domain : a virtual multicast address. Indeed, if multicast aggregation is deployed in a MPLS domain, MPLS labels are used and distributed with LDP (Label distribution protocol). With IP encapsulation, the packets are forwarded with a multicast routing protocol according to the virtual multicast address added to the packet.

To summarize, the edge router, at the domain entry, pushes a label into incoming multicast packets and keeps a group-label table. Within the domain, core routers uses only this label to route the packet and only maintain forwarding states for these labels. When leaving the domain, the edge router removes the label and the packet is forwarded as any other packet. This label based mechanism is depicted in Figure 1.

**Different types of group-tree matching.** In aggregated multicast, groups must be matched to aggregated trees. The group-tree matching can be done differently as defined in [19]. The matching is perfect for a group if all the tree covers exactly the group members and the tree cost (in terms of edge number) is no more than the native tree. It may also be leaky, if there are tree leaves that are not attached to group members. In that case, bandwidth is used to deliver data to nodes that are not in the multicast group but this bandwidth loss is usually not high. It may also be an incomplete match if the tree doesn't cover all the members of the group and it may be an incomplete leaky match if there are tree leaves that are not attached to members and moreover, there are some members not covered by the aggregated tree. All these aggregation types are illustrated in Figure 2. The choice of the aggregation type depends on what will be privileged bandwidth utilization or number of entries in routers. Perfect aggregation reduces the number of states without any overhead in terms of bandwidth. Adopting a leaky match aggregation with a bandwidth

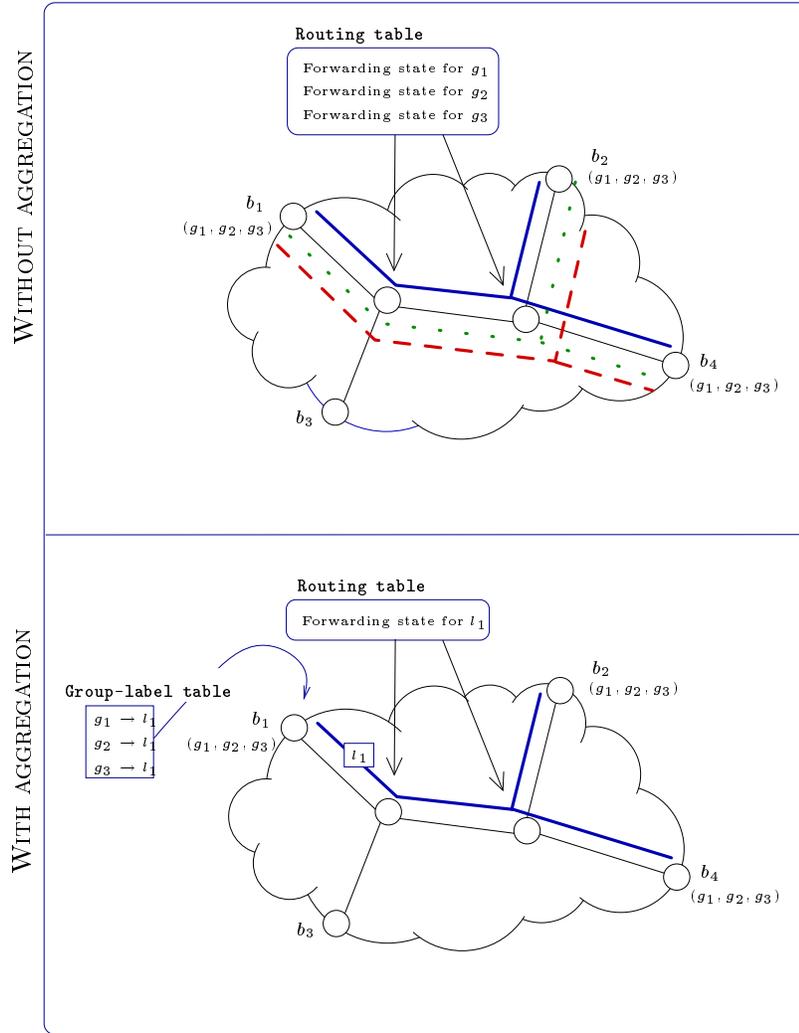


Figure 1: The groups  $g_1$ ,  $g_2$  et  $g_3$  share the same tree of label  $l_1$ .

threshold reduces more the number of trees. In the extreme case, if the bandwidth threshold is set to infinity, only one tree is needed for all the multicast groups.

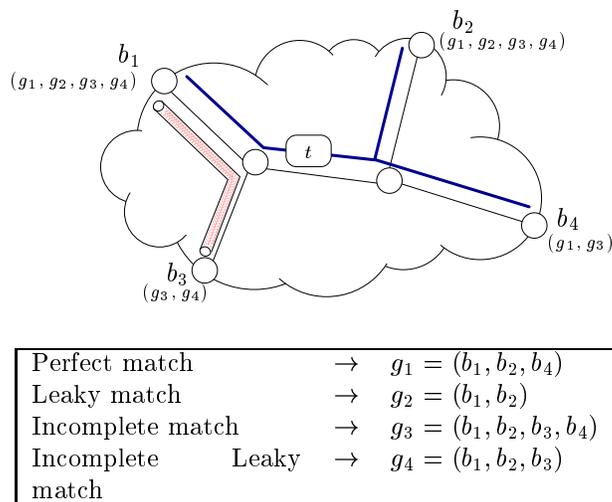


Figure 2: Different types of aggregation

### 2.1.2 Multicast Tree Aggregation proposals

Several protocols have been proposed dealing with multicast tree aggregation.

**Centralized multicast aggregation.** The first protocol is Aggregated Multicast (AM) and is proposed in [5] and in [6]. In AM, a centralized entity called tree manager performs the group-tree matching by assigning labels to groups. This tree manager stores the information concerning the group memberships and the trees maintained in the domain. Whenever a multicast group arrives, the edge router contacts the centralized entity which is in charge of finding an adequate tree for the given group. In this phase, if the tree manager finds a tree respecting the bandwidth assumed threshold, then the aggregation is performed. If not, the tree manager configures a new tree for the group. This new tree will be a candidate for further aggregations. In all the cases, the tree manager informs the border router of the label found. Then, the edge router is able to route packets for the group onto the aggregated tree. To achieve aggregation, the AM algorithm estimates all the trees currently configured into the domain. This process slows down the aggregation particularly when the number of tree configured is important and when the multicast groups change frequently inducing many queries to the tree manager. The protocol Scalable Tree Aggregation (STA) is proposed in [13] to speed up the aggregation algorithm performed by the tree manager. The ideas developed are a fast selection function based on an efficient sorting of the trees and a low storage of the trees. Unlike AM, the algorithm STA evaluates a sub-set of the currently configured trees. This is made possible by sorting the configured trees according

to their cost. In this way, when a group arrives, the tree manager evaluates only trees having costs close to the group native tree (the tree computed by the multicast routing algorithm deployed into the domain). To speed up the trees selection process and to reduce the trees storage, each configured tree is represented as a bitmap. Each bit of the bitmap corresponds to a domain router. When the router is within the tree, the corresponding bit in the bitmap is set to 1 otherwise it is set to 0. The bitmap mechanism reduce the tree storage cost and facilitates finding trees covering multicast groups to achieve aggregation.

**Distributed multicast tree aggregation.** Achieving aggregation in a centralized way raises many problems. When the tree manager fails new groups can no longer be aggregated and control information sent by edge routers concerning the currently aggregated multicast groups may be lost. Moreover, the centralized approach may induce an overload on links taking to the tree managers increasing congestion probability of these links. To face all these problems, the authors in BEAM [8] distribute the task of the tree manager among several routers, called core routers. Each core router has a local view of the configured tree in the domain. This protocol requires control messages between the core routers in order to perform the aggregation. Indeed, when a core router does not find an adequate tree for a group, it requests the other core routers for a possible aggregation.

**Multicast aggregation in large domains.** The protocol AMBTS [18] splits the native tree in sub-trees before aggregating, while the protocol TALD [21] performs tree aggregation in large domains by splitting the network in several sub-domains and by managing the sub-domains independently. This enhances the aggregation ratio as aggregating sub-groups to sub-trees is easier than aggregating the whole tree. AMBTS and TALD can be applied in large domains, where other approaches fail and perform almost no aggregation for the groups. All these protocols focus rather on the aggregation ratio with no regards to the bandwidth capacity.

**Multicast aggregation with QoS constraints.** AQoSM [7] and Q-STA [20] achieves aggregation under bandwidth constraints; namely when links have different limited bandwidths and groups are served by sensitive bandwidth applications. In these protocols, the tree manager maintains a table which indicates the bandwidths available on the different network links and which is updated whenever a multicast group arrives or leaves the domain. While AQoSM tests several sources for the native trees in order to build one that can accept the group according to the requested bandwidth, Q-STA builds native trees maximizing the bandwidth available on the links in order to achieve load balancing and to use in priority the links that are not heavily loaded.

## 2.2 Multi-constrained Multicast Routing

The problem of QoS routing, even in the unicast case, is known to be NP-complete and has been extensively studied by the research community [24]. Before presenting the mechanisms used to construct multi-constrained QoS multicast delivery structure, we must first specify some hypothesis used to solve these problems and the notation used throughout this paper. Then we present existing approaches to achieve multicast QoS routing.

### 2.2.1 Hypothesis

QoS routing approaches assume that the network-state information is temporarily static and has been distributed throughout the network. This network-state information is accurately maintained at each node and we assume that this information can be collected by any appropriate traffic engineering mechanism. The QoS metrics are categorized into additive (e.g., delay, jitter,..) and bottleneck (min or max) metrics (e.g. bandwidth). The constraints on min (resp. max) QoS measures can easily be treated by pruning all links (and possibly disconnected nodes) which do not satisfy the requested min (resp. max) QoS constraints. In contrast, constraints on additive QoS measures cause more difficulties. Hence, without loss of generality, all QoS measures are assumed to be additive.

A network topology is modeled as an undirected graph  $G = (V, E)$ , where  $V$  is the set nodes and  $E$  is the set of links. Each link is characterized by  $m$  additive QoS metrics. So we associate to each link an  $m$ -dimensional link weight vector of  $m$  non-negative QoS weights ( $w_i^j$ , for  $i = 1, 2, \dots, m$  and  $j$  is a link in  $E$ ). The  $m$  QoS constraints (limits)  $L_i$  for  $i = 1, 2, \dots, m$  are represented by the constraint vector  $\vec{L}$ . A multicast group  $g$  is composed of a source  $s$  and a set of members  $D = \{d_1, d_2, \dots, d_k\}$  where  $k$  is the number of multicast members.

### 2.2.2 Multi-constrained multicast routing proposals

For the multicast case, a number of QoS routing algorithms based on single, dual and multiple metrics have been proposed. Single metric QoS multicast routing algorithms have been proposed for cost [26, 4, 25] and delay [11, 15]. Dual metric based routing algorithms have been proposed for the following combinations: cost-delay [16, 14] and delay-jitter [22, 25]. Few works have dealt with multiple QoS metrics. One way to tackle the multicast QoS routing problem is to compute a set of unicast paths from the source to the multicast members using a unicast QoS routing algorithm. Then, the obtained sub-graph is reduced to optimize network utilization without violating the constraints. This strategy is adopted by the Multicast Adaptive Multiple Constraints Routing Algorithm (Mamcra) which is proposed by [17] as the multicast extension of Samcra, a unicast QoS routing algorithm [23] and the Taboo based QoS Multicast Routing algorithm (Taboo-QMR) [3].

Mamcra proceeds in two phases: path computation and path reduction. In the first phase, Mamcra computes multi-constrained shortest paths from the source node to each destination using Samcra, the QoS unicast routing algorithm (detailed in appendix). In the second phase, the set of obtained paths is then reduced to obtain a multicast sub-graph that eliminates as many cycles from the first paths set as possible. The aim of this reduction phase is to minimize the number of duplicated packets in links and to have a structure that approximates a tree structure. Mamcra proposes a heuristic approach based on a greedy algorithm to solve this problem which is efficient but the quality of the approximation isn't proved. In fact, the QoS multicast routing graph structure computed in these two steps may not be a tree.

In [3], the authors propose Taboo-QMR, a QoS multicast routing algorithm where they adopt the same strategy to achieve the first phase but allows using any unicast QoS routing algorithm during this step and they propose improvements to the reduction phase of Mamcra by adopting a meta-heuristic approach based on taboo search algorithm to provide a sub-optimal solution.

### 3 The mQMA algorithm

In this section, we describe our proposed mQMA algorithm to achieve multi-constrained multicast routing. In addition of hypothesis and notations specified in the previous section, we need to present more definitions before depicting the mQMA algorithm.

#### 3.1 Definitions

Given a graph  $G=(V,E)$ , a path  $p(s,d)=(P_v,P_e)$  where  $P_v \subset V$  and  $P_e \subset E$  are the sets of vertices and edges of the graph  $G$  connecting node  $s$  to node  $d$ . Consider a tree,  $T=(T_v,T_e)$  where  $T_v$  is the set of vertices included in  $V$  and  $T_e$  is a set of edges included in  $E$ . Consider a multicast group  $g$ , as a set of nodes of  $G$ , so  $g \subseteq V$ .

**Definition 1:** A path  $P = (P_v, P_e)$  can be grafted on a tree  $T$ , if the graph  $U=(U_v, U_e)$  is a tree where  $U_v = P_v \cup T_v$  and  $U_e = P_e \cup T_e$ . We note  $U=P \cup_g T$ .

**Definition 2:** The set  $S=g \cap T$  denotes the nodes that are both in  $g$  and in  $T$ .

**Definition 3:** Given and a set of  $i$  additive constraints  $\vec{L} = (L_1, L_2, \dots, L_i)$ , the Feasible Paths Set  $FPS$  refers to the set of paths from source node to the multicast destination nodes that are within constraints  $L_i$ .

#### 3.2 Algorithm mechanisms

The algorithm mQMA achieves multicast tree aggregation in a centralized way: a tree manager maintains a Multicast Tree Set ( $MTS$ ) which contains all the multicast trees configured into the domain. The algorithm mQMA proceeds in three main steps described below. When a new multicast group  $g$  arrives, the tree manager is contacted by the router that has received the IGMP message and after executing the first two steps, namely the computing path step and the tree decomposition, the tree manager tackles the aggregation step. In mQMA, tree aggregation may be achieved when a perfect or a leaky match exists, as it is defined in section 2.1.1. When the leaky match is used, the bandwidth loss threshold must be specified. In the following, we detail each of the three steps of the algorithm mQMA.

**Step (1): Computing paths.** The aim of this step is to compute multi-constrained paths from the source node to each destination node of the group. To perform this step, any unicast QoS routing algorithms can be used. For example, the algorithm Samcra<sup>1</sup> [23] can be used. This first step is mandatory to determine the multi-constrained Feasible Paths Set ( $FPS$ ) for a given group. The quality of the paths in  $FPS$  depends on the unicast routing algorithm used. If Samcra is used, feasible paths from source node to each destination node are found if they exist. Samcra algorithm chooses the optimal path among all paths from the source to the destination according to the objective function used by Samcra. If no path is found for a destination, then it is not included in the group as QoS requirements cannot be fulfilled. In this case, the  $FPS$  contains no path for this destination.

<sup>1</sup>Description of Samcra's mechanisms together with the objective function denoted Samcra length are presented in appendix.

**Step (2): Tree decomposition.** The *FPS* obtained in the first step isn't necessarily a tree and as it is explained in section 2.2, it may contain cycles. The aim of this step is to decompose the *FPS* into several trees. The obtained trees will form the Feasible Trees Set (*FTS*). If the *FPS* contains no cycle, the *FTS* contains only one tree:  $FTS = FPS$  and the algorithm goes directly to step 3. Note that if the same link is present in several *FPS*, it is kept only once in the *FTS*. In the worst case, the *FTS* contains  $|FPS|$  trees, where  $|FPS|$  denotes the number of paths in *FPS*. If the *FPS* contains cycles, the tree decomposition works in a greedy manner by attempting to graft iteratively the paths for the members onto a tree of the *FTS*. If during an iteration, the graft of a path implies a cycle in all trees of the *FTS*, then this path is added to the *FTS* as an another tree. In this paper and in the simulations, paths of the *FPS* were arranged according to the number of members. So, the path covering the most number of members is considered as the first tree in the *FTS*. Another issue that can be investigated is the choice of the tree of the *FTS* to be used to graft paths when the path can be grafted on more than one tree of the *FTS*. In this paper, the path is grafted on the first examined tree.

**Step (3): Aggregation.** After executing the first two steps, the tree manager attempts to achieve a global aggregation of the whole multicast group. It consists on finding from the *MTS* a multicast tree  $MT_i$  that covers all the members covered by the *FPS* without violating the constraints and with cost (in terms of number of links) at most the cost of all the trees (if several) in *FTS*. If  $MT_i$  exists, it is used by the multicast group  $g$  and all data for  $g$  are multiplexed onto that tree. If  $MT_i$  doesn't exist and if the *FTS* contains only one tree, then the aggregation of  $g$  isn't possible and the tree in *FTS* must be added to the multicast tree set *MTS*. If the *FTS* contains more than one tree and if the global aggregation isn't possible, mQMA proceeds by a partial aggregation. Indeed, each tree  $FT_i$  ( $1 \leq i \leq |FTS|$ ) of the *FTS* covers a sub-group  $g_i$  of the initial multicast group  $g$ . The algorithm mQMA attempts to aggregate all the obtained sub-groups  $g_i$  and if the aggregation is not possible, the corresponding tree  $FT_i$  is added to the multicast tree set (*MTS*) until all the members of  $g$  are satisfied. In this case, a multicast group is aggregated to a set of trees and the tree manager is in charge to add information that allows routers to multiplex data onto these trees.

Algorithm 1 describes formally mQMA during the two first steps while Algorithm 2 describes the third step of mQMA.

---

**Algorithm 1** The multi-constrained QoS Multicast Aggregation protocol (mQMA):  
step (1) and step (2)

---

**Input:** The network  $G = (V, E)$ , a group  $g = (d_1, d_2, \dots, d_{|g|})$  with a source  $s$ , constraints vector  $\vec{L}$ .

**Output:** A set of trees  $FTS$  covering, if possible, all the members of  $g$ .

### Step (1): Computing paths

```

FPS ← ∅;
For i from 1 to |g| do
    Compute the (optimal) path  $P_i = p(s, d_i)$  considering a unicast QoS routing
    algorithm;
    Add  $P_i$  in FPS;
end For

```

### Step (2): Tree Decomposition

```

FTS ← ∅;
If (FPS contains no cycles) then
    | FTS = FPS;
else
    Arrange the FPS according to the number of members covered by each path;
    For i from 1 to |FPS| do
         $P_i$  is the first path in FPS;
        If ( $P_i$  contains members not yet in FTS) then
            Grafted = false; j = 1;
            While (Grafted is false and ( $j \leq |FTS|$ )) do
                If ( $FT_j \cup_g P_i$  doesn't form cycle) then
                    |  $FT_j = P_i \cup_g FT_j$ ;
                    | Grafted = true;
                end If
                j ++;
            done
            If (Grafted is false) then
                | Add  $P_i$  to FTS as  $FT_{|FTS|+1}$ ;
            end If
        end If
    end For
end If

```

---

---

**Algorithm 2** Step (3): Aggregation
 

---

**Input:** The network  $G = (V, E)$ , a group  $g$  with source  $s$ , the constraint vector  $\vec{L}$ , the bandwidth threshold  $t_b$ , the current multicast tree set  $MTS$ , the  $FTS$  computed in the step 2.

**Output:** The updated multicast tree set  $MTS$ .

**Step (3): Aggregation**

```

candidates  $\leftarrow \emptyset$ ;
For  $i$  from 1 to  $|MTS|$  do
  If ( $MT_i$  can cover  $g$  and fulfills the constraints and  $\text{cost}(MT_i) -$ 
     $\sum_{1 \leq k \leq |FTS|} \text{cost}(FT_k) \leq \sum_{1 \leq k \leq |FTS|} \text{cost}(FT_k) \times t_b$ ) then
    | Add  $MT_i$  to candidates;
  end If
end For
If (candidates  $\neq \emptyset$ ) then
  | Aggregate  $g$  to the tree  $t$  among the candidates of minimum cost;
else
  If ( $|FTS| == 1$ ) then
  | Add  $FT_1$  to the  $MTS$ ;
  else
  For  $i$  from 1 to  $|FTS|$  do
    Let  $g_i = FT_i \cap g$  the subset members covered by tree  $FT_i$ ;
    For  $j$  from 1 to  $|MTS|$  do
      If ( $MT_j$  can cover  $g_i$  and fulfills the constraints and
         $\text{cost}(MT_j) - \text{cost}(FT_i) \leq \text{cost}(FT_i) \times t_b$ ) then
        | Add  $MT_j$  to candidates;
      end If
    end For
    If (candidates  $\neq \emptyset$ ) then
    | Aggregate  $g_i$  to the tree  $t$  among the candidates of minimum
      cost;
    else
    | Add  $FT_i$  in  $MTS$ 
    end If
  end For
end If
end If

```

---

### 3.3 mQMA on an example

Let's consider the topology presented in (Figure 3) composed of 7 nodes and 7 links where  $s$  is the source node and nodes  $d_1$  and  $d_2$  members of the multicast group. Two additive metrics are considered and the constraint limits are fixed to  $(6, 6)$ .

**Step (1) : computing paths.** Execution of the first step of mQMA provides the set  $FPS$  of shortest paths to  $d_1$  and  $d_2$ . Here  $FPS = \{(s-b-c-e-d_1), (s-a-c-e-d_2)\}$ . To compute these paths we adopt for this example Samcra algorithm but any other unicast QoS routing can be used.

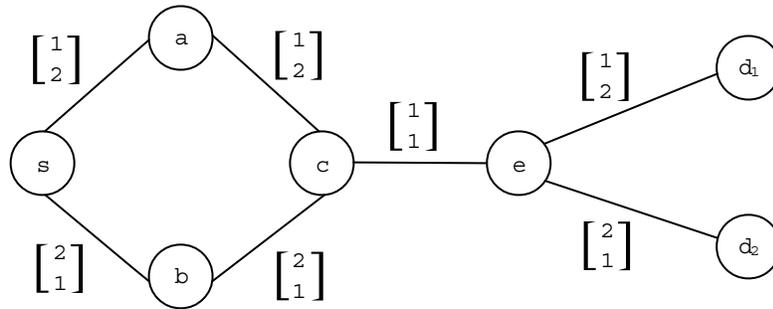


Figure 3: If the constraints are set to  $(7,7)$ , the routing structure for the group of source  $s$  and members  $(d_1, d_2)$  contains cycles.

**Step (2) : tree decomposition.** During the second step of mQMA, the graph structure obtained as the union of paths in the  $FPS$  is decomposed into trees indeed, the  $FPS$  contains one cycle. The tree decomposition step gives the set  $FTS$  composed of two trees,  $FTS = \{FT_1 = \{(s-b-c-e-d_1)\}, FT_2 = \{(s-a-c-e-d_2)\}\}$ .

**Step (3) : aggregation step.** The last step of mQMA, aims to find trees in the  $MTS$ , the multicast tree set, configured into the domain covering members and fulfilling constraints. The global aggregation attempts to find one multicast tree MT to aggregate the entire group. If, it is not possible, mQMA attempts to aggregate the sub-groups derived from the  $FTS_i$ . During the aggregating step and according to the already accepted and configured groups, feasible but not optimal tree, covering the new coming group, may exist in the  $MTS$  and may be used to achieve global aggregation.

In the example, if the constraint limits are fixed to  $(7, 7)$ , the  $MTS$ , may contain the tree  $MT_i = \{(s-b-c-e-d_1), (s-b-c-e-d_2)\}$ . This tree is feasible so can be used to aggregate the group  $(d_1, d_2)$ . If global aggregation is not possible, partial aggregation is executed in order to find in the  $MTS$ , two trees on which  $g_1 = (s, d_1)$  and  $g_2 = (s, d_2)$  can be aggregated.

If the *MTS* does not contain any tree that can be used to aggregate one of *FTS* trees, then the considered tree is added to the *MTS*. So, to recapitulate, for this example, four scenarios are possible:

- The *MTS* contains one tree covering member groups and fulfilling constraint limits;
- if not, the *MTS* contains two trees covering the sub-groups  $g_1$  and  $g_2$ ;
- if not, the *MTS* contains one tree that covers one of the two sub-groups, so the  $FT_i$  of the other sub-group must be added to the *MTS* and must be configured into the domain;
- if the two sub-groups can not be aggregated to any of the *MTS* trees, the two trees are added to the *MTS* and must be configured.

## 4 mQMA: a proposal to enhance multi-constrained routing structure

mQMA algorithm performs multicast aggregation taking into account multiple members requirements. In this section, we detail the cycles problem and how mQMA contributes in the reduction of forwarding structure.

### 4.1 Problem statement

Multicast routing aims to forward multicast data efficiently by sending single packets through the shared links and duplicating them if it is necessary through a tree structure. When multi-constrained multicast routing is considered, this philosophy may be no longer respected. Indeed, multi-constrained multicast routing computes a routing structure that satisfies multiple additive QoS metrics and this structure is not necessarily a tree. Not to affect the multicast philosophy, multi-constrained proposal takes into account the reduction of the computed routing structure in order to eliminate as many redundancies as possible without violating the members requirements.

**Cycles problem on an example:** Figure 4 illustrates the cycle problem. If the constraint limits are (8, 8), optimal paths ( $FP_1$  and  $FP_2$ ) for  $d_1$  and  $d_2$  form the cycle ( $s-a-d_0-b-s$ ). When rerouting data for  $d_2$  through  $FP_1$  or data for  $d_1$  through  $FP_2$ , the cycle ( $s-a-d_0-b-s$ ) is removed leading to a better use of network resources. When the constraint limits are fixed to (6,6). The path ( $s-b-d_0-e-d_1$ ) does not respect the constraints for  $d_1$  and the path ( $s-a-d_0-e-d_2$ ) does not respect the constraints for  $d_2$ . Consequently, the structure used for the group of source  $s$  and of members  $d_1$  and  $d_2$  contains the cycle ( $s-a-d_0-b-s$ ) which can not be removed. In this case,  $d_0$  receives packets twice, once from  $a$  and once from  $b$  and the link ( $d_0-e$ ) transfers duplicated packets. A specific routing has to be in place as traditional IP routing cannot support that case.

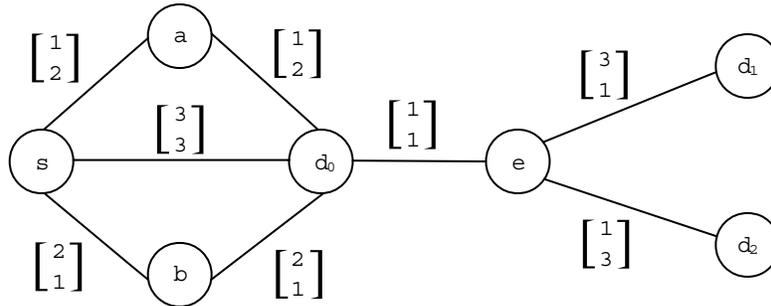


Figure 4: Network topology used to illustrate different mQMA mechanisms to handle cycles

## 4.2 mQMA contribution to solve cycles problem

Besides the reduction of routing entries, the algorithm mQMA can be considered as an alternative to solve the cycles problem. In addition to the aggregation benefit, the routing structure used by mQMA to deliver multi-constrained multicast data has been implicitly reduced.

mQMA removes cycles during the two last steps. (1) During the tree decomposition, some cycles can be removed. Indeed, the *FTS* grows in a greedy manner with the add of paths. When a path is added, it may cover some members not already considered. In that case, and if the path is feasible for these intermediate members, it will be used also to forward data to these members and the shortest paths for these members won't be considered at all and will be dropped. If these dropped paths formed some cycles, then with tree decomposition, these cycles are removed. (2) During the phase of aggregation, one global tree that is feasible and that covers the members can exist even if the structure *FTS* contains more than one tree. In that case, the cycles are removed from the group quite easily as global aggregation is made.

To apprehend how mQMA eliminates cycles, let us consider Figure 4 where  $s$  is the source node and nodes  $d_0$ ,  $d_1$  and  $d_2$  the multicast members.

If the constraints are fixed to (7,7), and if Samcra is used as the unicast QoS routing, the first step of mQMA algorithm returns the set  $FPS = \{FP_1 = (s-a-d_0-e-d_1), FP_0 = (s-d_0), FP_2 = (s-b-d_0-e-d_2)\}$ . The paths in the *FPS* do not form a tree, it contains 3 cycles which are  $(s,a,d_0,s)$ ,  $(s,b,d_0,s)$  and  $(s,a,b,d_0,s)$ .

**Cycles reduction during the tree decomposition step.** The decomposition step consider as a first path, the path containing the most member nodes. So, the first computed tree is  $FT_1 = \{s-a-d_0-e-d_1\}$ . The path  $FP_2 = (s-a-d_0-e-d_2)$  can not be grafted into the  $FT_1$  due to the cycle  $(s,a,b,d_0)$ , so it must be added into the *FTS* as  $FT_2$  and finally the path  $FP_0 = (s-d_0)$  used to forward data to  $d_0$  is already in  $FT_1$ , so this path will not be added to the *FTS* as a third tree. Thanks to this step, the cycles  $(s,a,d_0)$ ,  $(s,b,d_0)$  are removed and then the *FTS* contains only 2 trees. The third step in mQMA attempts to aggregate these two trees to exiting trees of *MTS*.

**Cycles reduction during the aggregation step.** If the *MTS*, contains the tree  $\{s-d_0-e-d_1-d_2\}$  and as the constraint limits are fixed to (7,7), this tree can be used to aggregate the multicast group. So, the routing structure used to forward data is now a tree and does no longer contain cycles. If the constraints are fixed to (6,6), global aggregation is not possible. In fact, the path  $(s-d_0-e-d_1)$  and the path  $(s-d_0-e-d_2)$  are not within constraints and so the tree  $\{s-a-d_0-e-d_1-d_2\}$  can not be used to aggregate the whole group.

Through this example, we show that although mQMA starts with a routing structure containing cycles, it can remove cycles from this structure during the tree decomposition step and during the aggregation step.

The number of trees found at the end of the decomposition step depends on many criteria: it depends on the number of disjoint paths in the graph topology and the number of metrics used to express the member constraints. On the example of Figure 5, if we consider 3 QoS metrics and constraint limits of (5,5,5), the global aggregation is not possible as cycles can

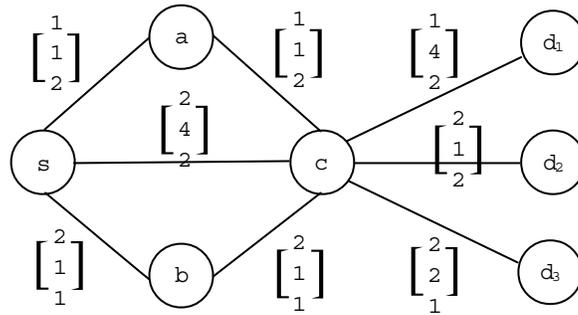


Figure 5: Example topology.

not be removed. The tree decomposition step computes 3 trees in the *FTS*. Indeed, to respect member requirements, 3 trees must be considered:

- $FT_1 = \{(s-b-c-d_1)\}$
- $FT_2 = \{(s-c-d_2)\}$
- $FT_3 = \{(s-a-c-d_3)\}$

We can state that: when  $n$  metrics are used, the tree decomposition step decomposes the routing structure to at most  $n$  trees.

## 5 Simulations

This section presents the results of the simulations comparing the algorithm mQMA to the algorithm Mamcra for the number of trees and the network resource usage. In the first step of mQMA, we have to use a unicast QoS routing algorithm. In the experimental study we adopt Samcra as the unicast QoS routing algorithm to compute the set of feasible paths *FPS*. Samcra which is described in appendix A is a unicast QoS algorithm that computes multi-constrained optimal path. The optimality is obtained according to the Samcra length function. We adopt this length function to compare mQMA to Samcra for the optimality of the routing structure. Moreover, we compare mQMA and Mamcra for the number of cycles removed from the multi-constrained multicast routing structure and for the cyclomatic number of the routing structure used effectively to forward data (we recall that this routing structure is a set of one or many trees for mQMA and a sub-graph containing eventually cycles for Mamcra).

### 5.1 Parameters of the simulations

The simulations were ran on Abilene network [1] which contains 11 routers and 14 links as depicted in Fig. 6. In this network, 1 000 groups were generated randomly with size between 2 and 11 members. The groups were source-oriented and the source for these groups was chosen among four routers which are known as specific delivering sources. The links  $l$  on the network were given values  $w_i^l$  (for the  $i$ -th metric) randomly from 1 to the fixed `MaxValueMetric` which was 5 during the simulations. We considered 2 additive metrics and we generated the constrained limits  $L_i$  (for the  $i$ -th metric) randomly for all the groups and these values were fixed to a maximum value varying from 5 to 500. For mQMA, leaky match aggregation is adopted and the bandwidth threshold  $t_b$  is varied from 0% to 40%. We repeat each scenario 1000 times. The simulation tool used to achieve this simulation is a graph simulator developed at IRISA and can be found at [12]. This simulator implements graph generating algorithms and several multicast routing and multicast aggregation algorithms.

### 5.2 Results of the simulations

This subsection presents the results of the simulations considering the number of trees, the network resource usage, the mean Samcra length per member.

#### 5.2.1 Number of trees

Figure. 7 shows the reduction of the number of trees for mQMA. This performance is due to the multicast tree aggregation. We recall that Mamcra routing structure is not always a tree. That's why in these simulations, the multicast routing structure given by Mamcra is decomposed on trees that must be taken into accounts when counting trees that must be effectively configured in the network. While, in general, Mamcra needs to maintain, at

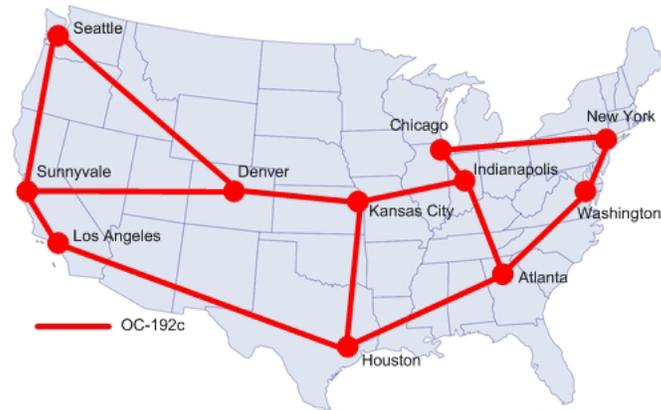


Figure 6: Abilene Network with 11 routers and 14 links.

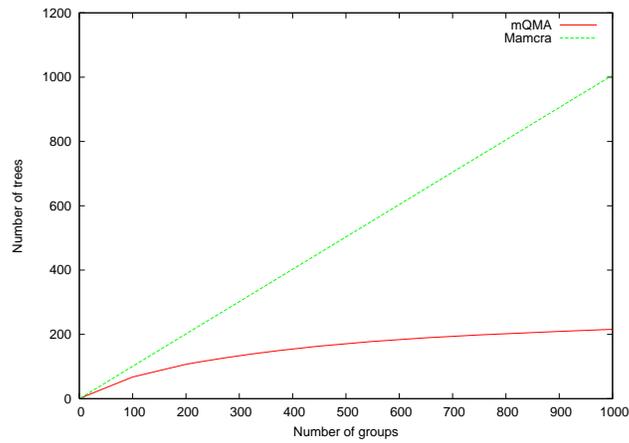


Figure 7: Reduction of the total number of trees

least, as many trees as groups in the domain, i.e. 1 000 trees (in the best case, all the multi-constrained multicast routing structure are trees), mQMA reduces the number of trees to only 200. As mQMA builds few trees, few forwarding states are maintained and the control messages to maintain the trees are reduced.

Figure. 8 plots the number of trees for mQMA considering the maximum value of constraints fixed for the groups and a bandwidth threshold loss fixed to 40%. When the constraints are strict and hard to fulfill considering the value metrics of links, the number of satisfied members decreases and the trees built often have the same structure. Then, the aggregation can be made more easily. Fig. 9 plots the number of members for which fea-

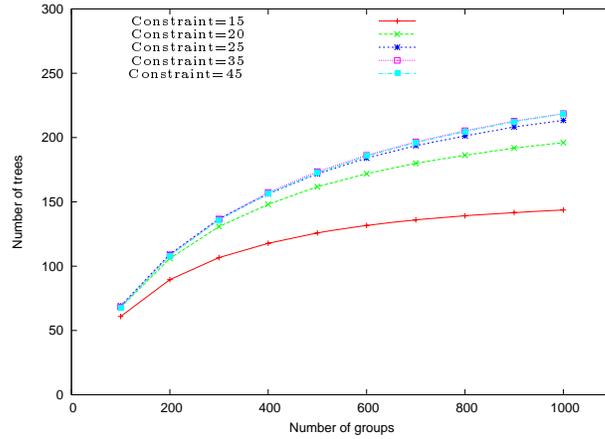


Figure 8: Number of trees.

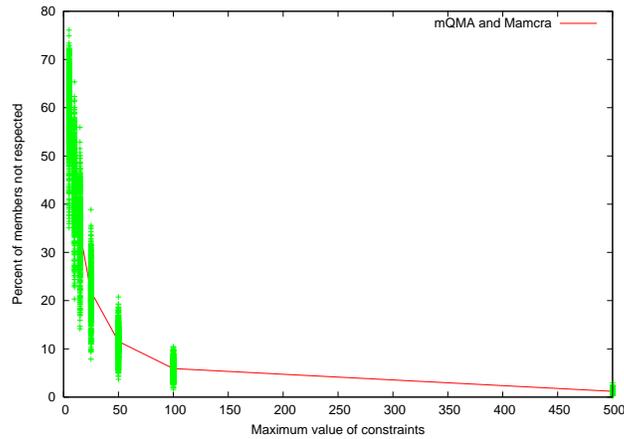


Figure 9: Multicast member satisfaction.

sible multi-constrained paths are not found and are not considered in the two last steps of mQMA. This number influences the number of trees and even the tree structure as shown in Fig. 8.

The number of tree is also affected by the type of aggregation adopted by mQMA. Fig. 10 shows that the number of trees decreases considerably when leaky aggregation is authorized. Using this results, a network manager will be able to decide according to the bandwidth availability whether to privilege aggregation rate or bandwidth losses. This number can reach only one tree if there is an infinite bandwidth threshold: in that case, only one tree covering all the routers of the network is configured.

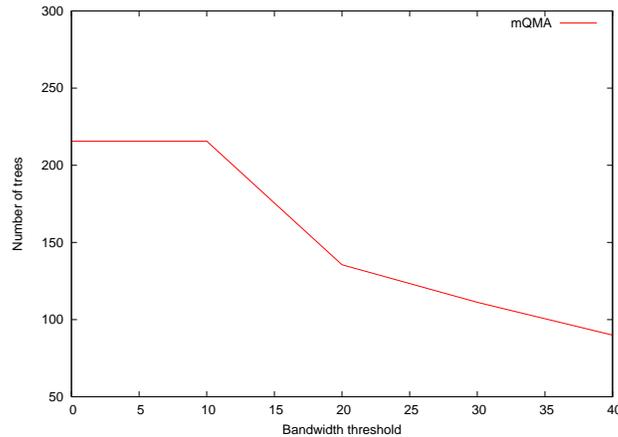


Figure 10: Number of tree according to the allowed bandwidth threshold.

### 5.2.2 Network resource usage

Figure 11 shows the network resource usage in terms of relative utilization. This metric is denoted by

$$U_R = \frac{C(T_{mQMA}) - C(T_{Mamcra})}{C(T_{Mamcra})} = \frac{C(T_{mQMA})}{C(T_{Mamcra})} - 1$$

where  $C(T_{algo})$  represents the cost the routing structures of the algorithm *algo* used to forward data for all the generated groups. In the simulations, the cost is measured in terms of number of links used for each trees set.

$$C(T_{algo}) = \sum_{g \in \mathcal{G}} |t(g)|,$$

where  $\mathcal{G}$  represents the set of all the groups generated,  $t(g)$  represents the structure used for  $g$  (which can be a set of trees for mQMA or a structure with cycles with Mamcra), and  $|t(g)|$  represents the number of links of  $t(g)$ . In mQMA, if the same tree is used to aggregate 2 groups, it is counted twice.

The relative utilization depends on the tolerated bandwidth threshold. Fig. 11 shows the network resource usage in terms of number of links. The results show that mQMA uses network resources slightly more efficiently than Samcra and Mamcra. This is mainly due to the phase of aggregation. Indeed, mQMA can find an already existing tree that covers the members, that is feasible and that has a lower cost. This occurs when mQMA lists all the trees and chooses among the ones that are feasible, the tree that has the lowest number of links. The results show that even when leaky aggregation is adopted and for lower threshold ( $\approx 17\%$ ), the aggregated trees used by mQMA to forward multicast traffic have lower cost than the Mamcra structure used when no aggregation is performed. So, even when leaky aggregation, mQMA may involve an effective gain in resource usage without violating

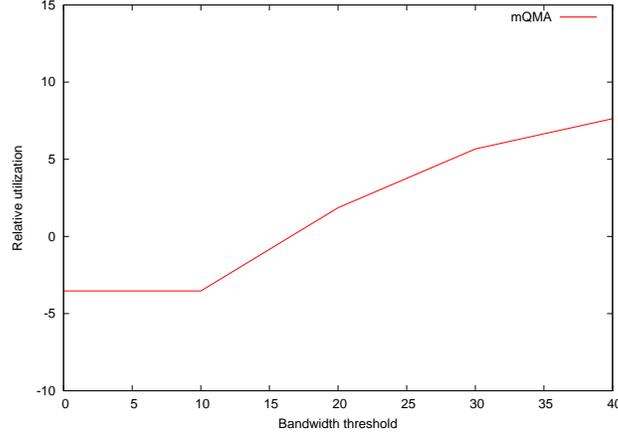


Figure 11: Relative resource utilization : mQMA vs Mamcra

the multi-constraint requirements of multicast groups. When the threshold increases, the network resource usage is not so high, the network resources waste for an infinite threshold does not exceed 35%.

### 5.2.3 Mean Samcra length per member

Figure 12 shows the mean Samcra length for the members, which is the main metric used for multi-constrained routing algorithm. This metric reflects the quality of the communications for the members of the group, expressed in function of the different QoS metrics. The Samcra length from  $s$  to  $d$  is denoted by :

$$\text{Samcra length}(s,d) = \max_{i < \# \text{ metrics}} \sum_{e_j \in \text{path}(s,d)} \frac{w_i^{e_j}}{L_i},$$

where  $w_i^{e_j}$  is the value of the metric number  $i$  for the edge  $e_j$  ( $e_j \in P$ ), and  $L_i$  is the requirement of the group for the metric  $i$ . The algorithm Samcra computes the optimal structure in minimizing the value of this metric. The results show that mQMA behaves in the same way as Samcra with slightly higher results for this metric. Then, we can say that the two algorithms are comparable and that mQMA gives good performance in terms of Samcra length as its results are close to the optimal.

### 5.2.4 Number of cycles removed

To achieve multi-constrained multicast routing, the routing structure is not always tree as it is detailed in section 2.2.2. The graph structure computed by the unicast QoS routing algorithm may contain cycles. Figure. 13 illustrates the number of cases with cycles contained on

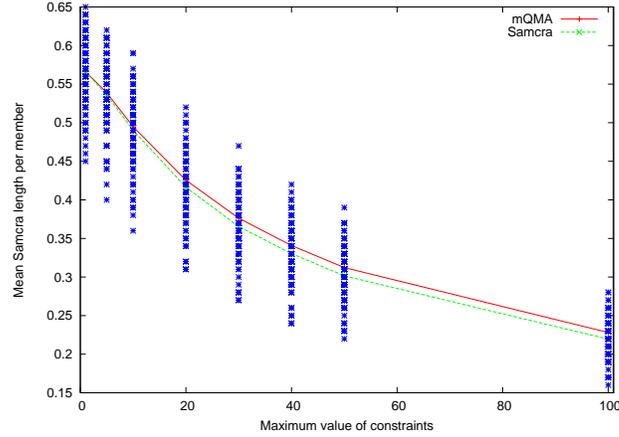


Figure 12: Mean Samcra length per member

the structure computed by the unicast algorithm to reach all the multicast members (namely Samcra). It represents the number of cases in which the set  $FPS$  of feasible paths contains cycles. Figure. 13 compares the number of cases in which mQMA eliminates cycles to the number of cases in which Mamcra succeed to do it. We recall, that cycles reduction obtained by mQMA include the one achieved by the tree decomposition and the one achieved by the aggregation (see section 1. In some cases, neither Mamcra nor mQMA removes the cycles found in  $FPS$ . In most of these cases, the cycles cannot be removed and the structure  $FPS$  is the only one possible that respects the constraints fixed for the group. The results show that the two algorithms are comparable in terms of cycle reduction as they behave in the same way.

The table 1 shows the number of cycles removed by Mamcra, mQMA and tree decomposition when generating 1000000 test cases. At the end of the simulations, mQMA has removed slightly more trees than Mamcra, that means that in 95.59% of cases, Mamcra has removed all the cycles from the  $FPS$  while mQMA has removed all the cycles in 96.84% of cases. From these 96.84% cases obtained by mQMA, 68.72% are implied by the tree decomposition step. The other remaining cases are implied by global aggregation. In 73% of the generated cases, the structure  $FPS$  found by Samcra contains at least one cycle. This confirms the necessity of finding an algorithm that removes the cycles.

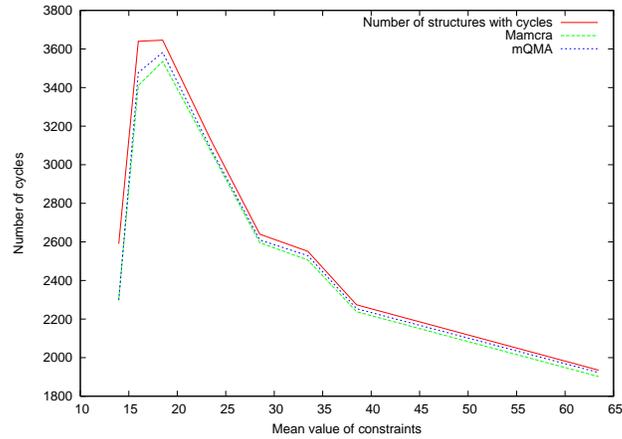


Figure 13: Number of cycles removed

	Number of structures with cycles	Structures without cycles with Mamcra	Structures without cycles with mQMA	Structures without cycles with Tree Decomposition
Maximum Value metric = 5	22402	21559	21754	15757
Total number of cycles removed (in %)		95.59% of cycles removed	96.84% of cycles removed	68.72% of cycles removed

Table 1: mQMA removes more cycles from the structures than Mamcra does

## 6 Conclusion

We presented in this paper a new algorithm mQMA that deals with the two main problems of multicast deployment: multicast forwarding state scalability and multi-constrained QoS routing. Our protocol is based on the techniques of multicast aggregation and on some ideas developed by Mamcra algorithm. To the best of our knowledge, mQMA is the first protocol that deals with these two main problems. With mQMA, in the worst case, a group can be associated to several trees and the data for the group are multiplexed onto these trees.

During the simulations, we show that our protocol achieves comparable performances in terms of Samcra length as Samcra which found the optimal routes for this metric. Moreover, our protocol mQMA spares the network resources by using less links than Mamcra and Samcra. Finally, mQMA gains all the benefits of multicast tree aggregation by building very few trees (at maximum 200 trees for 1 000 groups for Abilene network) and maintaining consequently few forwarding states for the groups.

## A Appendix

This section presents Mamcra and Samcra, the QoS routing used to solve respectively multicast and unicast multi-constrained routing problems. First, we give an overview of Samcra as it is used as a basis for the multicast QoS routing algorithm (Mamcra), next, we present, the Mamcra algorithm.

### A.1 Samcra algorithm

Samcra returns the path between a given source and a destination node respecting end-to-end constraints and optimizing an objective function. Samcra, like it is described in [23], is based on three fundamental concepts: a non-linear measure for the path length, the k-shortest path approach and the principle of dominated paths. All m additive QoS metrics are equally important. Each link is specified by a m-dimensional weight vector  $\vec{w} = [w_1, w_2, \dots, w_m]$ . The path vector  $\vec{w}(P) = [w_1(P), w_2(P), \dots, w_m(P)]$  is the vector sum of the link weights along this path. The path length is a vector norm and given by  $l(P) = \max_{1 \leq i \leq m} (\frac{w_i(P)}{L_i})$  where

$$w_i(P) = \sum_{l \in P} w_i(l)$$

This length function obeys the criteria for "distance" in vector algebra and is motivated by the geometry of the constraints surface in m-dimensional space. Samcra algorithm proceeds in a Dijkstra-like manner but by using this length function to explore nodes. Another result of this non linear function is that the subsection of shortest paths in a m-dimensional space are not necessarily shortest paths. That's why the k-shortest path approach is adopted and leading authors to consider not only the shortest path like it is done in Dijkstra's algorithm but also the 2nd shortest, etc, up to the k-th shortest path. Non dominance concept is used to reduce the space search by eliminating dominated paths and exploring only non dominated ones. Using this concepts Samcra find not only feasible path according to the required constraints but also optimal one according to the non linear length function defined above.

### A.2 Mamcra algorithm

Mamcra proceeds in two steps :

- **Step (A)**: Compute the set S of shortest paths from source node s to all p multicast members
- **Step (B)**: Add paths of S to M and optimize M without violating constraints

Step A is obtained by applying Samcra algorithm to find the set of paths from s to all destination. The set of paths S may contain overlaps, that's why, the step B attempts to remove some overlap in the set S in order to reduce overlap. This step proceeds in a greedy

manner to eliminate overlaps if it is possible. This step can be summarized by the meta-code in algorithm 3.

---

**Algorithm 3** Step B : Optimizing delivery structure

---

**Step (B) of Mamcra**

**Input:** The network  $G = (N, E)$ , a group  $g$  with a source  $s$ , constraints  $L_i$ , the set  $S$  of optimal path computed by Samcra

**Output:** A set  $M$  of paths

```

While ( $S \neq \emptyset$ ) do
  | add the path with the most members( $d_j$ ) to  $M$ ;
  | If (many) then
  | | choose the one with smallest length
  | end If
  | If (the added path forms a cycle in  $M$ ) then
  | | optimize  $M$  by rerouting the new path through an already existing path
  | | without violating constraints;
  | end If
  | If (cycle is not removed) then
  | | Check if the new path does not violate the min/max constraints
  | end If
  | Remove from  $S$  all nodes that are already visited by  $M$ ;
done
return ( $M$ );

```

---

## References

- [1] Abilene network. <http://abilene.internet2.edu>.
- [2] K. Almeroth. The Evolution of Multicast: From the Mbone to Inter-Domain Multicast to Internet2 Deployment. *IEEE Network*, September 1999.
- [3] N. Ben Ali, M. Molnar, and A. Belghith. Résolution du probleme de routage multicast multicritere. In *Metaheuristics Workshop*, 2006.
- [4] S. Chen and K. Nahrstedt. An overview of quality-of-service routing for the next generation high-speed networks: Problems and solutions. *IEEE Network, Special Issue on Transmission and Distribution of digital Video*, pages 64–79, November/December 1998.
- [5] J.-H. Cui, J. Kim, D. Maggiorini, K. Boussetta, and M. Gerla. Aggregated multicast — a comparative study. In *IFIP Networking*, number 2345 in LNCS, pages 1032–1044, May 2002.
- [6] J.-H. Cui, J. Kim, D. Maggiorini, K. Boussetta, and M. Gerla. Aggregated Multicast — A Comparative Study. *Special issue of Cluster Computing: The Journal of Networks, Software and Applications*, 2003.
- [7] J.-H. Cui, L. Lao, M. Faloutsos, and M. Gerla. AQoS: Scalable QoS multicast provisioning in Diff-Serv networks. *Computer Networks*, 2005.
- [8] J.-H. Cui, L. Lao, D. Maggiorini, and M. Gerla. BEAM: A Distributed Aggregated Multicast Protocol Using Bi-directional Trees. In *IEEE International Conference on Communications (ICC)*, May 2003.
- [9] S. E. Deering. *Multicast Routing in a Datagram Internetwork*. PhD thesis, Stanford University, December 1991.
- [10] C. Diot, B.N. Levine, B. Lyles, H. Kassem, and D. Balensiefen. Deployment issues for the IP multicast service and architecture. *IEEE Network*, 14(1), 2000.
- [11] M. Faloutsos, Banerjea A., and Pankaj R. Qosmic: Quality of service sensitive multicast internet protocol. In *ACM SIGCOMM*, pages 144–153, 1998.
- [12] Graph Simulator. <http://www.irisa.fr/armor/lesmembres/Moulhierac/outil.html>.
- [13] A. Guitton and J. Moulhierac. Scalable Tree Aggregation for Multicast. In *8th International Conference on Telecommunications (ConTEL)*, June 2005. Best student paper award.
- [14] L. Henrique, Costa M. K., Fdida S., O. Carlos, and M.B. Duarte. Hop-by-hop multicast routing protocol. In *ACM SIGCOMM*, pages 249–259, 2001.

- 
- [15] X. Jia, Y. Zhang, N. Pissinou, and K. Makki. A distributed multicast routing protocol for real-time multicast applications. *Computer Networks*, pages 101–110, 1999.
- [16] V.P. Kompella, J.C. Pasquale, and C. Polyzos. Multicast routing for multimedia communication. *IEEE/ACM Transactions on Networking*, 1(3):286–292, 1993.
- [17] F.A. Kuipers and Van Mieghem P. Constrained-based multicast routing algorithm. *Computer Communications*, 25(8):801–810, 2002.
- [18] Z.-F. Liu, W.-H. Dou, and Y.-J. Liu. AMBTS: A Scheme of Aggregated Multicast Based on Tree Splitting. In *IFIP Networking*, number 3042 in LNCS, pages 829–840, May 2004.
- [19] J. Moulierac. *Agrgation des communications multicast*. PhD thesis, Rennes I University, November 2006.
- [20] J. Moulierac and A. Guitton. QoS Scalable Tree Aggregation. In *IFIP Networking*, number 3462 in LNCS, pages 1405–1408, May 2005.
- [21] J. Moulierac, A. Guitton, and M. Molnár. Multicast Tree Aggregation in Large Domains. In *IFIP Networking*, 2006.
- [22] G.N. Rouskas and I. Baldine. Routing with end-to-end delay and delay variation constraints. *IEEE JSAC*, 15(3):346–356, 1997.
- [23] P. Van Mieghem, H. De Neve, and F.A. Kuipers. Hop-by-hop quality of service routing. *Computer Networks*, 37(3–4):407–423, 2001.
- [24] P. Van Mieghem, F.A. Kuipers, T. Korkmaz, M. Krunz, M. Curado, E. Monteiro, X. Masip-Bruin, J. Solé-Pareta, and S. Sánchez-López. *Quality of Future Internet Services*, chapter 3, pages 80–117. EU-COST 263 Final Report. Springer LNCS 2856, smirnov and al. edition, 2003.
- [25] B. Wang and C.J. Hou. Multicast routing and its qos extension: Problems, algorithms and protocols. *IEEE Network*, pages 22–36, 2000.
- [26] L. Wei and D. Estrin. The trade-offs of multicast trees and algorithms. In *International Conference on Computer Communications and Networks*, pages 17–24, 1994.

## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Related work</b>	<b>5</b>
2.1	Multicast Tree Aggregation . . . . .	5
2.1.1	Multicast Tree Aggregation Principles . . . . .	5
2.1.2	Multicast Tree Aggregation proposals . . . . .	7
2.2	Multi-constrained Multicast Routing . . . . .	8
2.2.1	Hypothesis . . . . .	9
2.2.2	Multi-constrained multicast routing proposals . . . . .	9
<b>3</b>	<b>The mQMA algorithm</b>	<b>11</b>
3.1	Definitions . . . . .	11
3.2	Algorithm mechanisms . . . . .	11
3.3	mQMA on an example . . . . .	15
<b>4</b>	<b>mQMA: a proposal to enhance multi-constrained routing structure</b>	<b>17</b>
4.1	Problem statement . . . . .	17
4.2	mQMA contribution to solve cycles problem . . . . .	18
<b>5</b>	<b>Simulations</b>	<b>20</b>
5.1	Parameters of the simulations . . . . .	20
5.2	Results of the simulations . . . . .	20
5.2.1	Number of trees . . . . .	20
5.2.2	Network resource usage . . . . .	23
5.2.3	Mean Samcra length per member . . . . .	24
5.2.4	Number of cycles removed . . . . .	24
<b>6</b>	<b>Conclusion</b>	<b>27</b>
<b>A</b>	<b>Appendix</b>	<b>28</b>
A.1	Samcra algorithm . . . . .	28
A.2	Mamcra algorithm . . . . .	28