Design of an Emulation Framework for Evaluating Large-Scale Open Content Aware Networks

Steven Latré*, Jeroen Famaey*, Tim Wauters*, Werner Van Leekwijck†, and Filip De Turck*

* Ghent University – IBBT – Department of Information Technology
Email: steven.latre@intec.ugent.be
†Alcatel-Lucent Bell Labs, Copernicuslaan 50, B-2018 Antwerpen, Belgium

***Abstract*—The popularity of multimedia services has resulted in new revenue opportunities for network and service providers but has also introduced important new challenges. The large amount of resources and stringent quality requirements imposed by multimedia services has triggered the need for open content aware networks, where specific management algorithms that optimize the delivery of multimedia services can be dynamically plugged in when required. In the past, a plethora of algorithms have been proposed ranging from specific cache algorithms to video client heuristics that are optimized for a specific multimedia service type and its corresponding delivery. However, it remains difficult to accurately characterize the performance of these algorithms and investigate the impact of an actual deployment in multimedia services. In this paper, we present a framework that allows evaluating the performance of such algorithms for open content aware networks. The proposed evaluation framework has two important advantages. First, it performs an emulation of the novel algorithms instead of using a simulation approach, which is often carried out to characterize performance. Second, the emulation framework allows evaluating the impact of combining different multimedia algorithms with each other. We present the architecture of the emulation framework and discuss the main software components used. Furthermore, we present a performance evaluation of an illustrative use case, which identifies the need for emulation-based evaluation.**

I. Introduction

In the last decade, multimedia services have become the most important type of services supported in the Internet. Because of this popularity, there is also a growing attention towards management approaches that specifically optimize the delivery of multimedia services. Given the increasing popularity of multimedia services, it is quickly becoming inviable to support the delivery of multimedia services by simply applying over-dimensioning. Instead, intelligent multimedia algorithms are investigated and proposed that try to deliver multimedia services at the highest quality possible, but with the lowest delivery cost possible for network and service providers. Hence, there is a need for a new management paradigm for multimedia services, which we call open content aware networks: network and service providers should be able to deploy specific management algorithms that optimize the delivery of multimedia services. In the past, a myriad of Quality of Experience optimizing multimedia services have been proposed, both in network and at the edges of the network. For example, in the past, a wide range of caching algorithms, which have been optimized for multimedia delivery [1], [2], have been proposed. Other examples include (i) specific video streaming solutions at the video server [3], (ii) the scheduling of deadline-aware content in the network [4], (iii) admission control techniques [5] and (iv) adaptive streaming solutions such as the recently proposed Dynamic Adaptive Streaming over HTTP standard (DASH) [6]. Open content aware networks bring these solutions together to provide an end-to-end management of multimedia services.

The performance of these solutions is often limited for two reasons. First, they are often evaluated using network-based simulation tools such as NS [7] and Omnet++ [8]. While such type of simulation allows to quickly perform a large parameter sweep of the problem domain, it also typically simplifies the problem domain leading to a reduced accuracy. For example, network-based simulation tools can provide an insight in the performance of multimedia algorithms in terms of network metrics but typically ignore node-oriented metrics such as the load of a server. Second, the evaluation of novel multimedia solutions typically only focuses on the performance of a single multimedia algorithm. However, the combined impact of different multimedia algorithms is typically of more importance for a network and service provider.

Building efficient experimentation facilities has recently gained increased attention, partly due to the European FIRE [9] framework and the United States GENI [10] project. However, many of the emulation-based facilities (e.g., Ofelia [11], Crew [12]) and automated deployment approaches [13] are often very generic and therefore not immediately applicable to the evaluation of multimedia streaming scenarios. On the other hand, a wide variety of extensions to common simulation software tools can be found that specifically focus on multimedia delivery. An example of such extension is a simulation environment for Content Delivery Networks, proposed by Stamos et al [14]. However, these extensions have the downside of simulations, described above: they only focus on one aspect of the performance characterization and sometimes have a reduced accuracy caused by simplifying assumptions.

In this paper, we aim to close the gap between the multimedia-optimized simulation tools and generic emulation frameworks. We present an emulation framework that has

978-3-901882-50-0

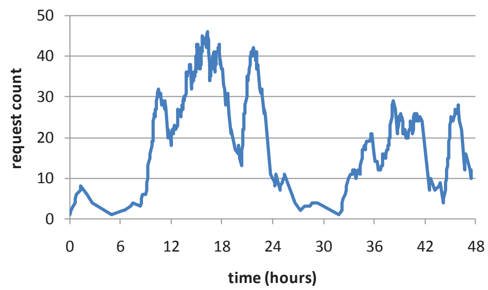

Fig. 1. Exemplary request pattern used by the emulation framework.

been developed in the course of the FP7 OCEAN project, which focuses on a multimedia optimized type of network management. The goal of the emulation framework is to provide a network and service provider a more accurate view on the performance of the different multimedia algorithms, which is needed in order to assess the viability of a possible deployment in a production network.

The remainder of this paper is structured as follows. We first provide an overview of the main validation components of the OCEAN emulation framework in Section II. Next, we describe an illustrative use case and discuss the functional components that were deployed into the emulation framework to support this use case in Section III. Next, the added benefit of applying emulation instead of simulation is shown through an evaluation of the use case scenario in Section IV. Finally, Section V concludes this paper.

II. Overview of the validation software components

A. Network infrastructure

The OCEAN emulation framework is run on top of the iLab.t Virtual Wall testbed facility, which is a large-scale network emulation environment that consists of 100 nodes interconnected via a non-blocking 1.5 Tbps VLAN Ethernet switch, and a display wall (20 monitors) for experiment visualisation. Each server is connected with 4 or 6 gigabit Ethernet links to the switch. The Virtual Wall nodes can be assigned different functionalities ranging from terminal, server and network node to impairment node. The management of the physical topology is carried out by the Emulab [15] testbed management software.

Furthermore, specifically for the emulation framework, an automation tool has been developed, called Wall Experiment Organizer (WExO). WExO allows alleviating the problems associated with manual experiment setup on the Virtual Wall testbed. WExO allows defining planned experiments and parameter sweeps via an XML file or by using a graphical user interface. It uses a set of plugins to define the required emulation: a plugin consists of a Java JAR file, which defines the actual start and deployment of software components. An overview of these functional components is discussed in the next section. Additionally, WExO allows easy repeatability of the experiments and automated log and result processing.

B. Traffic shaping

In order to limit the traffic between the links in the Virtual Wall testbed facility, a set of traffic shapers are used. These traffic shapers allow limiting the throughput, increasing the round trip time and increasing the packet loss, experienced on the link. The required traffic shaping can be configured using WExO and is achieved through the `tc` tool. Their role is to emulate realistic conditions that correspond with actual production networks.

C. User behavior emulation

The OCEAN framework allows emulating the behavior of users through request patterns. A request pattern is a function that defines the time, client, and content item of every request. An example of such a request pattern is shown in Figure 1. As emulations are typically performed in real-time, it may occur that request patterns span multiple days, which significantly increases the time required for performing a single emulation run. If wanted, the OCEAN emulation framework allows increasing the speed of emulations by (i) increasing the frequency of request patterns, (ii) shortening the length of each video and (iii) configuring the network link configurations accordingly.

D. Experiment monitoring

A third an final set of validation software components allow monitoring the status of the experiment. A set of monitoring probes is active throughout the emulated network, to allow gathering and interpreting the emulation results. The monitoring probes allow to (i) capture network-related statistics such as the experienced packet loss and observed bandwidth consumption of each video, (ii) collect access statistics such as when each request occurred and the name of the video requested and (iii) monitor performance of the specific nodes such as CPU and memory consumption.

III. Use Case: Cache-enabled video streaming

In this section, we present an illustrative use case that was evaluated using the emulation framework. We first discuss the scenario and then detail the software used for emulating this scenario.

A. Scenario overview

The use case represents a scenario in which a set of clients consumes a multimedia service from a service originator. We assume that the video is delivered using the HTTP Adaptive Streaming (HAS) delivery technique and that a set of caches can be placed inside the network to optimize the delivery of the multimedia services. In this use case, we specifically focus on the performance of the cache replacement algorithms.

Figure 2 shows a high level overview of the investigated topology. At one end it consists of a single HTTP web server, which is responsible for hosting the HAS video content. This server is directly connected to an intermediary HTTP proxy. This intermediary proxy is the cache which is under investigation and is hence responsible for temporarily caching

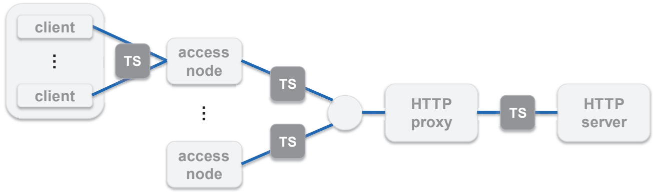

Fig. 2. Overview of the investigated topology.

popular content in order to reduce bandwidth load on the upstream link and resource load on the source server. Further down is a set of access nodes. They form the gateway of clients to the access network. These clients represent end-users interested in the multimedia content hosted by the server. In order to be able to emulate larger scenarios, the clients located behind a single access node are emulated by one physical node. However, the access nodes have no caching functionality and merely fulfill the role of monitoring point for the experiments presented in this document.

The network topology contains several configurable parameters, which can be varied throughout the emulations. The delay and bandwidth can be configured on the upstream link of the proxy, access nodes and clients. Additionally, a request trace can be configured on each client emulator, to indicate when client requests should be sent and which content item they should request. Finally, the number of access nodes can be varied. The network consists of a web server which hosts the HTTP video content, an intermediary HTTP proxy and several sets of clients connected through an access node. Network link characteristics, such as bandwidth, delay and packet loss, are emulated via traffic shaping (TS) nodes.

B. *Functional software components for multimedia delivery*

To enable the actual emulation of the use case described above, a set of functional software components is deployed into the emulation framework. We distinguish between server, in-network and client components.In order to support the HAS delivery of video streams at the server, a standard Apache web server is constructed. The OCEAN emulations focus on HTTP Adaptive Streaming-based delivery in which the video content is segmented in time and encoded into different video qualities. For this, the video content is encoded using the Microsoft Expression Encoder version 2.1 into Smooth Streaming format 1.0. Seven different bitrates are supported ranging from 300 kbps to 2.5 Mbps. Inside the network, a set of caches are deployed, each with their cache replacement algorithm. The cache replacement algorithms are implemented in the SQUID proxy software. For the scope of this paper, we focus on SQUID's default cache replacement algorithm (called segment-based LRU), which is the standard implementation of LRU on the SQUID proxy. To emulate the clients, an Advanced Video Coding (AVC) client emulator is used. It was developed within the context of the OCEAN project and behaves as a regular Microsoft Smooth Streaming client. However, video decoding and rendering has been disabled. This allows a large number of clients to be emulated on a single physical machine and in turns allows us to perform larger-scale experiments, using a relatively limited amount of physical resources.

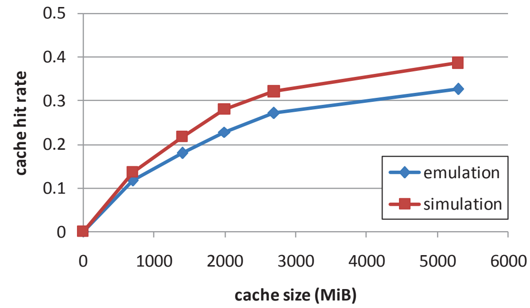

Fig. 3. Comparison of the observed cache hit rate for an increasing cache size. The results were obtained either through simulation or emulation.

IV. Performance Evaluation

A. *Accuracy of simulation*

In a first experiment, we compare the emulation results, achieved by the emulation framework, with simulation results of the same algorithm set. The experiment compares simulation results for SQUID's segment-based LRU with emulation results for Smooth Streaming with LRU. However, as in the simulations no quality adaptation was allowed, only the 300 Kbps quality layer was enabled. Additionally, all traffic shaping was turned off, resulting in 0 ms delay and 1 Gbps bandwidth on every link. The goal is to determine whether or not the simulation and emulation results differ, and if they do, what the reason is behind the difference.

Figure 3 depicts the comparison results: the cache hit rate is characterized for an increasing cache size. The graph shows a significant difference in cache hit rate. However, both curves do show a similar trend. There are several possible explanations for the lower cache hit rate in the emulation results. An important factor is the perfect conditions under which the simulations were performed. First, delivery conditions are perfect, meaning there is no packet loss, delay or synchronization issue. Second, in a real deployment, video segments contain protocol overhead. More specifically, every segment contains a HTTP header. As a consequence, a larger cache is needed to be able to cache the same number of segments. Another difference is the fact that in the simulations, all segments are identical, while in the emulations this is far from the case. All segments are 2 seconds long, but as the video fragments are VBR, their size differs significantly. Hence, different segments of the same video no longer have the same caching value. If two segments are equal in popularity, but one of them takes significantly less space in the cache, prioritizing the smaller segment might yield better caching performance. LRU does not take this aspect into account.

These results clearly show the added benefit of applying emulation. While the simulation results identified the same trend, the emulation results are more accurate as they take into account more complex interactions between the entities in the network, which was not taken into consideration in the

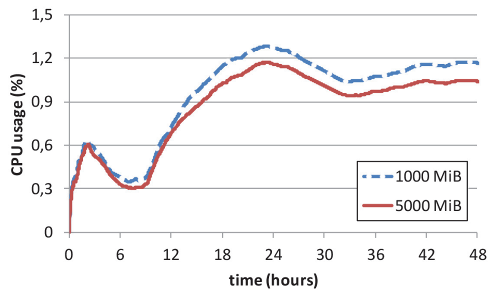

Fig. 4. Running average of the CPU usage over time.

simulations. On the other hand, emulations are more complex to setup and are more prone to failures.

B. *Impact on the CPU consumption*

As discussed, a downside of typical network-based simulation tools, is that they ignore non-network related metrics such as CPU and memory consumption. As an example, we investigate the CPU utilization of an experiment, which was executed in the emulation framework. The CPU utilization of the Proxy cache is an effective measure for the caches scalability. In this section we study the effects of increasing the cache size on CPU utilization.

Figure 4 depicts the running average CPU usage over time for a cache size of 1000 and 5000 MiB. The increases in average CPU usage correspond to times of high activity in the request trace (as illustrated in Figure 1). As expected, more simultaneous requests lead to high CPU consumption. Nevertheless, the consumed CPU values are very low, even when up to 50 simultaneous requests are active. For reference, the maximum CPU consumption for a cache of 1000 MiB, is 2%. When comparing the performance of the different cache sizes, it is shown that a 1000 MiB cache size has a higher CPU consumption than a 500 MiB cache size. This is because a smaller cache, leads to more replacements in the cache and thus a higher computational complexity.

V. Conclusions

We presented an emulation framework for evaluating multimedia networks, which was designed in the course of the OCEAN project to confirm earlier obtained results through simulation. The proposed framework is specifically intended for characterizing the combined impact of several algorithms for managing the delivery of multimedia services. It allows to make a detailed and accurate comparison of the performance of these algorithms in realistic scenarios. The emulation framework features a high level of automation as it allows new components to be easily plugged into existing experiments: as such, the flexibility of the emulation framework is high and the time required to develop and evaluate new algorithms is small. We illustrated the working of the emulation framework through an exemplary use case, in which the performance of several cache algorithms in a HAS delivery scenario is evaluated. The results showed the most important benefits of the emulation framework compared to traditional simulation-based evaluation. First, it features a higher level of accuracy as it emulates the transmission of real video data and all the interactions between distributed components and thus makes fewer simplifying assumptions that reduce the realism of the experiments. Second, in contrast to simulation, it can focus both on network related statistics and node related statistics.

Acknowledgment

Steven Latré and Tim Wauters are funded by grant of the Fund for Scientific Research, Flanders (FWO-V). The research was performed partially within the framework FP7 STREP project OCEAN (under grant agreement no. 248775).

References

[1] X. Li, W. Tu, and E. Steinbach, "Dynamic segment based proxy caching for video on demand," in *Multimedia and Expo, 2008 IEEE International Conference on*, 23 2008-april 26 2008, pp. 1181 –1184.

[2] Z. Zeng, B. Veeravalli, and K. Li, "A novel server-side proxy caching strategy for large-scale multimedia applications," *J. Parallel Distrib. Comput.*, vol. 71, no. 4, pp. 525–536, Apr. 2011. [Online]. Available: http://dx.doi.org/10.1016/j.jpdc.2010.06.008

[3] S. Fernandes, J. Kelner, and D. Sadok, "An adaptive-predictive architecture for video streaming servers," *J. Netw. Comput. Appl.*, vol. 34, no. 5, pp. 1683–1694, Sep. 2011. [Online]. Available: http://dx.doi.org/10.1016/j.jnca.2011.05.007

[4] K. De Schepper, B. De Vleeschauwer, C. Hawinkel, W. Van Leekwijck, J. Famaey, W. Van de Meerssche, and F. De Turck, "Shared content addressing protocol (scap): Optimizing multimedia content distribution at the transport layer," in *Network Operations and Management Symposium (NOMS), 2012 IEEE*, april 2012, pp. 302 –310.

[5] B. Meskill, A. Davy, and B. Jennings, "Revenue-maximizing server selection and admission control for iptv content servers using available bandwidth estimates," in *Network Operations and Management Symposium (NOMS), 2012 IEEE*, april 2012, pp. 319 –326.

[6] T. Stockhammer, "Dynamic adaptive streaming over http - standards and design principles," in *Proceedings of the second annual ACM conference on Multimedia systems*, ser. MMSys '11, ACM. New York, NY, USA: ACM, 2011, p. 133–144. [Online]. Available: http://doi.acm.org/10.1145/1943552.1943572

[7] S. Papanastasiou, J. Mittag, E. G. Strm, and H. Hartenstein, "Bridging the gap between physical layer emulation and network simulation," in *Proceedings of IEEE WCNC Conference, 2010*, April 2010.

[8] "Omnet++ network simulation framework," http://www.omnetpp.org - Last accessed on 30 August 2012.

[9] A. Gavras, A. Karila, S. Fdida, M. May, and M. Potts, "Future internet research and experimentation: the fire initiative," *SIGCOMM Computer Communication Review*, vol. 37, no. 3, pp. 89–92, 2007.

[10] GENI Project Office, "GENI Control Framework Architecture," October 20, 2008, document ID: GENI-ARCH-CP-01.4.

[11] A. Kpsel and H. Woesner, "Ofelia pan-european test facility for openflow experimentation," in *Towards a Service-Based Internet*, ser. Lecture Notes in Computer Science, W. Abramowicz, I. Llorente, M. Surridge, A. Zisman, and J. Vayssire, Eds. Springer Berlin Heidelberg, 2011, vol. 6994, pp. 311–312. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-24755-2_30

[12] L. A. DaSilva, L. Doyle, D. Finn, J. Tallon, I. Moerman, and S. Bouckaert, "Crew: Building a cognitive radio federation," in *IC0902*, Bologna, Italy, 11/2010 2010.

[13] S. Latré, W. Van de Meerssche, S. Melis, D. Papadimitriou, F. De Turck, and P. Demeester, "Automated management of network experiments and user behaviour emulation on large scale testbed facilities," in *Network and Service Management (CNSM), 2010 International Conference on*, oct. 2010, pp. 254 –257.

[14] K. Stamos, G. Pallis, A. Vakali, D. Katsaros, A. Sidiropoulos, and Y. Manolopoulos, "Cdnsim: A simulation tool for content distribution networks," *ACM Trans. Model. Comput. Simul.*, vol. 20, no. 2, pp. 1–40, 2010.

[15] B. White, J. Lepreau, L. Stoller, R. Ricci, S. Guruprasad, M. Newbold, M. Hibler, C. Barb, and A. Joglekar, "An integrated experimental environment for distributed systems and networks," in *Proc. of the Fifth Symposium on Operating Systems Design and Implementation*. Boston, MA: USENIX Association, Dec. 2002, pp. 255–270.