Multicast Admission Control in DiffServ Networks

Olli Alanen, Mikko Pääkkönen, Timo Hämäläinen and Jyrki Joutsensalo
Department of Mathematical Information Technology
University of Jyväskylä
40014 Jyväskylä, FINLAND
{opalanen,mikpaak,timoh,jyrkij}@cc.jyu.fi

***Abstract*— Multicast admission control in Differentiated Services network is an important but shortly researched subject. Our admission control method rejects new multicast join requests that would otherwise decrease the quality experienced by the existing receivers. Edge nodes filter join requests and generate new requests. The method was developed as an extension to the DSMCast protocol but could also be adapted to other protocols. It decreases delays, jitter and losses and maintains them within the desired constraints.**

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*) [1] has been the main interest of QoS researchers because it is currently the best solution for unicast QoS. For some new applications like IPTV and video conferencing, multicasting is another way to provide better QoS by saving bandwidth. However, there are three major problems of transporting multicast traffic in DiffServ networks. They should be solved to attain guaranteed quality for multicast traffic.

The first problem is the heterogeneous trees. Multicast trees should have different QoS-levels on different branches for the customers demanding different levels of quality. The feature is not included in any of the multicast protocols at present but there are some proposals to solve this problem with new protocols in [2], [7], [8], [10], [12].

The scalability issue between multicast and DiffServ is the second problem. The main idea behind the DiffServ concept is the good scalability achieved by maintaining the core routers stateless. The IP-multicast on the other hand is based on routers' forwarding states for each group and source. Therefore the current IP-multicast protocols should somehow be made more scalable. Some theoretical solutions and protocols for the scalability problem are presented in [2], [11], [12].

Neglected Reserved Sub-tree (*NRS*) problem is the third problem to be considered. The problem exists because one data flow in to the network can be replicated into many egress nodes. The admission control in such situations can be difficult and current policing methods are not intelligent enough. Sufficiency of the resources should be somehow checked every time a new receiver joins a group. A bandwidth broker is one solution for this, but then the broker should know quite a lot about the network. Its huge databases would make the solution less scalable. A distributed method is a faster and more scalable way to do it. Then the edge routers can make the decisions locally based on some algorithms and measured network statics. Some solutions and protocols to the NRS problem are presented in [2], [7], [8], [10], [12].

DSMCast [2] 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 an 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 simulated with a network simulator [3] and the results show the strengths of it.

The rest of the paper is organized as follows. Section II describes our admission control method and how it works. Simulations ran in network simulators and their results are shown in Section III Section IV then concludes the paper.

II. The Method

We propose a method to be used in multicast admission control decisions in DiffServ networks. It 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. The method consists of three phases and all of these have been explained in the next subsections and sequence diagram in Figure 1.

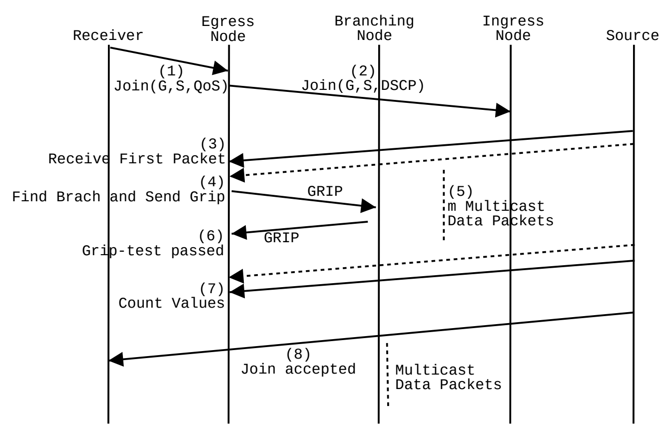

Fig. 1. Sequence diagram of the method.

The method was developed as an extension to the DSMCast protocol [2], but it could also be adapted to other protocols. DSMCast was chosen as the protocol in these simulations, because it has the multicast tree encapsulated in packet headers. The header can then be used to find the branching node. There has also been no previous proposals for its admission control method. In DSMCast joining and leaving the group is done by the egress edge, which also defends the use of a distributed method.

The method's purpose 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. The parameters are defined in the extended join-message `Join(S,G,QoS)` where the QoS parameter includes the average bandwidth usage of the flow, DSCP and the maximum acceptable values for delay, jitter and loss (*Step 1*).

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 and if not, the edge joins the group and moves to the next test before making any decisions (*Step 2*).

B. GRIP Test

The GRIP-test is based on the MGRIP protocol introduced in [7]. The idea is to check if the network has enough resources for a specific flow. In multicast, it is important to check only the path from the branching node to the destination. The branching node is a node where the packets are replicated for the last time. If the receiver is the first and only receiver of the group the ingress node is the branch node. This way, only the links whose utilization will increase, are measured.

The GRIP-test is done simultaneously with the measurement test. When the first packet of the group arrives to the egress edge (*Step 3*), the edge node checks the DSMCast header and finds the branching node. Then the egress edge sends a GRIP-packet to the branching node (*Step 4*). GRIP-packet includes the average bandwidth usage of the flow. At the branching node the packet is returned and on its way back, all the nodes on the path check if they have enough resources for the new flow. The check is done against link's few seconds' average usage of bandwidth. The packet is forwarded to the next node if the link has the capacity for the flow. Otherwise the packet is dropped. If the egress edge receives the GRIP-packet (*Step 6*), the GRIP-test is passed. Otherwise the egress node leaves the group, informs the receiver about rejecting and stops the method.

C. Measurement Test

In the measurement test, the egress edge receives the first *m* of the group's packets (*Step 5*) and then counts the values for end to end delay, jitter and packet loss from them. The values verified against the acceptable parameters are then calculated with the equation of exponential average (Equation 1) from the current (P_{curr}) and the history (P_{exp}) results (*Step 7*). If the results are below the acceptable maximum values, the measurement test is accepted. If not, the edge leaves the group and informs the client about the negative decision. If both the GRIP-test and the measurement test are passed, the joining can be accepted (*Step 8*). If either of the tests fail, the join is rejected and the edge node leaves the group.

$$P_{exp} = (1 - w) \cdot P_{exp} + w \cdot P_{curr} \qquad (1)$$

The measurement test gets its information from the packet's UDP and RTP headers. Sequence numbers are provided by the UDP and timestamps by RTP. The test measures the actual multicast traffic, because it gives the most realistic results.

III. Simulation Studies

Simulations were done with the Network Simulator 2 [3] and a GenMCast extension [5]. The topology used is presented in Figure 2. All the measured traffic flows 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.

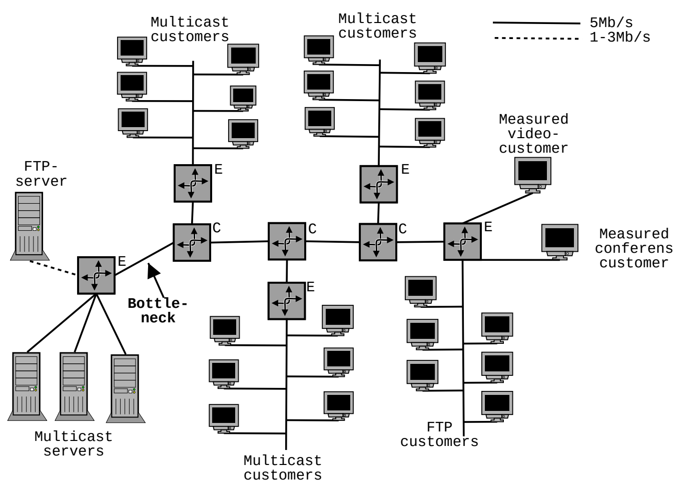

Fig. 2. Simulation's network topology.

The traffic 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 [6]. Ten FTP clients were used and their total utilization was limited by defining the link between FTP servers and the core router to small enough. The profiles of the measured traffic are shown in Figure 3.

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. The limits are defined by ITU-T and they were found from [4].

Weighted Round Robin (*WRR*) was used in every nodes' output queues' scheduling. The weight of each queue was

TABLE I
SIMULATIONS TRAFFIC.

Traffic	Bandwidth	Packet-size	DSCP	Delay (UB)	Jitter (UB)	Loss (UB)
Control messages	-	-	5	-	-	-
Video conference (H.263)	∼68kbit/s	∼800	6	400ms	17ms	10^{-6}
IPTV (H.263)	∼270kbit/s	∼950	10	150ms	50ms	10^{-4}
FTP	-	900	99	-	-	-

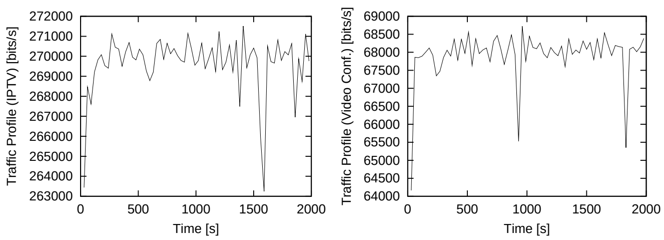

Fig. 3. Profiles of the measured traffic.

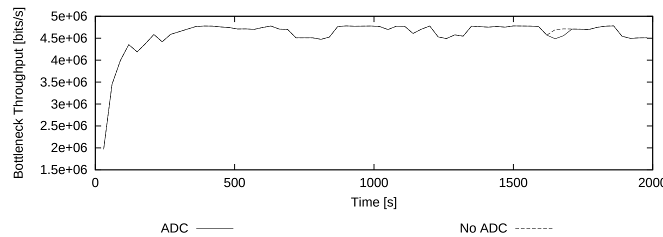

Fig. 5. Bottleneck link's throughput in the low utilization simulation.

defined as a bit smaller than the amount of expected traffic of the class. The purpose of the admission control is to handle these kind of cases. The queue lengths were defined to 25, 50 and 100 respectively to conference, IPTV and FTP traffic. This was because of the delay bounds of the applications.

Joins and leaves to the groups were randomized by certain random distributions. 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 duration simulated with Pareto distribution.

An example of how the join requests were made is presented in Figure 4.

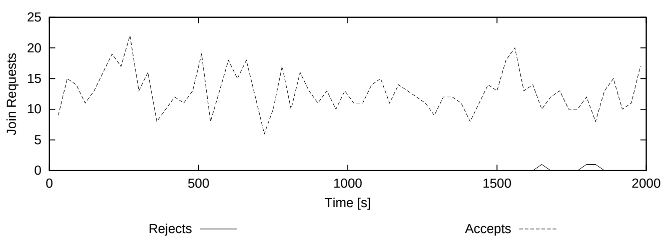

Fig. 4. Join requests.

60 simulations with different FTP throughput were run and their results are discussed in the following subsections.

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's link in this simulation was 1.4Mb/s. It can be seen from Figure 5 that the method does not bother in a lightly loaded situation where it works in exactly same manner than without it. No requests are rejected and the throughput of the bottleneck link is utilized as well as in graphs without the method.

Delays of measured flows are shown in Figure 6. The delays are naturally at the same level in the graphs with and without the method. There is no loss at all, which also defends the decisions made by the method.

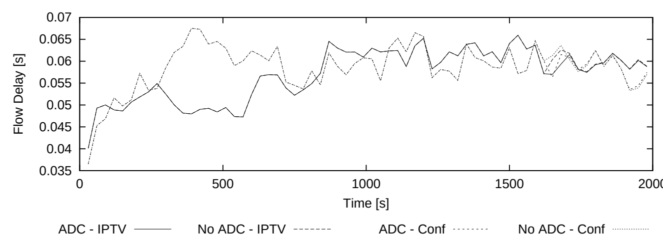

Fig. 6. Measured traffic's delay in the low utilization simulation.

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 1.8Mb/s. The throughput of the bottleneck link is presented in the Figure 7. Throughput with the admission control is a bit lower than without it, but only about ten percent or less.

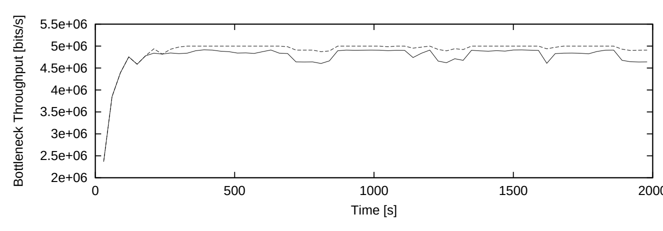

Fig. 7. Bottleneck link's throughput in the high utilization simulation.

The significance of the admission control can be seen very well in Figure 8. The delays are quite stable and a lot smaller than without the method. The Figure also shows that the jitter stays down with the admission control.

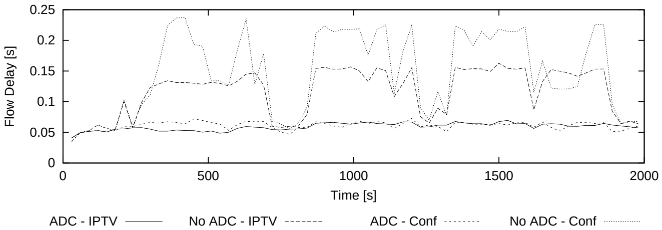

Fig. 8. Measured traffic's delay in the high utilization simulation.

Losses of the measured flows are shown in Figure 9. It shows that in this simulation there are no losses at all with the method. Without the admission control even 15 % of packets are sometimes lost.

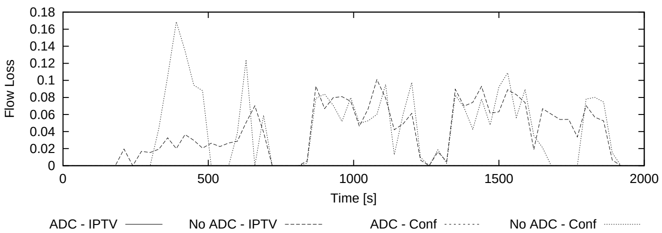

Fig. 9. Measured traffic's loss in the high utilization simulation.

C. Utilization Effect

The effects of bottleneck link's utilization are considered in more detail in the following results. The effect of FTP's throughput to the whole throughput is shown in Figure 10. It shows that the throughput stays almost as good everywhere and in 1.6Mb/s it starts to get a little behind the situation without the method. The 1.6Mb/s is a critical point because then the peak rates of the throughput exceed the link's capacity. Then the queues begin to fill and some of the requests have to be rejected.

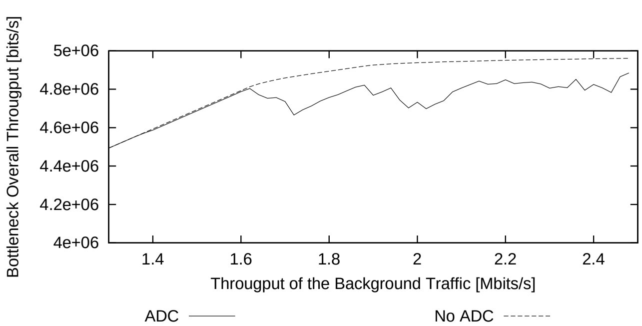

Fig. 10. Utilization effect on the throughput of the bottleneck link.

Figures 11 and 12 show clearly the critical point at 1.6Mb/s where the losses and delays start to grow without the method. At 1.9Mb/s the amount of traffic is so big, that the queues are full all the time and delays are at a maximum. Losses stay at zero in all the cases except for two, where few packets are dropped and delays do not variate at all with the method.

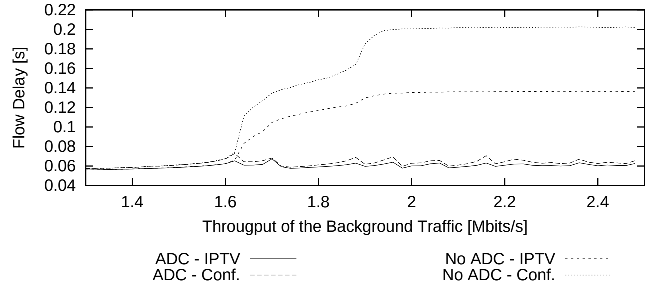

Fig. 11. Utilization effect on delays.

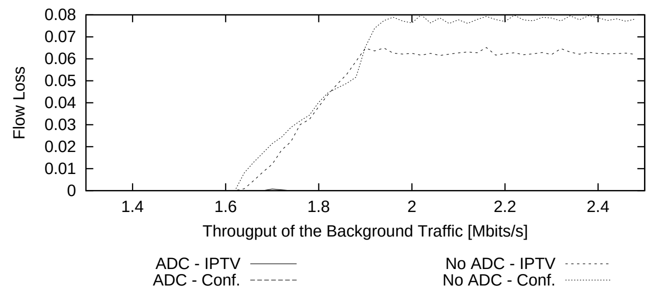

Fig. 12. Utilization effect on losses.

Figures 13 and 14 present the accepted and rejected join requests. They show how the amount of rejected requests slowly increases as the FTP-traffic grows. Figure 14 shows how the results of both tests are quite equally used to make admission control decisions. Accepted and rejected joins in some simulations are also presented in Table II

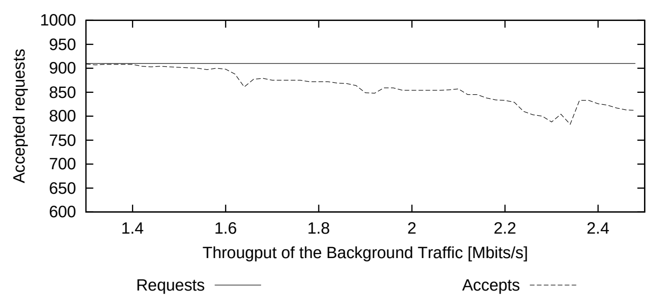

Fig. 13. Utilization effect on accepted joins.

TABLE II

ACCEPTED AND REJECTED JOINS.

FTP	1.4Mb/s	1.5Mb/s	1.6Mb/s	1.7Mb/s	1.8Mb/s
Requests	884	884	884	884	884
Accepts	881	879	880	840	860
Rejects (Measurement)	3	4	1	29	11
Rejects (GRIP)	0	1	3	15	13

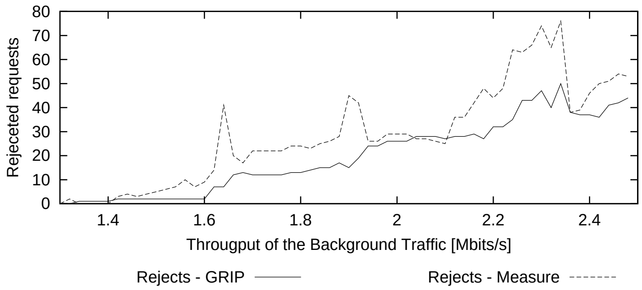

Fig. 14. Utilization effect on rejected joins.

Figure 15 presents the average extra delay that admission control produces to joins. It seems that the delay decreases as the utilization grows, but that is not the whole truth. When the utilization grows, also the amount of rejects grows. When the amount of accepted joins decreases the proportion of edge test acceptions grows. When the requests are accepted in the edge test, the extra delay is almost zero and the average of join delay decreases. Anyway, the delays do not grow and the average of $\frac{1}{2}$ seconds is not much.

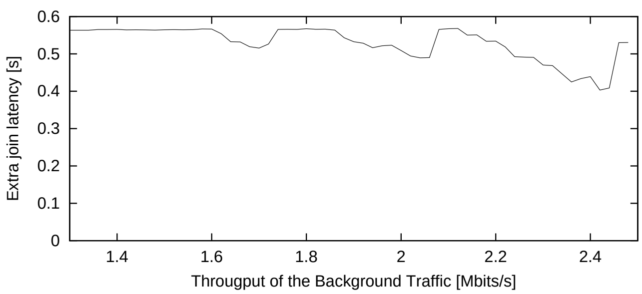

Fig. 15. Utilization effect on extra join delay.

IV. Conclusions

Although the research of multicast in DiffServ networks is only starting, both techniques are already widely used. 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.

Our admission control method is a competent solution that rejects the unwanted join requests. The method decreases delays, jitter and losses below the constraints and still utilizes the network well. The method does not produce more than an average of 0.6 second of extra joining delays which also defends the method.

In further development, the method could begin to co-operate with a bandwidth broker entity.

References

[1] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, "An Architecture for Differentiated Services" IETF RFC 2475, 1998.

[2] A. Striegel, G. Manimaran, "Dynamic DSCPs for Heterogeneous QoS in DiffServ Multicasting", Proc. of IEEE GLOBECOM, 2002.

[3] Network Simulator, Available at: <URL:http://www.isi.edu/nsnam/ns>, 8.4.2004.

[4] S. Lima, P. Carvalho, A. Santos, V. Freitas, "A Distributed Admission Control Model for Class-based Networks Using Edge-to-Edge QoS and SLS Monitoring", Proc of IEEE ICCS, 2002.

[5] A. Striegel, "GenMCast Software Extension for ns-2, ND CSE Technical Report", Available at: <URL:http://www.cse.nd.edu/ striegel/GenMCast/>, 8.4.2004.

[6] F. Fitzek and M. Reisslein, "MPEG-4 and H.263 Video Traces for Network Performance Evaluation", IEEE Network, vol 15, no 6, pages 40-54, November/December 2001.

[7] G. Bianchi, N. Blefari-Melazzi, G. Bonafede and E. Tintinelli, "QUASIMODO: QUAlity of ServIce-aware Multicasting Over DiffServ and Overlay networks," IEEE Network, 2003.

[8] B. Yang, P. Mohapatra, "Multicasting in Differentiated Service domains", Proc. of IEEE GLOBECOM, 2002.

[9] G. Bianchi, N. Blefari-Melazzi, "Admission Control over Assured Forwarding PHBs: A Way to Provide Service Accuracy in a DiffServ Framework," IEEE GLOBECOM, 2001.

[10] R. Bless, K. Wehrle, "IP Multicast in Differentiated Services Networks", IETF (In Progress) Draft, 2003.

[11] Z. Li and P. Mohapatra, "QoS-Aware Multicasting in DiffServ Domains", Proc. of Global Internet Symposium, 2002.

[12] A.Striegel, A. Bouabdallah, H. Bettahar, G. Manimaran, "EBM: A New Approach for Scalable DiffServ Multicasting", Proc. of Fifth International Workshop on Network Group Communications (NGC), 2003.