STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LIV**, Number 2, 2009

MACRO-ROUTING. PERFORMANCE EVALUATION

SANDA DRAGOŞ AND MARTIN COLLIER

Abstract. QoS routing is used in networks to find feasible paths that simultaneously satisfy multiple (QoS) constraints. The scaling difficulties in conventional shortest-path routing can be addressed using hierarchical routing and state aggregation. State aggregation gives rise to an approximate representation of the network, which can lead to inaccurate path selection. We evaluate in this paper our hierarchical routing protocol, called *Macro-routing*, that we introduced in [4]. *Macro-routing* can distribute the route computation efficiently throughout the network using mobile agents. This allows it to process more detailed information than in conventional hierarchical routing protocols and so increases the likelihood of finding the best path between source and destination.

1. Introduction

The migration of all types of communications services to packet-switched networks (notably the Internet) means that the proportion of traffic generated by real-time applications continues to rise. The Quality of Service (QoS) requirements of such applications can be met either by overprovisioning of the network, or by service differentiation. The latter solution will require ubiquitous network support for QoS routing. QoS routing is the process of identifying efficient paths that can satisfy QoS constraints (e.g. bandwidth, delay, delay variation).

The primary issue for QoS routing solutions in very large networks is *scalability* [10, 9, 2]. Unlike routing protocols such as BGP that are based on relatively static information (e.g. path hop count or AS count), QoS routing requires the frequent updating of dynamic network state information. The update messages consume significant network bandwidth. This amount of state information needs considerable storage space, while processing it requires significant computational power. A large scale deployment of QoS routing

Received by the editors: September 30, 2009.

2010 *Mathematics Subject Classification.* 68M12, 90B20.

1998 *CR Categories and Descriptors.* C.2.1. [**Network Architecture and Design**]: Network topology – *Hierarchical topology*; C.2.2. [**Network Protocols**]: Routing protocols.

Key words and phrases. QoS routing, hierarchical routing.

generates three main types of overhead: *communication*, *computation* and *storage*.

Topology aggregation [8] reduces the amount of routing information and the routing table sizes by orders of magnitude. However, aggressive aggregation methods may have a negative impact on routing performance [7].

2. Hierarchical QoS routing protocols

Private Network-to-Network Protocol (PNNI) [3] is the only QoS-aware hierarchical routing protocol standardized and implemented. It is used in ATM networks and allows up to 104 hierarchical levels. A drawback of PNNI is that the route computation load is distributed unevenly among the network nodes. Also, the aggregation process used in PNNI leads to inaccurate state information advertisements [7, 11] which can result in the inefficient utilization of network resources.

HDP [5] is a proposal for a hierarchical routing protocol within Multiprotocol Label Switching (MPLS) networks. It uses cluster-based server farms as managing nodes which collect all the routing information from their domains and compute them centrally. The setup time of a path is reduced by computing the routes within different domains on one level in parallel, at the expense of an increase in the number of messages [5]. Also, by starting the path computation at the top of the hierarchy and progressing downwards, the aggregation strategies may be used inefficiently as some routing information can already be obsolete by the time the protocol reaches the lower levels of the hierarchy.

In Viewserver [1], the path computation is done by the source node, which gathers centrally all the required routing information by traversing the hierarchy upwards (to find the parent "*view server*") and downwards (to collect detailed routing information about transit and destination domains). However, the setup time is long as the whole path is computed on a single node, and there are scalability concerns regarding the amount of state information gathered.

3. Macro-routing

We proposed in [4] a new protocol that addresses the problem of hierarchical QoS routing within MPLS networks. It is called *Macro-routing* because, being an inter-domain routing protocol, its routing decisions at the higher levels are macro-decisions, as opposed to the detailed or micro-decisions made at the lowest level of the hierarchy.

Macro-routing is capable of both routing and signalling functionalities which are accomplished by the use of mobile agents. Thus, instead of advertising state information, small mobile agents are dispatched to process such information at each node. The advantage of this approach is that the information used to compute routes can be much more detailed that 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 analyse 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.

3.1. **Protocol description.** The hierarchical organization of *Macro-routing* consists at the lowest level of a number of domains which are typically independent administrative areas. The nodes within such domains are physical network nodes (i.e. routers or switches). Each domain has a *managing node*, which must be able to interpret mobile agent code. It can either be selected from the nodes of the domain (as with PNNI) or it can be a distinct node (as in HDP). Its main function is to maintain an aggregated representation of the domain it is managing.

As the hierarchy is decided administratively, each domain at the lowest level of the hierarchy may choose its own routing strategy. For example it may use standard link-state methods, or may use mobile agents for route discovery. The latter method implies the existence of a mobile agent interpreter on each router or switch. The only *Macro-routing* requirement is that the managing node must contain an Full-Mesh aggregate representation of the managed domain. The maintenance of that aggregate representation is the responsibility of the domain administrator.

For the higher levels of the hierarchy the managing node creates an aggregated representation in four steps:

(1) Each border node and the source and destination nodes activates a mobile agent that floods the domain by replicating itself at each node. Its goal is to find all possible paths to all the other border nodes within the same domain. Each mobile agent records the path it follows and processes the routing information at each node. If one mobile agent is revisiting a node, or the path it has traversed to date does not satisfy the given QoS constraints, it will be discarded. If it reaches another border node it will transmit the path used and its cost to the managing node.
(2) The managing node chooses optimal paths between each pair of border nodes.

(3) A Full-Mesh aggregation topology is created using the selected paths. The costs of the selected paths will become *nodal costs* when computing paths at the next level of the hierarchy.
(4) Some or all of the other computed paths, which have not been selected for the aggregate representation, can be cached for recovery purposes or as alternative paths.

There are three major phases in the *Macro-routing* protocol whereby it finds and selects a QoS path from a given source to a given destination.

3.1.1. *Determination of participant domains.* The first phase involves determining the domains through which the path is likely to pass. It develops in two stages.

In the first stage, the source node initiates an "*upwards search*" in the hierarchy for the lowest level parent node, known as the root node, which has a view of both source and destination, as in HDP and Viewserver.

In the second stage, the parent node initiates a "*downwards search*" 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 the physical domains reached by this search will be *participant domains*.

3.1.2. *Path computation.* The second phase involves the determination of the path.

Every managing node in the *participant domains* will create its own aggregate representation by calculating routes between all domain border nodes.

Mobile agents released from each border node traverse the domain multiplying themselves in search of all possible paths that satisfy the QoS constraints. At the end of its journey (i.e. upon reaching a border node different from the starting one) each mobile agent would send the gathered information to the local managing node. Sending the information to the managing node can be done either by piggybacking on legacy messages or using mobile agents.

Starting from the second level of the hierarchy, *nodal costs* will be considered as well as link costs when computing the path cost[1]. The topmost domain will have as border nodes only the aggregated representation of the source and destination domains. The managing node of this domain will determine all the possible paths between its border nodes (the source and the destination) and

[1]Costs include link costs and nodal costs. Nodal costs represent the cost of traversing a virtual node. This nodal cost is zero at the lowest level of the hierarchy.

based on their costs it can determine the optimal path. The other paths found during this process can be used by fast recovery mechanisms or as alternative paths.

3.1.3. *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 overall path can be determined by traversing the hierarchy downwards and interrogating all the managing nodes along the chosen path about the detailed sub-paths across their domains.

The path set up and resource reservation can be done either by existing signaling 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. Advantages of using mobile agents instead of RSVP for the reservation process are:

- *availability:* Mobile agent support is already available in the nodes, so there is no need to deploy or configure additional software;
- *parallelism:* 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. Mobile agents can do the reservation in a parallel and distributed manner so that, from each hierarchical level, within each domain, a mobile agent can be dispatched to reserve resources corresponding to each logical/aggregated link.
- *hierarchical reservation:* We give the name "hierarchical reservation" to the process of reserving the resources for the overall path in a manner which corresponds with the hierarchical representation used by the routing protocol (i.e. Macro-routing). The first resources to be reserved are those corresponding to the links within the topmost (e.g., level k) domain within the hierarchy. If such resources are not available another level-k path will be selected until there is a path with available resources. Then, the process continues within the $k-1$ level domains which are represented by the virtual nodes traversed by the selected path at level k. The process stops when $k=0$. The advantage for such reservation strategy is that any unavailable sub-path can be substituted on the spot while all other resources (previously addressed) remain reserved. RSVP, however, performs sequential reservation. This means that any failure is reserving a resource will result in an overall failure, which requires finding a new source-to-destination path and starting the reservation process all over again.

Macro-routing works best in the MPLS context as the separation between the control and forwarding planes allows the coexistence of complex or different routing strategies. Setting up a hierarchical path in an MPLS network is straight-forward. That is because MPLS has the label stack capability. Thus, every sub-path within every domain and every hierarchical level can be treated independently. This can be done either by using a mobile agent-based "hierarchical reservation" or by using any other label distribution protocol.

When all the resources have been successfully reserved and the overall path has been set up, the request is served and the traffic may flow. In the case of resource unavailability or link/node failure, alternative paths which are already computed can be used for a fast recovery.

3.2. **Implementation details.** The *Macro-routing* protocol can be implemented using any mobile agent technology. We have chosen the Wave technology [12]. The main reason for selecting this particular technology is that its syntax make Wave code very compact, perhaps 20 to 50 times shorter that equivalent programs written in C++ or Java [13]. The use of Wave in MPLS networks for routing purposes has already been advocated in [6] to discover multi-point to point trees. Here it is used for hierarchical routing.

The two phases called (*Determination of participant domains* and *Path reservation and set-up*) have a straightforward implementation. The implementation details of the initial *Path computation* phase of the protocol were presented in [4].

4. Test results and performance evaluation

4.1. **The simulation model.** For all tests the *Georgia Tech Internetwork Topology Models* (GT-ITM) [14] was used to generate random network topologies. One example of such topologies is presented in Fig. 1. The corresponding values of number of nodes, number of links and connectivity degree for each topology used in our simulations are shown in Table 1.

One or multiple constraints were associated to each link. The main metrics used were *administrative cost* $\in [1, 15]$ and *hop count*[2] $\in [20, 30]$. For each link, the corresponding metric was randomly chosen.

The results in [4] were obtained by deploying real wave agents on a virtual (i.e., simulated) network running on a single host. Due to the complexity of the virtual networks required to evaluate multi-constrained Macro-routing (hierarchical networks with hundreds of nodes), we developed an application

[2]For the first hierarchical level, the hop count metric would be 1 for each link. However, for the upper hierarchical levels, this hop count can have greater values and differ from one link to another due to the aggregation process.

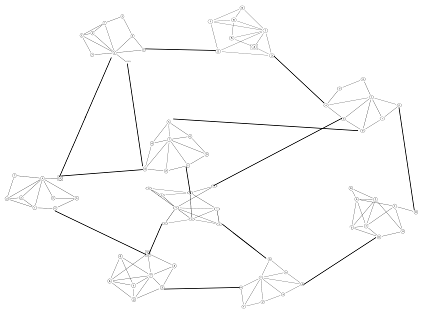


FIGURE 1. Two-level hierarchical topology similar with the ones used in simulation

TABLE 1. Details about two-level hierarchical topologies used in simulation

No. of nodes (N)	**No. of links (L)**	**Connectivity degree ($2L/N$)**
$4 \times 4 = 16$	24	3.00
$9 \times 9 = 81$	124	3.06
$10 \times 10 = 100$	149	3.98
$12 \times 12 = 144$	[216, 372]	[3, 5.166]
$13 \times 13 = 169$	282	3.33
$15 \times 15 = 225$	368	3.27
$20 \times 20 = 400$	600	3.00

to generate a list of the paths the waves would discover, rather than deploying waves on such networks.

4.2. **Macro-routing's communication overhead.** The communication overhead incurred by Macro-routing was evaluated in [4]. It was shown that many of the waves Macro-routing generates within high connectivity topologies end in cycles, and thus do not contribute to route discovery. We introduced a parameter called *lifespan*, which resembles the TTL field used in the IP protocol. Its purpose is to limit the number of *waves* generated during route search by reducing the number of generations which the parent *wave* can produce. The rationale for this is that the law of diminishing returns is assumed to apply - it is unlikely that an exhaustive search of every possible path is necessary to find the optimal path. The modified algorithm is no longer *guaranteed* to find the optimal path (and indeed will find *no* path if the destination is more than *lifespan* hops away). It was shown, however, that in a representative[3] network comprising $9 \times 9 = 81$ nodes, the optimal paths were found provided that $lifespan \geq 5$.

The following tests further explore the influence of the *lifespan* parameter on the performance of Macro-routing. Before presenting our test results, we first introduce some notation and terminology.

Definition Effort

Macro-routing's *effort* is the ratio of the number of ineffective waves which end up in cycles after n nodes have been visited to the number of waves which might find a path.

Definition Efficiency

Let C_{opt} be the optimal path cost between a specific source and destination, and C_{act} the actual path cost obtained by the (sub-optimal) lifespan-limited Macro-routing algorithm. Macro-routing's *efficiency* is then:

$$E = \frac{C_{opt}}{C_{act}}, \tag{1}$$

Definition Failure, success, best

Let E be Macro-routing's efficiency as defined in Definition 4.2. We define Macro-routing's results as:

- *failure*: no path is found (because of too low a lifespan value): $E = 0$
- *success*: paths satisfying the requirements are found, and: $E > 0$
- *best*: the paths found include the optimal path, i.e. : $E = 1$.

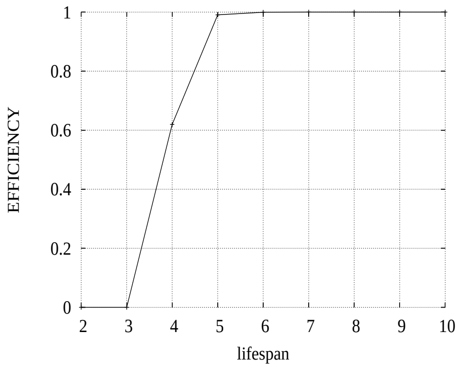
The tests were performed on two-level hierarchical networks with connectivity varying from 3 to 5.166 and $12 \times 12 = 144$ nodes. These topologies were divided into three different classes based on their connectivity degree[4] (c_d) varying in the following intervals:

(1) $c_d \in [3, 3.66]$
(2) $c_d \in [4, 4.5]$
(3) $c_d \in [5, 5.166]$

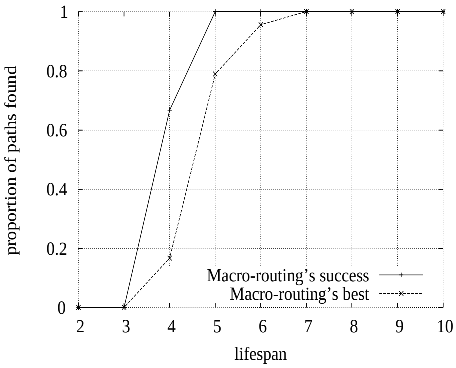
[3]Each domain from the two level hierarchical networks has an average node degree of 3.5 (i.e. $2 \cdot L/N = 3.5$).

[4]We will refer to the average node degree (i.e. $2 \cdot L/N$, where L is the number of links and N is the number of nodes) as *connectivity degree*.

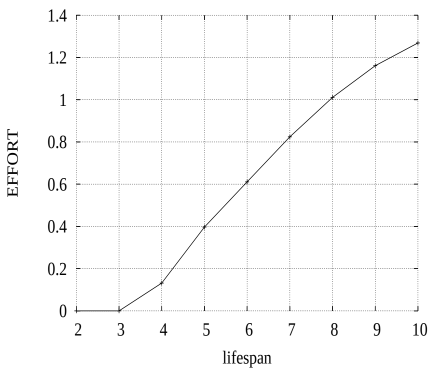
We ran 30 sets of tests on each hierarchical topology class. The mean values of the results obtained across the three classes are depicted in Fig. 2. They show that on the given topology (with $12 \times 12 = 144$ nodes) a lifespan value above 5 does not affect Macro-routing's efficiency in a significant manner (see Fig. 2(a) and 2(b)), while the number of waves is greatly reduced (see Fig. 2(c) for the ratio between the cycle and the alive waves and Fig. 2(d) for the total number of waves/link). Moreover, when the value of lifespan is set to 7 there is no loss in Macro-routing's efficiency, while its overhead (i.e. the number of waves) is considerably reduced.



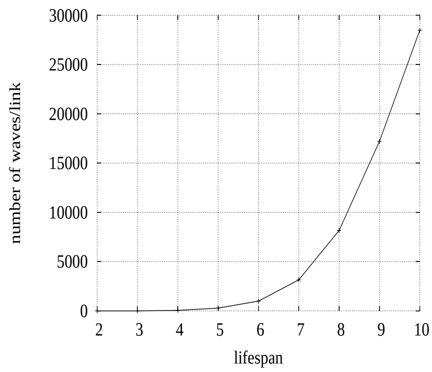
(a) Macro-routing's Mean **Efficiency** (see Definition 4.2)



(b) Mean percentages of **success** and **best** Macro-routing paths found (see Definition 4.2)



(c) Macro-routing's Mean **Effort** (see Definition 4.2)



(d) Mean number of waves/link generated

FIGURE 2. Macro-routing's performance while applying the *lifespan*

The average communication overhead generated by waves, as depicted in Fig. 2(d) is significant for lifespan values above 5, considering the relatively small size of the network. This is because of the high network connectivity in these cases, as can be seen in Fig. 3, which shows the considerable variation in the number of waves generated on three different classes of topologies.

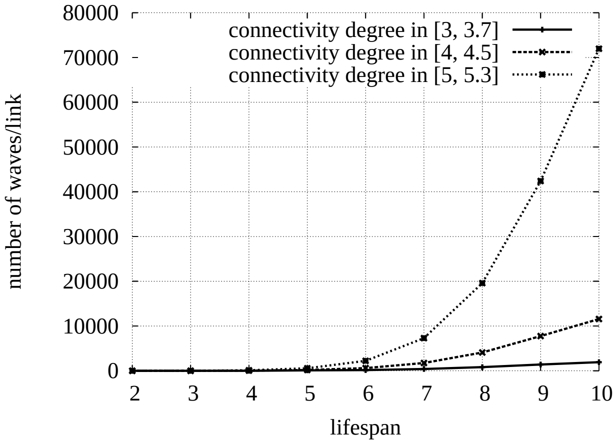


FIGURE 3. Macro-routing's communications overhead on different topologies

The number of waves generated by Macro-routing is overestimated in these results due to the assumption that all paths satisfy the constraints. Demanding constraints would significantly limit the number of compliant paths, and thus the wave population.

5. Conclusions

In this paper we proposed solutions for *Macro-routing*'s *scalability* in terms of storage, computational and communication overhead introduced by QoS routing. By using mobile agents, *Macro-routing* allows routes to be discovered rapidly without the imprecision introduced by topological state aggregation in other approaches. The price paid for this level of performance is that a large number of mobile agents (implemented as *waves*) traverse the network. However, the level of wave traffic can be restricted by limiting their *lifespan*. Tests run on the two level hierarchical networks showed that there exists a threshold over which this parameter significantly reduces the communication overhead without impairing the performance of our protocol. Moreover, the size of one *wave* packet is significantly smaller (i.e., less than one KB) compared with the link-state packets used to distribute QoS routing information (see details about the PNNI packet size in [3]). Also, due to the dynamic nature of the routing information, such link-state packets need to be broadcast frequently. In contrast, waves are only dispatched when needed. Thus, it can be tentatively assumed that the communication overhead generated by the wave population is, at worse, no greater than the overhead generated by the link-state packets when centralized QoS mechanisms are used. This quantitative argument will need to be confirmed by quantitative studies, which is a topic for future research.

Acknowledgements

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

References

[1] C. Alaettinoglu and A. U. Shankar, *The Viewserver Hierarchy for Interdomain Routing: Protocols and Evaluation*, IEEE Journal of Selected Areas in Communications, 13 (1995), pp. 1396–1410.

[2] G. Apostolopoulos, R. Guerin, S. Kamat, and S. K. Tripathi, *Quality of service based routing: A performance perspective*, in ACM SIGCOMM 98, Vancouver, Canada, August 1998, pp. 17–28.

[3] ATM Forum, *Private network-network interface specification , version 1.0*, Tech. Report af-pnni-0055.000, ATM Forum, March 1996.

[4] S. Dragos and M. Collier, *Macro-routing: a new hierarchical routing protocol*, in Proceedings of the IEEE Global Telecommunications Conference (Globecom), November/December 2004.

[5] M. El-Darieby, D. C. Petriu, and J. Rolia, *A Hierarchical Distributed Protocol for MPLS path creation*, in 7th IEEE International Symposium on Computers and Communications (ISCC'02), July 2002, pp. 920–926.

[6] S. Gonzalez-Valenzuela and V. C. M. Leung, *QoS Routing for MPLS Networks Employing Mobile Agents*, IEEE Network Magazine, 16 (2002), pp. 16–21.

[7] R. Guerin and A. Orda, *QoS-based Routing in Networks with Inaccurate Information: Theory and Algorithms*, in IEEE Conference on Computer Communications (INFOCOM'97), April 1997, pp. 75–83.

[8] F. Hao and E. W. Zegura, *Scalability techniques in qos routing*, Tech. Report GIT-CC-99-04, College of Computing, Georgia Institute of Technology, 1999.

[9] ———, *On Scalable QoS Routing: Performance Evaluation of Topology Aggregation*, in 21st IEEE Conference on Computer Communications (INFOCOM'00), vol. 1, June 2000, pp. 147–156.

[10] S. S. Lee, S. Das, G. Pau, and M. Gerla, *A Hierarchical Multipath Approach to QoS Routing:Performance and Cost Evaluation*, in IEEE International Conference on Communications (ICC'03), Anchorage, AK, USA, May 2003.

[11] W. C. Lee, *Topology Aggregation for Hierarchical Routing in ATM Networks*, ACM SIGCOMM Computer Communication Review, 25 (1995), pp. 82–92.

[12] P. S. Sapaty, *Mobile Processing in distributed and Open Environments*, Wiley, 2000.

[13] 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.
[14] 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.

Babeş-Bolyai University, Department of Computer Science, 400084 Cluj-Napoca, Romania
E-mail address: sanda@cs.ubbcluj.ro

Dublin City University, School of Electronic Engineering, Dublin 9, IRELAND
E-mail address: collierm@eeng.dcu.ie