Abstract—Multicast admission control in Differentiated Services network is an important but shortly researched subject. We propose a parameter-based admission control method. The method rejects new multicast join requests that would otherwise decrease the quality experienced by the existing receivers. DiffServ network edge nodes filter join requests and generate new requests. The proposed method is developed as an extension to the DSMCast protocol but could also be adapted to other protocols. In this paper the parameter-based admission control is compared to earlier created, measurement-based admission control methods, as well to situation when no admission control is used.

I. INTRODUCTION

Different Quality of Service (QoS) architectures are nowadays widely used and researched. They provide better quality in terms of delay, jitter and packet loss. Differentiated Services (DiffServ) [3] has been the main interest of QoS researchers because it is currently the best solution for unicast QoS. For some applications like IPTV and video conferencing, multicasting is another way of providing better QoS by using efficiently network bandwidth. However, there are three major problems of transporting multicast traffic in DiffServ networks. The problems are heterogeneous trees [4]–[8], scalability [4], [8], [9] and neglected reserved sub-tree (NRS) [4]–[8] and they should be solved to attain guaranteed quality level for multicast traffic.

DSMCast [4] is an exceptional solution for these problems. It solves the problems of heterogeneous trees and scalability and gives a competent framework for solving the NRS-problem. This paper proposes admission control extension to the DSMCast protocol. The method is a distributed solution to the NRS-problem and it rejects the joining attempts that would degrade the QoS of the existing receivers more than is allowed. The method could be changed to inter-operate with any other protocol with only small modifications. The method was tested with a network simulator [10] and the results show the strengths of this method.

II. THE METHOD

The proposed method for multicast admission control solves the NRS problem described earlier. The method is distributed to the egress nodes of the network, which makes the core nodes and the whole network scalable. The basic idea is to filter the join requests arrived to the edge nodes and send some extra leave messages when necessary. A measurement based method is presented in [1] and in this paper a new method, parameter-based, is presented and described in detail.

The purpose of the method is to check if a specified flow can reach the requirements demanded. This means that client must define the traffic profile of the flow and the QoS parameters that it should fulfill. When a client wants to receive a multicast transmission, it first defines the limits for the maximum acceptable delay and the amount of bandwidth needed for the flow.

The admission control method can be divided in three phases and next subsections describe the tests, related to these phases. In the case no admission control is used, there won’t be any tests when joining to the multicast groups meaning that all the new members are welcome.

A. Edge Test

In the first test, the egress edge checks from its group states if it is already forwarding the group. If it is, the join request is accepted without any other tests. Otherwise, the edge moves to the next test before making any decisions.

B. Delay Test

In the delay-test the maximum delay that can occur, is approximated. Since each class is assumed to be regulated by the Token Bucket scheme with the bucket depth $\sigma_i$ and token rate $\rho_i$, the maximum end-to-end delay is calculated using equation 1 [2].

Parameter $F_j$ defines the frame size of node $j$ (i.e. the number of bits taken from the queues to the output link at one scheduler cycle), $\phi_{h,j}$ the amount of frame for class $i$ on node $j$, $L_i$ the maximum packet size in session $i$, $r_j$ the outgoing link capacity on node $j$ and the last sum defines the total propagation delay of the path.

$$D_i = \frac{\sigma_i}{\rho_i} + \sum_{j=1}^{k} \left( \frac{F_j - \phi_{h,j} + L_i}{r_j} \right) - \frac{L_i}{\rho_i} + \sum_{j=1}^{k} \frac{p_j}{\rho_i}$$  (1)
If the calculated maximum delay is less than the predefined maximum value, the delay test is passed and the edge moves to the next test.

On the other hand, in the measurement-based methods 1 and 2 the delay-test is done by measuring the current average-delay in the group that we want to join. This is done by joining the group and receiving the 20 first packets. If the calculated average delay is below the maximum acceptable one, the test is passed.

C. Bandwidth Test

The available bandwidth is approximated with equation 2 rather than measuring it. The available bandwidth $B_i$ in class $i$ can be denoted as follows:

$$B_i = (w_i \cdot B_{tot}) - (N_{G,i} \cdot R_{flow}) \quad (2)$$

Parameters $w_i$, $B_{tot}$, $N_{G,i}$ and $R_{flow}$ denote the weight of the class in WRR scheduler, total bandwidth of the link, number of groups in the class and the average arrival rate of the flow in a group $G$, respectively.

As it can be seen from the delay formula 1 above, the admission control decision is made upon the worst-case that can occur at the moment of calculating the delay value. The bandwidth test, on the other hand, just checks if there is any bandwidth available within the class. Therefore these two tests guarantee really strict QoS guarantees.

However, the two measurement-based methods differ from each in this bandwidth test from each other in the following manner: in our earlier method 1 [1] the total available bandwidth in the link is monitored to define if the new flow will be accepted, while in method 2 the available bandwidth within each class is monitored. The measuring is done with so called GRIP-test.

The test goes as follows: when joining the group, the edge-node sends a GRIP-packet with the wanted QoS parameters to the branching node. When the Grip-packet returns from the branching node to the edge node, on the path every node examines, based on the method used, if the new flow can be accepted according the bandwidth requirements. At the edge node, the packet is examined and if the test was accepted at each node, the bandwidth test succeeded. If the test failed, even in one single node, the result of the test was a failure and the admission control rejects the joining.

III. Simulation Studies

Simulations were done with the Network Simulator 2 [10] with a GenMCast extension [11]. The topology used is presented in Fig. 1. All the measured traffic flowed through all the core routers in the simulation’s topology. Only two customers and their QoS were measured in the simulations and other multicast and FTP clients just make background traffic and join requests. In this case we do not have to care about multicast tree or routing and it is enough to examine only the path from source to destination.

The traffic characteristics used in the simulations is presented in Table I. IPTV and video conference traffic were used as traffic sources and there were 10 groups for both of them. The video traffic was captured from real video streams compressed to H.263 format [12]. Ten FTP clients were used and their total load was limited by defining the link between FTP servers and the core router to small enough. The profiles of each flow of video- and conference customers are depicted in figures 2 and 3, respectively.

The upper bounds for QoS metrics are also presented in Table I. These upper bounds were used in the measurement test of the admission control method.

Weighted Round Robin (WRR) scheduler was used in the output queues of each node. The weight of each queue was defined a bit smaller than the amount of expected traffic in the class. The purpose of the admission control is to handle these
kinds of cases. The queue lengths were defined to 25, 50 and 100 packets for conference, IPTV and FTP traffic, respectively. This was because of the delay bounds of the applications.

Joins and leaves to/from the groups were randomized by certain random distributions. The overall amount of requests in all simulations were set to 765. The duration of the membership was simulated with exponential distribution because it seems to describe the duration of the connection in a most realistic way. After leaving the group, the customers waited another Pareto-distributed time-interval. 70 simulations with different FTP throughput were run.

IV. RESULT ANALYSIS

In total there were three different simulation scenarios and in the following subsections the results of the simulations with explanations are presented.

A. Low Utilization

The first results are from a simulation where links were lightly loaded and the queues were not full at any time. The bandwidth of the FTP link in this simulation was 1.325Mb/s.

In the figure 4 can be seen that FTP throughput remains the same no matter what admission control method is in use. This is because in this scenario the bottleneck link is not fully utilized at any time.

The effects of these methods to the delay can be seen in figure 6. The figure shows the end-to-end delay experienced by the video stream receivers in this lightly loaded scenario. Now it can be seen clearly that the parameter-based - and measurement-based method 2, which were wasting bandwidth by rejecting join requests, provide lower end-to-end delays. This is simple: when there is less traffic in the network, the delays remain low because the packets will be handled faster.

B. High Utilization

The second results are from a simulation where utilization of the network was much higher. The bandwidth of the FTP's link in this simulation was 2.775Mb/s.

In this highly loaded scenario FTP throughput was more than twice as much as in the low utilization scenario. Now,
figure 7 shows clearly that background traffic rate differs significantly when using different methods. One clear exception is measurement-based method 1 where background traffic is at its maximum. Method 1 calculates the current average delay and the available link bandwidth before making decisions whether or not to accept the new client. In this scenario background FTP traffic uses a great amount of the bandwidth available on the links hence leading to a situation where no new multicast clients will be accepted.

Figure 8 indicates that when not using admission control at all, some amount of the background traffic is being dropped. This is because all the multicast clients are automatically accepted when they wish to join the groups, and as a more valuable users their packets are being prioritized. Therefore the queues of the background traffic will start filling up, and in the end leading to the dropping of the packets.

The two other methods provide a compromise between the methods described earlier. Their admission control decision is based on the bandwidth shares given to each class, so no matter if the background traffic has filled the links - no join requests are being rejected for that. In measurement-based method 2, the background traffic has an effect on the measured current average delay, hence leading to the greater amount of rejections. In parameter-based method the delay estimate is based on the amount of the connections sharing the same priority, and therefore the background traffic does not have any effect on the calculated delay.

Delays can be seen in figure 10. In case no admission control is used, the queues will fill up because all the connections are accepted, and that is why the delays start to increase significantly. Measurement-based method 1, on the other hand, produces the smallest delay in this case. This can be explained with the help of figure 9. In a highly loaded situation this method does not fully utilize the bandwidth of the network, hence leading to the situation where the queues are not that full. This again leads to smaller end-to-end delays.

Figure 9 shows that all the other methods seem to use the bandwidth of the network better than measurement-based method 1 with the cost of the increased delay, respectively.

C. Utilization effect

For this subsection, a total of 70 simulations were run, every round increasing the amount of the background FTP-traffic. In the figure 11 it can be seen how the bottleneck links’ utilization increases as a function of the FTP-traffic from 1.325Mb/s all the way up to 3.025Mb/s.

As the amount of background traffic increases, the bottleneck links’ utilization level increases as well. It can be seen that the maximum utilization is reached when using no admission control at all. Also the maximum delay experienced by the multicast receivers is reached at 1.9 Mbit/s. This is the point where the background traffic has used all the bandwidth dedicated to it. In this case of no admission control, the network is very well utilized from the very beginning.

Measurement-based method 1 starts to reject joining requests beginning at 1.6 Mbit/s. From this point onward, the utilization level does not increase significantly. There will be more rejects on the video customers than on the conference
Fig. 11. Bottleneck link’s utilization as a function of the background traffic

Fig. 12. Accepted requests as a function of the background traffic

Fig. 13. The number of video customers throughout the simulations

Fig. 14. The number of conference customers throughout the simulations

Fig. 15. End-to-end delay of conference customer throughout the simulation

customers because of the stricter QoS requirements of the first ones. This can be seen on the figures 13 and 14. The delay remains low throughout the simulation because the admission control maintains the number of multicast receivers at low all the time.

On the other hand, when using method 2 or parameter-based method, the utilization level reaches the maximum possible at point 2.4 Mbit/s - from this point onward the available bandwidth is used as well with the case of no admission control. This can be explained as follows. At point 2.4 Mbit/s the background traffic has used all the bandwidth of its own plus the bandwidth which was left over from the multicast receivers. Like mentioned earlier, these two methods tend to leave some amount of bandwidth dedicated to that class unused. It can be noted from the figure 15 that at point 2.2 Mbit/s the delays increase very steeply due the packets being queued until point 2.4 Mbit/s. After the point 2.4 Mbit/s, the delays settle down to a certain value. This refers that multicast receivers have reached their maximum queuing delay because the amount of background traffic is at its maximum.

As mentioned earlier, the parameter-based and measurement-based 2 approach waste bandwidth. That is why the utilization level starts from such a low level in comparison to cases of no admission control or measurement-based method 1 (figure 11).

V. CONCLUSIONS

Although the research of multicast in DiffServ networks is only starting, both multicast and DiffServ are already widely used. The co-operation of them is not however a trivial task. An obvious need for multicast admission control in DiffServ networks exists as it solves one of three main problems in the merging of multicast and DiffServ.
Presented admission control methods provide competent solution for different requirements at the context of multicast in DiffServ networks. If the quality of the service perceived by the users is not the issue while the efficient network utilization is, then there is no need for admission control. However, admission control should be used in case there is need to guarantee some level of QoS for the end users. Our parameter-based approach utilizes the network pretty well while keeping the delays low. The measurement-based method 1 does not utilize the network so efficiently, but maintains the delays low whether the network is highly loaded or not. The measurement-based method 2 uses the bandwidth well but causes the delays to grow more than in method 1.

In future, the development work would involve integrating a pricing model for CAC decision making, as well as testing our methods in a larger scale network environments. In addition, development of the methods to make them utilize the network resources more efficiently will also be of a great interest. One possible solution for this would be to examine the bottleneck link's utilization level which would be used as a threshold value. Below this threshold the CAC decision would be made on a link basis, where as above the threshold the decision would be classbased.

REFERENCES