



Signaling performance of UPnP QoS architecture

Brewka, Lukasz Jerzy; Wessing, Henrik; Dittmann, Lars

Published in:
Advanced Networks and Telecommunication Systems (ANTS),

Link to article, DOI:
[10.1109/ANTS.2009.5409891](https://doi.org/10.1109/ANTS.2009.5409891)

Publication date:
2009

Document Version
Publisher's PDF, also known as Version of record

[Link back to DTU Orbit](#)

Citation (APA):
Brewka, L. J., Wessing, H., & Dittmann, L. (2009). Signaling performance of UPnP QoS architecture. In *Advanced Networks and Telecommunication Systems (ANTS)*, (pp. 1-3). IEEE.
<https://doi.org/10.1109/ANTS.2009.5409891>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Signaling Performance of UPnP QoS Architecture

Lukasz Brewka, Henrik Wessing and Lars Dittmann

Department of Photonics Engineering,

Technical University of Denmark, 2800 Lyngby, Denmark

Phone: (+45) 45253620, Email: {ljbr, hewe, ladit}@fotonik.dtu.dk

Abstract—Provisioning QoS within home networks becomes a common requirement. This paper evaluates one of the candidates for home network control and management protocol suites, namely UPnP. Using network modeling tools we identify highlights and drawbacks of UPnP QoS Architecture.

I. INTRODUCTION

Providing QoS within the home networks has attracted a lot of attention when automated and intelligent home is considered. Similar views on the network control and management might also be seen in small office environments. This is mainly caused by the growing number of flows in home networks and diversity of traffic types with clear differentiation of the importance of particular traffic flows. At the same time, there is a great emphasis on introducing a management system in the home environment that can handle the dynamic character of the network, where devices leave and join the network frequently. Naturally, service based platforms are often a choice for organizing and controlling the described networks. There is a number of protocols designed for dynamic service discovery that can be used for establishment of home network; UPnP [1], DPWS [2], Bonjour [3], IGRS [4], Jini [5]. While Bonjour does not explicitly consider QoS, IGRS and Jini are more focused on the end devices' resources than network's resources, UPnP and DPWS are clearly defining network QoS mechanisms. The scope of listed protocols is usually quite broad, nevertheless this paper concerns mainly the chosen QoS provisioning mechanism, namely the reservation procedures, keeping the context of home networking in focus. UPnP together with its QoS Architecture specification provides a good environment for evaluation of signaling procedures in service based architecture. That is why, in the remaining part of this paper, we will consider only UPnP QoS Architecture [6] and the analysis will be based on its signaling model. However the analysis made here are generic enough that it could be used in any QoS architecture where resource reservation procedure is similar (e.g. DPWS – Device Profile for Web Services). Some evaluation of UPnP QoS framework was done in [7] where authors concentrate on performance in WLAN environment, whereas here we show how UPnP QoS Architecture reservation procedures treat flows depending on defined importance.

The remainder of this paper is organized as follows. Section II describes the basis of the performed UPnP QoS Architecture modeling, section III addresses the simulations. Finally, conclusions and future work is placed in section IV.

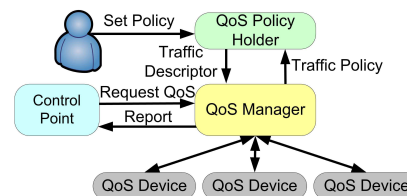


Fig. 1. UPnP architecture

II. MODELING UPnP AND IN-HOME QoS

UPnP is a set of protocols defining addressing, discovery, description, presentation, eventing and control of network devices [1].

Usually, UPnP is used in a network where different devices and services share the resources. For such network, introduction of QoS capabilities seems to be natural. In [8] extensions to the protocol are proposed, which indicate need for further development of new functionalities. That is why UPnP Forum (the initiative behind UPnP) defined the UPnP QoS Architecture.

In this paper we demonstrate a model, which represent the signaling procedures of UPnP QoS Architecture in accordance with UPnP QoS Architecture version 3 [6].

A. UPnP Entities

The UPnP QoS entities and their relations are depicted on Fig. 1. The Control Point (CP) is an instance that has the knowledge of the source and destination of a particular flow and the Traffic Specification (TSPEC) of the content. It is not a separate UPnP module as such. CP sends the information to the QoS Manager (QM) that communicates with the rest of the services/devices on the network.

QM [9] is the coordination unit responsible for requesting, updating and releasing QoS assigned to various traffic streams. The QoS Policy Holder (QPH) [10] is responsible for providing the traffic policies to requesting QM that provides the Traffic Descriptor as input parameter. The QoS Device (QD) service resides on the source, destination or intermediate node of the particular flow. QD is administering its resources and reporting the state according to its configuration set by QM [11]. During our work the administration of the resources within the device model was performed on certain level of abstraction that allows later specification of mapping mechanisms that could be used with multiple Layer 2 technologies.

The UPnP QoS Architecture defines three types of Quality of Service provisioning; prioritized, parameterized and hybrid. Prioritized QoS is a default approach and means end-

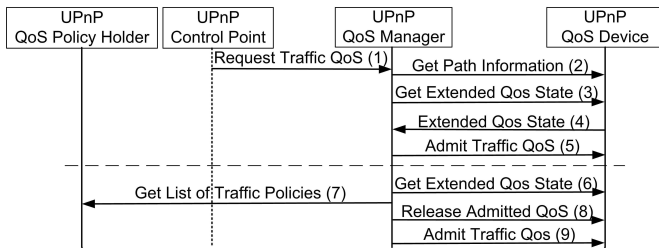


Fig. 2. Interaction diagram for Traffic QoS request with preemption

to-end traffic prioritization. Prioritization is performed based on the policy stored in the QoS Policy Holder. Parameterized QoS requires resource reservation on the entire traffic path - if not all segments in the network support this type of QoS, the attempt of its establishment will fail. Hybrid QoS admission can take place for situation when some of the segments on the flow's path are not supporting parameterized QoS and Control Point allows use of prioritized QoS on those segments. [6]

B. Signaling in UPnP

The analysis in this paper mainly considers UPnP signaling during establishment of parametrized QoS. The interaction diagram for reservation is presented on Fig. 2. Its upper part shows the reservation for a case without preemption which is a case when there are enough available resources or when preemption was not requested. The diagram present the interaction between the UPnP QoS services when preemption is enabled. [6]

The basic procedure for traffic QoS establishment (see Fig. 2) starts with CP's QoS request for a given flow sent to QM (1). The request contains the Initial Traffic Description and Resource parameters, which define flow's basic parameters and resources that are requested. Later QM, based on the source and destination addresses contained in the Initial Traffic Descriptor, determines the path for the flow and devices that needs to be configured for incoming traffic (2). This is followed by determination of the state of QDs with Get Extended QoS State action (3), in response the QDs provide information about their capabilities and current state (4). Afterwards the QM will invoke the Admit Traffic QoS action (5). [6][10]

In case the attempted reservation fails and CP request preemption the QM will continue the reservation procedure. QM will request from QDs the information about blocking flows with second Get Extended QoS State request (6). Once QM have knowledge of blocking flows it can send the Get List of Traffic Policies request to QPH for the list of traffic policies (7). Based on obtained policies QM can make decision if and what should be preempted. Later QM tries to release some resources (8) and admit the newly requested flow again (9). Normally preemption will take place in the situation when the requested resources are not available on one of the network devices, but some of the resources are occupied by the flows that according to the traffic policies are of the less importance than the newly arriving flow. The specification does not define details for the process of choosing flow to be released in case there are multiple candidates for preemption.

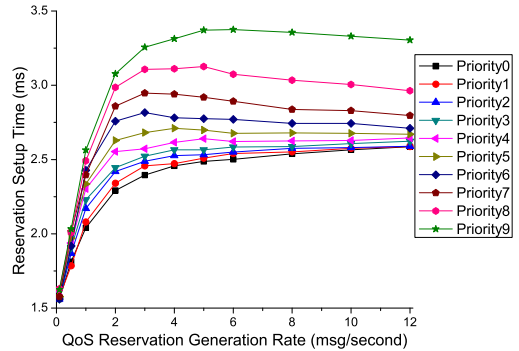


Fig. 3. Setup time for flows of different priority in function of traffic QoS request message generation rate

III. SIMULATIONS

During the modeling activities we focused our attention on three major aspects of the reservation procedure that influence the overall performance and quality of network architecture. The model used for the work presented here is developed in the OPNET modeling tool [12]. The priority and resources occupied by particular reservations are chosen probabilistically. The interval between reservation generations is exponentially distributed with parametrized mean value. We performed modeling on the architecture similar to generic UPnP QoS Architecture. We implemented three QDs connected to QM, which process random requests. The arrival rate of QoS requests is adjustable parameter changed during the simulations (we present results for request rates between 0.1 to 12 requests per second).

First we look at reservation setup time within different classes of service, second rejection ratio of arriving reservation requests and finally the rate with which reservation messages are rejected. Fig. 3 shows that the average setup time for higher priorities reservations is higher. This is an expected tendency, simply because the reservation with preemption takes longer and this kind of reservation is less likely to happen for low priority traffic. Another observation is that this tendency does not change for growing traffic QoS rate. The graph also shows that higher generation rate causes extension of setup time - which increases around 50% for low priority flows and 100% for high priority packets. Higher impact on the setup time of high priority flows can be explained by the fact that preemption is occurring more often for higher message generations rates and mainly affects the higher priority traffic flows which cause preemption more likely than the low priority reservations. Setup time observed for all proprieties and requests rates, for modeled 100Mb/s links, are on level that should not influence network performance visible for the users.

In Fig. 4 we present rejection ratios for different priority flows measured as a number of reject notification for particular priority over total number of notifications received. The results of the simulation clearly show that the ratio of rejected messages for lower priorities are higher. One can

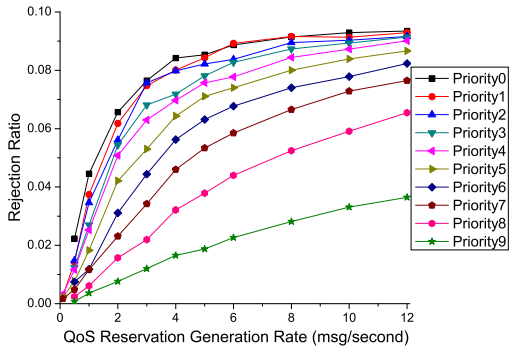


Fig. 4. Rejection ratio for different priority flows as a function of traffic QoS request message generation rate

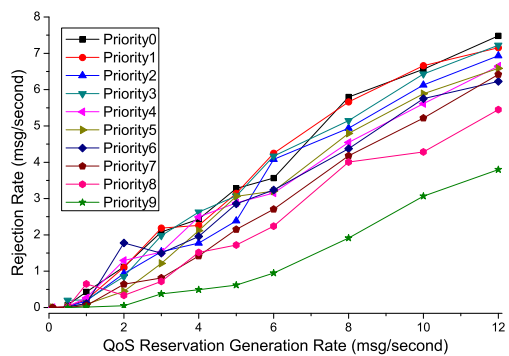


Fig. 5. Rejection rate for different priority flows as a function of traffic QoS request message generation rate

also see that the rejection ratios for all the priorities grows with growing rate of message generation. Another noticeable fact is that lower priority rejection ratios seem to converge at the higher request rates. That can be explained by the fact that at high rates devices are accommodating almost only flows with priority 9 and almost all lower priority flows are rejected. Analysis of the data also shows that the reservation rejection ratio for lower generation rates is well distributed between classes, providing good separation between different priorities.

Finally, in Fig. 5 the rejection rate characteristics are presented. The graph shows that the average rate for rejection of particular flows is dependent on their priority. The results confirm that packets with higher priority are less often rejected. The results also show that the rejection rate is much influenced by the rate of message generation. The data presented reveal that separation between classes obtained for rejection rates are not as distinct as for rejection ratios. That is because the measurements for rejection rate are more influenced by the "dynamics" of the simulations. One should notice that though on the average high priority reservations are rejected fewer times – it can happen that two consequent high priority reservations are rejected due to either insufficient resources (even after preemption) or simply not sufficiently high priority.

IV. CONCLUSIONS

The modeling discussed in this paper shows that UPnP QoS Architecture is capable of providing different levels of QoS depending on the flows importance - user priority. This is best shown by parameters like rejection ratio and rejection rate that show that higher importance number causes flows to be rejected on much lower rates. On the other hand one has to be aware that higher priority flows have on the average higher setup times. Despite the reasons for this might be obvious, i.e. higher priority is more often causing preemption which extends the setup time, it still is not desired as usually a traffic with higher priorities should be serviced with better quality. One can still argue that extension of waiting time that avoids rejection is a fair solution. It is also often a case that setup time might not be that important, in some cases small delay at the beginning of the traffic flow is not significant comparing to later in-time delivery.

UPnP signaling is well defined and performs well even using the simplest – sequential reservation and preemption approach (meaning no parallel resource management is performed). But we would like also to notice that the full specification of the UPnP QoS Architecture is quite heavy and the signaling overhead could be limited as the simulations were possible with very limited subset of original UPnP messages complexity. Signaling overhead is on the other hand not an issue for the protocol that is mainly intended for private networks where data price and volume usually is not of concern.

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7) under project 212 352 ALPHA "Architectures for fLexible Photonic Home and Access networks".

REFERENCES

- [1] "Upnp technology the simple, seamless home network a white paper," December 2006.
- [2] *Devices Profile for Web Services Version 1.1*, OASIS, <http://docs.oasis-open.org/ws-dd/dpws/wsdd-dpws-1.1-spec.html>, July 2009.
- [3] *Bonjour Overview*, Apple Inc., <http://developer.apple.com/documentation/Cocoa/Conceptual/NetServices/NetServices.pdf>, May 2006.
- [4] *Internet Grouping and Resource Sharing*, IGRS Information Industry Association, <http://www.igrs.org/en/index/index.asp>.
- [5] *Jini Architecture Specification*, http://www.jini.org/wiki/Jini_Architecture_Specification, March 2007.
- [6] *UPnP QoS Architecture:3 Service Template Version 1.01 For UPnP Version 1.0*, UPnP Forum, November 2008.
- [7] J. P. Laulajainen, P. Perala, and A. Laikari, "Evaluation of upnp qos framework performance in wireless lan," in *Consumer Electronics, 2008. ISCE 2008*, April 2008.
- [8] F. Giovanelli, G. Bigini, M. Solighetto, and P. Maggi, "A upnp-based bandwidth reservation scheme for in-home digital networks," in *ICT 2003. 10th International Conference on Telecommunications*, vol. 2, February 2003.
- [9] *UPnP QoSManager:3 Service Template Version 1.01 For UPnP Version 1.0*, UPnP Forum, November 2008.
- [10] *UPnP QoSPolicyHolder:3 Service Template Version 1.01 For UPnP Version 1.0*, UPnP Forum, November 2008.
- [11] *UPnP QoSDevice:3 Service Template Version 1.01*, UPnP Forum, November 2008.
- [12] *OPNET Modeler Ver. 14.5.A*, <http://www.opnet.com>, March 2007.