AquareYoum: Application- and Quality of Experience-Aware Resource Management for YouTube in Wireless Mesh Networks

Barbara Staehle, Florian Wamser, Matthias Hirth, David Stezenbach, Dirk Staehle
University of Würzburg, Institute of Computer Science, Chair of Communication Networks, Würzburg, Germany
{bstaehle, florian.wamser, matthias.hirth, stezenbach, dstaehle}@informatik.uni-wuerzburg.de

***Abstract*—The browser has become the users' interface to a plethora of Internet applications which are accessible from nearly everywhere and every device. The price for the simple and cheap access over the Internet is often a reduced end-user quality of experience (QoE). The reason for this is that the network ignores the content of the packets it transports and thereby neither knows which services it supports, nor if and which quality requirements have to be given. In addition, the needs of the applications can be time varying, and the network might not be able to give strict quality guarantees. We therefore advocate the idea of an application-network interaction in order to dynamically adapt the network resources if a QoE degradation is imminent. The software suite *AquareYoum* implements this approach and enables a smooth YouTube video playback in a congested wireless mesh Internet access network by dynamically selecting the least congested Internet gateway.**

I. Introduction

Nowadays, a variety of applications like Cloud Storage, Skype, IPTV, YouTube, or Massive Multiplayer Online Gaming are accessed over the Internet [1]. The application variety translates to highly varying quality requirements. This is an enormous challenge for the Internet access networks which are the bottleneck of today's communication infrastructure and consequently mainly responsible for ensuring the end-user satisfaction, or quality of experience (QoE). The problem is that the access networks do not know which applications they are transporting and can hence not give application-specific quality of service (QoS) guarantees, even if different QoS classes would be supported. Moreover is it not possible to map applications to static QoS classes as the requirements of many contemporary Internet applications are time varying. As a result, the QoE of the end user is in many cases not optimal.

We are convinced that the *Aquarema* concept [2], which is short for "Application- and Quality of Experience-Aware Resource Management" is a solution to both issues. Aquarema uses information from the application layer to dynamically adapt the network resources and thereby ensures the user satisfaction. Previous works achieve the application network interaction e.g. by modifying the entire communication stack and implementing a cross-layer solution [3]. A method applicable for a given communication infrastructure is deep packet inspection [4]. This provides a network management tool with information about the application running in the network, but is not viable for large networks supporting many applications. Detecting applications is moreover only half the coin as the network has also to know about appropriate QoS parameters. Deriving such application-specific and sometimes time-varying parameters from an observed traffic flow is even more complex, in particular for reactive TCP traffic.

QoE-based resource management (RM) which has e.g. been proposed by [5] is a solution to this problem and therefore a more promising option, as the network resources are continuously adapted to quality feedback. QoE is a measure for the subjective quality that a user experiences [6] and depends not only on the network but also on external and user specific factors. It is hence difficulty to assess, but a large number of models exist which allow to derive the QoE from network parameters, e.g. for video streaming [7]. For ensuring the user satisfaction, however, a *prediction* of the QoE is necessary in order to trigger adequate RM actions before the user gets annoyed. Therefore, Aquarema monitors the *Application Comfort* (AC) which characterizes the application performance and in particular allows a QoE prediction. Communicating this information to the network avoids the overhead of deep packet inspection and allows to react prior to a QoE degradation.

Ensuring the application QoE is a challenge for all kinds of networks but is especially problematic in IEEE 802.11 wireless mesh networks (WMNs) where the complex multi-hop topologies together with the random channel access scheme make it impossible to give fixed quality guarantees such as bandwidth, or delivery delay [8]. Hence, sophisticated RM algorithms are mandatory and we will concentrate on this type of networks in the following, although the Aquarema idea is applicable to all other kind of networks, too.

Today, roughly one third of all consumer Internet traffic is due to video transmissions, a share which will increase to nearly two thirds by the end of 2014 [1]. This is our motivation for choosing YouTube as a representative example for an Internet application and implementing the software suite AquareYoum which enables an application- and QoE-aware RM for YouTube in WMNs. In an earlier work [2] we showed that the cooperation of the YouTube AC monitoring tool YoMo and a traffic shaper which is able to restrict the bandwidth of concurring best effort traffic is suitable for maintaining the YouTube QoE. In this work, we report on a more sophisticated implementation where the used resource management tool

David Stezenbach is now with the University of Vienna: david.stezenbach@univie.ac.at.

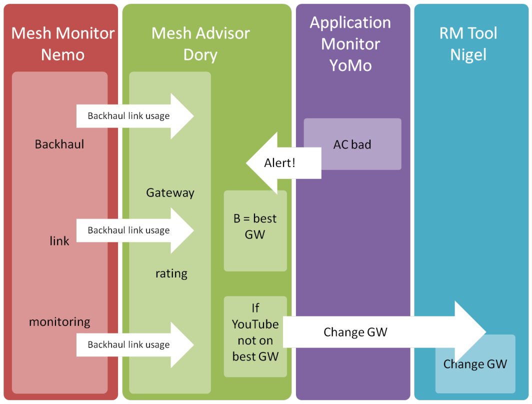

Fig. 1. Interaction between the AquareYoum components

(a) YoMo parameters for a YouTube video (b) YoMo GUI

Fig. 3. The application monitor YoMo

allows to seamlessly switch the Internet access gateway.

The remainder of this work is structured as follows. Section II is dedicated to describing the Aquarema concept for WMNs. Implementation details on AquareYoum are given in Section III and results on its performance are discussed in Section IV. In Section V we summarize our contributions and give an outlook on future work.

II. Aquarema for WMNs

Resource management in WMNs, see e.g. [8], covers routing including gateway selection, channel and interface allocation, prioritization of medium access through contention parameters, and finally traffic shaping. The core component of Aquarema is hence the *Mesh Advisor* which is informed about the applications running in the network and the current status of the network. Therefore it is able to trigger a suitable *Resource Management tool* if necessary. The network-application interaction is assured by the *Application Monitor*, a light-weight tool installed at the client. It signals the presence of an application to the Mesh Advisor and constantly monitors the AC. In case of an imminent QoE degradation, the AC of an application falls below a threshold. This is immediately reported to the Mesh Advisor which evaluates this warning together with the information about the network status it periodically receives from the *Mesh Monitor* tools and decides about a RM action. For a more elaborate description of Aquarema please refer to [2]. The interaction of the Aquarema components at the example of the subsequently discussed AquareYoum suite is shown in Fig. 1.

The key benefit of Aquarema is the creation of a win-win situation for both users and network operators: The user experiences a better QoE, because the network operator can systematically support dedicated applications. The network operator on the one hand can use resources more efficiently and on the other hand achieve a higher user satisfaction, which in turn leads to competitive advantages.

III. AquareYoum - Implementation Details

The objective of the AquareYoum software suite is to assure the YouTube QoE in large WMNs with more than one Internet gateway. The setup of AquareYoum in the i3 mesh testbed at the University of Würzburg together with its basic functionality is shown in Fig. 2. The interaction of the tools is depicted in Fig. 1.

Let us start the discussion of the tools with the Mesh Advisor *Dory* (Decider over rerouting YouTube). Dory is written in C++ and constantly receives information about the amount and type of traffic on the different backhaul links from the C++ Mesh Monitor *Nemo* (Network Monitor) whereof an instance runs on each gateway node. Fig. 1 illustrates that Nemo periodically sends its measurement data to Dory which consequently always knows which gateway is the momentarily least congested one.

The application information is provided by the Java-based YouTube application monitor *YoMo* (YouTube Monitor) which runs at the client. For more details on YoMo and the technology behind YouTube, please see [2], we just summarize the most important details. The YouTube player is a proprietary Flash application which concurrently plays a Flash video (FLV) file and downloads it via HTTP. At the beginning of the download, the client fills an internal buffer and starts the video playback as soon as a minimum level of buffered playtime is reached. During the time of simultaneous playback and downloading, the buffer grows until a certain larger threshold is reached and as long as the download bandwidth is larger than the video rate. If the buffer runs empty, the video *stalls*, i.e. the video playback is interrupted.

Determining the YouTube QoE is out of scope of this work, but what in any case negatively affects the user satisfaction is a stalling of the video. Therefore, we define the YouTube AC as the amount of playtime, the YouTube player has buffered, i.e. the amount of time, β, the player can continue playing if the connection to the server is interrupted. Fig. 3(a) shows for a sample YouTube video[1] how β can be calculated as the difference between the amount of already downloaded playtime T and the current time of the video t which are retrieved by an FLV parser and a FireFox plugin respectively. YoMo constantly monitors and visualizes β in a GUI (cf. Fig. 3(b)).

If β drops below a critical threshold, YoMo notifies Dory which now evaluates the information it received from Nemo to find the least congested gateway. Subsequently, it instructs the RM tool *Nigel* (New Internet Gateway Elector) to move the YouTube flow to this gateway and thereby avoids the stalling and the QoE degradation (cf. Fig. 1). Nigel is written in Python

[1]http://www.youtube.com/watch?v=VuQkDaKaPgk, last accessed 04/11

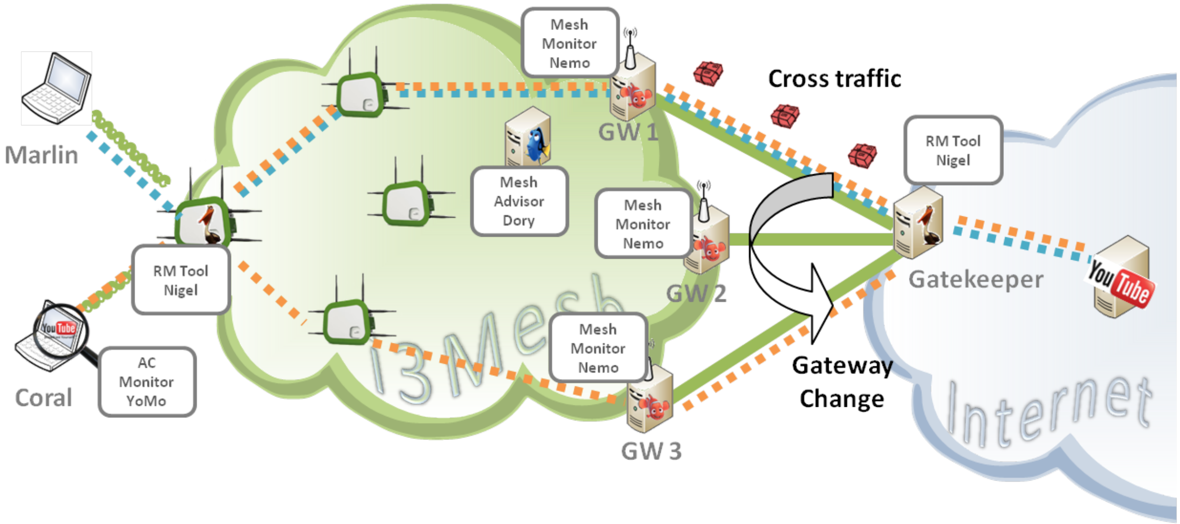

Fig. 2. AquareYoum setup and demonstration of its functionality in the testbed of the University of Würzburg

and runs on the access point of the client and the *Gatekeeper*, a machine located in the public Internet playing a similar role as the mobile IP mobility anchor. Nigel establishes an overlay network including the mesh network and the gatekeeper. This allows to seamlessly relocate a YouTube TCP flow which must not experience a change of the source IP address.

All components of AquareYoum have been developed in the context of diploma or bachelor theses. They are under constant development and available upon request. The YouTube monitor YoMo is already available for download[2].

IV. Experimental Evaluation

During the Euroview 2010 and the KiVS 2011, the functionality of AquareYoum has been demonstrated in the i3 mesh with a setup captured by Fig. 2. The laptops Coral and Marlin are both displaying a YouTube video and are faced with cross traffic congesting their Internet connection. On Coral, AquareYoum is enabled which allows Dory to move the corresponding YouTube flow to a non-congested gateway. Hence, Coral experiences a smooth YouTube video playback while the video stalls on Marlin. Screencasts of this experiment are available on the AquareYoum project site[3].

In order to prove the scalability of our concept, AquareYoum has additionally been set up in the DES-testbed of the Free University of Berlin[4], spanning three buildings of the university. We used one of the mesh nodes as access point for a client displaying a YouTube video. Three mesh nodes assured the connection to the G-Lab experimental facility[5] where the three Internet gateways and the gatekeeper were located, five more mesh nodes were used for establishing a WMN multi-hop topology. Deploying the gateways in the G-Lab experimental facility, consisting of over 100 nodes spread on 20 sites all over Germany, enables to emulate different realistic Internet qualities of the backhaul links.

[2]www.german-lab.de/go/yomo
[3]http://www3.informatik.uni-wuerzburg.de/research/mnrg/aquareyoum
[4]http://www.des-testbed.net
[5]http://www.german-lab.de

Results of this experiment are shown in Fig. 4. It consists of three subfigures which correspond to the three Internet gateways. Each of the subfigures shows the time of the video download on the x-axis. This starts with the beginning of the YouTube page load and ends when the video is completely downloaded. In all subfigures the same curve representing the playtime buffered by the YouTube player is shown on the y-axis on the right. The load on the gateways is different in contrast and illustrated on the y-axis on the left. All backhaul links are shaped to 3 Mbps, an upper bound which is indicated in all subfigures. The second horizontal line shows the alarm threshold for the buffer level which indicates the time when YoMo sends an alarm message to Dory. The shaded areas represent the time during which the client's YouTube flow is routed over the corresponding gateway. The icons indicate the time when cross traffic is started or stopped, or when Dory decides to move the YouTube flow to another gateway.

The top-most subfigure of Fig. 4 shows that at the beginning of the experiment, the YouTube flow is routed over gateway 1. During the first seconds it consumes much bandwidth as the Flash player has to be loaded and the buffer has to be filled, but as soon as a certain buffer level is reached, the YouTube bandwidth consumptions go down while β still increases. After a while a 3 Mbps CBR UDP cross traffic flow is added which completely congests this backhaul link and causes β to decrease. The shaded area ending as soon as β falls below the critical threshold of 15 sec, represents that at this time YoMo sends an alarm message to Dory which decides to move the YouTube flow to the non-loaded gateway 3. Note two things: first, the load of the backhaul link shortly exceeds the upper limit. The reason for this is that the 3 Mbps are not a hard, but a soft limit realized by a shaping script. Next, the TCP congestion control is responsible for the buffered playtime to decrease only slowly.

The bottom-most subfigure of Fig. 4 shows that after the change to gateway 3, the YouTube flow gets enough bandwidth and the buffer level increases. After some time, we congest gateway 3 by a cross traffic flow of 3 Mbps and the previously

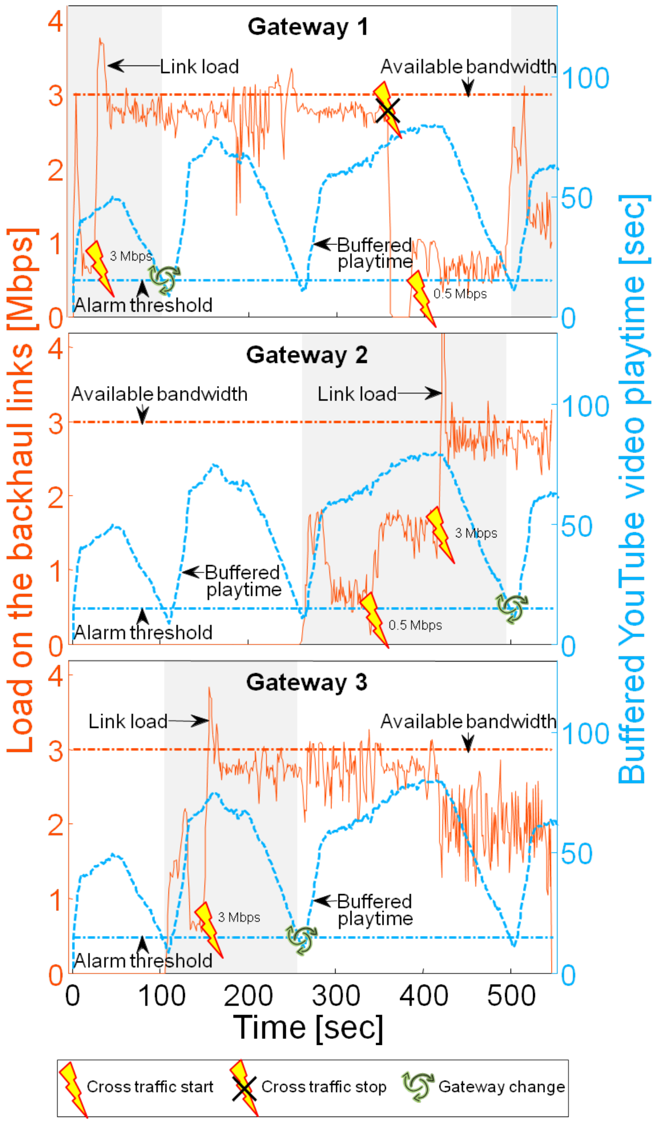

Fig. 4. Results from an AquareYoum Experiment in the DES-Testbed

described procedure repeats: after some time, the buffer level decreases, and as soon as it falls below the critical threshold, the YouTube flow is relocated to gateway 2 where it can grab enough bandwidth and the YouTube buffer level recovers.

After some seconds, we increase the load on backhaul link 2 by adding a 0.5 Mbps cross traffic flow. The capacity for the YouTube flow is however still large enough and the buffer level continues to increase. Hence, we add more cross traffic in order to completely congest the link and, as a consequence, the buffer level falls below the critical threshold some seconds later. Meanwhile the 3 Mbps cross traffic on backhaul link 1 has been replaced by a smaller traffic flow of 0.5 Mbps. When now receiving the alarm message from YoMo, Dory correctly recognizes that none of the backhaul links is completely free, but that the capacity on backhaul link 1 is sufficient for supporting the YouTube flow. Hence, the YouTube flow is moved to this link and the video download and the video playback can be terminated successfully.

V. Conclusion and Outlook

In this work, the software suite AquareYoum, which is an implementation of the application- and QoE-aware radio resource management concept Aquarema, has been introduced. The core idea of Aquarema is application comfort monitoring. The experienced AC is measured at the client and communicated to a resource management instance, hence a cross-layer resource management approach becomes possible. AC goes beyond QoE because it allows to predict the future development of the QoE and consequently enables triggering resource management decisions in order to avoid QoE degradations.

We evaluated the functionality of AquareYoum as an implementation of Aquarema both in a small and in a large WMN. We were able to show that the cooperation of the four different AquareYoum components is able to improve the QoE of a YouTube user by seamlessly moving the YouTube flow to the least congested gateway if necessary. The experimental results underline that Aquarema offers an enormous potential for WMNs and other types of networks. Our future work will therefore be dedicated to developing Aquarema in two directions. Firstly, we will develop AC monitoring tools similar to YoMo for other popular applications. Secondly, we will extend the Aquarema architecture by incorporating intelligent access mechanisms, gateway selection, channel selection, and routing for WMNs in the presence of various applications.

Acknowledgments

The authors are very grateful towards Phuoc Tran-Gia and Robert Henjes for their support and valuable comments.

Many thanks go to Sebastian Deschner, Andreas Wendl, and Andreas Blenk for their invaluable implementation work. The authors are moreover much obliged to the DES-Testbed team for their help during the experiments.

This work was funded by the Federal Ministry of Education and Research of the Federal Republic of Germany (Förderkennzeichen 01 BK 0800, GLab). The authors alone are responsible for the content of the paper.

References

[1] Cisco Systems Inc., “Cisco Visual Networking Index - Forecast and Methodology, 2009-2014,” White Paper, June 2010.

[2] B. Staehle, M. Hirth, R. Pries, F. Wamser, and D. Staehle, “Aquarema in Action: Improving the YouTube QoE in Wireless Mesh Networks,” in *BCFIC*, Riga, Latvia, February 2011.

[3] S. Khan, Y. Peng, E. Steinbach, M. Sgroi, and W. Kellerer, “Application-Driven Cross-Layer Optimization for Video Streaming over Wireless Networks,” *IEEE Communications Magazine*, vol. 44, no. 1, 2006.

[4] “Traffic Inspection for Visibility, Control and New Business Opportunities,” Ericsson White Paper, February 2008.

[5] R. Pries, D. Hock, and D. Staehle, “QoE based Bandwidth Management Supporting Real Time Flows in IEEE 802.11 Mesh Networks,” in *Special Issue PIK on Mobile Ad-hoc Networks*, January 2010.

[6] A. Van Moorsel, “Metrics for the Internet Age: Quality of Experience and Quality of Business,” in *PMCCS 5*, Erlangen, Germany, September 2001.

[7] U. Engelke and H. J. Zepernick, “Perceptual-based Qality Metrics for Image and Video Services: A Survey,” in *3rd EuroNGI Conference*, Trondheim, Norway, May 2007.

[8] I. Akyildiz, X. Wang, and W. Wang, “Wireless Mesh Networks: A Survey,” *Computer Networks*, vol. 47, no. 4, 2005.