Design and Evaluation of a Multi-class Based Multicast Routing Protocol

[1] Departamento de Sistemas de Informação,
Universidade do Minho, Campus de Azurém,
4800 Guimarães, Portugal
`joao@dsi.uminho.pt`
[2] Departamento de Informática,
Universidade do Minho, Campus de Gualtar,
4710 Braga, Portugal
`{costa,alex}@di.uminho.pt`

Abstract. Most of current multicast QoS routing proposals are based on the principle that QoS routes must be computed for each request, where requests explicitly express their resource requirements. As a result, within this environment, the goal of QoS routing is to satisfy individual request requirements, resorting to resource reservation to maintain those requirements after a feasible path has been found. This type of strategy is suited within the IntServ model but does not seem adequate in presence of DiffServ networks. According to DiffServ model, traffic flows are aggregated into specific classes-of-service and each flow receives a specific treatment accordingly to its class-of-service. There are no per flow guarantees, only per class differentiation. In this environment instead of per flow path computation, per class path calculation should be made, and so, within multicast scenarios, multiple multicast trees must be computed in order to satisfy different QoS requirements of different traffic classes.

This paper presents a new multicast routing protocol 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. The proposed protocol is implemented and simulated using Network Simulator. A set of simulation results are presented, analyzed and compared against PIM-SM, a widely deployed multicast routing protocol.

1   Introduction

The main goal of multicast routing protocols is to build a distribution tree or a set of trees in order to deliver data packets from sources to a set of receivers in an efficient manner, without incurring into network overloads. To minimize the resource usage in the network the multicast tree built should be the tree with minimum cost. The problem of finding such a tree is NP-complete and

is called *Steiner Tree Problem*[1]. There are many heuristics in finding a sub-optimal Steiner tree [2]. When QoS is considered, besides the connectivity, the tree branches between the sources and each receiver should satisfy the QoS constraints which turns the problem of build a multicast tree even more complex.

Several strategies have been proposed to implement QoS Multicast Routing, most of them relying on flooding in order to find a feasible tree branch to connect a new member. QoSMIC[3] and QRMP [4] are examples of those strategies. The underlying idea is to obtain multiple paths where a new member may connect to the tree. Typically, multiple probe messages are sent over different possible routes collecting QoS information on the path. Among candidate paths the new member selects the one that is able to satisfy its QoS requirements. This type of strategy is better suited within the IntServ model[5]. The main strength of the IntServ model is its ability to provide service guarantees by means of (state-full) resource reservation. However it has several weaknesses too. Each router is required to maintain state information for each flow, thus, scalability problems do arise in operational environments. In addition a significant amount of processing overhead is required within each router, and the connection setup time may even sometimes be greater than the time required for the transmission of all the packets belonging to a specific flow. The alternative model is called DiffServ[6]. According to this new model, the traffic is aggregated into specific classes-of-service thus changing the scope from per flow guarantees to per class differentiation. Before entering a DiffServ domain packets are marked by border routers (or ingress routers) in one of the available classes-of-service. Inside domain, core routers just give them a specific treatment accordingly to its class-of-service. There are difficulties and challenges when trying to adequate multicast protocols to DiffServ model. The main assumption behind this work is that in presence of DiffServ networks, per flow computation is not adequate. Instead of that a per class path calculation must be made.

In this paper a new multicast routing protocol is proposed enabling per class multicast routing implementation. The proposed protocol takes link asymmetry into account as it defines a *shortest-path-tree* based routing strategy as opposite to a *reverse-path-tree* based one. This is an important feature because when routing constraints are introduced links become asymmetric in terms of the quality of service they may offer, thus link costs are likely to be different in each direction.

2   A Model for Multi-class Based Multicast Routing

The main objective of this work is to propose a new multicast routing protocol that enables per class-of-service multicast routing implementation. The key ideas of the protocol are:

- Build multiple trees, one per class of service. Within a DiffServ multicast scenario, multiple multicast forwarding trees may be found, one per Class of Service (CoS), in order to comply with different per-class Quality of Service

(QoS) requirements. The main objective of this work is to study the viability
and efficiency of such an approach.

– Implementation of a directed-tree based routing strategy, instead of a
reverse-path-tree one. We believe that reverse path routing is not adequate
to address Quality of Service Routing. Links are asymmetric in terms of the
quality of service they offer, which makes reverse path routing not suited to
implement QoS routing.

– Use both shared trees and source based trees. In PIM-SM the use of both,
shared and source based trees, is proposed. It allows nodes to initially join a
shared tree and then commute to source based trees if necessary. The same
idea is used in the proposed protocol.

Besides, the proposed protocol is aligned with current IP multicast model
since it allows that sources and receivers may join or leave at any time and no
previous group membership knowledge is assumed.

2.1   MCMRP Tree Construction Algorithm

First, a multiple shared tree mechanism is proposed in order to give receivers
the ability to join the group without knowing where the sources are located.

The multiple shared tree mechanism proposed is inspired in Protocol Indepen-
dent Multicast-Sparse Mode (PIM-SM)[7] with trees rooted at a Rendez-Vous
Point (RP) router. 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 when compared to the number of participants. Data packets
originated by sources are sent towards the RP router, previously marked ac-
cording 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.
When a new receiver decides to join, the designated router sends an explicit
join request towards the RP router. The routers along the way between the new
receiver and the RP just forward the join request message and no sate informa-
tion is introduced in these routers. When the RP receives a join request message
from a new receiver it must send one join acknowledge message per class of
service. These messages must travel towards the new receiver through the best
unicast path per each class of service. Routers, along those paths, receiving such
acknowledge message may then update their routing tables in order to build
new multicast trees branches. Updating is done basically by registering with the
multicast routing entry for that tree, the acknowledge message's incoming and
outgoing router interfaces.

The process of joining the shared tree in MCMRP is detailed in Figure 1,
where variables and flags have the same meaning as defined in PIM-SM[7]. In
the illustrated scenario there are two different classes of service ($CoS = 1$ and
$CoS = 2$) and router A (the designated router of the new receiver) issues a join
request message. The routing table entries have the same fields as the PIM-SM
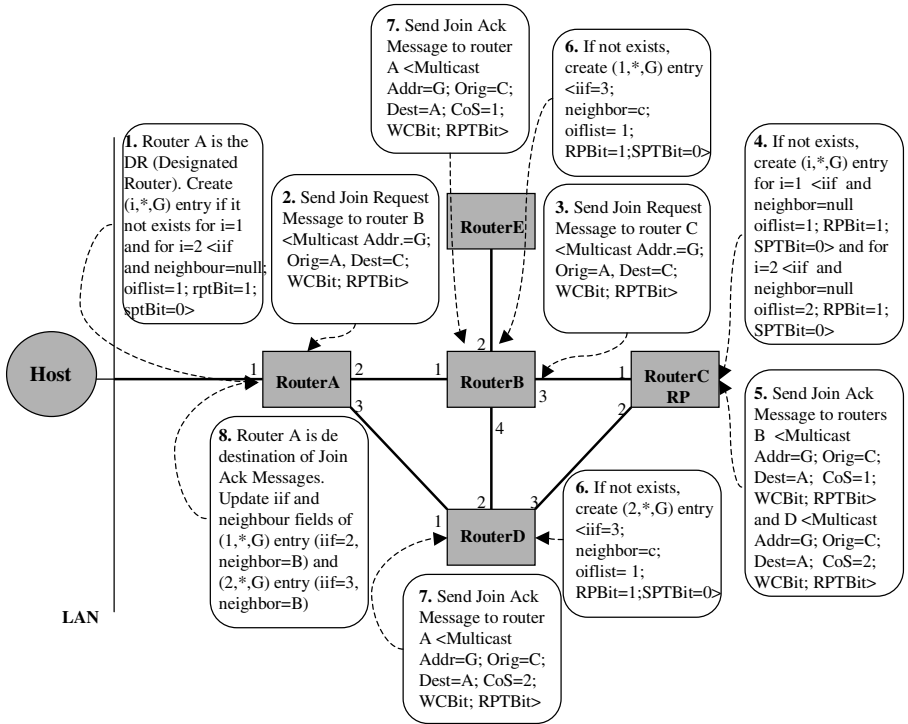
Fig. 1. Shared Trees Set Up. Actions are numbered in the order they occur.

ones, and an extra one: the upstream neighbor in the tree. This field has been introduced in order to implement the prune mechanism.

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. After a starting period a receiver may demand for a reclassification of 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 to source 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. 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.

When accepting a join for a new Class of Service, a source must generate an acknowledge message, addressed to the corresponding receiver. This procedure is similar to the one described for the construction of the shared trees. But in this situation only one join acknowledge message is generated per join request. Two different situations may occur. The receiver may decide to switch to a source based tree in the same class used by the source, or it may want to switch to a source based tree requesting a different class of service. In the first case, when a router in the path between the source and the receiver receives the join acknowledge message, if it is not already in the source based tree it must create a (i,S,G) entry and copy the outgoing interfaces list from the (i,*,G) entry to the outgoing interfaces list of the (i,S,G) new entry. This is because, in the future, packets from source S will be forward based on this new entry. Besides, when a router lying between the source and the receiver starts to receive data from that source, it must issue a prune of that source on the shared tree of that class. This prune indicates that packets of the class of service i from this source must not be forwarded down this branch of the shared tree, because they are being received by means of the source based tree. This mechanism is implemented by sending a special prune to the upstream neighbor in the shared tree of the class i. When a router at the shared tree of the class i receives this type of prunes, it creates a special type of entry (an (i,S,G)RPT-bit entry) closely like a PIM-SM router. In MCMRP the outgoing interface list of the new (i,S,G)RPT-bit entry is copied from the (i,*,G) entry and the interface deleted is the one being used to reach the node that had originated the prune, which may not be the arriving interface of the prune packet. This is because in MCMRP there are directed trees not reverse path ones. These (i,S,G)RPT-bit entries must be updated too when a join acknowledge message arrives in order to allow the join of a new receiver on a shared tree with source-specific prune state established.

When a receiver decides to join a source requesting a different class of service, the process is different. When a new (i,S,G) entry is created, the outgoing interface list should not be copied from the (i,*,G) entry, because in this case the other receivers connected through the corresponding shared tree still want to receive data packets in the source's default class of service. For the same reason these entries should not be updated when a posterior join to shared tree acknowledge message is received. In addition, the "prune of source in the shared three" mechanism must be triggered by the Designated Router when it receives the join acknowledge message. The prune messages must be sent to the shared trees of all classes except to the shared tree of the class for which the receiver commuted. This is because the receiver will start to receive the source's packets through the source tree in the desired class, so it can not continue to receive it by the shared tree of the source's default class of service.

The process of switching from the shared tree to a source based tree in MCMRP is detailed in Figure 2. In the illustrated situation the receiver decides to switch to a source based tree in class of service 1 ($CoS = 1$), supposing the source's default class of service is 2.
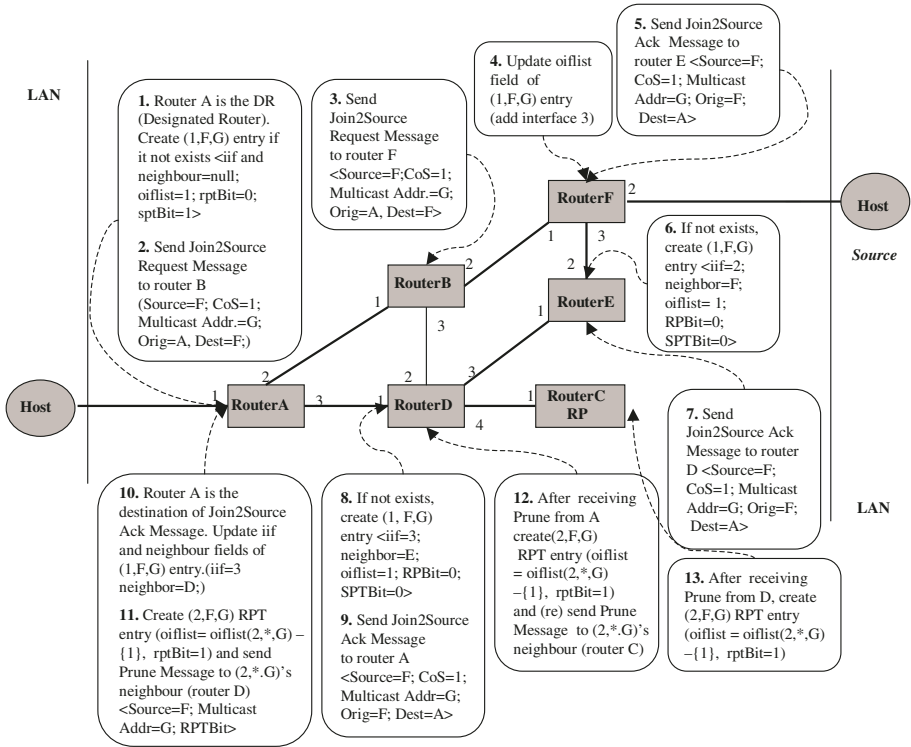
LAN

1. Router A is the DR (Designated Router). Create (1,F,G) entry if it not exists <iif and neighbour=null; oiflist=1; rptBit=0; sptBit=1>

2. Send Join2Source Request Message to router B (Source=F; CoS=1; Multicast Addr.=G; Orig=A, Dest=F;)

3. Send Join2Source Request Message to router F <Source=F;CoS=1; Multicast Addr.=G; Orig=A, Dest=F>

4. Update oiflist field of (1,F,G) entry (add interface 3)

5. Send Join2Source Ack Message to router E <Source=F; CoS=1; Multicast Addr=G; Orig=F; Dest=A>

RouterF

Host

Source

6. If not exists, create (1,F,G) entry <iif=2; neighbor=F; oiflist= 1; RPBit=0; SPTBit=0>

RouterB

RouterE

Host

RouterA

RouterD

RouterC RP

7. Send Join2Source Ack Message to router D <Source=F; CoS=1; Multicast Addr=G; Orig=F; Dest=A>

LAN

10. Router A is the destination of Join2Source Ack Message. Update iif and neighbour fields of (1,F,G) entry.(iif=3 neighbor=D;)

11. Create (2,F,G) RPT entry (oiflist= oiflist(2,*,G) – {1}, rptBit=1) and send Prune Message to (2,*.G)'s neighbour (router D) <Source=F; Multicast Addr=G; RPTBit>

8. If not exists, create (1, F,G) entry <iif=3; neighbor=E; oiflist=1; RPBit=0; SPTBit=0>

9. Send Join2Source Ack Message to router A <Source=F; CoS=1; Multicast Addr=G; Orig=F; Dest=A>

12. After receiving Prune from A create(2,F,G) RPT entry (oiflist = oiflist(2,*,G) –{1}, rptBit=1) and (re) send Prune Message to (2,*.G)'s neighbour (router C)

13. After receiving Prune from D, create (2,F,G) RPT entry (oiflist = oiflist(2,*,G) –{1}, rptBit=1)

Fig. 2. Switching from Shared Tree to Source Based Tree

3 Multi-class Based Multicast Routing Protocol Implementation

MCMRP implementation is based on Directed Trees Multicast Protocol (DTMP). DTMP [8] is a multicast routing protocol that builds directed trees instead of reverse-path-ones. A complete description of this protocol, with implementation details and comparative results with PIM-SM may be found in [8]. DTMP uses both shared trees and source based trees, like PIM-SM, in order to get the advantages of the both strategies. It is suited for use in asymmetric networks where link costs between any two nodes are different in each direction.

Another major element of MCMRP is the unicast routing protocol in use. Although MCMRP is independent of the underlying unicast routing protocol, it must be a multi-class enabled unicast routing protocol. In other words, the unicast routing protocol must be able to find the unicast routes that can meet the QoS requirements of each Class of Service. In order to build a new tree branch for each Class of Service the multicast routing protocol will search the unicast routing table for the unicast path that is more adequate to satisfy the QoS

requirements of each class. To accomplish this new feature, a new unicast routing protocol called CoSLSP (Class of Service Link State Protocol) was implemented.

CoSLSP aims to provide a class based unicast routing mechanism. The basic idea is to find one route per class-of-service, able to satisfy the QoS requirements of that class. Apart from the goal of satisfying the QoS requirements of each class, this protocol also addresses the problem of optimizing network utilization. Therefore, instead of computing just the routes that might meet the QoS requirements of each class, CoSLSP tries to find the shortest path that might satisfy those requirements. It is a unicast link-state protocol that uses a modified Dijkstra algorithm capable of finding the shortest path routes, if they exist at all, that can meet the QoS requirements of different classes of service. In few words: the path calculation algorithm starts by finding the shortest path, whose feasibility is then verified against the QoS requirements. If infeasible, the next shortest path is then iteratively verified, until a feasible path is found or a configured threshold is reached. In this way, a different route is found for each class of service and it is installed in the routing table. The packet forwarding process has been modified too in order to lookup for the appropriate route depending on the class of service of each packet.

CoSLSP has been implemented and evaluated with Network Simulator (NS-2)[9]. The simulations results show that CoSLSP in case of network congestion is able to find "better" routes in respect to the QoS metrics of each class of service.

4   Simulation Analysis

NS[9] has been used in order to simulate MCMRP and its results have been compared with a PIM-SM implementation. In our simulations we used MCMRP with CoSLSP, and an implementation of PIM-SM with LS (a link-state unicast routing protocol implementation included in NS-2 distribution).

To evaluate MCMRP we used two different metrics. To measure the quality of the multicast trees built by MCMRP we used a metric combining the number of data packet replicas with the cost associated to each link traversed by each packet. The second thing we intend to measure is the gain of using class-of-service multicast routing. For this purpose we used the average packet drops occurred in the flows of each class-of-service.

4.1   Simulation Scenarios

The topology used in a simulation scenario is a typical large ISP network[10]. It includes 36 nodes, 18 of them are core nodes, and the other 18 are edge nodes. Each of the core nodes is connected with one edge node by a symmetric link with the cost 1. The core nodes are inter-connected with each other by 30 asymmetric links. There are different alternative paths with different costs between any pair of core nodes. Link costs are integers randomly chosen from the interval [1, 10].

Three different classes of service with different QoS requirements in terms of losses were considered. Class 1 does not support any losses, class 2 supports well 25% of losses and finally class 3 can deal with 50% of losses. Each link

has 3 physical queues (one per class) and two virtual queues corresponding to two different drop precedences. All queues are configured exactly in the same way in order to prevent inside node differentiation. Therefore the only class differentiation that can occur is caused by the action of the routing protocol.

For each simulation run, only one group is considered and the RP is randomly chosen within the set of all nodes. There are four fixed sources and each source generates traffic in a class-of-service randomly chosen. It is assumed that a single receiver is connected to each edge node in the topology and that all edge nodes have one potential receiver attached. At the beginning of the simulation there are no receivers joining the group. After an initial period, 9 receivers start to join the group building three shared trees rooted at RP, one per class of service. After all the receivers have joined 8 receivers randomly chosen, join the four different sources requesting a class-of-service randomly chosen too. This scenario is then kept till the end of the simulation. Before the simulation ends, all the receivers leave the group.

4.2   Simulation Results

Simulations results are presented in Figures 3 and 4, The results shown reflect the computed average after 100 independent simulations.

Figure 3 show the characteristics of the trees built with the two protocols (MCMRP and PIM-SM). The curves presented in Figure 3a, show the average tree cost in function of number of receivers. The tree cost is measured in term of number of replicas `times` the link cost. The curves presented in Figure 3b, show the total number of links in the topology that are involved in the multicast trees as a function of number of receivers.

The results shown in 3a bring to evidence that MCMRP constructs trees with lower costs than those created by PIM-SM. This is because MCMRP builds directed trees instead of reverse path trees. Note that CoSLSP does not choose the *best* unicast routing path, it chooses the *best* unicast path that can meet the QoS requirements of each class-of-service. Even with this characteristic the trees
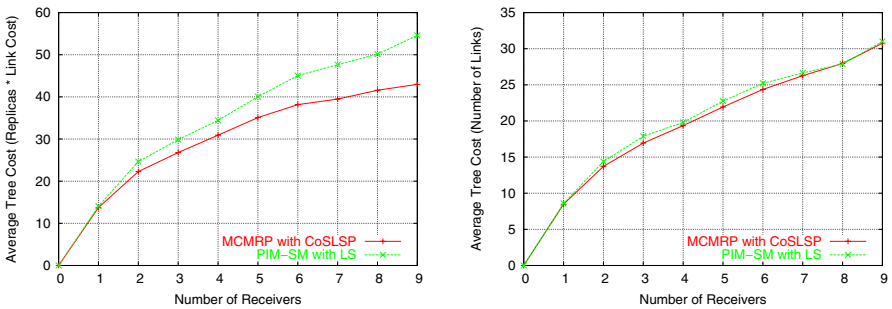
(a) *Tree Cost (packet replicas × link cost)* **(b)** *Number of Links in the multicast trees*

Fig. 3. Characteristics of the multicast trees

(a) *Flow Drops - Class 1*

(b) *Flow Drops - Class 2*

(c) *Flow Drops - Class 3*
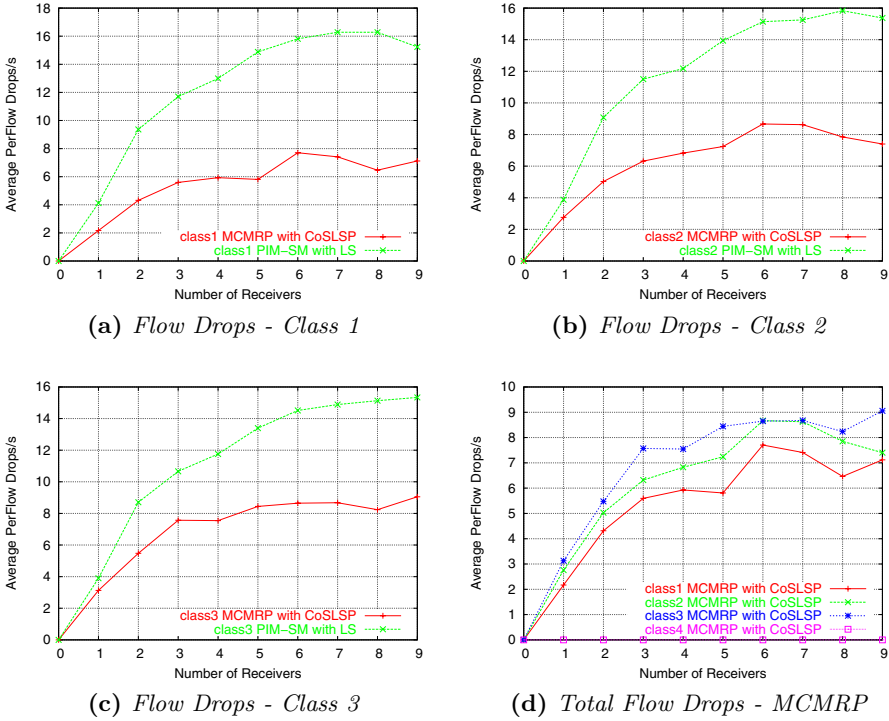
(d) *Total Flow Drops - MCMRP*

Fig. 4. Average packet drops

built by MCMRP are *better* in terms of total costs than trees built by PIM-SM. In addition, observing figure 3b we conclude that MCMRP is able to build *better* trees than PIM-SM without enlarging their size.

Figure 4 shows the average packet drops suffered in function of number of receivers. Figures 4a, 4b, 4c show the average packet drops occurred in the flows of each class of service when using the two protocols (MCMRP and PIM-SM). Figure 4d shows the results obtained for all the three classes, in terms of packet drops per flow, when using MCMRP.

These results demonstrate that when MCMRP is used a considerable less amount of drops is verified. This is because MCMRP try to find routes less congested when links became bottlenecks. In addition results show that MCMRP routing strategy promotes the expected differentiation between classes. Observing figure 4d we conclude that the average packet drops suffered by class 3 is greater than the average packet drops suffered by class 2 and the average packet drops suffered by class 2 is greater than the average packet drops suffered by class 1. This is because class 1 have the highest QoS requirements, followed by class 2 and finally class 3 is the least demanding one.

5 Discussion

A new protocol is presented in this paper, MCMRP - a multicast routing protocol that implements multi-class based multicast routing, to be used in a DiffServ environment. Because class differentiation is inherently unidirectional, we propose the usage of source and shared 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.

MCMRP has been implemented and tested with Network Simulator. The simulations results show that in presence of asymmetries within the network the MCMRP is a promising approach, enabling the establishment of directed multicast distribution trees with significant lower tree costs than either shared and source based trees created by PIM-SM. In addition, MCMRP, is able to find "better" trees in respect to the QoS metric of each class of service.

References

1. Winter, P.: Steiner problem in networks: A survey. Networks 17, 129–167 (1987)
2. Berman, P., Ramaiyer, V.: Improved approximations for the Steiner tree problem. In: Proceedings of the Third Symposium on Discrete Algorithms, pp. 325–334 (1992)
3. Faloutsos, M., Banerjea, A., Pankaj, R.: Qosmic: Quality of service sensitive multicast internet protocol. In: SIGCOMM, pp. 144–153 (1998)
4. Chen, S., Nahrstedt, K., Shavitt, Y.: A qos-aware multicast routing protocol. In: INFOCOM (3), pp. 1594–1603 (2000)
5. Mankin Ed., A., Baker, F., Braden, B., Bradner, S., O'Dell, M., Romanow, A., Weinrib, A., Zhang, L.: Resource ReSerVation protocol (RSVP) – version 1 applicability statement some guidelines on deployment. Request for Comments 2208, Internet Engineering Task Force (September 1997)
6. Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., Weiss, W.: An architecture for differentiated service. Request for Comments 2475, Internet Engineering Task Force (December 1998)
7. Estrin, D., Farinacci, D., Helmy, A., Thaler, D., Deering, S., Handley, M., Jacobson, V., Liu, C., Sharma, P., Wei, L.: Protocol independent multicast-sparse mode (PIM-SM): protocol specification. Request for Comments 2362, Internet Engineering Task Force (June 1998)
8. Nicolau, M.J., Costa, A., Santos, A., Freitas, V.: Directed Trees in Multicast Routing. In: Ajmone Marsan, M., Listanti, G.C.M., Roveri, A. (eds.) QoS-IP 2003. LNCS, vol. 2601, pp. 321–333. Springer, Heidelberg (2003)
9. Fall, K., Varadhan, K.: The NS Manual (January 2001), http://www.isi.edu/nsnam/ns/ns-documentation.html
10. Apostolopoulos, G., Guerin, R., Kamat, S., Tripathi, S.K.: Quality of service based routing: A performance perspective. In: SIGCOMM, pp. 17–28 (1998)