Open ContEnt Aware Networks
www.ict-ocean.eu

Deliverable D6.6

Definition of Evaluation Scenarios – Phase II

Public deliverable, Version 1.1, 31 January 2012

Authors	
FT	
AL-BELL	Werner Van Leekwijck
IBBT	Jeroen Famaey
IDATE	
N2Nsoft	
UVS	
FRAUNHOFER	
TP	Łukasz Oleszkiewicz, Paweł Grochocki, Marcin Pilarski, Wojciech Reichel, Krzysztof Wójcik, Bogdan Banasiak
UERT	
Reviewers	Steven Latré (IBBT)

Abstract

This document describes a selection of technical use cases for the PHASE II emulations and field trial, based on the status of deliverables and algorithms worked out in other work packages.

Further, the objectives used for validation, and the requirements and platforms (middleware, OS versions, end-user devices, etc.) for the test beds (both emulation platform and field trial) are described.

Project funded by the European Union under the
Information and Communication Technologies FP7 Cooperation Programme
Grant Agreement number 248775

EXECUTIVE SUMMARY

This document describes a selection of technical use cases for the PHASE II emulations and field trial, based on the actual status of deliverables and algorithms worked out in other work packages. The main focus of the technical use cases will be on adaptive streaming of segmented video items, where each client is served by a dedicated (personal) stream. Clients are connected via a cache enabled network. Emulation experiments will incorporate both single and double layer of caching using a combination of intermediate and leaf caches. Field experiments will focus on the longer term single layer architecture with small leaf caches only. For the caching algorithms, Least Recently Used (LRU) wil be used as the benchmark, and mLRU will be evaluated both in the emulation and field experiments. In addition, the emulation experiments will also further evaluate Chunk-based Caching (CC). A thorough comparison of AVC-based and SVC-based adaptive streaming will be made in both emulation and field testing.

Further, the objectives used for validation, and the requirements and platforms (middleware, OS versions, end-user devices, etc.) for the test beds (both emulation platform and field trial) are described. On one hand, the performance criteria include criteria that were also taken into account in the simulation experiments in WP4 and WP5, with the aim to verify the simulation results in an emulation environment with real data files, chunks, flows, and network protocols.

On the other hand, specific criteria that have not been taken into account in the simulations like processor load required for running the specific caching and streaming algorithms will be investigated and compared for the different algorithms.

The IBBT Virtual Wall will be used to execute the emulation experiments, and several functional components for HTTP adaptive streaming, caching, and emulation will be integrated. For the field trial, PlanetLab nodes will be used.

TABLE OF CONTENTS

INTRODUCTION

The activities in WP6 focus on the design, construction, integration and evaluation of various experimental tools, in a two-iteration approach, in order to validate the OCEAN project concepts and prototype implementations, as well as to feed project dissemination actions.

In task 6.1 "Definition of Evaluation Scenarios", a first activity has identified the interesting use cases for validation, based on the business and architectural requirements determined in WP2 and WP3. The actual validation of the work, in addition to the in-lab simulations performed on the single technological components in WP4 and WP5, will be performed via emulations. The emulation driven experimentations in Task 6.3, using the integrated prototype software developed in Task 6.2, will verify the scalability of the approach taken in WP4 and WP5 in a distributed hardware environment. Task 6.4 will perform field trial validation through an actual, physical system deployment on the Internet via PlanetLab nodes. Its purpose is to complement the other validation efforts.

This document describes a selection of technical use cases for the PHASE II emulations and field trial. Based on the actual status of deliverables and algorithms worked out in other work packages, it defines the use cases, with the rationale for their selection. Obviously, the number of use cases selected for emulation testing is smaller that the number of cases that can be handled in simulations in the other work packages. Selection criteria include the (business) relevance of certain cases, promising simulation results, and emulation feasibility. The field trial validation consists of a further subset of the use cases.

Further, it describes the objectives and scenarios used for validation, and the requirements and platforms (middleware, OS versions, end-user devices, etc.) for the test beds (both for emulation and field trial).

1. TECHNICAL USE CASE SELECTION

1.1 Service Use Cases

The selection of use cases for the Phase II emulations and field trial is based on D2.2 "Final requirements for Open Content Aware Networks", D2.3 "Use cases definition and market scenarios for future OCEAN services", and D2.4 "Business models for future OCEAN architectures".

1.1.1 Requirements

D2.2 [1] gives the definition of a comprehensive set of technical requirements for the architecture and functions to be developed by OCEAN. These requirements cover network-related aspects such as video delivery related requirements and Quality of Experience (QoE)/Quality of Service (QoS) related requirements, but also more content oriented aspects such as requirements related to content creation and consumption-related services. The main conclusions of D2.2 include:

- Consumption of online video services is already significant and will become even more important in the future. Online **video will become the primary type of Internet traffic**.
- The biggest driver so far for video traffic has been the availability of free content, through free short clip platforms and streaming platforms, and more recently catch-up TV and social networking platforms. For most of those platforms, end-users are not really expecting a high level of QoS, except for catch-up TV. However, with more content being made available and users migrating progressively to paid videos, the duration of the videos consumed will become longer, the number of viewed videos will rise, and **better QoS** and better encoding (HD and/or 3D) will be expected. Adaptive technologies and higher bandwidth availability on the last mile (e.g. fiber) will be enablers for these services.
- To offer the appropriate QoS, interconnecting CDNs of traditional providers, telcos and content providers could be a good approach to solve some of the issues for QoS, whereby traditional Global CDNs would provide efficient global distribution and telco CDNs take care of efficient local delivery to end-users. This **CDN interconnection** is challenging both from a business and a technical perspective with standardized interfaces to be defined between all involved players to manage data and metadata.
- QoS is more and more considered as a major stake for professional content, paid or not, with major interests for time to start or seamless experience. As an open platform, OCEAN should support as much as possible all of the main video technologies, with a focus on the most popular current and future technical choices, including **H.264 and adaptive streaming**.

Based on the above conclusions on technical requirements, Phase II emulation and field trial will focus on **adaptive streaming of high quality video**. QOS parameters will be an important aspect of the metrics in the evaluation.

The network last mile will not be considered as a bottleneck, but congestion could happen on aggregation or other network links. Emulation experiments will allow a deterministic control over packet loss and congestion points in the (emulated) network, while field trial experiments will use the real Internet status in terms of congestion and delay at different times of the day and week.

CDN interconnection will not be validated through the experiments as the main challenges are related to business models and interface specification agreements.

1.1.2 Use cases and market scenarios

D2.3 [2] first provides a view on the overall landscape and evolution of the TV and video in order to put the evolution of the online video industry (and underlying technologies like OCEAN) in perspective with major trends that are expected to unfold and entice all players to develop IP-based digital distribution strategies. Then, it proposes three “extreme” scenarios for the online video market development, as well as an intermediate “reference scenario” to be used as a basis for further evaluation of the CDN market.

The main findings related to the overall landscape and evolution of TV and video include:

- While television viewing time is globally stagnating, the use of Time Shifting (TS), i.e. on-demand access to television programming on a Personal Computer (PC) or a television (TV) set is growing fast. Moreover, new personal multimedia devices such as tablets are likely to further increase **personal, non-linear uses.** Additionally, online video will accelerate **segmentation.**
- The process of making video **content available illegally** is becoming more and more professional. However, new legal frameworks, technical solutions, and especially the increasingly widespread availability of legitimate services on the Web are decreasing the window of opportunity for piracy.
- New approaches to **the way viewers select their programming** are emerging. (e.g. search engines, social network sites, ..)
- **3D** is being gradually introduced but is likely geared toward proposing a specific experience for a **limited number of genres of programs**, rather than becoming the next standard for watching video.
- Television advertising revenue is at risk from Internet competition. Managed television networks are challenged by the open Internet, as the Internet is becoming a full video distribution platform. **Online video challenges business models and the value chain.**

Based on the above, the Phase II emulations and field trial will focus on **personal video streams** (i.e. no multicast scenarios), where video content is made available to individual users via a specific flow.

Security and DRM topics are out of scope of the emulation and field testing, as no dedicated investigation was done on these topics in the OCEAN project.

1.1.3 Business models

D2.4 [3] assesses business models for the content provider and telco for several scenarios and discusses several new options for the future. On the technical side, it builds further on the requirements and scenarios discussed in D2.2 and D2.3.

1.2 Architecture

The architecture for Phase II emulation and field trial is based on D3.1 “OCEAN functional architecture and open interface specification”, and D3.2 “OCEAN architecture for FTTH and DSL access networks”.

1.2.1 Functional architecture

D3.1 [4] focuses on the functional architectural framework and open interfaces. It investigates several interconnection models.

For Phase II, again an architecture in which users are connected to a **cache network** is selected. The proxies either serve the end-users directly, or forward the request upstream to the next level of caches or the origin server. This architecture does not require the implementation of the request routing mechanism, and allows testing to focus on the caching algorithms and adaptive streaming behavior. No specific NNI interface will be required in Phase II.

Initial simulation results have shown significant benefits of caching and adaptive delivery, without specific resource management or other functions at the network layer. Phase II experimentation will focus on this lightweight architecture without TNI interface or usage of underlying functions.

1.2.2 CDN topology

D3.2 [5] identifies the mapping of the most relevant Telco CDN topologies on top of a network using xDSL and FTTx access technologies. It analyzes a number of basic scenarios, as well as some optimized and experimental scenarios. The main longer term results relevant for the definition of emulation scenarios are:

- A further increase in service uptake, higher steaming bitrates, and stronger FTTH penetration, favors architectures with Leaf Cache Servers (LCS) at the network edge nodes over Intermediate Cache Servers (ICS) at the regional POPs.
- Architectures with a single level of LCS at the network edge nodes perform about the same as architectures in which a second level of ICS at the regional POPs is added.
- Experimental scenarios with small integrated caches offer interesting perspectives provided that the integrated caches can meet certain cost targets.

Based on the above findings, the **Phase II emulation** tests will incorporate both single and double layer of caching architectures using **a combination of LCS and ICS**.

The **field trial testing** will focus on the longer term single layer architecture with **LCS**.

1.3 Caching

The caching algorithms and criteria for evaluation are based on D4.3 “Evaluation of optimized caching for a highly distributed CDN”, and D4.7 “Optimized on-line learning algorithms for a highly distributed CDN”.

1.3.1 Optimized caching

D4.3 [6] summarises the results obtained in task T4.1, with aim to evaluate reactive caching algorithms and cache collaboration strategies for a set of caches deployed in typical tree-based access network topologies over which a highly dynamic content library is offered. The conclusions are:

- There is no caching algorithm that outperforms the others by a considerable margin.
- Modified Least Recently Used (mLRU), the scoring-based SC caching algorithm and the Cache Management based on Temporal Pattern Solicitation (CMTPS) algorithm perform a little better than Least Recently Used (LRU), the most commonly used caching algorithm today.
- Chunking increases the performance for small caches.
- Borrowing caches increase the performance with respect to hierarchical caching on the link upstream of the intermediate cache. Federated caches are probably too complex to implement for the gain they bring in tree-based access networks.

Based on these conclusions, Phase II (emulation and field tests) will use **LRU as the benchmark** caching algorithm. As mLRU is only marginally more complex than LRU, while SC and CMTPS are more difficult to implement, **mLRU** will be used to validate and refine the simulation results of WP4. All video **content will be chunked** and caching algorithms will operate on individual chunks as cacheable items.

In addition, the Phase II emulation experiments will also evaluate Chunked-based Caching (CC), a refined version of the Scoring-based Caching (SC) algorithm, and use hierarchical caching.

1.3.2 Online learning

D4.7 [7] concluded that chunking and predicting the probability that a chunk will be visited, is still beneficial when some users with some probability abort the video they are watching. Further, it concluded that real implementations of the advanced predictions algorithms (that can only rely on past data) perform better than standard LRU, but are outperformed by LFU with the optimal window size. It also demonstrated that exploiting the diurnal pattern for the peak traffic is not enough to have a gain in hit ratio.

As a result, the phase II test will not invest in implementing and testing more advanced popularity prediction algorithms, as the potential benefits over the chunk-based algorithms mentioned in 1.3.1 are small.

1.4 Adaptive Delivery

The delivery mechanisms for Phase II evaluation are based on D5.2 “Evaluation of congestion control via the scalable video codec”, D5.4 “Media adaptation and transmission control in a highly distributed CDN”, and D5.7 “Protocol implementation for highly distributed content delivery”.

1.4.1 Congestion control via SVC

D5.2 [8] evaluates the use of the scalable video codec (SVC) for congestion control. It analyses several techniques in combination with SVC: Priority Media Delivery (PMD) with HTTP streaming, TCP-friendly congestion control algorithms, the Pre Congestion Notification (PCN) mechanism, and graceful video rate adaptation. The main conclusions include:

- Although SVC imposes a coding efficiency penalty, SVC allows overcoming of longer interruptions than the actual pre-buffer time, which makes it suitable for scenarios with constrained bitrates, potential link interruptions and transmission rate variations. The application of SVC in HTTP streaming has furthermore the additional benefit that the network and caching infrastructure is used more efficiently when compared to adaptive HTTP streaming based on bit stream switching.
- The combination of SVC with Priority-based Media Delivery enables a longer, smoother playback when interruptions and transmission rate variations affect the channel resulting in an improved user experience.
- The combination of SVC with two TCP-friendly congestion control protocols (SVC-TFRC and SVC-TEAR) shows alternating results in terms of number of interruptions due to congestion and level of throughput.
- Dynamic video rate adaptation for SVC with a PCN system including a buffering mechanism improves network utilization in terms of number of admitted sessions and QOE.
- Graceful adaptation of SVC encoded video streams can provide gains of up to 100% compared to traditional Resource Admission Control (RAC) systems.

Considering the promising simulation results of using **SVC in combination with (popular) HTTP streaming and priority-based delivery**, Phase II experiments (emulation and field tests) will focus on validating and refining those results.

SVC-TFRC and SVC-TEAR results are less conclusive on potential advantages, and will be omitted for Phase II.

PCN systems are not widely deployed yet, and RAC systems are specific to more managed operator networks. Additionally typical RAC systems are often centralized and hard to couple with a dynamic video rate adaptation algorithm as detailed in D5.2. Hence, focus of Phase II experiments will be on regular IP network technology.

1.4.2 Media adaptation

D5.4 [9] describes client emulation software and a network proxy for emulation testing and field testing of HTTP adaptive streaming. The modified Squid proxy used in Phase I, will also still be used for Phase II emulation experiments. In addition, D5.4 also presents new results on SVC-based PCN and media rate adaptation which further refine those of D5.2.

However, as stated above in section 1.4.1, PCN systems will not e part of the Phase II experiments.

1.4.3 Highly distributed delivery

D5.7 [10] describes the technique considered to be the most promising solution for highly distributed media delivery to different device types including media scalability with SVC. It also describes the implementation of the OCEAN SVC HTTP Streaming client to be used in Phase II experiments. Main conclusions of the simulations include:

- SVC-based HTTP-Streaming outperforms AVC-based HTTP-Streaming for live IPTV
- by using SVC, receivers can request a representation that approximates more to the average TCP throughput than when AVC is considered
- the receivers can react much faster to network variations using SVC, which prevents the video from being disrupted when the traffic in the network unexpectedly increases

Both Phase II emulation experiments and field testing will thoroughly evaluate **SVC-based HTTP adaptive streaming** and compare with AVC-based HTTP adaptive streaming. Criteria for evaluation include required client buffer size, throughput, reaction to network variations, and also network proxy criteria such as cache hit ratio, processing requirements, etc.

2. VALIDATION SCENARIOS

2.1 Emulation experiments

2.1.1 Experimental set-up

As in Phase I, the emulation experiments are different from the simulations of WP4 and WP5 as they use real data files, processors, cache memories, network links, switches and routers, and (TCP) video flows. While the simulator is essentially event-driven, fed by a list of events, and using a time-stepped (e.g. 10 sec) simulation approach to speed up the simulations, the emulation is running experiments in real time on a real network, be it constrained and scaled-down in bandwidth and possibly memory sizes to maximize the number of experiments that can be carried out in a given time period.

Figure 1 shows the functional components of the emulation set-up, which is similar to Phase I. Clients, proxies, and servers have to support both AVC-based and SVC-based HTTP adaptive streaming functionality. More details can be found in section 3.

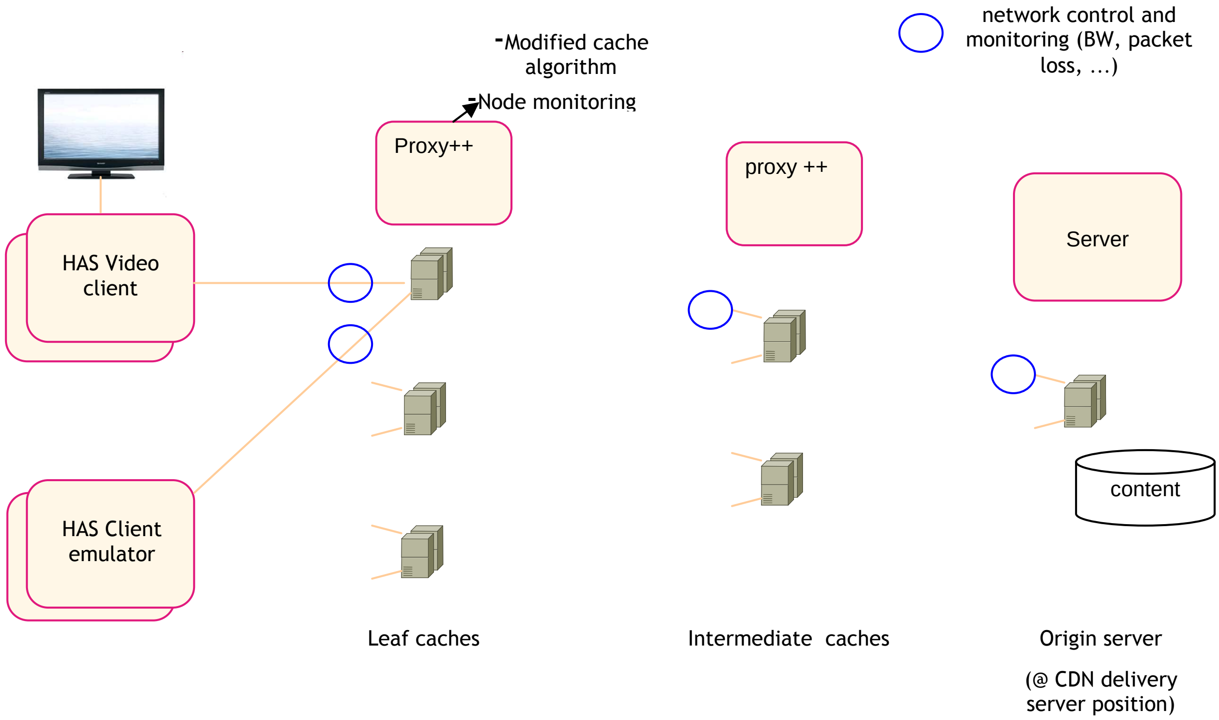


Figure 1: Functional blocks for emulation set-up

2.1.2 Objectives and rationale

- The Phase II emulation experiments aim to answer several questions not addressed by the Phase I emulations. More specifically, the following new topics will be addressed: In Phase I, all clients were emulated on a single physical node. However, due to memory constraints and other overhead, this allowed only up to around 40 simultaneous clients to be emulated in SVC-based experiments. In order to verify the results under heavier concurrent loads, multiple physical nodes will be used to emulate more simultaneous video clients. This also allows taking into account heterogeneous clients (i.e., clients with heterogeneous network constraints with respect to the last mile).

- In contrast to simulations, emulations need to be performed in (near) real-time. In the Phase I experiments, time was sped up 9 times, allowing a 48 hour trace to be emulated in just over 5 hours, and reducing the length of 90 minute movies to 10 minutes. Increasing the speed-up factor would allow longer traces to be used, but would lead to loss in precision. The goal in Phase II is to evaluate longer traces (up to 1 week), without further increasing the speed-up factor. Such experiment would take around 18 hours to emulate using a 9 times speed-up factor.
- During Phase I, the bandwidth bottleneck was located between the proxy and clients. However, a major advantage of proxy caches is the fact that they reduce bandwidth consumption on the link between server and proxy. As such, we aim to explore the effects of a bandwidth bottleneck between proxy and server and the cache size on the delivered quality of both SVC and AVC-based HAS.
- During Phase I, the effects of bandwidth and delay on the different HAS protocols was thoroughly investigated. Our goal is to further explore the effects of various network conditions on these protocols. More specifically, packet loss and jitter will be studied this time around.
- The caching proxy used in Phase I only supported the Smooth Streaming protocol for the custom caching strategies (mLRU and CC). This will be remedied in Phase II, allowing us to evaluate the effect of these novel caching strategies on SVC-based delivery protocols as well.
- The Phase I evaluated topology consisted of a single proxy cache. In Phase II, more elaborate network topologies, consisting of a hierarchy of proxy caches, will be studied. This will allow us to investigate the efficiency of hierarchical caches (as compared to a single cache) and cache collaboration.

2.2 Field experiments

2.2.1 Experimental set-up

The field experiments use a subset of the components used in the emulation experiments (notably the AVC-based HTTP adaptive streaming client emulator, the SVC-based HTTP adaptive streaming client emulator, and the simplified network proxy for small caches).

The field experiments will use PlanetLab nodes, connected over the Internet. Figure 2 shows the functional architecture. More details can be found in section 4 .

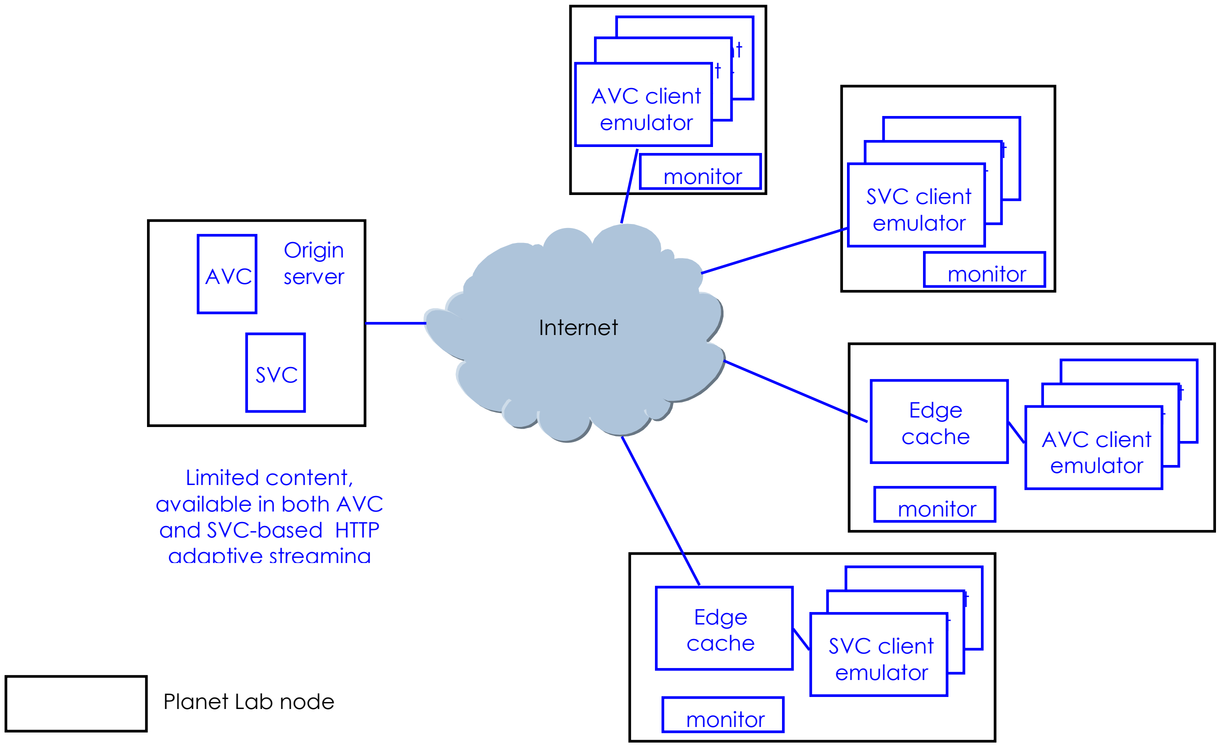


Figure 2: Functional blocks for field experiments set-up

2.2.2 Objectives and rationale

The objectives of the field experiments are twofold:

- Measure and (statistically) compare obtained qualities by AVC-based versus SVC-based HTTP adaptive streaming clients, with and without intelligent edge cache systems, over PlanetLab.
    - Simple metrics that can be used include number of segments downloaded of each quality, number of quality variations, number of buffer under-runs, …
- Compare traditional LRU with OCEAN cache algorithms (mLRU) in terms of cache hit ratio and used bandwidth on the cache uplink for small edge caches.

The rationale for these objectives is to validate the OCEAN simulation results which predict that:

- The SVC-based system will provide better quality than the AVC-based system
- Edge cache systems will improve the obtained video quality
- SVC will provide a better caching efficiency than AVC
- The OCEAN-specific caching algorithm will perform better than LRU

2.2.3 Demonstrator

For demonstration purposes, a real video client using SVC-based HTTP adaptive streaming client will be used over the same test bed.

3. EMULATION PLATFORM

The evaluations are performed on the IBBT Virtual Wall from its iLab.t Technology Center (http://ilabt.ibbt.be). iLab.t interconnects its test network (>200 nodes) and measurement equipment to this Virtual Wall facility, a generic emulation environment for advanced network, distributed software and service evaluation, supporting scalability research. A more thorough description of the Virtual Wall and its functionality can be found in Section 3 of D6.1 [11].

The Phase I emulations have lead to the identification of several limitations and shortcomings of the evaluated functional software components. As such, extended and improved versions of the Phase I software components will be used for the Phase II emulations.

4. FIELD TRIAL PLATFORM

4.1 Introduction

PlanetLab is a global research network that supports the development of new network services. Since the beginning of 2003, more than 1,000 researchers at top academic institutions and industrial research labs have used PlanetLab to develop new technologies for distributed storage, network mapping, peer-to-peer systems, distributed hash tables, and query processing.

4.1.1 A collection of machines distributed over the globe

Most of the machines are hosted by research institutions, although some are located in co-location and routing centers (e.g., on Internet2's Abilene backbone). All of the machines are connected to the Internet. The goal is for PlanetLab to grow to 1,000 widely distributed nodes that peer with the majority of the Internet's regional and long-haul backbones.

4.1.2 A software package

All PlanetLab machines run a common software package that includes a Linux-based operating system; mechanisms for bootstrapping nodes and distributing software updates; a collection of management tools that monitor node health, audit system activity, and control system parameters; and a facility for managing user accounts and distributing keys. This software is distributed as a package, called MyPLC, which others can use to build and deploy their own "private PlanetLabs."

The key objective of the software is to support distributed virtualization—the ability to allocate a *slice* of PlanetLab's network-wide hardware resources to an application. This allows an application to run across all (or some) of the machines distributed over the globe, where at any given time, multiple applications may be running in different slices of PlanetLab.

4.1.3 An overlay network testbed

One of PlanetLab's main purposes is to serve as a testbed for overlay networks. Research groups are able to request a PlanetLab slice in which they can experiment with a variety of planetary-scale services, including file sharing and network-embedded storage, content distribution networks, routing and multicast overlays, QoS overlays, scalable object location, scalable event propagation, anomaly detection mechanisms, and network measurement tools. There are currently over 600 active research projects running on PlanetLab.

The advantage to researchers in using PlanetLab is that they are able to experiment with new services under real-world conditions, and at large scale. The example services outlined above all benefit from being widely distributed over the Internet: from having multiple vantage points from which applications can observe and react to the network's behaviour, from being in close proximity to many data sources and data sinks, and from being distributed across multiple administrative boundaries.

4.1.4 A deployment platform

In addition to supporting short-term experiments, PlanetLab is also designed to support long-running services that support a client base. That is, rather than view PlanetLab strictly as a testbed, we take the long-term view in which the overlay is both a research testbed and a deployment platform, thereby supporting the seamless migration of an application from early prototype, through multiple design iterations, to a popular service that continues to evolve. Using an overlay as both a research testbed and a deployment platform is synergistic. As a testbed, the overlay's value is to give researchers access to:

1. A large set of geographically distributed machines.
2. A realistic network substrate that experiences congestion, failures, and diverse link behaviours.
3. The potential for a realistic client workload.

Its value as a deployment platform is to provide researchers with a direct technology transfer path for popular new services, and users with access to those new services. We believe that supporting both roles is critical to the success of the system.

Services currently running continuously on PlanetLab include the CoDeeN and Coral CDNs; the ScriptRoute network measurement service; the Chord and OpenDHT a scalable object location services; and the PIER, Trumpet, and CoMon network monitoring services.

4.1.5 A microcosm of the next Internet

Not only are researchers evaluating and deploying end-user services on top of PlanetLab, but we also expect them to develop foundational sub-services that can be folded back in to PlanetLab, thereby enhancing the facility for others. Our long-term goal is to identify the common building block services upon which other services and applications can be constructed, or said another way, our goal is to understand how the Internet can be architected to better support overlays.

This perspective is motivated by the general question of how the networking research community can best impact the global Internet. Unfortunately, the very commercial success that has fuelled our increased dependency on the Internet has also reduced our ability to evolve its underlying architecture to meet new demands and correct emerging vulnerabilities. This is because, as a recent National Research Council report notes,

...successful and widely adopted technologies are subject to ossification, which makes it hard to introduce new capabilities or, if the current technology has run its course, to replace it with something better. Existing industry players are not generally motivated to develop or deploy disruptive technologies...

Overlay networks provide an opportunity to introduce disruptive technologies. Overlay nodes can be programmed to support the new capability or feature, and then depend on conventional nodes to provide the underlying connectivity. Over time, if the idea deployed in the overlay proves useful, there may be economic motivation to migrate the functionality into the base system, that is, add it to the feature set of commercial routers. On the other hand, the functionality may be complex enough that an overlay layer may be exactly where it belongs. Our overarching goal is to support the introduction of disruptive technologies into the Internet through the use of overlay networks. PlanetLab is an essential element of this vision.

4.2 Hardware configuration

Planetlab environment installed in TP network is built from three types of hardware platform HP DL160, HP DL320 and IBM x3550. There is 50 nodes connected in 8 different geographical sites. PlanetLab operating system is based on Linux CentOS distribution. Kernel has implemented modules for support vserver architecture which is kind advanced chroot. Each virtual server can have allocated appropriate amount of memory or disk space. In case of planned tests of proxy cache software it will be size of memory for cache or storage for data. Distribution data is with using eth0 interface which called primary interface in planetlab terminology.

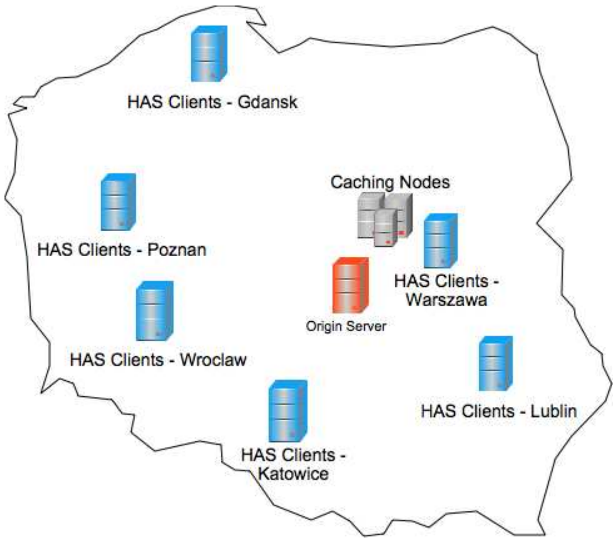
Server model	HP 320G5
CPU	Xeon Quadcore 2Ghz
RAM	4GB
Storage	2x250 GB
OS	CentOS 5.6
Interfaces	2x1Gbps

Server model	HP 160G6
CPU	Xeon Quadcore 2,2Ghz
RAM	16GB
Storage	1TB
OS	CentOS 5.6
Interfaces	2x1Gbps

Server model	IBM x3550M3
CPU	Xeon Quadcore 2,4Ghz
RAM	32GB
Storage	1TB
OS	CentOS 5.6
Interfaces	4x1Gbps

4.3 Network overview

PlanetLab nodes in Polish Telecom are installed in 7 different sites. All nodes are connected to core routers via switches.



4.3.1 Technical Overview

Each of the clusters belonging to the PlabetLab is connected to the backbone network routers (Juniper MX and M series) in local PoP through uplinks from 1Gbit to 4gbits. Diagram below shows typical PoP in TP infrastructure (Figure 4.1).

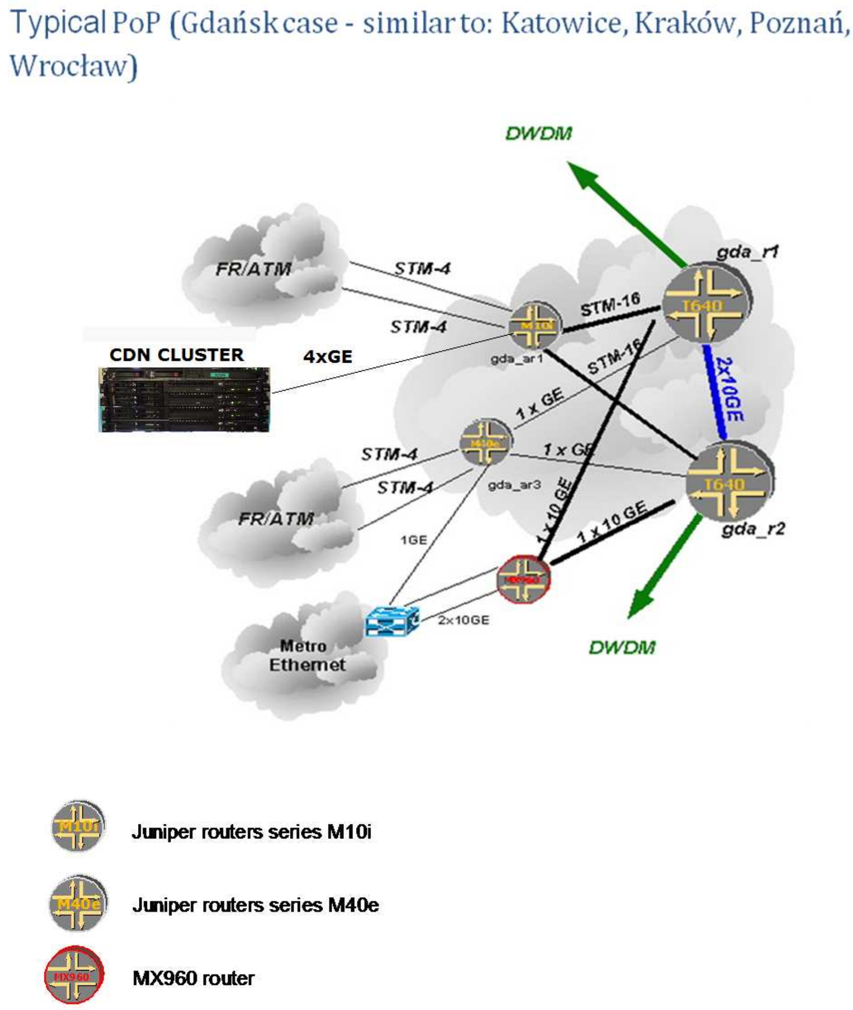


Figure 4.1: Example of a Typical PoP in TP infrastructure

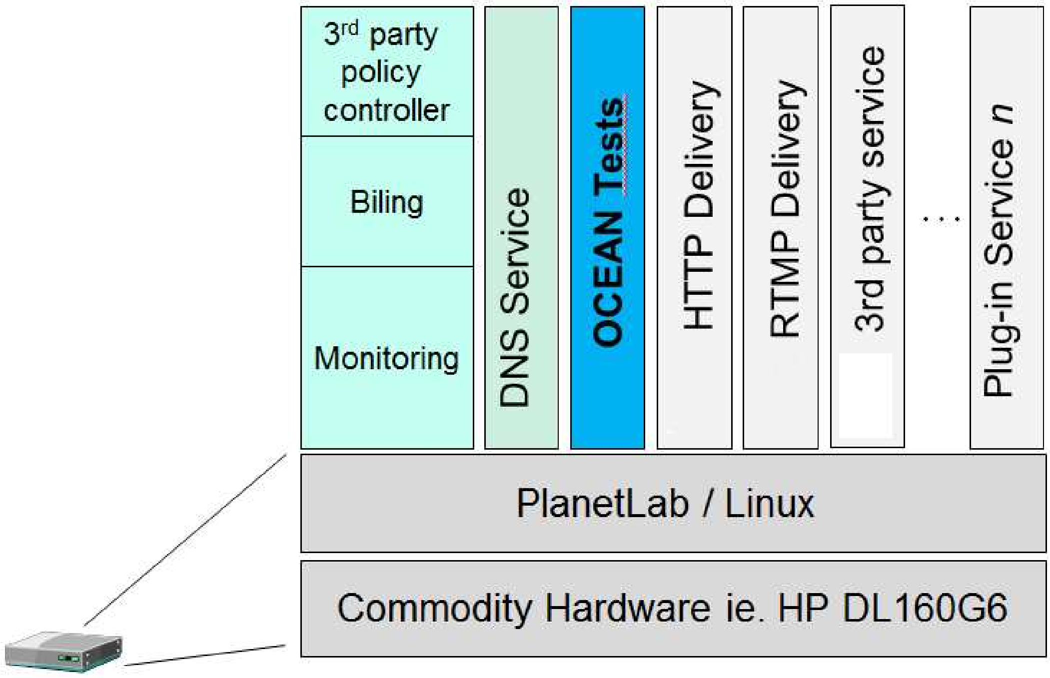
Each cluster of PlanetLab consists of 8 servers connected via a switch to the network backbone.

Optical connections are made using a single mode patchcords. Other connections are made using CAT 5e UTP cable. The data are designed interfaces eth0 and eth1 each node. Interfaces for data transfer (eth0 and eth1) of individual nodes are connected to the switch CDN. Interfaces for managing (ILO) are connected to switch.

Access routers (war_r3 - Juniper M, MX series) to provide the service "content servers, etc." are configured with the principles of service "the ACCESS website". Because the service is used more as a one Gigabit Ethernet interface, these interfaces should be aggregated. This will help avoid problems when the expansion of services will need to add another interface GE.

4.4 Node architecture

Scheme below presents typical node architecture. Particular nodes can differ from each other, depending on what components are currently installed on them.



Each software package will be installed at separate vservers. Following software requirements:

- RPM packages or/and the source code
- Linux CentOS 5.6 compatibility
- x86_64 platform compatibility

4.5 PlanetLAB limitations

There are two types of limits on the bandwidth usage of each slice:

- 100 Mbps cap is on the instantaneous bandwidth usage of the slice.
- The 488 kbps cap is on the average daily bandwidth usage of the slice.

This amounts to a daily Tx byte limit of 10 gigabytes* over the 24hour recording period. At 80% (8GB) of the Tx byte limit, a slice is capped at the byte limit minus the current byte total over the time left in the recording period. Which means:

500000 bits per second / 8 bytes per bit * 24 hours per day * 60 minutes per hour * 60 seconds per minute = 5400000000 bytes = 5.4 gigabytes in the context of data transmission.

REFERENCES

[1] “Final requirements for Open Content Aware Networks”, D2.2, January 2011.
[2] “Use case definition and market scenarios for future OCEAN services”, D2.3, July 2011.
[3] “Business models for future OCEAN architectures”, D2.4, October 2011.
[4] “OCEAN functional architecture and open interface specification”, D3.1, draft December 2011.
[5] “OCEAN architecture for FTTH and DSL access networks”, D3.2, draft December 2011.
[6] “Evaluation of optimized caching for a highly distributed CDN”, D4.3, July 2011.
[7] “Optimized on-line learning algorithms for a highly distributed CDN, D4.7, draft December 2011.
[8] “Evaluation of congestion control via the scalable video codec”, D5.2, July 2011.
[9] “Media adaptation and transmission control in an highly distributed CDN”, D5.4, draft December 2011.
[10] “Protocol implementation for highly distributed content delivery”, D5.7, draft December 2011.
[11] “Definition of evaluation scenarios – Phase I”, D6.1, February 2011
[12] http://www.squid-cache.org

ACRONYMS

AVC	Advanced Video Coding
CC	Chunk-level Caching
CDN	Content Delivery Network
CMTPS	Cache Management based on Temporal Pattern Solicitation
CPU	Central Processing Unit
DSL	Digital Subscriber Line
FTTH	Fiber To The Home
HD	High Definition
HTTP	Hyper Text Transfer Protocol
ICS	Intermediate Cache Server
LCS	Leaf Cache Server
LFU	Least Frequently Used
LRU	Least Recently Used
MPEG	Moving Pictures Expert Group
NNI	Network to Network Interface
OCEAN	Open ContEnt Aware Networks
OCIG	OCEAN Content Interworking Gateway
OS	Operating System
PCN	Pre Congestion Notification
PMD	Priority based Media Delivery
QOE	Quality of Experience
QOS	Quality Of Service
RAC	Resource Admission Control
SC	Scoring-based Caching
SVC	Scalable Video Coding
TCP	Transmission Control Protocol
TEAR	TCP Emulation At Receiver
TFRC	TCP Friendly Rate Control
TNI	Transport Network Interface
UNI	User to Network Interface
VOD	Video On Demand

WP	Work Package