Quality of Service for Multicasting in Content Addressable Networks

Marc Brogle, Luca Bettosini, and Torsten Braun

Institute of Computer Science and Applied Mathematics
Universität Bern, Neubrückstrasse 10, 3012 Bern, Switzerland
brogle@iam.unibe.ch, bettosin@iam.unibe.ch, braun@iam.unibe.ch

Abstract. Multicasting is an efficient mechanism for one to many data dissemination. Unfortunately, IP Multicasting is not widely available to end-users today, but Application Layer Multicast (ALM), such as Content Addressable Network, helps to overcome this limitation. Our OM-QoS framework offers Quality of Service support for ALMs. We evaluated OM-QoS applied to CAN and show that we can guarantee that all multicast paths support certain QoS requirements.

1 Introduction

The OM-QoS framework [1,2] aims to enable different Application Layer Multicast (ALM) protocols to support Quality of Service (QoS). In [1], we focused on how to make Scribe ALM, which runs on-top of Pastry Peer-to-Peer (P2P), QoS aware. To support more P2P/ALM protocols, we also analyzed in [2] how to enable QoS for NICE, Content Addressable Networks (CAN) [3], and how to create a general solution using a layered approach, in order to support different P2P/ALM networks.

As described in [1,2], the multicast tree has to be built in such way that the QoS requirements or capabilities are monotonically decreasing from the root to the leafs. To manage different QoS parameters, we introduced the concept of QoS classes, which need to have a natural order. Such a QoS class could be a combination of bandwidth, jitter and CPU requirements. An example of a QoS aware tree is depicted in Fig. 1.

In this paper, we evaluate OM-QoS for CAN, which is the reference implementation for the OM-QoS framework's general solution (layered approach). For each QoS class, we use a dedicated CAN. Those CAN layers are then ordered and interconnected with each other using gateway links, to build a QoS aware aware multicast tree. For an overview of CAN, see [3]. A more detailed description of OM-QOS is available in [2].

2 Evaluation of QoS Enabled Content Addressable Network

We evaluated the application of OM-QoS to CAN using the OMNet++[1] simulator. Therefore, we implemented the CAN protocol and made some enhancements to fully support join and leave functionality. Due to the possibility of nodes leaving at any time during the simulation, we also needed to support temporary key-space assignment as

[1] Official OMNet++ simulator website: http://www.omnetpp.org

T. Pfeifer and P. Bellavista (Eds.): MMNS 2009, LNCS 5842, pp. 170–175, 2009.

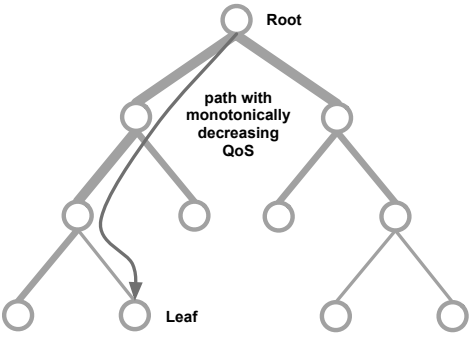


Fig. 1. QoS Supporting Multicast Tree

well as zone handover and merge mechanisms. If the multicast sender leaves its CAN, its responsibility to transmit continuously multicast messages has also to be passed to another CAN node. The CAN protocol was implemented in a fully decentralized manner, the network was built and maintained using only message transmission among the nodes. This original "native" CAN implementation has no QoS awareness/support. Therefore, it would be quite simple to modify our solution to also work in real networks.

We analyzed the CAN implementation and our OM-QoS extensions to CAN using various scenarios. We present the results for a scenario consisting of a CAN network without QoS support, where the nodes have one of 64 possible QoS classes assigned, in order to verify the QoS properties of the multicast paths for "native" CAN. Further, we present the results for a scenarios consisting of an OM-QoS enabled CAN with one of 64 QoS classes assigned to a node. All nodes get their QoS class (out of 64 possible QoS classes, depending on the scenario) assigned on startup. We then compare how many of the multicast message paths fulfill the previously described QoS property in native CAN (without any QoS mechanisms for multicast tree construction) and in OM-QoS enabled CAN with a QoS aware multicast tree construction mechanism. The latencies between all the nodes are determined using distance matrices, defining the latencies for each possible node pair connection. The distance matrices were built using topologies generated by Brite[2]. The minimum, maximum and median delay values of those matrices are depicted in Table 1. For each of the scenarios, we looked at various networks ranging with a node count from 100 to 2000 in steps of 100. Each node step was evaluated using 13 different distance matrices, which we evaluated each with 20 different random seeds influencing arrival, departure and other random based decisions. Therefore, each scenario consists of 5200 simulations. We removed 1% of the outliers (0.5% of the min. and max. values each) for the figures and results in this paper. Figure 2 shows the simulation results for a native CAN scenario without any QoS support but where each node gets one of 64 QoS classes assigned at startup. All nodes reside in one single CAN consisting of up to 2000 nodes. In Fig. 3, we show the simulation results for an OM-QoS enabled CAN scenario with 64 QoS classes. For the OM-QoS enabled CAN scenario with 64 QoS classes, we have 64 distinct CAN layers consisting

[2] Website for Brite topology generator: http://www.cs.bu.edu/brite/

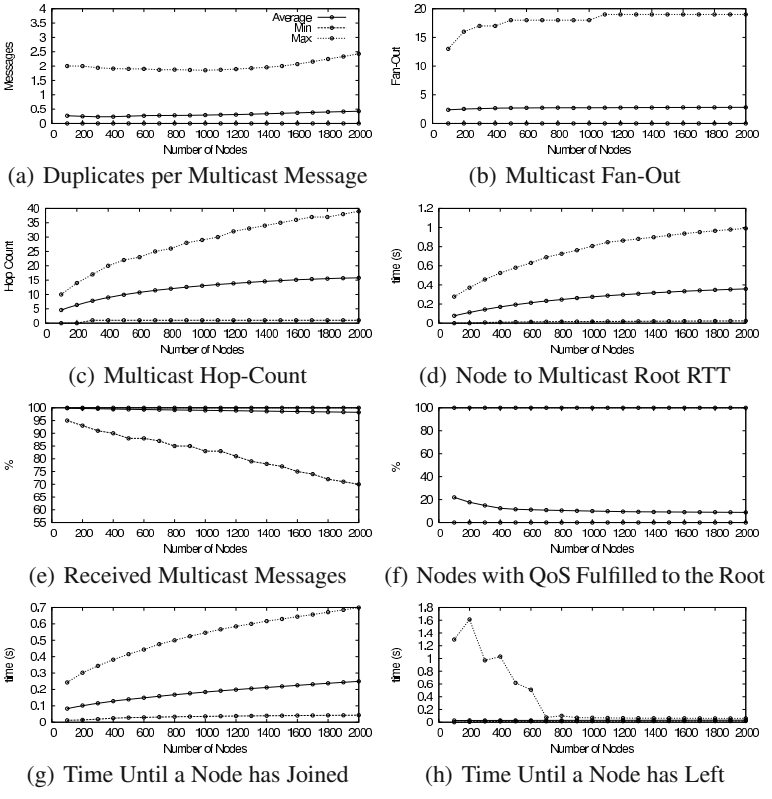
Table 1. Delay Properties of Distance Matrices in ms

Matrix	**min RTT** (ms)	**mean RTT** (ms)	**max RTT** (ms)
Matrix 0	0.08	22.47	48.44
Matrix 1	0.09	30.35	90.48
Matrix 2	0.05	30.56	94.58
Matrix 3	0.05	29.76	90.23
Matrix 4	0.07	23.26	57.52
Matrix 5	0.09	22.78	51.82
Matrix 6	0.04	22.77	49.24
Matrix 7	0.08	23.27	52.30
Matrix 8	0.05	22.91	53.92
Matrix 9	0.05	23.27	50.83
Matrix 10	0.08	22.47	48.44
Matrix 11	0.05	22.91	54.00
Matrix 12	0.01	23.13	54.17

of an estimated average of 1.56 to 31.25 nodes for the scenarios with 100 to 2000 nodes in total. We compare the average, minimum and maximum values for the OM-QoS enabled CAN scenario with the results for the native CAN scenario without QoS support. Figures 2(a) and 3(a) present the duplicates received per multicast message. The average for the native CAN scenario is between 0.3 to 0.5 duplicates per message, which is due to the nature of CAN. As the minimum shows, duplicates can be often avoided. For some nodes the worst case is between 2 to 2.5 duplicates per multicast message. Comparing the average with the OM-QoS enabled CAN scenario, we see that the values behave now more or less constant due to the fact that we have smaller CAN networks. On the other hand, the average for less nodes has risen slightly, because of the duplicates that can be introduced by inter-layer forwarding. Our mechanisms between layers focuses more on reliability than on duplicate avoidance. This has the same effect on the maximum and minimum values.

Figures 2(b) and 3(b) show the multicast fan-out, the number of direct downstream children of a node. For the native CAN scenario, the average fan-out of 3 per node is very acceptable and remains more or less constant for the different amount of nodes. As expected, there are also leaf nodes that do not forward any messages (minimum of 0). For the worst case, the maximum is at 19, which seems to grow only very slowly for more nodes. Looking at the OM-QoS enabled CAN scenario, the average has been increased by 1 to 2 children. These additional children are introduced by the inter-layer forwarding mechanism. The maximum has been reduced because of having smaller CAN on the different layers, but the values grow more steadily with more nodes in a scenario. The minimum did not change, because there will always be pure leaf nodes.

In Figures 2(c) and 3(c), we present the hop-count of multicast messages. The hop-count on average grows slowly when there are more nodes in a CAN. Minimum and maximum values for the hop-count behave as expected, with the minimum being more or less constant while the maximum grows steadily when more nodes are in a CAN. The average hop-count in the OM-QoS enabled CAN scenario is generally increased due to inter-layer forwarding, but behaves now more or less constant due to smaller CAN sizes. This has a similar effect on the maximum. Figures 2(d) and 3(d) show the RTT of nodes to the root of the multicast tree. This value behaves the same as the hop-count with an average between 0.1 to 0.35 seconds and a maximum of 1 second for



(a) Duplicates per Multicast Message (b) Multicast Fan-Out

(c) Multicast Hop-Count (d) Node to Multicast Root RTT

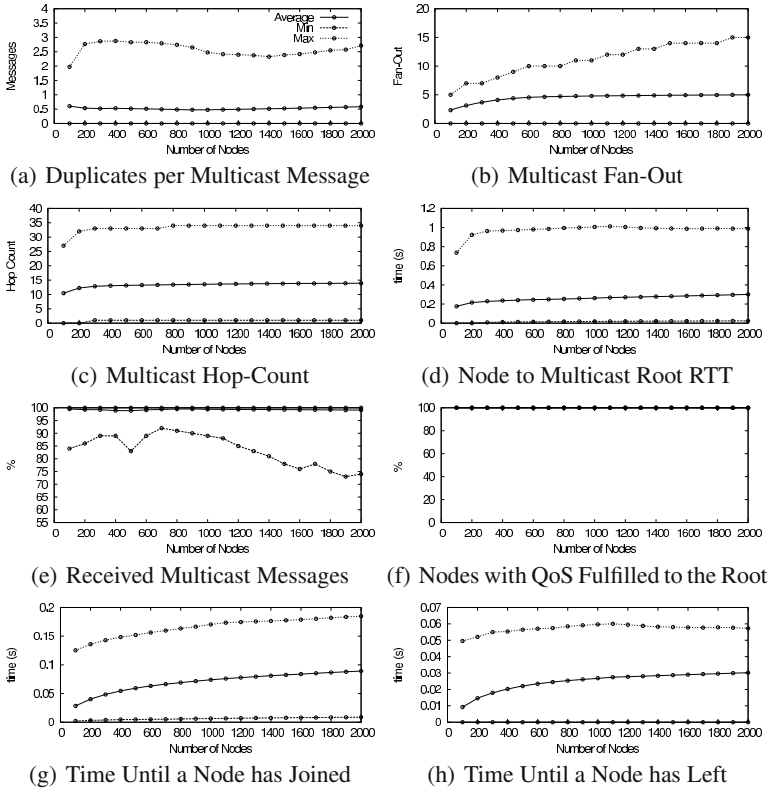
(e) Received Multicast Messages (f) Nodes with QoS Fulfilled to the Root

(g) Time Until a Node has Joined (h) Time Until a Node has Left

Fig. 2. Simulation Results for a Native CAN without QoS Support (with 64 virtual QoS classes)

the native CAN scenario. In the OM-QoS enabled scenario, the RTT values receive the same impact as for hop-count. They are on average a bit higher than in the native CAN scenario but behaving more or less constant with more nodes in the scenario.

In Figures 2(e) and 3(e), we depict the percentage of received multicast messages. More nodes in a CAN result in less multicast messages being successfully delivered to every node, as can been seen looking at the behavior of the minimum values for the native CAN scenario. The range for the average is between 98 to 100 percent, with a maximum of 100 percent as expected. The amount of the received multicast messages in the OM-QoS enabled CAN scenario is improved, because there are only small number of nodes in a CAN and due to the reliability optimized inter-layer forwarding mechanism. The minimum value behaves less predictably, because there are now multiple CAN in parallel behaving completely different than when just having one big CAN for all nodes. In Figures 2(f) and 3(f), we present the percentage of nodes that hold the previously described property on their full path to the root of the multicast tree. Although the paths in the multicast tree are not constructed in a QoS aware manner in the native CAN scenario, on average 10 to 20 percent of the paths may hold the previously mentioned QoS property. As expected, all the paths fulfill the QoS property in the OM-QoS



(a) Duplicates per Multicast Message (b) Multicast Fan-Out

(c) Multicast Hop-Count (d) Node to Multicast Root RTT

(e) Received Multicast Messages (f) Nodes with QoS Fulfilled to the Root

(g) Time Until a Node has Joined (h) Time Until a Node has Left

Fig. 3. Simulation Results for an OM-QoS Enabled CAN with 64 QoS Classes

enable CAN scenario. Figures 2(g) and 3(g) show the time a node required to fully join a CAN. The values in the native CAN scenario grow steadily with the amount of nodes, having an average ranging between 0.1 and 0.25 seconds, while the maximum is still in an acceptable range from 0.25 to 0.7 seconds. The minimum remains more or less constantly low close to 0.05 seconds, which reflects the early moments of a CAN, where there are only a few nodes in the network and join related messages have only to be passed among few nodes. The OM-QoS enabled CAN scenario has smaller CAN in the different layer. Therefore, joining time on average is reduced significantly. The maximum and minimum values behave accordingly.

In Figures 2(h) and 3(h), we present the time a node required to successfully leave. On average, nodes can quickly leave a CAN, but sometimes the handover of zones to other nodes may take more time if multiple nodes want to leave at the same time. With less nodes in the native CAN scenario having only one single CAN, the effect of multiple nodes wanting to leave at the same time is stronger and can result in leaving times up to 1.6 seconds. In the OM-QoS enabled CAN scenario, nodes can leave faster due to having a lower probability of nodes simultaneously leaving from the same CAN layer. Therefore, the average and maximum values grow slowly and turn to behave almost

constant with more nodes being distributed evenly over the multiple layers. Since there are also CAN layers that only consist of one node when having multiple layers, no handover of zones has to be done and leaving is performed instantly (0s minimum). We also ran the simulations for an OM-QoS enabled CAN with 256 QoS classes (4 times more layers). The duplicates increased for small number of nodes. The average fan-out for small amount of nodes in a scenario is reduced, because only one or very few nodes are in each CAN. Therefore, multicast messages mainly have to be forwarded to other layers and rarely inside the same CAN. With more nodes in the system, the average and maximum match the behavior of the scenario with 64 QoS classes in relation to the amount of nodes per CAN in a layer. The average hop-count is increased by more than the factor 2, while the maximum is more than 3 times higher, the average RTT to root values are tripled, while the maximum is almost 4 times larger. The average of lost multicast messages is smaller because of our inter-layer forwarding improvements.

3 Conclusions

In this paper we have presented OM-QoS (Quality of Service for Overlay Multicast) applied to CAN (Content Addressable Networks), which enables CAN to support Quality of Service (QoS) enabled multicasting. The concept of QoS classes allows an abstract view of QoS requirements. For each QoS class, a dedicated CAN is established. These layers of CAN are interconnect using gateway nodes. Multicast messages are passed from a higher QoS class layer to a lower QoS layer, therefore building multicast trees that support QoS requirements. We have evaluated our approach using OMNET++. Simulations of networks with 100 to 2000 nodes have been performed and analyzed. OM-QoS is a self-managing middleware, which supports efficient and reliable usage of content distribution networks and other multimedia applications such as (real-time) A/V streaming based on Application Layer Multicast (ALM).

Enabling QoS for CAN using OM-QoS slightly increases the average fan-out of nodes and duplicates received per multicast message, but in consequence, this improves the percentage of multicast messages received. Other properties of CAN are not changed significantly when using OM-QoS.

References

1. Brogle, M., Milic, D., Braun, T.: QoS enabled multicast for structured P2P networks. In: P2PM 2007 at IEEE Consumer Communications and Networking Conference, January 2007. IEEE, Los Alamitos (2007)
2. Brogle, M., Milic, D., Braun, T.: Quality of service for peer-to-peer based networked virtual environments. In: P2P-NVE 2008 Workshop at the 14th IEEE International Conference on Parallel and Distributed Systems, Melbourne, Victoria, Australia, December 2008. IEEE, Los Alamitos (2008)
3. Ratnasamy, S., Francis, P., Handley, M., Karp, R., Schenker, S.: A scalable content-addressable network. In: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications, vol. 31. ACM, New York (2001)