A Framework for Multi-Class Based Multicast Routing

Maria João Nicolau[1], António Costa[2], and Alexandre Santos[2]

[1] Departamento de Sistemas de Informação,
Universidade do Minho, Campus de Azurém,
4800 Guimarães, Portugal
Phone: +351 253 604442
Fax: +351 253 604471
joao@uminho.pt
[2] Departamento de Informática,
Universidade do Minho, Campus de Gualtar,
4710 Braga, Portugal
{costa,alex}@uminho.pt

Abstract. The goal of Differentiated Service Architecture is to provide the benefits of different CoS levels while avoiding the limitations of the IntServ model. This is accomplished by aggregating traffic into specific classes, thus changing the scope from QoS (per flow) to CoS (per class) guarantees.
In presence of DiffServ networks, per flow path computation is not adequate. Instead, per class path calculation must be made and so multiple paths (unicast routing) or trees (multicast routing), must be computed in order to satisfy the different QoS requirements of different traffic classes.
This paper presents a new multicast routing strategy enabling per class multicast tree computation. The proposed heuristics enable directed trees establishment, instead of reverse path ones, due to the importance of link asymmetry within an environment which is, essentially, unidirectional.

1 Introduction

As well as for unicast routing, there are two different approaches in order to provide QoS to multicast routing: per flow and per class routing.

The first one performs routing at flow level. Several strategies have been proposed [1] [2], [3], most of them relying on flooding in order to find a feasible tree branch to connect a new member. The underlying idea is to obtain multiple paths where a new member may connect to the tree. Among candidate paths the new member selects the one that is able to satisfy its QoS requirements.

This strategy is suited within the Integrated Services model (IntServ) [4] that aims to provide QoS service guarantees for each individual flow crossing the network, by means of resource allocation, but it does not fit in presence of DiffServ networks[5].

Most differentiated services implementation proposals make use of control algorithms for aggregating service levels, packet marking and policing, and preferential treatment of marked packets within the network. The issue of routing as a means to enhance aggregate QoS has not yet received the necessary attention.

In this paper a new multicast routing strategy is proposed enabling per class multicast routing implementations. Class based multicast routing is only accomplished within an Autonomous System and questions related with inter-AS routing are considered to be dealt by Border Gateway Multicast Protocol (BGMP) [6], MBGP [7] (Multiprotocol Extensions for BGP-4) or related protocols. The proposed model takes link asymmetry into account as it defines a *shortest-path-tree* based routing strategy as opposite to a *reverse-path-tree* based one.

2 A model for Multi-Class Based Multicast Routing

Sources are the elements in better conditions to define QoS requirements since they are the ones generating traffic. Having multiple sources per group, with perhaps different QoS requirements, one may have different data flows of different classes of service. In this situation, receivers must join a group with no restrictions in terms of traffic classes they are able to receive. Furthermore, they must be able to receive all classes of all sources, at least in the starting period of group membership.

A multiple shared tree mechanism is proposed, with trees rooted at a Rendezvous Point (RP) router, inspired in Protocol Independent Multicast-Sparse Mode (PIM-SM) [8]. A shared tree per class of service available is needed, in order to give sources the ability to start sending data in any class. It is assumed that the total number of classes of service "available" has a pre-established upper limit and is small as compared to the number of participants.

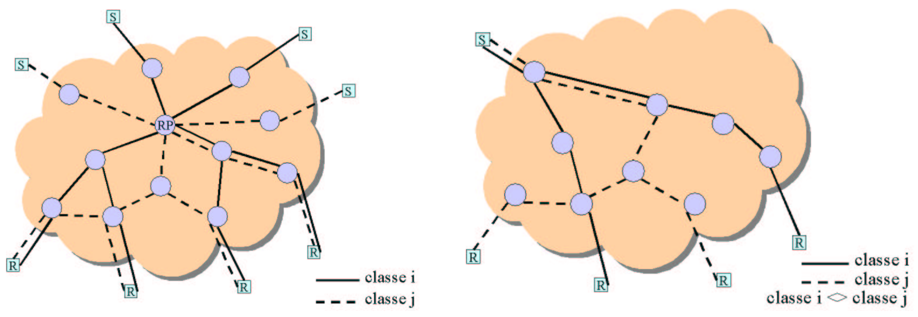
Data packets originated by sources are sent towards the RP router, previously marked according to source defined QoS parameters. The RP router forwards data packets from sources through one of the shared trees, based on their class of service. Receivers must connect to all of the RP shared trees when joining the group. This mechanism is illustrated in figure 1(a).

At this point, the question lies on how to build several distinct shared trees, one per class of service. Explicit join requests must be sent by the receivers towards the Rendezvous Point router. When RP router receives a join request it must send back to the new receiver an acknowledgment packet per class through the best path for that class. Routers, along those paths, receiving such an acknowledgment packet may then update their routing table in order to build new multicast trees branches. Updating is done basically by registering with the multicast routing entry for that tree, the acknowledge packet's incoming and outgoing router interfaces.

The multiple RP shared tree mechanism, presented so far, does not really allow receivers to specify their own QoS requirements. Traffic flows from sources to receivers through one of the shared trees, according only to the QoS parameters defined by sources. How can a receiver, after a starting period, specify a given requirement? Can a receiver demand for a reclassification of a source multicast traffic?

This issue cannot be accomplished by a shared tree, but it may be met if the receiver joins a source-based tree. When initiating the join procedure, the receiver should include in the join request the desired Class of Service. It is up to the source to decide whether or not to accept the join, knowing that when accepting a join, traffic in the requested class of service must be generated. Of course it is up to the source to define the maximum number of classes it will be able to handle.

In this situation, each source may face several distinct requests of several distinct receivers for different classes of service within the same group. At the limit, for larger groups, there may be requests for all classes. Even with this worst case situation scalability problems do not arise because the total number of different classes will be much smaller than the total number of receivers. In practice this implies one source-based tree per class of service, unless some order relationship between the classes can be established.



(a) QoS parameters defined by sources

(b) QoS parameters defined by receivers

Fig. 1. *QoS parameters defined sources or receivers*

When accepting a join for a new Class of Service, a source must generate an acknowledgment marked with the same class mark, addressed to the corresponding receiver. This procedure is similar to the one described for the construction of the shared trees. If no relationship can be established between different classes of service, the mechanism will construct multiple source trees, one per class of service as illustrated in figure 1(b).

In a more realistic scenario, sources and receivers will both want to specify their own QoS requirements. Since anyone can join and leave the group at any time, QoS requirements cannot be pre-negotiated. In addition, not much knowledge about group is available at join time in order to allow for a reasonable QoS specification. This means that receivers will join the group without knowing exactly what classes of service are currently being provided by sources.

Therefore, the multiple Rendezvous Point shared tree mechanism described is the preferable default mechanism. Receivers initially join all the available RP shared trees, one per Class of Service, in order to receive traffic from any source in their classes. Sources flow their data packets to RP, marked according to their own requirements. Then receivers may try to join any source tree, specifying their own requirements, which may or may not be accepted by sources.

Figure 2 illustrates a scenario with four receivers initially connected to a RP router by two different shared trees, constructed for *class i* and *class j* respectively. There are also four sources, two of them generating traffic marked for *class i* and the other two generating traffic for *class j*. After a short period of time, the receiver R2 decides to

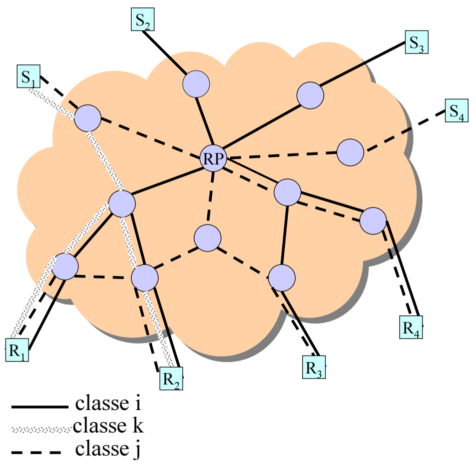


Fig. 2. *QoS parameters defined by sources and receivers*

join source `S1`, requesting a new class of service `k`. That single request originates a new source tree, rooted at source `S1` for Class of Service `k`.

This strategy, fully presented in [9], is now being tested with Network Simulator extended with PIM-SM implementation[10].

3 Discussion

The model proposed in this paper addresses multicast routing with a new approach: not only it was designed with the Class of Service paradigm in mind, but was also based upon the idea that both sources and receivers may dictate their own CoS parameters in that paradigm. Flexibility is therefore the main advantage of the model: it fits into distinct environments.

Because Differentiated Services are inherently unidirectional, we propose the usage of source and Rendezvous Point (RP) directed trees instead of typical reverse path forwarding ones. The heuristic is based upon explicit join acknowledges sent by either source or RP routers in response to explicit join requests sent by receivers. Furthermore the multicast framework presented tends to distribute multicast traffic in a evenly fashion as multiple shared-trees will distribute traffic along different links, instead of concentrating traffic on a smaller number of links (those in the single tree). So, although indirectly, this framework is also useful for network traffic engineering proposes as multicast load balancing becomes manageable.

Class differentiation is mainly achieved by means of usual DiffServ mechanisms *plus* routing differentiation, making it possible to use different routes for different Classes of Service. Generic class characteristics and identifications are directly derived from its DiffServ counterparts. Far from conflicting with traditional inside node differentiation strategies, this proposal is their routing-level complement.

The focus presented here is the functional model in what it respects to integrate CoS routing announcements and path calculation in a class based DiffServ-like environment.

Metrics to be used to characterize class based routing are being detailed, although scenarios taking into account important tuples like $< bandwidth, delay >$ have been used.

References

1. Shigang Chen, Klara Nahrstedt, and Yuval Shavitt. A qos-aware multicast routing protocol. In *INFOCOM (3)*, pages 1594–1603, 2000.
2. Michalis Faloutsos, Anindo Banerjea, and Rajesh Pankaj. Qosmic: Quality of service sensitive multicast internet protocol. In *SIGCOMM*, pages 144–153, 1998.
3. L. Guo and I. Matta. QDMR: An efficient qoS dependent multicast routing algorithm. In *Proceedings of the Fifth IEEE Real-Time Technology and Applications Symposium (RTAS '99)*, pages 213–223, 1999.
4. A. Mankin, Ed., F. Baker, B. Braden, S. Bradner, M. O'Dell, A. Romanow, A. Weinrib, and L. Zhang. Resource ReSerVation protocol (RSVP) – version 1 applicability statement some guidelines on deployment. Request for Comments 2208, Internet Engineering Task Force, September 1997.
5. S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An architecture for differentiated service. Request for Comments 2475, Internet Engineering Task Force, December 1998.
6. D. Thaler, D. Estrin, and D. Mayer. Border gateway multicast protocol (bgmp): Protocol specification, November 2000. Internet draft-ietf-bgmp-spec-02.txt.
7. T. Bates, R. Chandra, D. Katz, and Y. Rekhter. Multiprotocol extensions for BGP-4. Request for Comments 2283, Internet Engineering Task Force, February 1998.
8. D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobson, C. Liu, P. Sharma, and L. Wei. Protocol independent multicast-sparse mode (PIM-SM): protocol specification. Request for Comments 2362, Internet Engineering Task Force, June 1998.
9. Maria João Nicolau, António Costa and Alexandre Santos. Multi-Class Multicast Routing. Technical report, Universidade do Minho, 2001.
10. Maria João Nicolau and António Costa. Experimentando o PIM-SM no Network Simulator. Technical report, Universidade do Minho, 2001. *In Portuguese.*