

Hierarchical QoS Routing by Using Multi-Constrained Macro-Routing

Sanda-Maria Dragoş

Chair of Computer Systems, Department of Computer Science

Faculty of Mathematics and Computer Science

“Babes-Bolyai” University 400084, Cluj-Napoca, Romania

Email: sanda@cs.ubbcluj.ro

Abstract—QoS routing is used for finding feasible paths that satisfy simultaneously multiple constraints and it is a NP-complete problem. The difficulty of solving such problems increases in the hierarchical context, where aggregation techniques influence the path computation process. This paper offers a solution for hierarchical QoS routing by introducing the protocol called *Multi-constrained Macro-routing*. This protocol uses *Macro-routing* protocol with the *Extended Full-Mesh* aggregation technique to determine multiple hierarchical QoS routes. Tests presented here prove that *Multi-constrained Macro-routing* outperforms any other hierarchical routing protocol that uses the *Full-Mesh* aggregation by finding better and more QoS paths.

Keywords—hierarchical routing, topological aggregation, MPLS, mobile agents

I. INTRODUCTION

The task of finding a path in the network satisfying multiple constraints is particularly challenging. This problem is even more complex in the case of hierarchical networks [1], [2], [3]. The problem here is that since a logical link at one level of the hierarchy corresponds to P paths in the physical network, a complete characterization of the logical link, where there are M constraints to be met, requires $M \times P$ parameter values to be recorded. To do so would negate the benefits of topology aggregation that a hierarchical routing scheme brings, and so some means must be found to reduce the amount of data used to characterize the logical links.

In the case where $M = 1$ (the single constraint problem) this is straightforward. The lowest cost path among the set of P paths is used to represent the full set, since this is the “best” path. This solution is *not* available when $M > 1$, since identifying the “best” path is problematic. The optimal path may be a compromise between the various constraints, and it will not be apparent *a priori* which of the P paths this is.

The obvious solution is to record the metrics of a candidate set of best paths, in the hope that one of these will be the appropriate path. This causes two difficulties:

- 1) the amount of data used to describe a logical link, while reduced, is still considerable;
- 2) some mechanism is needed to record and select the candidate paths.

This paper offers a solution for both of these challenges.

II. BACKGROUND

A. The hierarchical multi-constrained problem

Within a hierarchical routing context, the multi-constrained problem is very complex [1], [2], [3]. That is because it is very difficult to designate appropriate values for the multiple QoS parameters that are associated with a logical¹ link in order to build the aggregate nodal representations. That is because one path may be the best for one constraint but may not satisfy other constraints.

Figure 1 presents a network topology with each link having associated two additive weights. The issue here is what values should be associated with the logical link from the aggregated representation which connects nodes a and c .

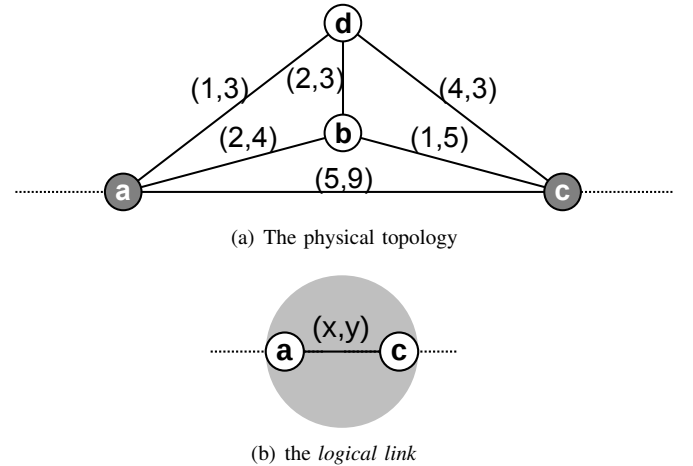


Fig. 1. Example of topology aggregation with two additive QoS metrics

As depicted in Figure 2, there are five possible paths between the border nodes a and c . Their representation in the bi-dimensional space determined by the two weights is also depicted in Figure 2.

The conventional methods for solving the “best” path problem are described in [1], [2], [4].

- *Single path metric*: The path selection is made based on a single QoS metric w_i . The representative path is then

¹A *logical link* is an aggregation of all physical paths between two border nodes.

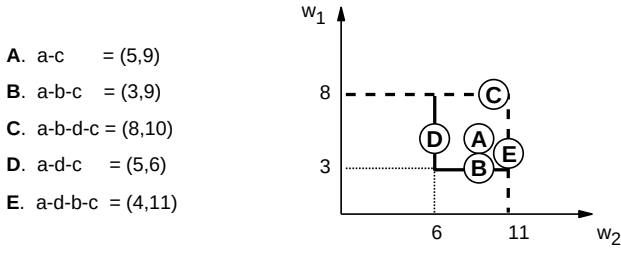


Fig. 2. Paths represented into the QoS-metric bi-dimensional space

the one that satisfies: $w_i^* \leq w_i^{(j)}$ for all $j = \overline{1, P}$. The issue remains to select the actual metric. For example in Figure 1 the chosen path will be (B) if w_1 is the representative QoS metric, or (D) if w_2 is the representative QoS metric.

- *The best case approach:* The aggregated link will have as QoS metrics $(w_i)_{i=\overline{1, M}}$ the *best* metric values amongst all paths, i.e. for all $i = \overline{1, M}$:

$$w_i = \min_{j=\overline{1, P}} w_i^{(j)} \quad (1)$$

where P is the total number of paths. In Figures 1-2 the M-tuple corresponding to the QoS constraints for the *best case approach* is (3, 6).

- *The worst case approach:* The aggregated link will have as QoS metrics $(w_i)_{i=\overline{1, M}}$ the *worst* metric values amongst all paths, i.e. for all $i = \overline{1, M}$:

$$w_i = \max_{j=\overline{1, P}} w_i^{(j)} \quad (2)$$

where P is the total number of paths. In Figures 1-2 the M-tuple corresponding to the QoS constraints for the *worst case approach* is (8, 11).

Works in [4], [5], [6] consider only one metric for QoS information aggregation. The best case approach is studied in [7], where the QoS parameter of each logical link between any two borders is set to be the minimum delay and cost metrics of all paths between those border nodes. In [8], the QoS parameter of each logical link between any two border nodes is set to be the worst one, that is the maximum delay and cost metrics for all the paths between two border nodes. However, it has been shown in [9] that the best case approach or the worst case approach do not work well.

Other research papers consider only two constraints such as cost and delay [10], [11], bandwidth and cost [12], [13], capacity and bandwidth [14] or bandwidth and delay [2], [3], [15], [16].

However, the most important requirement in multi-constrained hierarchical routing is not to find a “best” representative for the logical link, but to find the appropriate sub-path for the *best* final path. A method to find such sub-paths by considering more than one representative for a logical link is presented in this paper.

B. Macro-routing

Macro-routing [17] is a protocol that addresses the problem of hierarchical routing within MPLS networks. It uses small mobile agents that are dispatched to process the state information at each node. Thus, no state information advertising is required. The advantage of this approach is that the information used to compute routes can be much more detailed than in traditional link-state protocols (e.g., it can feature multiple QoS constraints, or a Full-Mesh aggregated representation). Moreover, by using mobile agents which can replicate at each node and therefore analyze a large set of paths, route computations are done in a distributed and parallel manner which reduces the time required for path setup and distributes the processing burden amongst mobile agents.

Another notable feature of *Macro-routing* is that it uses Wave [18] as the mobile agent technology. The advantage is that the Wave syntax generates code that is 20 to 50 times shorter than equivalent programs written in C++ or Java [19]. Figure 3 shows the Wave implementation of the path computation algorithm used by *Macro-routing* at the first level of the hierarchy, where the nodal costs are zero.

```

Falg=RP (OS (Fcost==0, A/~Fborders) . \
  A/~Fpath.Fcost<Frequired.Fpath&A.Fcost+L.#) . \
Fborders=ARG1.Frequired=ARG2.Fmanaging=ARG3. \
Falg. \
Fpath&A.Fcost+L.@#Fmanaging.Npath&Fpath.Ncost&Fcost

```

Fig. 3. The Wave implementation of the path computation algorithm (explained in [17])

Macro-routing is capable of both routing and signalling functionalities which may be both accomplished by the use of mobile agents.

C. The extended Full-Mesh (EFM) aggregate representation

The key tradeoff in topology aggregation is between performance and accuracy. As many works [20], [21], [4] proved, the Full-Mesh representation offers the most accurate representation. However, this aggregation is not used because the amount of information to be advertised in link-state routing increases with the square of border nodes. By using mobile agents that consult the state information *in situ*, no state information advertising is required. Thus, the Full-Mesh representation can be used [17].

A new topology aggregation scheme, called *extended Full-Mesh* (EFM), is introduced in [22]. It is intended to be used with *Macro-routing* in order to compute hierarchical multi-constraint routes within MPLS networks.

The EFM extends the Full-Mesh by allowing not only one “best” path between any two border nodes but multiple satisfactory paths.

To build an EFM aggregation, all (or multiple) border-to-border paths satisfying the required QoS constraints have to be determined. During this process (done by using mobile agents, which in this case are called *waves*), path costs corresponding to each QoS constraint are also determined for each such

path. To create a port-to-port link connection in the EFM representation, there are considered all paths found between the two corresponding border nodes. These P paths found, each having M costs corresponding to the QoS metrics form a matrix of $M \times P$. Each path can be represented in the M -dimensional space as a “dot” placed corresponding with the values of its QoS costs. Such “dot” is called *EFM path*. The smallest M -dimensional “volume” that contains all the EFM paths viewed as “dots” is called the *EFM interval*.

The main factors that govern the number of EFM paths are:

- the network connectivity degree;
- the number and severity of the constraints.

Thus, highly connected networks, fewer constraints and resource availability are contexts in which the number of EFM paths may be so high that it can negatively influence the path computation process. In [22] there are presented 8 different techniques to reduce the number of EFM paths, which are grouped into three classes:

- **Truncation** - This class contains greedy methods through which best resources are occupied first;
- **Dispersal** - Is a class of methods that try to keep a larger spectrum of alternative paths, but require a lot of computation and are therefore impractical;
- **Practical** - This class contains simpler methods/algorithms.

A selection of these techniques (slightly modified) are presented in Section III-A, and are used in this paper to test the EFM concept within the multi-constraint Macro-routing context.

III. MULTI-CONSTRAINED MACRO-ROUTING

The protocol that extends *Macro-routing* to find multi-constraint paths by using the EFM aggregate representation is called *Multi-constrained Macro-routing*. Similar with *Macro-routing*, it finds and selects a hierarchical QoS path in three steps:

- 1) **Determination of participant domains** - The source node initiates an “upwards search” in the hierarchy for the lowest level parent node which has the view of both source and destination. From this parent node a “downwards search” is initiated in parallel to all its children. Recursively, the nodes reached will continue the search to all their children until they reach the lowest level of the hierarchy. All physical domains reached by this search are the *participant domains*. All these searches are done by mobile agents called *waves* that are propagated in the hierarchical network.
- 2) **Path computation** - All *participant domains* have to create (in parallel) their own EFM aggregate representation. This can be done either on-line and/or off-line. After the EFM representations for all participant domains are completed, the process recursively repeats with the next levels of the hierarchy where nodal costs

(i.e. the costs of traversing a virtual/aggregated node) are also considered when computing a path. The topmost hierarchical level will contain a single domain that will have as border nodes the aggregate representation of source and destination domains. All paths found at this level represent the aggregate representation of final paths and have the whole path costs for the corresponding QoS constraints. One or more final paths can be selected to be used. All the other already computed paths can be used either by fast recovery mechanisms or as alternative paths.

The paths for permanent and/or long-term connections can be determined off-line and only updated if/when necessary, while the paths used by short-term connections are computed online (i.e. on-demand). However, pre-computations or capacity planning can also be performed off-line.

- 3) **Path reservation and set up** - To accommodate the traffic for which the request has been made, the final path must be set up and the resources reserved. The path set up and resource reservation can be done either by existing signalling protocols if they can set explicit paths, by existing resource reservation protocols (such as the Resource reSerVation Protocol (RSVP)), or by using suitably programmed mobile agents. The main advantage of using mobile agents instead of RSVP for the reservation process is that mobile agents can do the reservation in a parallel and distributed manner. From each hierarchical level, within each domain, a mobile agent can be dispatched to reserve resources corresponding to each logical/aggregated link by duplicating at each *parent*² by the number *hierarchical children*³, while RSVP has to traverse the path in a sequential manner twice: by using *PATH* messages, from source to destination, to determine the path and then by using *RESV* messages, from destination to source, to reserve the path.

Setting up a hierarchical path in an MPLS network is straight-forward, using the MPLS label stack capability. Thus, every sub-path within every domain and every hierarchical level can be treated independently. Using mobile agents for setting up the LSP (Label Switched Path) is also preferred as mobile agents assigned to reserve the resources can also set up MPLS labels and then replicate and dispatch to advertise/distribute them. This can be done in a distributed and parallel manner using multiple mobile agents as described above. Any other label distribution protocol has to do the path set-up sequentially.

²Each *parent* in the hierarchy is an aggregation of a domain at the next lower hierarchical level.

³All *children* in the hierarchy are the nodes in the same domain aggregated by the *parent*.

A. Decreasing the number of EFM paths

During the *path computation* phase, considering more candidate paths require more computations, and the size of a mobile agent (e.g., *wave* in this case) may increase so that the algorithm generates too much control traffic. The information a *wave* has to carry, in addition to that of a standard *Macro-routing wave*, is $M \times (P-1) \times c$ bytes where M is the number of metrics, P is the number of *EFM paths*, and c is the number of bytes occupied by a metric value.

To control the *wave* size, the number of *EFM paths* can be limited by a threshold T . This however is not a trivial problem as it is not obvious *a priori* which EFM path to discard when there are too many.

The remainder of this paper investigates possible selection criteria in order to determine their effectiveness.

Two classes of techniques are presented below for reducing the number of considered EFM paths. One (*truncation*) is a greedy method through which the best resources are occupied first; the other method (*random*) tries to keep a larger spectrum of alternative path vectors in order to provide a better load balancing.

In describing these techniques, the following special⁴ cases are considered:

- The methods are applied to a two-dimensional EFM interval. (although they are applicable on any M -dimensional EFM interval with $M \geq 2$).
- All metrics are normalized to lie in the range from 0 (*best*) to 1 (*worst*).

1) *Truncation of the EFM interval*: It eliminates paths starting from the “*worst*” and progressing through the “*best*” until only T EFM paths remain.

A difficulty is that it is not possible to clearly distinguish the *worst* multi-constraint paths. Three possible methods to select these paths are considered below. They are schematically represented in Figure 4.

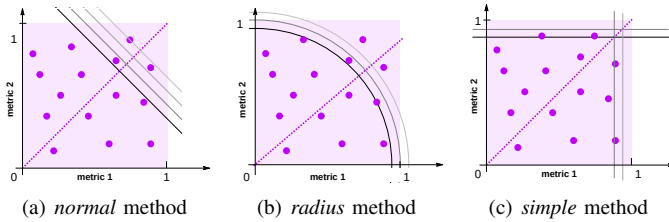


Fig. 4. Methods for truncating a two-dimensional EFM interval

The first method, depicted in Figure 4(a) progressively eliminates the path vectors that are traversed by the line $f(x,t) = 2t - x$, which lies perpendicular to the diagonal from the *worst-worst* (1,1) corner down to the *best-best* (0,0), as the parameter t decreases from 1 towards 0, until (at some value $t = t_0$) (at most) T path vectors remain. The path vectors that satisfy the inequality $f(x, t_0) > y$ are the selected ones⁵.

⁴The restriction to these cases is purely for clarity of explanation.

⁵ x and y are the two metrics in this two-dimensional example.

Another method, presented in Figure 4(b), is similar but the delimitation between the selected and the unselected paths is not a line but an arc. This arc is part of the circle centered at $x = 0, y = 0$. The circle's radius, r , varies from $\sqrt{2}$ down to 0 until no more than T path vectors remain. When the appropriate radius $r = r_0$ is found, all path vectors inside the arc are selected. Thus, a path vector is retained if it satisfies the inequality $\sqrt{x^2 + y^2} < r_0$.

Figure 4(c) depicts the third truncation method, where the *worst* path with respect to a single metric is eliminated. The designation of the considered metric is done either starting with a randomly chosen metric and then choosing other metrics for the next path elimination in a round-robin fashion or based on a *priority* metric attribute. The chosen attribute may change with every path elimination allowing another metric with a higher priority to be the next decision factor. If only one metric is used in the selection process (without alternation), this method is referred to as *truncate single*.

2) *Random selection*: Another method for reducing the number of EFM paths is to reduce the density of path vectors within the EFM interval though random selection. Thus, T EFM paths are picked at random from the available paths. However, true randomness being difficult to implement in lightweight real-time processes, it will in practice be approximated using some cyclic selection process or selection based on some quasi-random attribute such as a wave checksum.

Another “random” path selection that can be implemented in a real system is to consider the first T paths found. This mechanism is inherently biased and can also be considered a truncation selection because it tends to favor paths with low delay and few hops.

B. The simulation model

All tests were performed on two-level hierarchical network topologies, randomly generated by the *Georgia Tech Internet-network Topology Models* (GT-ITM) [23]. One example of such topologies is presented in Fig. 5.

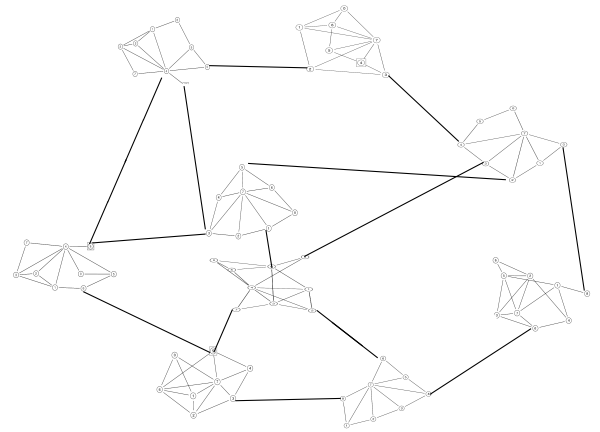


Fig. 5. Simulations-like two-level hierarchical topology

The GT-ITM utility was used to generate flat (single-level)

topologies. Using these flat topologies, the hierarchy is automatically created by taking one flat topology for level two and using the other flat topologies to expand its nodes to represent the first level of the hierarchy. The hierarchy is created in such a way that no node would have a *degree*⁶ smaller than two, otherwise the hierarchy was considered *invalid*. To increase the average node degree, the topologies from the first hierarchical level with the fewest number of links (i.e., the smallest average node degree) were associated with the nodes of highest degree from the topology at the second level of the hierarchy. Thus, domains corresponding with such topologies would have more border nodes (i.e. would be aggregated as nodes with highest degree).

A source and a destination node were randomly chosen for each test so that no direct connection exists between them (neither at the first nor at the second level of the hierarchy). The context of each test is defined by the hierarchy, the randomly assigned costs for each link, the source and the destination nodes.

Some preliminary tests deploying real wave agents on a virtual (i.e., simulated) network running on a single physical host were performed. However, due to the complexity of the virtual networks required to evaluate Multi-constrained Macro-routing (hierarchical networks with hundreds of nodes), an application⁷ was developed to generate the same list of the paths the waves would discover, rather than deploying waves on such networks.

Multiple sets of tests were run on different hierarchical topologies with two hierarchical levels as presented in Table I.

TABLE I
TOPOLOGIES USED IN THE SIMULATION TESTS

Topology /nodes	intra+inter domain links	connectivity $2L/N$
4 domains $\times$ 4 = 16	20+4= 24	3.00
9 domains $\times$ 9 = 81	110+14= 124	3.06
10 domains $\times$ 10 = 100	137+12= 149	2.98
12 domains $\times$ 12 = 144	208+15= 223	3.09
13 domains $\times$ 13 = 169	265+17= 282	3.33
15 domains $\times$ 15 = 225	343+25= 368	3.27
20 domains $\times$ 20 = 400	565+35= 600	3.00

There were implemented algorithms for the path selection mechanisms proposed in Section III-A. In the remainder of this paper the implemented path selection techniques will be referred to by the acronyms presented in Table II.

The quasi-random method selects the T paths for the EFM aggregate representation from the list of paths generated by the backtracking application. Preserving the order in which the paths are generated by the application, there are selected T paths from positions $l(i)$ in the list, where $l(i)$ is defined

⁶The *degree* for a specific node is considered to be the number of (inter- and intra-domain) links that are incident to it.

⁷An iterative backtracking algorithm was used to determine all paths that a wave traversing a network and duplicating at each node with the number of outgoing links would find.

TABLE II
THE ACRONYMS USED FOR THE PATH SELECTION TECHNIQUES USED FOR SIMULATION TESTS

candidate path selection technique	acronym
TRUNCATE SINGLE using metric j	$TS[j]$
TRUNCATE NORMAL	TN
TRUNCATE RADIUS	TR
QUASI RANDOM	QR
FIRST COME FIRST STORED	$FCFS$

as can be seen in (3):

$$l(i) = i \cdot \left\lfloor \frac{P}{T} \right\rfloor, \quad (3)$$

with $1 \leq i \leq T$ and P is the total number of paths found.

In order to emulate the FCFS mechanism, we introduced the *delay* metric in addition to the other metrics that were considered for all the other selection mechanisms. Thus, the paths selected by FCFS are the T paths with the smallest delay.

The metrics used for the tests that were performed on different hierarchical topologies were *administrative cost*, *hop count* and *delay*⁸. The *hop count* metric would be 1 for each physical link. Up in the hierarchy, the values of *hop count* will differ from one link to another due to the aggregation process. For the other two metrics, the values were randomly chosen, for each test, for every physical link within the hierarchical topology, from a given interval (i.e. *administrative cost* $\in [1, 15]$ and *delay* $\in [2ms, 45ms]$).

It was imposed, for all the tests, a limit T for the number of candidate paths to be selected (e.g., $T = 5$). This means that the EFM aggregate representation will have $\Omega_{u,v}$ EFM paths for every pair of border nodes u, v , where:

$$\Omega_{u,v} = \min\{T, n_{u,v}\} \quad (4)$$

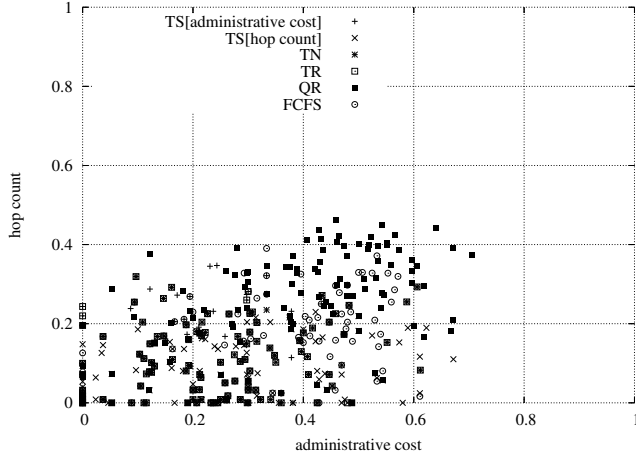
Here $n_{u,v}$ is the number of paths found between border nodes u and v .

Each test performed on one topology resulted in different values for the best or the worst costs of the designated metrics. In order to compare them, these results were normalized so that 0 represents the best value and 1 represents the worst value of each metric.

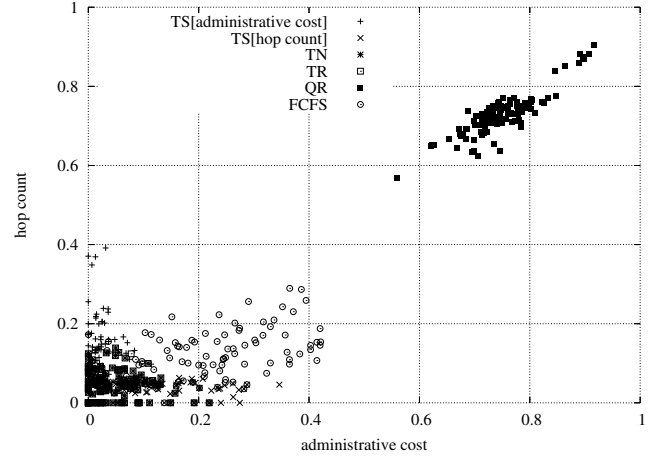
C. EFM's simulation results

The first set of tests were performed in order to determine the best method for decreasing the number of EFM paths. Tests started from the smallest topologies to the larger ones amongst the ones presented in Table I. As it can be seen in Figure 6(a), there were no significant differences between the 5 methods for the 4×4 topology. It can be conjectured that too few paths are available in such a small network for the method of path selection to be significant. In larger topologies each method developed patterns that are most visible for the results

⁸The *delay* metric was used only by the FCFS technique in order to reduce the number of EFM paths to T if more were found.

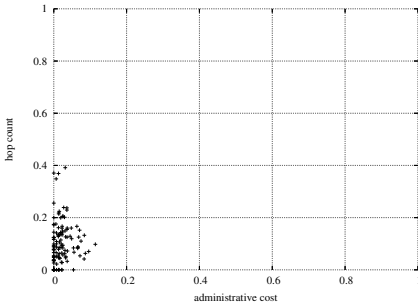


(a) 4x4 node topology

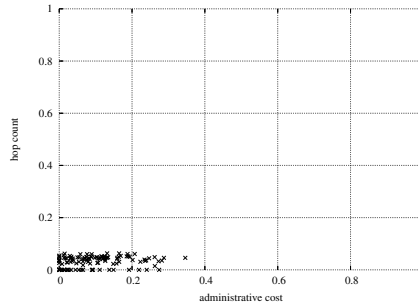


(b) 20x20 node topology

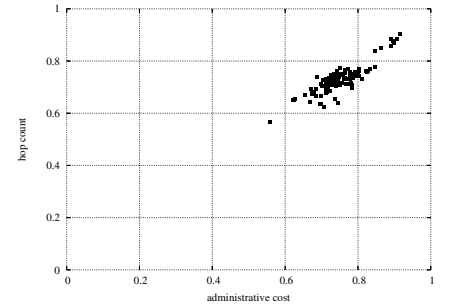
Fig. 6. Comparing the six EFM methods on a 4x4 node topology



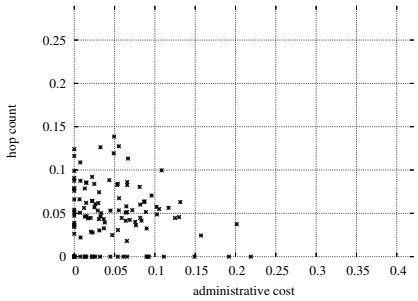
(a) TRUNCATE SINGLE using administrative cost



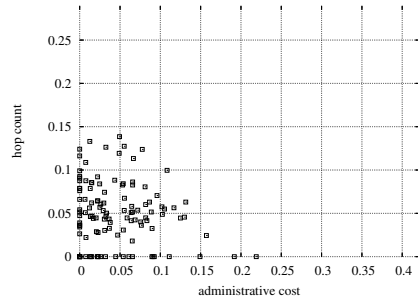
(b) TRUNCATE SINGLE using hop count



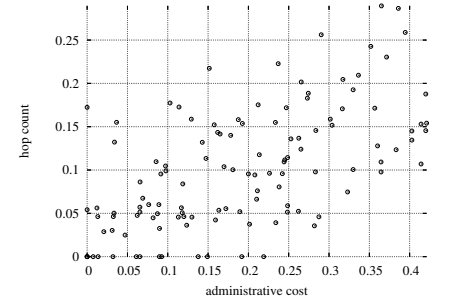
(c) RANDOM



(d) TRUNCATE NORMAL



(e) TRUNCATE RADIUS



(f) FCFS

Fig. 7. Detailed comparison of the six EFM methods on a 20x20 node topology

obtained for the 20x20 topology. These results are presented in Figure 6(b).

As it can be observed also in Figure 7 where the same results for all EFM methods are presented separately⁹, these patterns show that:

- $TS[j]$ leads to better results when considering only metric j , but may produce worse results overall compared with other truncation approaches (e.g., TN and TR);

⁹The results presented in Figures 7(d), 7(e) and 7(f) are enlarged so that they can be better viewed and because all paths selected by TN , TR or $FCFS$ are enclosed in this smaller interval.

- There are very few cases when the QR approach finds one or two very good path costs. However, such cases are rare and form no discernible pattern. Overall it can be ranked as the worst technique;
- $FCFS$ is the best random method, but performs worst than any truncation method;
- The difference between $TRUNCATE\ NORMAL$ and $TRUNCATE\ RADIUS$ methods on the tests performed on the 20x20 node topology are insignificant, as verified by the detailed/enlarged results in Fig. 7. This suggests that any truncation rule which is fair (i.e., which favors

no metric in its selections) will offer similar performance.

Next, more tests were performed in order to determine the best method amongst TRUNCATE NORMAL and TRUNCATE RADIUS. Thus, the next set of tests were performed on different topologies with the same number of nodes (i.e. $12 \times 12 = 144$) but with different connectivity degrees (i.e. $c_d \in [4, 4.5]$, and $c_d \in [5, 5.3]$).

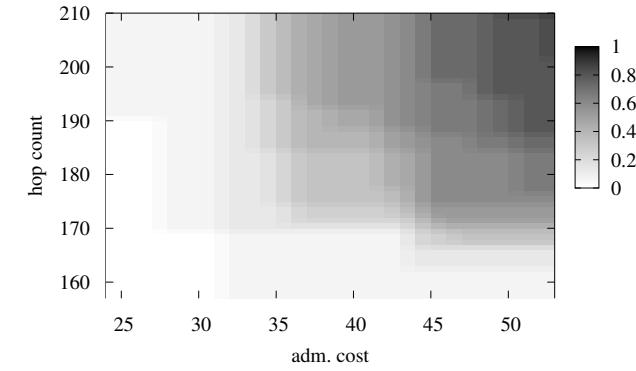
Figure 8 depicts the percentage of the number of paths that were found by both methods and satisfy the QoS constraints: *administrative cost* < 54 and *hop count* < 210 and their distribution on the bi-dimensional space determined by the two metrics. Differences between the two path

macro-routing algorithm was *Multi-constraint Macro-routing* with EFM aggregation (with $T = 4$) and the TRUNCATE RADIUS path selection mechanism. This was compared to a scheme with Full-Mesh aggregation and the *Lagrangian-based linear composition* [24] for selecting the representative path for each port-to-port connection used for building the Full-Mesh aggregate representation. All metrics were considered equally weighted with $d_i = \frac{1}{n}$, where n is the number of metrics. Thus, the *Lagrangian-based linear composition* considered was:

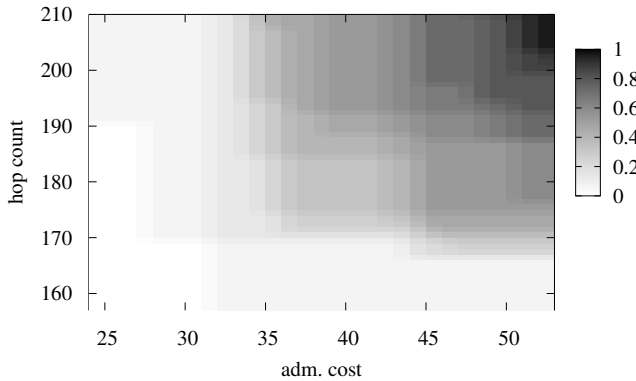
$$W_i = \frac{\sum_{k=1}^n c_i^k}{n} \quad (5)$$

The selection mechanism based on (5), however, leads to the same results as would *Multi-constraint Macro-routing* when the TRUNCATE RADIUS path selection mechanism is applied with $T = 1$. Thus, the two different protocol implementations will be referred to as $T = 4$ and $T = 1$.

The results depicted in Fig. 9 show the percentage of paths found by both methods that satisfy the constraints: *administrative cost* < 54 and *hop count* < 210 . Here it can easily be observed that $T = 4$ generates much better results than $T = 1$. This proves that *Multi-constraint Macro-routing* can find more and better paths than any other hierarchical routing protocol that uses the Full-Mesh topology aggregation.



(a) the proportion of TN paths which satisfy the constraints

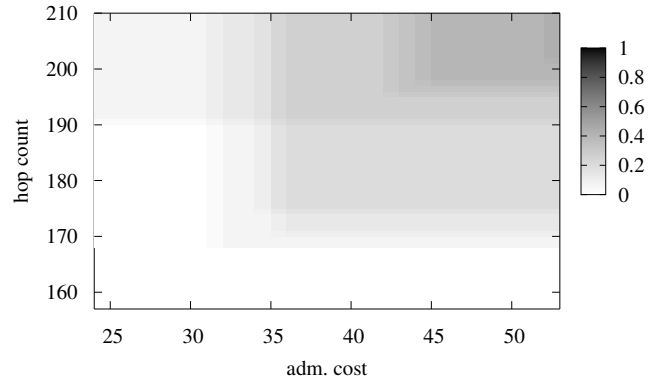


(b) the proportion of TR paths which satisfy the constraints

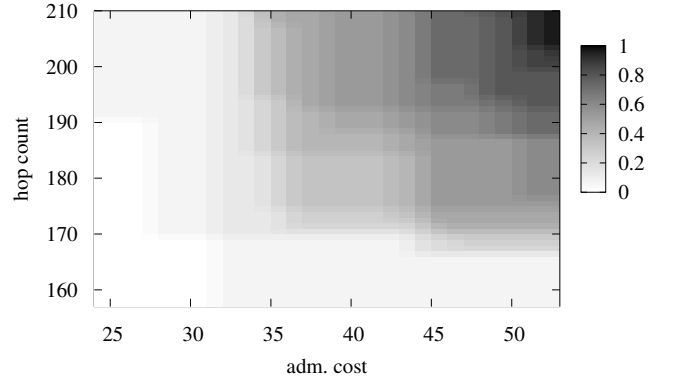
Fig. 8. Comparing TRUNCATE NORMAL and TRUNCATE RADIUS methods on 12×12 node topologies with connectivity degree $c_d \in [5, 5.3]$

vector reduction methods (i.e. TRUNCATE NORMAL and TRUNCATE RADIUS) occur only in topologies with higher connectivity degree (i.e. $c_d \in [5, 5.3]$), where most of the EFM paths found by the RADIUS method satisfy the established constraints in a larger proportion than the NORMAL method.

The final set of tests compared in terms of performance the macro-routing approach to the best hierarchical routing algorithm that can be created by using existing concepts. The



(a) the proportion of TR paths for $c_d \in [5, 5.3]$ and $T=1$



(b) the proportion of TR paths for $c_d \in [5, 5.3]$ and $T=4$

Fig. 9. Compare TRUNCATE RADIUS with $T = 4$ and $T = 1$ on 12×12 node topologies with different connectivity degrees

IV. CONCLUSIONS

This paper addresses the problem of hierarchical QoS routing. The *Extended Full-Mesh (EFM)* aggregate representation used by the protocol proposed here, called *Multi-constrained Macro-routing*, is a compromise between the Full-Mesh aggregation and no aggregation. The Full-Mesh representation works well for finding hierarchical paths based on a single metric, as presented in [17]. However, it does not guarantee to find a hierarchical path based on multiple metrics, even if such a path exists. That is because it is very difficult to designate a single “best” multi-constrained path for every pair of border nodes in order to build the Full-Mesh aggregate representation. EFM allows more paths to be considered (if they exist), and thus increasing the chances of finding a viable path. However, the EFM aggregation technique can potentially generate too much computational and communication overhead. Thus, the number of candidate paths has to be limited. A number of techniques for limiting the size of an EFM representation were presented and discussed in this paper. Moreover, tests proved that this representation offers better results than Full-Mesh. It was also determined the most efficient path selection mechanisms for obtaining the most appropriate multi-constrained paths for every border-to-border. This technique is called *TRUNCATE RADIUS*.

A real implementation of *Multi-constrained Macro-routing* protocol will allow the deployment and testing other path selection methods that were not implemented and tested in this paper.

ACKNOWLEDGMENT

This material is partially supported by the Romanian National University Research Council under award PN-II IDEI 2412/2009.

REFERENCES

- [1] W. C. Lee, “Spanning Tree Method for Link State Aggregation in Large Communication Networks,” in *IEEE Conference on Computer Communications (INFOCOM)*, 1995, pp. 297–302.
- [2] T. Korkmaz and M. Krunz, “Source-Oriented Topology Aggregation with Multiple QoS Parameters in Hierarchical ATM Networks,” in *7th International Workshop on Quality of Service (IWQoS)*, 1999, pp. 137–146.
- [3] K.-S. Lui, K. Nahrstedt, and S. Chen, “Hierarchical QoS Routing in Delay-Bandwidth Sensitive Networks,” in *25th Annual IEEE Conference on Local Computer Networks*, 2000, pp. 579–588.
- [4] W. C. Lee, “Topology Aggregation for Hierarchical Routing in ATM Networks,” *ACM SIGCOMM Computer Communication Review*, vol. 25, no. 2, pp. 82–92, 1995.
- [5] F. Hao and E. W. Zegura, “On Scalable QoS Routing: Performance Evaluation of Topology Aggregation,” in *21st IEEE Conference on Computer Communications (INFOCOM)*, 2000, pp. 147–156.
- [6] V. Sarangan, D. Ghosh, and R. Acharya, “Capacity-aware state aggregation for interdomain qos routing,” *IEEE Transactions on Multimedia*, vol. 8, no. 4, pp. 792–808, 2006.
- [7] S. Jeon, “Topology aggregation method for multiple link parameters,” in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM)*, 2004, pp. 1378–1382.
- [8] I. T. Okumus, H. A. Mantar, J. Hwang, and S. J. Chapin, “Inter-Domain QoS Routing on Diffserv Networks: A Region-Based Approach,” *Computer Communications*, vol. 28, no. 2, pp. 174–188, 2005.
- [9] K.-S. Lui, K. Nahrstedt, and S. Chen, “Routing with topology aggregation in delay-bandwidth sensitive networks,” *IEEE/ACM Transactions on Networking*, vol. 12, no. 1, pp. 17–29, 2004.
- [10] R. Hou, K.-S. Lui, K.-C. Leung, and F. Baker, “Routing with QoS information aggregation in hierarchical networks,” in *17th International Workshop on Quality of Service (IWQoS 2009)*. IEEE Computer Society, July 2009, pp. 1–9.
- [11] —, “Aggregation-Based QoS Routing in the Internet,” *Journal of Communications*, vol. 5, no. 3, pp. 239–246, 2010.
- [12] A. Orda and A. Sprintson, “Precomputation schemes for QoS routing,” *IEEE/ACM Transactions on Networking*, vol. 11, no. 4, pp. 578–591, August 2003.
- [13] W. S. Lai, “Hierarchical routing with QoS,” in *3rd International ICST Conference on Simulation Tools and Techniques (SIMUTools)*. Brussels, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2010, pp. 1–7.
- [14] Z. Li and P. Mohapatra, “QRON: QoS-Aware Routing in Overlay Networks,” *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 1, pp. 29–40, January 2004.
- [15] J. Zhang, Y. Han, and L. Wang, “New topology aggregation mechanisms for delay-bandwidth sensitive networks,” in *11th IEEE Singapore International Conference on Communication Systems (ICCS)*, November 2008, pp. 737–742.
- [16] W.-Y. Tam, K.-S. Lui, S. Uludag, and K. Nahrstedt, “Quality-of-Service routing with path information aggregation,” *Computer Networks*, vol. 51, no. 12, pp. 3574–3594, 2007.
- [17] S. Dragoş and M. Collier, “Macro-routing: a new hierarchical routing protocol,” in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM)*, November/December 2004, pp. 1510–1514.
- [18] P. S. Sapaty, *Mobile Processing in distributed and Open Environments*. Wiley, 2000, ISBN: 0-471-19572-3.
- [19] S. T. Vuong and I. Ivanov, “Mobile Intelligent Agent Systems: WAVE vs. JAVA,” in *1st Annual Conference of Emerging Technologies and Applications in Communications (etaCOM'96)*, Portland, Oregon, May 1996.
- [20] K. N. S. Uludag a K.-S. Lui and G. Brewster, “Analysis of Topology Aggregation Techniques for QoS Routing,” *ACM Computing Surveys (CSUR)*, vol. 39, no. 3, 2007, article No. 7.
- [21] Y. Tang, S. Chen, and Y. Ling, “State aggregation of large network domains,” *Computer Communications*, vol. 30, no. 4, pp. 873–885, February 2007.
- [22] S. Dragoş and M. Collier, “Multi-constraint Macro-routing by using the Extended Full-Mesh aggregation technique,” in *2nd International Conference on Intelligent Computer Communication and Processing (ICCP)*, Cluj-Napoca, România, September 2006, pp. 229–234.
- [23] E. W. Zegura, K. L. Calvert, and S. Bhattacharjee, “How to Model an Internetwork,” in *IEEE Conference on Computer Communications (INFOCOM'96)*, vol. 2, San Francisco, California, USA, March 1996, pp. 594–602.
- [24] A. Juttner, B. Szviatovszki, I. Mecs, and Z. Rajko, “Lagrange Relaxation Based Method for the QoS Routing Problem,” in *IEEE Conference on Computer Communications (INFOCOM)*, 2001, pp. 782–791.