***Enriched Network-aware Video Services over Internet Overlay Networks***
www.envision-project.org

# *Deliverable D6.1*

## Initial Testbed Description and Preliminary Evaluation Results of Content-aware Cross-layer Optimisations for Advanced Multimedia Applications

Public report, version 1.1, 26 February 2012

***Authors***

| | |
|---|---|
| *UCL* | Raul Landa, Richard G. Clegg, Christopher Pluntke, Lawrence Latif, Eleni Mykoniati, Miguel Rio, David Griffin |
| *ALUD* | Nico Schwan, Klaus Satzke, Ivica Rimac |
| *LaBRI* | Toufik Ahmed, Samir Medjiah, Abbas Bradai, Ubaid Abbasi, Damien Magoni. |
| *FT* | Bertrand Mathieu, Irène Grosclaude, Selim Ellouze, Anne Tréhu, Patrick Devos, Pierre Paris, Valery Bastide |
| *TID* | Oriol Ribera Prats, Nikos Laoutaris |
| *LIVEU* | Noam Amram, Liron Yadin |

***Abstract*** This deliverable presents work achieved in work package 6 (WP6) during M18 to M24 of the ENVISION project. This document elaborates on the specification of the different testbeds installed at the premises of the industrial partners, in particular the FT testbed at both Lannion and Paris locations, the ALUD testbed in Stuttgart, the TID testbed by Telefonica deploying CDN technology, and a small LIVEU testbed in Kfar Saba. The deliverable also elaborates on the integrated prototypes evaluation and the specification of the tests objectives, scenarios and attributes. Moreover, the deliverable elaborates on simulations and theoretical analyses and provides some initial test results.

University College London, UK (UCL)
Alcatel-Lucent Deutschland AG, Germany (ALUD)
Université Bordeaux 1, France (LaBRI)
France Telecom Orange Labs, France (FT)
Telefónica Investigacion y Desarollo, Spain (TID)
LiveU Ltd., Israel (LIVEU)

Project funded by the European Union under the
Information and Communication Technologies FP7 Cooperation Programme
Grant Agreement number 248565

# EXECUTIVE SUMMARY

This document is the first WP6 deliverable of the ENVISION project. The project advocates the cross-layer optimisation between network and application overlay functions through the *Collaboration Interface between Network and Applications* (CINA), documented in [D3.2]. The aim of WP6, the integration and experimentation work package is to evaluate the algorithms, techniques and tools developed in work packages 3-5 and to prove that they are valid, achieve the project objectives and the system requirements as specified in WP2. Thus the work we have done and documented in here is a result of half year duration work started from M18 where WP6 started its activity to M24 deliverable time.

This report documents the testbeds defined as part of the activities in T6.1, the functional validation and performance analysis tests as part of T6.2 and T6.3. The project has prepared specifications for four testbeds as defined in chapter 2. The FT testbed is considered as the main one, allowing for evaluation and demonstration of the integrated prototypes and most of the developed technologies. A description of the FT testbed can be found in chapter 2.1. The other testbeds by ALUD, TID and LiveU testbeds are more standalone types of testbeds and less considered for integrated prototypes evaluation. Rather, they are used to evaluate the technology developed at each of the associated industrial partners.

The FT testbeds consist of a number of wired network segments located in Lannion and a wireless 3G network segment located in Paris, the testbed network being a replica of the operational networks consisting same type of elements such as DSLAM, BRAS and routers.

This report also specifies a set of evaluation scenarios in chapter 3; these are further divided into prototype evaluation, simulations and theoretical evaluations. A large set of evaluations are detailed, each one consisting of its objectives, environment definition, its performance metrics, controlled and uncontrolled parameters and the evaluation scenario description.

Prototype-based tests are to support the evaluation and demonstration of the ENVISION prototypes. We have specified the planned evaluations for the following systems: *Live Content Distribution and Content Adaptation*, *Interactive Content Distribution, Multicast Network Service, Live Content Push via High capacity Node and Multi-link enabled peers*, *Caching, P4P CDN,* and *ENVISION-COMET joint evaluation*.

The simulation and theoretical analysis intend to complement the prototypes allowing simulation of larger scale overlay and network topologies that can be achieved in our testbeds. The following simulations and/or theoretical analyses have been specified: *ISP Preferences, Consolidated Overlay View – Overlay-ISP Cooperation Trade-off, Distributed Data Management System, Live Content Distribution, Distribution Tree Optimisation, Content Adaptation, and the Multi-Link Forward Error Correction and Rate Control.*[1]

Finally, in chapter 4 we are providing our initial evaluations results, consisting of preliminarily results for evaluation of the monitoring module in the multi-link enabled peer and results for the overlay consolidations and the content adaptation systems.

[1] Detailed descriptions of the simulation models and initial results have been suppressed in the public version of this deliverable as this is ongoing work which is planned for publication in scientific conferences and journals. The results will be made public in future deliverables.

# TABLE OF CONTENTS

# LIST OF FIGURES

# 1. INTRODUCTION

The ENVISION project addresses the cooperation between the overlay applications and the underlying networks through the CINA service. High-definition, highly interactive networked media applications pose major challenges to network operators. Multi-sourced content means higher quantities of data throughout the network, putting additional pressure at the network edge for unprecedented upload capacity in access networks. If the entire burden of supporting high volumes of HD/3D multi-media streams is pushed to the ISPs with highly concurrent unicast flows this would require operators to upgrade the capacity of their infrastructure by several orders of magnitude. Rather than simply throwing bandwidth at the problem, the ENVISION approach is to develop intelligent cross-layer techniques that, on the one hand, will mobilise network and user resources to provide network capacity where it is needed, and, on the other hand, will ensure that the applications adapt themselves and the content they are conveying to available network resources, considering core network capacity as well as the heterogeneity of access network and end-device capabilities.

Meeting these challenges requires a previously unseen amount of cooperation between application providers, users and the communications networks that will transport the application data. Applications need to be able to accommodate unpredictably large numbers of participants in a cost-effective way, while still maintaining high responsiveness to deliver a high Quality of Experience to the participants. Content, which itself is changing dynamically in scale and context according to user participation and behaviour, needs to be adapted to network capacity and capabilities, and networks need to be aware of the nature and needs of the content it is transporting.

The aim of WP6, the integration and experimentation work package, is to evaluate the algorithms, techniques and tools developed in previous work packages and to prove that they are valid and that they achieve the project objectives and the system requirements as specified in WP2.

WP6 will establish the experimentation environment that will undertake the functional validation and performance analysis tests and will provide the demonstrations required in order to evaluate and disseminate the project results.

This deliverable specifies the following ENVISION aspects:

- In chapter 2 we provide the specification of the ENVISION testbeds.
- In chapter 3 we specify the ENVISION evaluation plans for prototypes as well as simulation and theoretical analysis.
- In chapter 4 we provide the initial results.

The interim and final specifications and evaluation results produced by the project will be documented and reported in D6.2 in M36 respectively.

# 2. TESTBEDS SPECIFICATIONS

This section introduces the identified and built testbed resources that are available for experimental work by the project. These testbeds would be used to evaluate ENVISION developed mechanisms, algorithms and protocols, and the initial evaluations and test specifications to be executed are further detailed in section 3.

## 2.1 FT Testbed

In Lannion, Orange Labs has a specific unit which aims to set up and maintain an experimental network, representative of the commercial network, composed of network equipment deployed (or to be deployed) in the real Orange network. This experimental network is then a replica of real operational network, comprising different network equipment such as DSLAM, BRAS, core and edge routers, wireless access points and 3G equipment.

### 2.1.1 Testbed Specification

This network unit is spread over several locations and the department offers access (and configuration) to this representative network for experiments and evaluations of various Orange projects. For the ENVISION project, the objective is to use this rich test environment for the benefit of the project. Typically, it is aimed to deploy the multicaster, the network caches and the high capacity node in this experimental network and make some functional tests with real end-users, connected to the network via a representative access network. The ENVISION project shares time on this network with other Orange test projects.

For ENVISION, we specified the network equipment we need, their configuration, the network topology so that to set up the testbed according to our expectations. Figure 1 presents the current Orange testbed constructed for ENVISION.

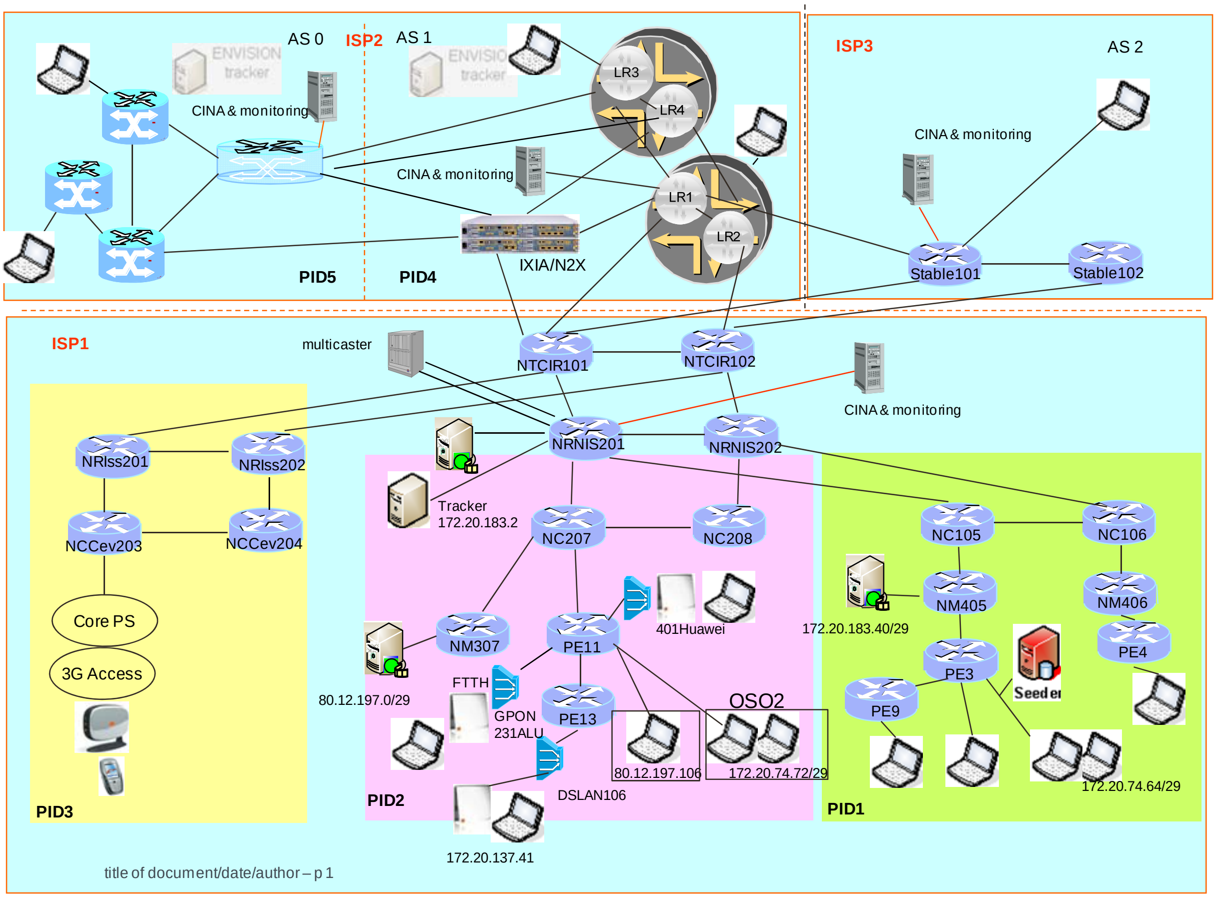

**Figure 1: Orange testbed network for ENVISION**

In this figure, we can see that we are able to emulate 3 different ISPs, in order to emulate an overlay application running on top of several ISPs). We have 2 ISPs having 2 and 3 different regions (named PIDs, to be compliant to the ALTO terminology). For ISP1, we have 2 PIDs being located in Lannion (the 2 on the right-hand side) and 1 located in Paris (connectivity to wireless network on the left-hand side). We will define priority (lower cost) so that clients in Lannion preferentially connect to other clients also in Lannion. For cross-ISP traffic, we define a lower cost between ISP1 and ISP2 than between ISP3 and ISP2, so that clients in ISP2 will preferentially connect to clients in ISP1 rather than clients in ISP3. Some tests also involve increased delay between ISP1 and clients in ISP2 than between clients in ISP1 and clients in ISP3, so that if the delay metric is selected by the overlay application for building the network map, the result will change (clients in ISP1 will connect to clients in ISP3 rather than clients in ISP2). This configuration shows the interest of having specific metric for the network map (added-value from ENVISION compared to ALTO) and highlights how the peer selection process is impacted by the network preferences of the ISP as well as by the critical metric for the application point of view.

## 2.1.2 Testbed value for evaluation and demonstration

We will deploy the multicaster in ISP1, as well as some network caches to perform some tests to highlight the benefit of the dynamic activation of network services requested by the overlay application. There will be one CINA server and one monitoring component per ISP. This testbed will be used in the evaluation scenarios described in section 3.

## 2.2 ALUD Testbed

In Stuttgart, Bell Labs Germany has set up a local experimental network test environment composed of standard network equipment and some special networking gear provided by the Alcatel-Lucent product divisions. The setup is aimed to demonstrate the functionality of the Interactive Content Distribution System and the CINA server on one hand, and to test and evaluate the instantiation of the ENVISION High Capacity Node (HCN) on the other.

### 2.2.1 Testbed Specification

For ENVISION, ALUD BL has specified the equipment needed for our testing purposes and has developed the configuration and the network topology so that the testbed setup will perform according to expectations. Figure 2 shows the local testbed configuration as it has been realised for the ENVISION project on the ALUD BL Stuttgart laboratory premises.

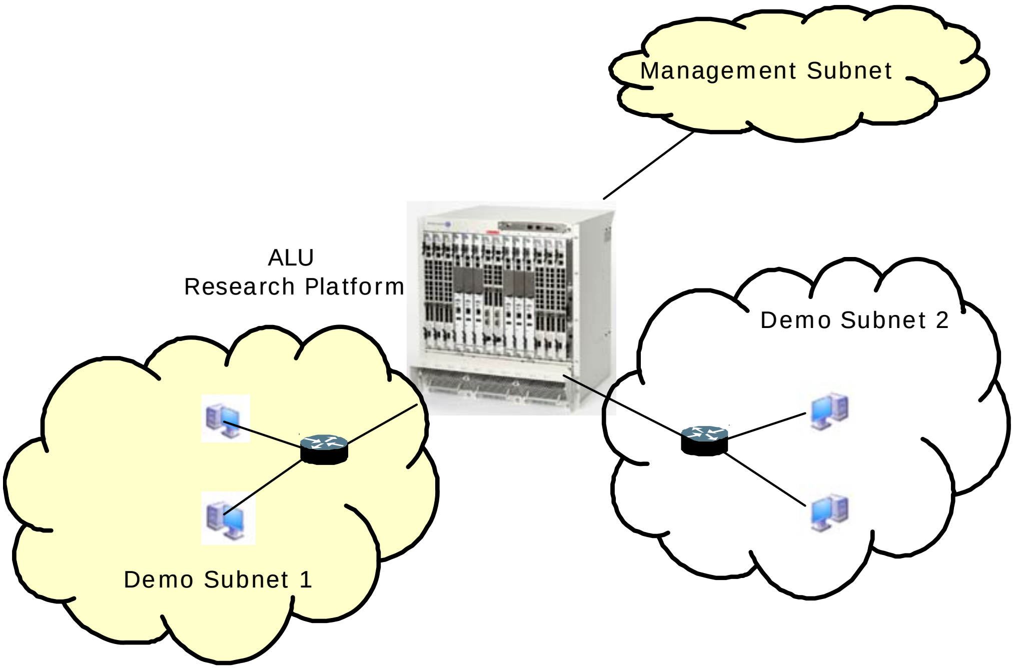

**Figure 2: ALU Testbed configuration**

The ALUD local testbed configuration consists of two Demo Subnets for experiments, the ALU research platform, and a Management Subnet. Both Demo Subnets contain a number of PCs acting as peers and hosting the CINA servers.

Also, a specific local test environment around the ALU Research Processing platform (see Figure 2) has been built mainly to enable proof-of-concept implementations and to enable the demonstration of the features and the instantiation of the ENVISION HCN. The Alcatel-Lucent Research Platform provides basic routing and switching functionality with extensible and advanced packet processing. These core switching functions can be complemented with extension cards to perform control and fast-path packet processing. Flows can be dynamically redirected to these cards through flow classification. The extension cards additionally support Linux with KVM-based virtual machines. The platform thus is a natural host environment for the HCN network service.

### 2.2.2 Testbed value for evaluation and demonstration

The testbed will be used for the evaluation scenarios as described in section 3. In particular we will deploy a set of overlay peers that distribute media streams, CINA servers that provision information on their view of the network topology, and deploy the HCN on the ALU Research Platform. Functionality tests will highlight the benefits of information provisioned by the CINA servers and the HCN network services once requested by the overlay application.

## 2.3 TID Testbed

### 2.3.1 Testbed Specification

It is first necessary to know about the main components of the CDN. This previous knowledge will help us to understand the test bed and the way ENVISION will improve the default CDN behaviour.

- **Topology Server**: It is the network data source. It gives the network cost of connecting a couple of internet endpoints. It is one of the tracker information sources for node selection decisions.
- **Tracker**: The main controller of the CDN. It continuously receives snapshots of the CDN nodes resource usage stats. With this data plus the network topology map, tracker can figure out which is the best node to serve a specific end user, both in terms of network and in terms of application resources.
- **CDN Nodes**: These are the content delivery nodes. The large amount of servers where the end users connect to for content downloading.

This section details a simplified workflow in order to understand the ENVISION improvement. Only a small part of the CDN workflow will be described (see figure 1 at the bottom of the section) to ease the understanding.

- A computer (e.g. Player1) asks the CDN tracker about a specific content by a CDN general URL (e.g. http://www.globalcdn.com/liveEvent1.avi).
- The tracker geo-locates the computer and chooses the best CDN Node for this customer (let’s say its CDN Node 1) thanks to the Topology server and resolves the URL name to the CDN Node 1 IP address.
- The computer asks CDN Node 1 about that content.
- CDN Node 1 establishes a connection with the Origin server (client side) and starts serving the content to the computer.

There are several protocols that can be used for streaming multimedia over computer networks. Each player supports a specific protocol and asks the CDN Node to serve the content in that protocol.

We have chosen two protocols to illustrate this testbed. A minimal (and very simplified) description is provided to understand the scenario.

- **Smooth streaming**. A player using this protocol will ask the CDN node about the content and receives a MANIFEST file with the different chunks (about 2 seconds each) to be downloaded.

The player establishes a unique session with the video server to get every chunk. That way the player only interacts with the tracker once and it also only establishes a connection with the selected CDN Node that lasts for all the video duration.

- **Dynamic streaming**. A player using this protocol will ask the CDN Node about the content and receives a MANIFEST file with several playlists. Then the player asks the server for the playlist that includes a list of chunks (about 2 seconds each). Finally the player tries to get every chunk from the server. The main difference is the player doesn't establish a unique session. Every time it needs a chunk it asks for it. Chunks are identified by its URL (e.g. http://www.globalcdn.com/liveEvent1.avi?chunk=chunk1). So, the player must ask the tracker for every chunk, which, in turn, asks the topology server. That way the CDN Nodes serving the content may change during viewing if topology information changes.

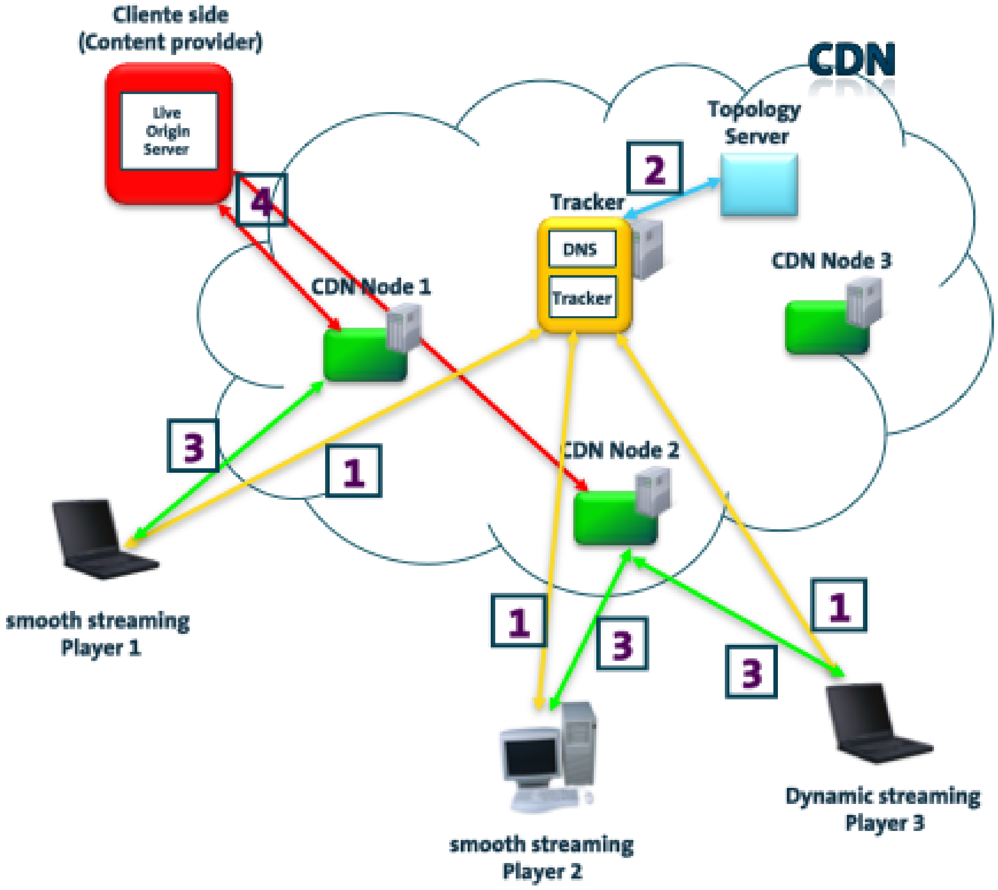

**Figure 3: Simplified CDN workflow**

### 2.3.2 Testbed value for evaluation and demonstration

We will deploy some elements of the CDN in Internet so we can simulate somehow a real behaviour. We will also build a network topology for the benefit of the test specified in 3.1.5. Both elements will enable us to evaluate the algorithm in a real environment as well as prove our theory about streaming protocol behaviour and CDN improvements.

## 2.4 LiveU Testbed

In Kfar Saba, Israel, LiveU has developed a local experimental Laboratory environment consisting of a Linux PC which is used to emulate multi-link environment to test technology developed by LiveU, i.e. data resource management, content generation, content adaptation and distribution from the source peer over multiple networks.

### 2.4.1 Testbed Specification

The laboratory is running a dedicated proprietary protocol over a single Ethernet link to emulate a multilink environment, the following capabilities are available:

- Adding modems: change and remove the number of modems available on the fly, to emulate the loss of a modem, or gain of new network.
- Configurations:
  - Configure each modem rate limit (bandwidth).
  - Configure each modem up & down delays.
  - Configure each modem loss rate.
- Running Scripts: in order to reproduce the same channels and behaviours and validate the algorithm under specific conditions, we have generated hundreds of scripts that represent real world environment conditions learned from our collected data in real networks.
- Analysis:
  - delay history
  - bandwidth history - for each modem and total bandwidth
  - loss history
  - traffic analysis - data vs. forward error correction FEC
- All configuration can be changed during streaming – simulating a real modem environment.
- There is an option to load a recorded modem file and stream using that modem behaviour.

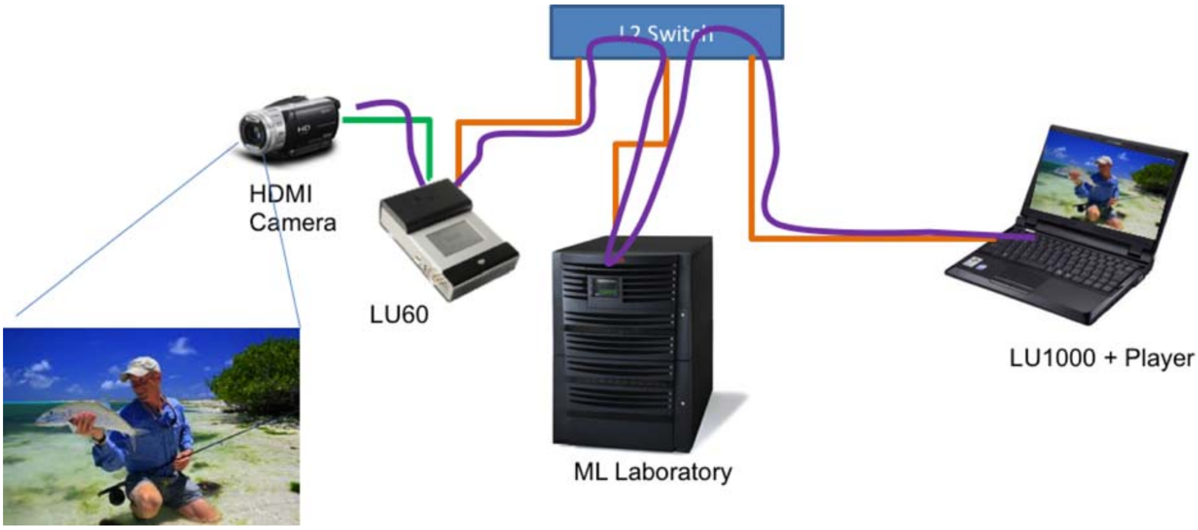

**Figure 4: ML Laboratory testbed**

### 2.4.2 Testbed value for evaluation and demonstration

These testbeds will be used to evaluate the multilink enabled peer performing its adaptation and optimisation as a source peer for ENVISION. In more detail, it will be used to test the resource data management module that is responsible for the monitoring and predictions of the networks behaviour, the content generation in various use case conditions, the content scheduling and protection.

# 3. EVALUATIONS

## 3.1 Prototype Evaluation

### 3.1.1 Live Content Distribution and Content Adaptation

The specifications of this system are provided in detail in section 4 of [D4.2] and section 4 of [D5.2].

#### *3.1.1.1 Objectives and Evaluation Requirements*

- Functional validation of the content distribution and adaptation functions interacting with the CINA server to retrieve different types of costs.
- Functional validation of the overlay topology construction mechanism using CINA costs that represent delay and loss probability.
- Functional validation of the content adaptation function for the selection of SVC layers according to the achieved throughput over all overlay links.
- Functional validation of the content adaptation function interoperating with topology construction to achieve increased throughput and therefore higher quality.
- Evaluate in a way that can be measured and demonstrated to observers the improvement in application performance and network utilisation achieved with the announcement and use of CINA costs.

#### *3.1.1.2 Evaluation Environment*

These tests will be performed in FT testbed (see section 2.1). The software prototype integrating live content distribution and content adaptation functions will be deployed in a number of terminals in ISP1, ISP2 and ISP3, and one application tracker maintaining the list of peers participating in different sessions of the application distributing different content objects will be deployed in one of these domains. This tracker will be implementing the CINA client and will be interacting with the CINA servers provided at each domain. The content source nodes will be deployed in the same location (PID1 of ISP1) to allow direct comparison between different application sessions.

The evaluation environment will be enhanced with visualisation tools to show the topology of the overlay application for each session and to report on the load conditions at specific network links. Indicators about the stream liveness and the stream quality for a particular consumer (see section 3.1.1.3) will be provided as part of the functionality of the overlay application prototype receiving and playing the video at the terminal devices.

Finally, in order to create the conditions required for triggering the content distribution and content adaptation logic in a controlled way, rate limiting rules may need to be installed for specific flows at particular network links to increase loss over these links. Moreover, additional delay may be introduced between two endpoints.

### 3.1.1.3 Performance Metrics

- *stream liveness*: The time difference between the playout point at a consumer and the time a chunk is produced at the source. For each user the average value during the lifetime of its session will be used. For the entire overlay application, a percentile (e.g. 90th percentile) of the values per user will be used.
- *stream quality*: The spatial, temporal and SNR resolution of the stream (i.e. the SVC layer) a user receives. The values per user and for the overlay will be calculated as above.
- *network load*: The average and 95th percentile of the utilisation at a particular interface. For the network overall, the maximum and average values per interface will be used for the inter-domain links, and/or other links that are part of overlay paths set with lower CINA costs in order to encourage traffic to shift away from these links.

### 3.1.1.4 Controlled and Uncontrolled Variables

Controlled variables:

- *use of CINA delay costs*: The overlay topology construction algorithm may establish overlay connections taking into account the CINA delay costs, or randomly.
- *use of CINA loss probability costs*: The overlay content adaptation functions may respond to decreased throughput over an overlay link taking into account the CINA loss probability costs, or it may select alternative links randomly.

Uncontrolled variables:

- *overlay link network delay*: The delay between any two overlay nodes in the testbed can vary in the range of [20,100] milliseconds. Additional delay may be introduced for specific connections in order to highlight the benefits achieved using the CINA delay costs.
- *overlay link throughput*: The throughput achieved between any two overlay nodes in the testbed, which can be controlled by rate limiting the data flow between them.

#### *3.1.1.5 Evaluation Scenarios*

The live content distribution and content adaptation evaluation scenarios are captured in Table 1.

| Description | Metrics | Controlled Variables | Uncontrolled Variables |
|---|---|---|---|
| Stream Liveness Increase:<br>Evaluate the impact on the stream liveness and the network load at inter-domain links from using the CINA delay costs | stream liveness, network load | use of CINA delay costs: one application session with use of CINA delay costs enabled and one without<br>use of CINA loss probability costs: disabled for both application sessions | overlay link network delay: selection of consumer nodes with highest network delay<br>overlay link throughput: maximum available |
| Stream Quality Increase:<br>Evaluate the impact on the stream quality from using the CINA loss probability costs | stream quality | use of CINA delay costs: as above<br>use of CINA loss probability costs: one application session with use of CINA loss probability costs and one without | overlay link network delay: as above<br>overlay link throughput: rate limited throughput at particular links |

**Table 1: Live Content Distribution and Content Adaptation Evaluation Scenarios**

### 3.1.2 Multicast Network Service

The specifications of the multicast network service are provided in detail in sections 4.2 in [D3.2].

#### *3.1.2.1 Objectives and Evaluation Requirements*

- Demonstrate the feasibility of an open multicast network service in a replica of real operational network.
- Evaluate the delay to activate the service for a stream through the CINA interface.
- Evaluate the benefits brought by the multicast network service in terms of stream quality and liveness.
- Evaluate the benefits brought by the multicast network service on intra-domain link load.

#### *3.1.2.2 Evaluation Environment*

These tests will be performed using the FT testbed described in section 2.1.

The multicast network service will be deployed in ISP1.

#### *3.1.2.3 Performance Metrics*

- *stream liveness:* as defined in section 3.1.1.3
- *stream quality*: as defined in section 3.1.1.3

- *induced network load:* The total bandwidth required to distribute a specific live content to N users. Obtained by monitoring the input/output traffic on overlay nodes.

#### *3.1.2.4 Controlled and Uncontrolled Variables*

Controlled variables:

- *multicast source selection policy*
- *multicast data selection policy*
- *number and types of nodes:* Number and types of nodes receiving simultaneously the same stream in an AS (if SVC coding: number of nodes per requested quality level)
- *multicast receivers:* number and types of nodes receiving the traffic in multicast
- *content quality*

Uncontrolled variables:

- *Platform Load*: The load on test platform ("real" network with several distinct tests)

#### *3.1.2.5 Evaluation Scenarios*

| **Description** | **Metrics** | **Controlled Variables** | **Uncontrolled Variables** |
|---|---|---|---|
| Time to activate the service through CINA | | | Platform load |
| Impact on stream delivery performance | stream liveness, quality | multicast source selection policy<br>multicast data selection policy<br>number and types of nodes<br>multicast receivers<br>content quality | Platform load |
| Impact on network load | induced network load | multicast source selection policy<br>multicast data selection policy<br>number and types of nodes<br>multicast receivers<br>content quality | Platform load |

**Table 2: Multicast Network Service Evaluation Scenarios**

### 3.1.3 Content Distribution via IVCD and Multi-link enabled peers

The following section describes the evaluation scenario of the integrated prototype between MLEP and the Interactive Content Distributions (IVCD) system. The scenario includes optimisation on the uplink done by boosting the capacity and proper selection of the best Multi-link Server (MLS). Further the IVCD system functionality is demonstrated, including its use of CINA information and the

High Capacity Node (HCN) CINA network service. The scenario fits well with the recent trends of Micro-Journalism and user generated live content as can be seen in many portals such as www.ustream.com, social networks, YouTube, www.justintv.com and more.

The scenario defines integration where the MLEP is connected to several access networks of generally different network operators/CINA domains, the MLEP shall invocate network queries from the different CINA servers, based on the retrieved information a selection of MLS, and links priorities are performed. The MLEP uses the CINA defined interfaces and protocols and performs uplink optimisation of the stream from anywhere anytime into the core network by the MLS pushing the content to the root peer of the IVCD which is responsible for further distribution of the retrieved stream in a P2P fashion.

The specifications and implementation details of the IVCD peer-to-peer system are provided in detail in section 5 of [D4.2].

The specifications of the CINA protocol and implementation details the of CINA library are provided in sections 2 of [D3.2].

The specifications of the High-Capacity Node service are provided in detail in section 4.4 of [D3.2].

#### 3.1.3.1 Objectives and Evaluation Requirements

- Demonstrate the functionality of the peer-to-peer video streaming system and its ability to integrate CINA network information and services.
- Demonstrate the functionality of the tree topology creation algorithm in the tree management module, incorporating optional information from the CINA server.
- Demonstrate the functionality of the CINA client and server library
- Demonstrate the functionality of the High-Capacity Node network service.
- Evaluate the performance improvements (latency, quality..) of the proposed solution for distribution of live content from anywhere and anytime to over the uplink path.

#### 3.1.3.2 Evaluation Environment

These tests will be performed in FT testbed, ALUD testbed, and LiveU labs, see section 2 for more information on the testbeds. The evaluation will be carried out in steps by ALUD and LiveU and will be integrated into FT testbed for final evaluation and demonstration of the technology. We have identified and analysed three integration options and will refine and select the most proper one for the evaluation in the near future.

#### 3.1.3.3 Performance Metrics

- *Latency*: The time difference between the playout point and the time a chunk is produced at the source. For each user the average value during the lifetime of its session will be used. For the entire overlay application, a percentile (e.g. 90-th percentile) of the value per user will be used.
- *Stream quality*: The spatial, temporal and SNR resolution of the stream uploaded to the network. The values per user and for the overlay will be calculated as above.

- *Robustness*: The ability to mitigate with varying network conditions, packet loss and connectivity loss of links.
- *Uplink throughput*: the aggregated effective data rate.

#### *3.1.3.4 Controlled and Uncontrolled Variables*

Controlled variables:

- *Number of domains*: The maximum number of domains the MLEP is connected to.
- *Number of CINA servers*: whether per domain, or not in all domains.
- *Number of MLS (LU1000)*.
- *Latency between peers.*
- Latency of links.
- Loss rate on links.
- Rate of links.
- *Node out degree*: The maximum number of outgoing overlay connections per node.
- *Number of nodes:* The number of nodes that join the overlay.
- *Network and cost map*: The network and cost map provided by the CINA server are given for the experiment.

Uncontrolled variables:

- *Resolution of the encoded stream (number or pixels).*
- *Generated video data rate.*
- *FEC overhead.*
- *The bitrate per link.*
- *Tree topology*: The topology created by the tree manager is partially based probabilistic decisions, thus the topology may vary for similar experiment setups. This includes the place of the HCN in the topology.

### 3.1.3.5 *Evaluation Scenarios*

The IVCD demonstration scenarios are captured in the following table.

| Description | Metrics | Scenario Steps |
|---|---|---|
| Demonstrate functionality of the peer-to-peer system without information from the CINA server | functionality of topology creation algorithm and streaming capabilities | • Source Peer is initialised to stream from a video source<br>• Consumer Peers join the tree topology and start displaying the content<br>• The overlay topology is visualised |
| Demonstrate the functionality CINA client and server by using network information for the overlay topology | functionality of CINA client and server<br>functionality of network information usage by topology creation algorithm | • Source Peer is initialised<br>• Consumer Peer A1 from network domain A is started<br>• Source Peer contacts the CINA Server and retrieves information about Consumer Peers in network domain A<br>• Consumer Peer A2 from domain A is started, no discovery or CINA messages are needed<br>• Consumer Peer B1 from domain B is started, which triggers another discovery and CINA message exchange<br>• Consumer Peers are started from both domains until amount of Scenario 1 is reached<br>• The topology is visualised and shows how the CINA information has influenced the topology |
| Demonstrate the functionality of the High Capacity Node network service | functionality of the instantiation of the network service<br>functionality of the integration into the overlay topology | • Source Peer contacts CINA Server requesting instantiation of HCN<br>• CINA Server instantiates and configures HCN<br>• Tree topology is reorganised accordingly<br>• Visualisation shows new topology with two branches, a flat one in domain A with HCN and a deeper one in domain B |

**Table 3: Content Distribution via IVCD Demonstration Scenarios**

The integrated test scenarios are captured in the following table:

| **Description** | **Metrics** | **Controlled Variables** | **Uncontrolled Variables** |
|---|---|---|---|
| IVCD demonstration scenario including the invocation of the high-capacity node | Functionality of demonstration scenario | - *node out degree*<br>- *number of nodes*<br>- *network and cost map* | - tree topology |
| MLEP uplink, selection of the MLS and improving the stream quality | stream liveness,<br>Quality | - *Number of domains*: The maximum number of domains the MLEP is connected to.<br>- Number of MLS (LU1000):<br>- *Number of HCNs*: The number of HCN participates in the evaluation scenario, could be one per domain or less, with a minimum of 1 per all.<br>- *Latency between peers*<br>- Latency of links<br>- Loss of links<br>- Rate of links | - *Resolution of the encoded stream (number or pixels)*<br>- *Generated video data rate*<br>- *FEC overhead*<br>- *The bitrates per link* |

**Table 4: IVCD Integrated Test Scenarios**

## 3.1.4 Caching

### *3.1.4.1 Objectives and Evaluation Requirements*

The testbed in FT Orange lab have the main goals of testing caching:

- Improving QoE for end-users.
- Saving BW for ISP preferring internal exchanges to limit interconnection.

To evaluate the cache impact the following are required:

- A tool to measure the gain in delay for end-users.
- The amount of bandwidth saved thanks to ENVISION ISP metrics
- The impact of cache presence on connectivity between ENVISION nodes

#### *3.1.4.2 Evaluation Environment*

We will use the FT testbed described in section 2.1.

Caches are deployed in ISP1, at backhaul and core level. This integration will permit us to check cache efficiency following location and if it is possible to check the hierarchical caching solution with specific communication inter caches.

Caches are ENVISION nodes and known as peers with a possibility to be a permanent and origin peer for new and popular contents.

#### *3.1.4.3 Performance Metrics*

First of all, tests will be specialised on individual tests and not on loading tests.

Cache can be evaluated on the following metrics:

- Capacity to ingest content/chunk (cache ingestion: content is stored in the cache)
    - *CacheContentTypeIngestion*: Following content type: live is not in the scope.
    - *CacheSeeders*: Number and location of ENVISION seeders seen by cache.
    - *CacheDelayIngest*: delay to get chunk available for delivery: to be able to ingest new content as fast as possible.
- *CacheDelayDelivery*: Inserted delay in content/chunk delivery: cache presence may add a specific delay.
- Capacity to deliver content/chunk:
    - *CacheContentTypeDelivery*: Delivering all ingested content types.
    - *CachePerturbationDelivery*: No perturbation propagated to end users.
    - *CacheQoEDelivery*: Gain in time to deliver chunk. Must appreciate difference between cache/no cache.
- *ConnectedNodes*: Number and location of ENVISION seeders seen by ENVISION leechers under cache influence.
- *NodesQoE*: quality of experience for internal FT nodes (freeze, etc.).
- *Bandwidth*: amount of traffic on different links.

#### *3.1.4.4 Controlled and Uncontrolled Variables*

Controlled variables:

- *ContentType*: content under distribution thanks to ENVISION network.
- *CacheLocation*: cache location is under ISP responsibility.
- *CacheNodesNumber*: Number of ENVISION nodes per cache (virtual machines).
- *ManagedNodes*: Number of ENVISION nodes inside our test platform managed by cache team.

Uncontrolled variables:

- *UnmanagedNodes*: Number of nodes which can be connected to in test scenarios.

- *PlatformLoad*: The load on the test platform ("real" network with several distinct tests): delay to get delivery from cache in end-users devices.

### 3.1.4.5 Evaluation Scenarios

The table below shows the different scenarios.

| **Description** | **Metrics** | **Controlled** | **Uncontrolled** |
|---|---|---|---|
| REFERENCE: no cache | *Bandwidth*<br>*ConnectedNodes*<br>*NodesQoE* | *ContentType* = VoD<br>*ManagedNodes*= 2 to all (*) | *UnmanagedNodes* = $x$ Labri nodes |
| Cache integration: content ingestion | *CacheContentTypeIngestion*<br>*CacheSeeders*<br>*CacheDelayIngest*<br>*NodesQoE* | *CacheLocation* = Backhaul/Core (**)<br>*CacheNodesNumber* = 2<br>*ContentType* = VoD<br>*ManagedNodes* = 2 to all | *UnmanagedNodes* = $x$ Labri nodes |
| Cache integration:<br>Content delivery | *CacheContentTypeDelivery*<br>*CacheSeeders*<br>*CacheDelayDelivery*<br>*CacheQoEDelivery*<br>*Bandwidth*<br>*ConnectedNodes*<br>*NodesQoE* | *CacheLocation* = Backhaul/Core<br>*CacheNodesNumber* = 2<br>*ContentType* = VoD<br>*ManagedNodes* = 2 to all | *UnmanagedNodes* = $x$ Labri nodes |

**Table 5: 3.1.4 Caching Evaluation Scenarios**

* *ManagedNodes* number is not yet fixed; ** currently location at core level is not authorised.

## 3.1.5 P4P CDN

### 3.1.5.1 Objectives and Evaluation Requirements

The testbed to be deployed by TID will have the main goals of demonstrating:

- Algorithm behaviour and tuning
- Algorithm integration inside CDN topology server and QoE improvement for End Users

To evaluate the QoE improvement the requirements are:

- A smooth streaming player on a PC with internet access
- A dynamic streaming player on a PC with internet access

### *3.1.5.2 Evaluation Environment*

#### 3.1.5.2.1 CDN scenario without ENVISION

Figure 3 also illustrates this scenario. Assuming there is a problem in the CDN node **2**: its network link has problems and it is only able to serve 10 Mbps because of network congestion. This would mean that the video runs choppy and player 2 and player 3 customers will perceive poor quality video.

Nevertheless, as neither the topology has changed nor the servers are overloaded, every time the topology server is asked it will keep choosing CDN Node 2 as the best server for player 2 and player 3.

#### 3.1.5.2.2 CDN scenario with ENVISION

ENVISION will improve the topology server selection algorithm and enable it to choose a better CDN node dealing with unexpected events. It will add another input to the current algorithm, making it much more reliable.

Figure 5 describes the new scenario.

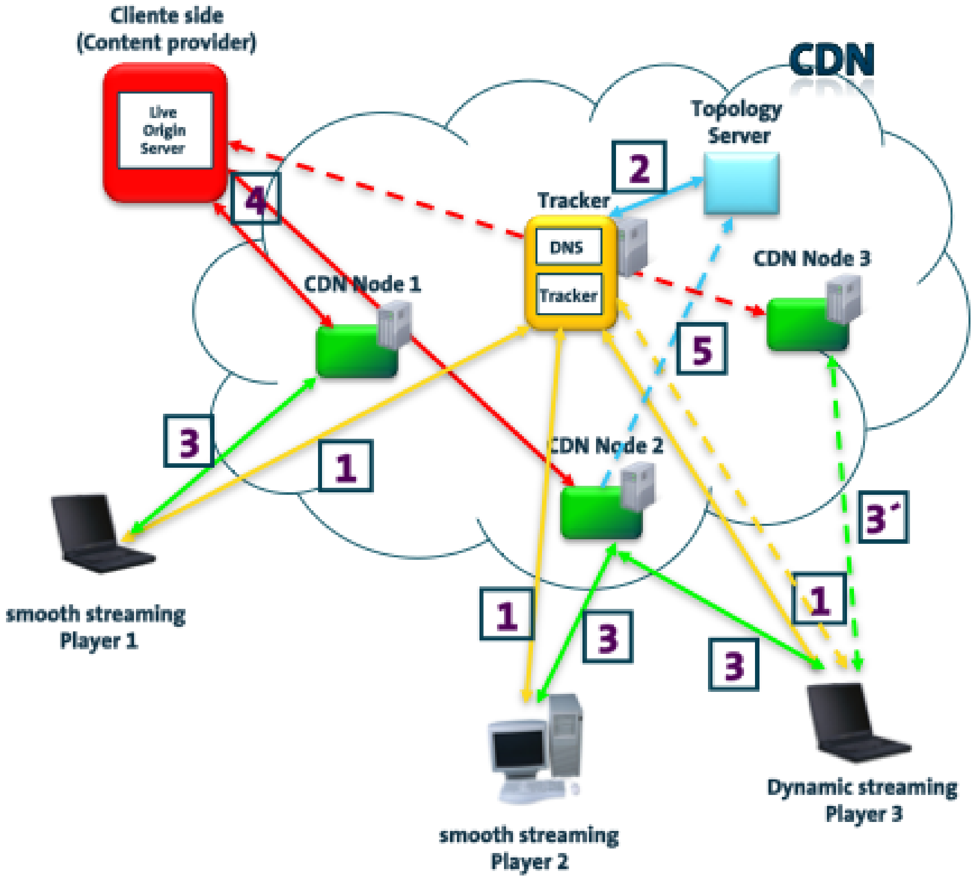

**Figure 5: Improved workflow with ENVISION**

The initial workflow is the same as the previously described:

1) A computer (e.g. Player1) asks the CDN tracker about a specific content by a CDN general URL (e.g. http://www.globalcdn.com/liveEvent1.avi).

2) The tracker geo-locates the computer and chooses the best CDN Node for this customer (let's say its CDN Node 1) thanks to the Topology server and resolves the URL name to the CDN Node 1 IP address.

3) The computer asks the CDN Node 1 about that particular content.

4) CDN Node 1 establishes a connection with the Origin server (client side) and starts serving the content to the computer.

   But then, all of a sudden a network problem arises. This problem has the following implications (broken lines):

   Both customers with player 2 and player 3 watch the video with very poor quality

5) Player 3 uses dynamic streaming, so a new connection is established for each trunk. CDN Node 2 detects, thanks to ENVISION, poor-quality TCP connections and reports this new data to the Topology server.

   The topology collects network quality key indicators. This new data will make the Topology Server choose CDN Node 3 for Player2 and Player3 when asked.

Player 2 uses smooth streaming. That means it won't ask the tracker again and it will keep using the same CDN Node while watching this video. So Player 2 will watch the video with poor quality.

Player 3, however, uses dynamic streaming. So, it will ask the tracker about the best CDN Node to serve the next chunk every 2 seconds. That way, after some minutes, the tracker will choose CDN Node 3 (**3'**) and Player 3 will watch the video with high quality again.

Please, note that even the Player 2 customer will make the most of this algorithm improvement the next time he watches a video.

#### *3.1.5.3 Performance Metrics*

Tests will be specialised on individual tests and not on loading tests. Metrics defined are related with quality of experience

We will evaluate the following metric:

- Quality of Experience perceived by the user

#### *3.1.5.4 Controlled and Uncontrolled Variables*

Controlled variables

- *EndPointNodeNumber*: number of CDN Nodes to serve content.
- *Topology*: initial topology.
- *playersNumber*: number of player.

Uncontrolled variables:

- *round-trip time between nodes*: real round-trip times and their changes during the testbed.

#### *3.1.5.5 Evaluation Scenarios*

The table below shows the different scenarios:

| **Description** | **Metrics** | **Controlled** | **Uncontrolled** |
|---|---|---|---|
| Communication between CDN nodes themselves and between nodes and customer players is OK | *Quality of experience* | *EndPointNodeNumber*<br>*Topology*<br>*playersNumber* | *Round-trip time between nodes* |
| Some network problems between customer nodes and CDN nodes: poor QoE | *Quality of experience* | *EndPointNodeNumber*<br>*Topology*<br>*playersNumber* | *Round-trip time between nodes* |
| CDN detects the problem and The Topology Server reacts and changes Server selection | *Quality of experience* | *EndPointNodeNumber*<br>*Topology*<br>*playersNumber* | *Round-trip time between nodes* |

**Table 6: P4P CDN  Evaluation Scenarios**

### 3.1.6 ENVISION-COMET

While both ENVISION and COMET focus on various aspects of digital data content in the Internet (content access, dissemination, delivery etc.), the high-level approaches employed are different. The COMET project tackles the issue of content dissemination via an overlay approach at the network level resulting in a 2-plane approach aiming to mediate the delivery of Internet content via native COMET network entities. On the other hand, the ENVISION project deals with the problem by developing techniques for the content delivery at the application layer and by fostering the collaboration between the applications and the underlying ISP networks to achieve the co-optimisation of the often misaligned application and network performance objectives.

We identify a possible collaboration between the two projects by exploiting the fact that both COMET and ENVISION propose the creation of novel entities owned by ISPs which can then collaborate to gain better network awareness and thus, achieve better content delivery performance.

#### *3.1.6.1 Objectives*

In COMET, content delivery paths are mediated by the content mediation entity (CME) selecting the server and the path and the content-aware forwarding entity (CAFE) forwarding the data to the appropriate next-hop selected by the CME. Each ISP operates at least one CME and CAFE. A highly extensible 2-phase multi-criteria decision algorithm [CMT3.2] has been developed to take into account various performance metrics when choosing the optimal content delivery path. This algorithm is implemented within the CME.

In ENVISION the CINA server is developed with the purpose of facilitating applications in finding better content delivery options. Each ENVISION-enabled domain deploys a CINA server through which it allows the applications to query about application connections, annotated or ranked using different metrics. These metrics may reflect network performance properties (e.g. routing hop count, estimated delay, etc.) and/or ISP preferences (e.g. cost of transit links) and are used to influence the decisions at the application layer. In COMET, these ENVISION costs [D3.2] can be used as additional criterion in the selection of a content delivery path. For a particular content item, several paths may exist to reach a server with the content, and as the forwarding is performed over CAFE nodes the connection to be ranked by the CINA server is the connection to the next-hop CAFE rather than the connection to the final destination. The CINA server would therefore provide costs for the next-hop domain for all of the candidate paths.

The COMET-ENVISION integration scenario aims therefore at enabling the CINA server to feed to its local CME the costs it maintains such that the decision algorithm within CME can exploit the added information when optimising the content delivery to end users in a more timely and informed manner.

### 3.1.6.2 Evaluation Environment

To achieve this integration, the COMET CME will need to implement the CINA Client protocol stack in order to discover the costs information provided by the CINA Server at each ENVISION-enabled domain. We illustrate the integration setting in Figure 6 within the COMET federated testbed, described in detail in [CMT6.1]. While the figure shows all domains with end users having a CINA server, in a general evaluation scenario not all domains need to be ENVISION-enabled. For example, if some content item requested by CC@ASWUT1 can be found through ASPT, ASWUT2 or ASWUT3, the CINA server would return the corresponding costs of receiving incoming traffic through these domains.

In the case of this particular integration scenario we will use the ALTO-compatible endpoint cost service, see section 7.7.5 of ALTO protocol (http://tools.ietf.org/html/draft-ietf-alto-protocol-10). The endpoint in this case is the IP address of the CAFE that is the next-hop on a candidate path.

In order to set the costs announced by CINA, we will use a configuration file mapping remote IP prefixes (or in our case simply the next-hop CAFE IP addresses) to inter-domain links, and assigning costs to these links. For any particular evaluation scenario we should be able to assign costs such that we can demonstrate the impact on the COMET path selection from taking into account the ENVISION costs.

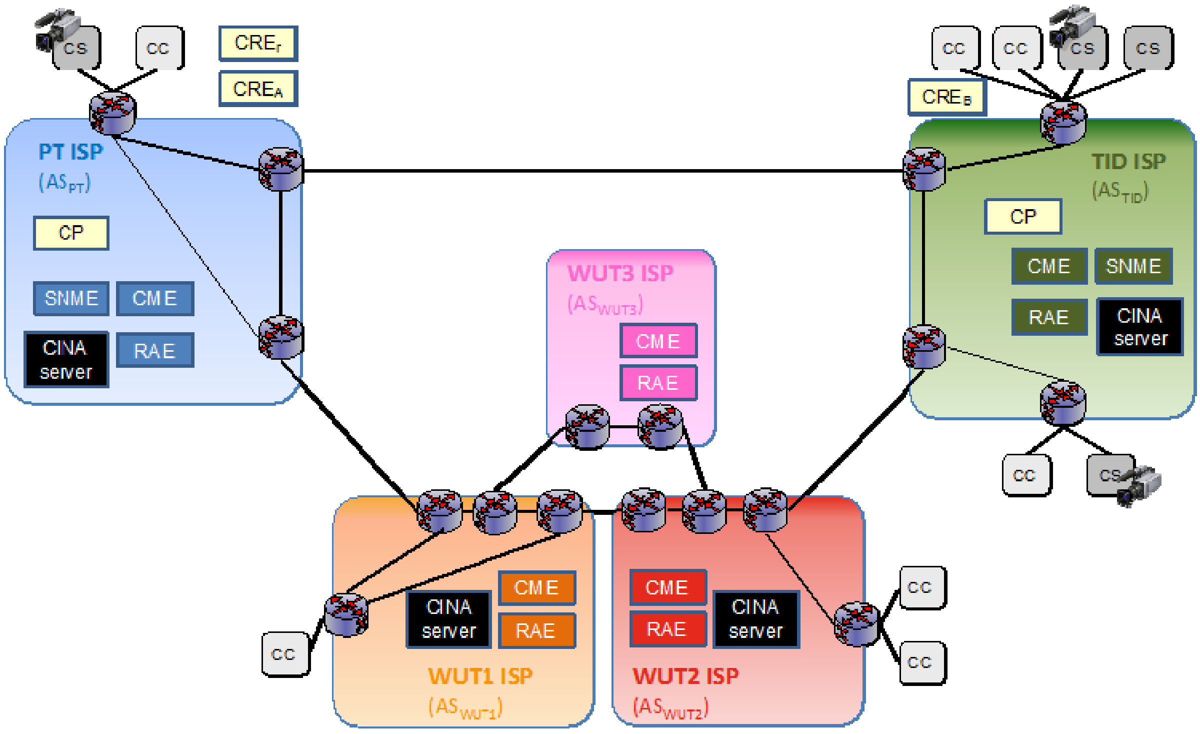

**Figure 6: COMET-ENVISION integration**

## 3.2 Simulations and Theoretical Analysis

### 3.2.1 Distributed Data Management System

The specifications of this system are provided in detail in section 3 of [D4.2].

#### *3.2.1.1 Objectives and Evaluation Requirements*

- Performance evaluation of the n-casting application overlay in terms of search completeness, accuracy and response time under various conditions and different popularity distributions for the resource groups.
- Evaluation of the scalability and the efficiency of the n-casting system in terms of protocol message overhead and its growth rate with the number of nodes and the frequency of updates in the system.
- Evaluation of the scalability and the efficiency of the n-casting system in terms of size of the routing tables at each node and the size of the routing protocol messages and how it increases with the number of nodes and the number of distinct resource group identifiers.
- Cost/benefit analysis of the performance and efficiency trade-offs associated with specific n-casting design parameters.

#### *3.2.1.2 Evaluation Environment*

A stochastic simulator programmed in Java has been implemented. The simulator takes input data produced by a separate clustering program and uses them to create the n-casting overlay over the given clusters, and the corresponding protocol structures including the n-casting routing tables and group membership data. Overlay messages and statistics are recorded including node discovery events when nodes join or leave the overlay, routing updates when nodes join or leave a particular group and query forwarding messages when nodes enquire for resources of a particular group. The size of clusters and the number of groups in the overlay can be varied, in order to provide realistic modelling of population density and regional group popularity.

As input to the simulator for the distribution of nodes in groups and geographical areas we will use the BitTorrent datasets obtained by [CLY+11], containing information about torrents and the peers participating in their distribution. Information about the peers includes their AS and geographical location.

### 3.2.1.3 Performance Metrics

- *query time*: The sum of the propagation delay along the overlay query forwarding path. The average and standard deviation values across all queries will be reported.
- *query completeness error*: The proportion of group members that exist in the system but were not discovered by a query. This is calculated as: $\frac{\min(N_e,N)-N_d}{\min(N_e,N)}$, where $N_e$ is the number of group members that exist in the system and $N_d$ is the number of group members discovered by the query. The average and standard deviation values across all queries will be reported.
- *query accuracy error*: The difference in distance of the discovered from the actual closest group members. This is calculated as: $avg_{i\in\aleph_d}(D_i) - avg_{j\in\aleph_c}(D_j)$, where $\aleph_d$ is the set of the discovered nodes, $\aleph_C$ is the set of the actual closest *N* nodes and $D_i$ is the propagation delay from the querying node to node *i*. The average and standard deviation values across all queries will be reported.
- *routing protocol overhead*: The average number of routing messages exchanged per node for each membership change. The average and standard deviation values across all nodes will be reported.
- *forwarding protocol overhead*: The number of forwarding messages per query. The average and standard deviation values across all queries will be reported.
- *routing table size*: The size of the routing table at each node in terms of number of entries for the local and remote areas and the size of the corresponding entries.

### 3.2.1.4 Controlled and Uncontrolled Variables

Controlled variables:

- *number of distinct members – N*: The maximum number of group members to be returned by a query.
- *maximum number of nodes per cluster*: The maximum number of nodes in a low level cluster.

Uncontrolled variables:

- *number of nodes*: The number of nodes participating in the n-casting overlay, starting from a few thousand and up to a hundred thousand nodes.

- *number of keys*: The number of distinct keys used to uniquely identify overlay application resource groups.
- *membership churn*: The frequency of nodes joining or leaving a particular group.
- *group popularity*: The distribution of the number of members per group and per geographical area, following the distribution of the number of peers per torrent in the dataset under consideration, and targeting specifically a uniform distribution with well-populated groups and a highly skewed distribution with some sparse and some highly populated groups.

### 3.2.1.5 Evaluation Scenarios

The distributed data management system evaluation scenarios are captured in Table 7.

| **Description** | **Metrics** | **Controlled Variables** | **Uncontrolled Variables** |
|---|---|---|---|
| Performance with Number of Nodes and Group Popularity:<br>Evaluate the n-casting performance under different conditions and group popularity distributions | query time, query completeness error, query accuracy error | number of distinct members<br>maximum number of nodes per cluster | number of nodes<br>group popularity |
| Scalability with Number of Nodes and Group Membership Churn:<br>Evaluate the protocol overhead in large scale setups and highly dynamic conditions | routing protocol overhead, forwarding protocol overhead | maximum number of nodes per cluster | number of nodes<br>membership churn<br>node churn |
| Scalability with Number of Nodes and Number of Resource Groups:<br>Evaluate the size of the routing tables as a function of the number of resource groups | routing table size | maximum number of nodes per cluster | number of nodes<br>number of keys |

**Table 7: Distributed Data Management System Evaluation Scenarios**

### 3.2.2 Live Content Distribution

The specifications of this system are provided in detail in section 4 of [D4.2].

The tests described here complement the prototype evaluation scenarios described in section 3.1.1. In this section we focus on experimentation with the performance and scalability properties of the live content distribution algorithms under various setups and dynamic conditions created in a controlled simulation environment.

#### *3.2.2.1 Objectives and Evaluation Requirements*

- Performance evaluation of the live video distribution system under various configurations, content bandwidth requirements and SVC profiles, number of participants and underlying network performance conditions.
- Evaluation of the impact on performance for increasing degrees of distribution of the coordination functions across smaller clusters of nodes to achieve scalability.
- Evaluation of the performance gains achieved with the deployment and use of CINA services and in particular multicast and high-capacity nodes, under different overlay and network conditions and for a range of coverage of these services.

#### *3.2.2.2 Evaluation Environment*

A custom simulator developed by UCL and brought to ENVISION as background software is used. This simulator allows for experimentation with arbitrary sets of overlay nodes and topology construction algorithms. Existing datasets with real node distributions and RTT measurements between them can be used as input, or the simulator can generate artificial sets for arbitrarily large number of nodes. Nodes can join or leave the overlay at predefined times or at randomised events following predefined profiles with different distributions including exponential, log-normal or Weibull [WEIB51]. The simulator models the protocol and data message exchange taking into account the RTT, upload and download capacities of the nodes in an overlay connection, but without modelling the behaviour of the network between the overlay nodes.

These capabilities will be enhanced in ENVISION to support the integration of ISP resources into the overlay topology, and in particular high-capacity nodes and multicast resources. The specific overlay topology construction policies for live SVC video studied in ENVISION will be implemented in the simulator, and their performance will be evaluated using the performance metrics specified below.

#### *3.2.2.3 Performance Metrics*

- *stream liveness*: The time difference between the playout point at a consumer and the time a chunk is produced at the source. For each user the average value during the lifetime of its session will be used. For the entire overlay application, a percentile (e.g. 90th percentile) of the values per user will be used.
- *stream quality*: The spatial, temporal and SNR resolution of the stream (i.e. the SVC layer) a user receives. The values per user and for the overlay will be calculated as above.

#### *3.2.2.4 Controlled and Uncontrolled Variables*

Controlled variables:

- *topology construction policy*: Several policies considered in [D4.2] will be tested for establishing overlay connections starting from the live video source node and selecting the next hops based on network delay, upload capacity, SVC layers to be consumed at each participant node, density of other nodes consuming the stream at a particular area, etc.
- *overlay node clustering*: The grouping of the overlay nodes to clusters of smaller manageable size to facilitate the overlay coordination functions. Several clustering methods and cluster sizes will be investigated in order to study the impact of clustering to the overlay performance metrics.
- *use of multicast service*: The support of multicast transmission for some subsets of overlay nodes corresponding to ENVISION-enabled ISPs. The overlay topology construction policies in this case will be enhanced to use multicast for these subsets and the experiments will assess the corresponding overlay performance gains.
- *use of high-capacity nodes*: The availability of high-capacity nodes in some locations corresponding to ENVISION-enabled ISPs. The overlay topology construction policies in this case will be enhanced to integrate these nodes to the overlay topology and determine which SVC layers or stream segments they will be distributing. This choice is not available for normal participant nodes that need to download the entire stream according to their quality requirements.

Uncontrolled variables:

- *number of nodes*: The number of nodes participating in the live video streaming overlay, starting from a few thousand and up to a hundred thousand nodes.
- *node delay and capacity distribution*: The distribution of the RTT between overlay nodes and their upload/download capacity. Realistic datasets will be used as much as possible, such as iPlane [MIP+06], Peerwise [LLS07], and measurements gathered with TurboKing [LL08] from sideground activities of the ENVISION partners.
- *stream SVC profile*: Streams of different bitrates will be used, with several levels of quality mapped to SVC layers under a set of predefined profiles.
- *multicast coverage*: Multicast coverage can be limited, with multicast support only for small subsets of nodes at few areas, or wider including a significant number of areas with large numbers of nodes.
- *high-capacity nodes availability*: Several degrees of high-capacity nodes availability will be investigated, from few locations and limited capacity to a larger set of nodes with substantial capacity.

### *3.2.2.5 Evaluation Scenarios*

The live content distribution evaluation scenarios are captured in Table 8.

| Description | Metrics | Controlled Variables | Uncontrolled Variables |
|---|---|---|---|
| Performance with Overlay Topology Construction Policy:<br>Evaluate the performance with different topology construction policies and under different overlay setups and stream profiles | stream liveness, stream quality | topology construction policy: all candidate policies<br>overlay node clustering: OFF<br>use of multicast: OFF<br>use of high-capacity nodes: OFF | number of nodes<br>node delay and capacity distribution<br>stream SVC profile |
| Performance with Node Clustering:<br>Evaluate the impact on the performance with different methods of clustering and cluster sizes | stream liveness, stream quality | topology construction policy: all candidate policies<br>overlay node clustering: all candidate methods and cluster sizes<br>use of multicast: OFF<br>use of high-capacity nodes: OFF | number of nodes<br>node delay and capacity distribution<br>stream SVC profile |
| Performance with Multicast Support:<br>Evaluate the performance gains achieved with the use of multicast under different multicast coverage conditions | stream liveness, stream quality | topology construction policy: the most successful policies from the first test suite<br>overlay node clustering: OFF<br>use of multicast: ON<br>use of high-capacity nodes: OFF | number of nodes<br>node delay and capacity distribution<br>stream SVC profile<br>multicast coverage |
| Performance with High-Capacity Node Availability:<br>Evaluate the performance gains achieved with high-capacity nodes under different availability scenarios | stream liveness, stream quality | topology construction policy: the most successful policies from the first test suite<br>overlay node clustering: OFF<br>use of multicast: OFF<br>use of high-capacity nodes: ON | number of nodes<br>node delay and capacity distribution<br>stream SVC profile<br>high-capacity nodes availability |

**Table 8: Live Content Distribution Evaluation Scenarios**

### 3.2.3 Content Adaptation

The detailed specifications of the proposed mechanisms are provided in sections 4.1 and 4.2 of D5.2.

Stream smoothing is the set of mechanisms used in order to enhance the user's quality of experience under content adaptation.

Scheduling is the mechanism used to efficiently request the chunks from the other peers with respect to links' capacity, chunks priority, and chunks availability in the neighbourhood.

#### *3.2.3.1 Objectives and Evaluation Requirements*

a) Stream Smoothing :
   - Evaluate the effect of bandwidth fluctuation on the smoothness of the stream adaptation.
   - Evaluate the stream continuity under content adaptation.

b) Efficient Scheduling
   - Evaluate the throughput of the system with respect to the overlay neighbour density and different streaming conditions.
   - Evaluate the efficiency of the proposed priority mechanism in terms of delivery ratio.
   - Evaluate the optimality of the proposed non-layered scheduling method and the heuristic proposed in the layered method.

#### *3.2.3.2 Evaluation Environment*

- The network simulator (NS2) and Simulink-Matlab simulator are used for evaluating the performance of the proposed mechanisms.

#### *3.2.3.3 Performance Metrics*

- ***Number of layer changes during video playback***: We measure the average number of transition from higher layers to lower layers.
- ***Stalling Events***: We are interested in the average number and duration of stalling events that occur during video playback due to the unavailability of the content. The shorter the stalling event, the better the session quality.
- ***Bandwidth utilisation***: The effective utilisation of the available bandwidth is an important consideration in P2P streaming. It measures the effective bandwidth used over the total available bandwidth in the network.
- ***Useless chunks ratio***: It represents the number of chunks that arrive at each peer after their playback deadline or higher layers chunks without their dependent lower layer chunks over the total number of encoded chunks.
- ***Delivery ratio at layer L***: It is defined as the average delivery ratio at layer l among all the peers that can play layer L. A chunk of layer L is considered as properly received if and only if all the related chunks from layers lower than L are already received before their playback deadline.

#### *3.2.3.4 Controlled and Uncontrolled Variables*

***1) Controlled variables***

- *Overlay size*: the number of peers participating in the overlay.
- *Peer degree*: the number of neighbours a peer is connected to.

- *Bandwidth fluctuation interval*: this is the time interval, in which the available bandwidth is supposed constant, and after which, it fluctuates to another value.
- *Streaming rate*: the bitrate of the overall video stream, i.e. the bitrate of the higher layer in layered-video, and the stream bitrate in non-layered video.
- *Stream composition*: the number of layers composing a video stream and their characteristics (bitrate, dependency ...).

***2) Uncontrolled variables***

- *Bandwidth fluctuation*: the distribution of the available bandwidth value.
- *Neighbour peer selection*: the manner how peers are connected to the other peers. Each peer selects randomly its neighbours among the other peers.
- *Packet delay*: the packet delay depends inherently on the network topology, the waiting time in the different queues across the protocols layers stack and the network elements.

### 3.2.3.5 Evaluation Scenarios

| Description | Metrics | Controlled Variables | Uncontrolled Variables |
|---|---|---|---|
| *Layer changes:*<br>Evaluate the impact on smoothing with varying the bandwidth | Number of layer changes | - Bandwidth fluctuation interval= 10seconds<br>- Overlay size = 300 peers.<br>- Streaming rate varies from 250kbps to 500kbps<br>- Total number of layers: 4 | Bandwidth fluctuation= random |
| *Stream continuity:*<br>Evaluate the impact on the continuity of stream with varying overlay size and bandwidth | Stalling events | - Bandwidth fluctuation interval= 10seconds<br>- Overlay size = 300 peers<br>- Streaming rate varies from 250kbps to 500kbps<br>- Total number of layers: 4 | Bandwidth fluctuation= random |
| *Bandwidth utilisation:*<br>Evaluate the impact of the number of the neighbours and the streaming rate on the bandwidth utilisation in the proposed scheduling mechanism. | Bandwidth utilisation | - Number of neighbours varies from 5 to 30<br>- Overlay size = 500 peers<br>- Streaming rate varies from 250kbps to 500kbps<br>- Total number of layers =12 | Download bandwidth of peers (40% users with 512Kbps, 30% with 1Mbps and 30% with 2Mbps) |
| *Useless chunks:*<br>Defined as the ratio of chunks that arrive after | Useless chunks ratio | - Number of neighbours varies from 5 to 30<br>- Overlay size = 500 peers | Download bandwidth of peers (40% users with 512Kbps, 30% |

| | | | |
|---|---|---|---|
| their playback deadline or without the corresponding lower layer chunks. | | - Streaming rate varies from 250kbps to 500kbps<br>- Total number of layers =12 | with 1Mbps and 30% with 2Mbps) |
| *Delivery ratio:* The delivery ratio at layer *l* is the average delivery ratio at layer *l* among all the nodes that can play layer | Delivery ratio | - Overlay size = 500 peers<br>- Total number of layers: 6 and 12 layers.<br>- Streaming rate= 250kbps | - Number of neighbours |

**Table 9: Content Adaptation Evaluation Scenarios**

# 4. EVALUATION RESULTS

## 4.1 Prototypes

### 4.1.1 MLEP

The MLEP as described in detail in ENVISION [D5.2] consists of technologies for data resource monitoring followed by predictions and estimations of links performances, content generation, protection and scheduling. The initial results of the technology evaluation are found in this chapter and detailed in the following subsections.

#### *4.1.1.1 Monitoring and Content Generation*

The target bitrate for the content generation is based on channel monitoring and predictions, In Figure 7 the blue line is the aggregated throughput generated by the LiveU testbeds described in section 2.4. The bitrate variations are artificially generated using a preconfigured script which is based on actual measurements in 3G networks. The brown line is the adaptive generated AVC stream which is controlled by the monitoring and prediction module. As seen, the generated data rate is somehow less than the actual data rates, to eliminate congestion and other phenomena which are hard to deal with in delay-sensitive applications. Whenever the brown line exceeds the blue line, a delay is generated and the probability of packet loss is increased. In AVC streams this could be critical to the QoE, thus a relatively large amount of FEC is required on the expense of the video rate.

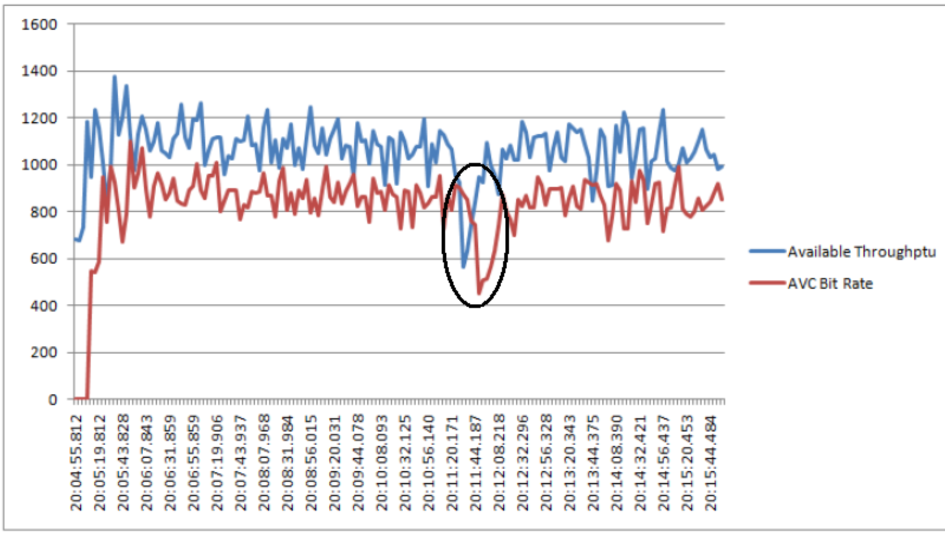

**Figure 7: Aggregated Available bitrate versus AVC bitstream**

The aggregated bitrate when applying the same channel conditions to SVC is shown in Figure 8, again the blue line is the aggregated available bandwidth, the brown is the base layer of the video stream, green is the sum of the base and one enhancement layer and the purple line is the entire stream (3 layers, in this example). The idea is to generate the stream in a way that the worst case channel conditions still enable a "safe" delivery of the base layer. In that case, using a smart conservative scheduler as described in ENVISION [D5.2] would allow an increase of the QoE relatively to AVC stream.

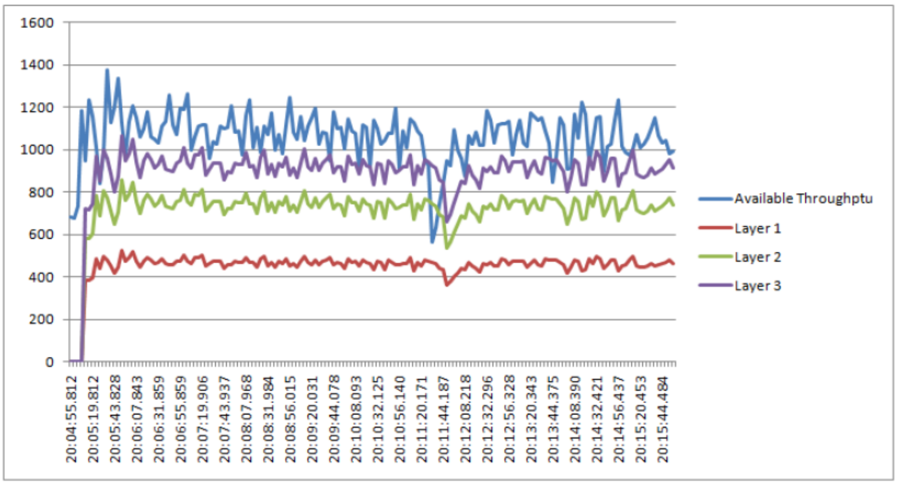

**Figure 8: aggregated bitrate versus SVC bitstream**

## 4.2 Simulation and Theoretical Analysis

### 4.2.1 Content Adaptation

### *4.2.1.1 Preliminary results*

#### 4.2.1.1.1 Smoothing mechanism

In this scenario, we evaluate the main smoothness metrics of our system: layer changes and stalling events. We consider a fixed size overlay with 300 peers having varying download capacity. During the session, the peers exchange the buffer map periodically. Each peer runs the smoothing algorithm as described in D5.2 Section 4.2.1. We compare our proposed mechanism with LayerP2P. We introduced sudden bandwidth changes in the network by varying the capacity of peers at random intervals. This allowed us to observe the effectiveness of our proposed smoothing mechanism.

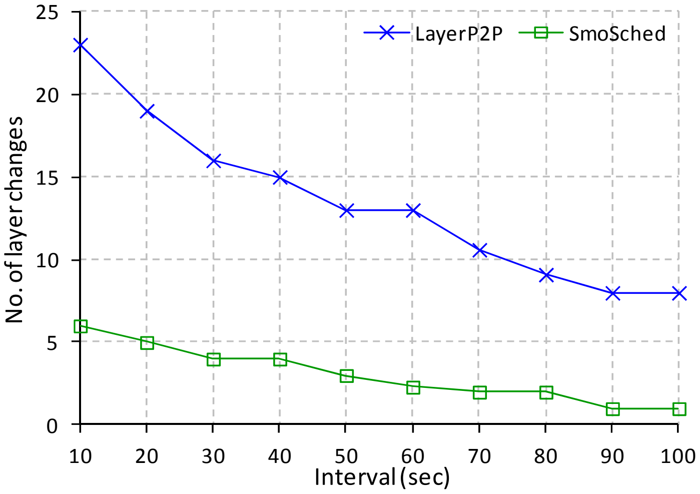

**Figure 9: Number of Layer Changes**

Figure 9 shows the number of layer changes at different intervals for both mechanisms. The objective was to minimise the number of layer changes that occur due to changing network conditions. The interval on x-axis represents the period of time between successive changes in the bandwidth. As expected, the number of layer changes decreases as this interval increases. Thus, it is found that bandwidth changes have an obvious effect on quality variation. When compared with the state of the art, our proposed mechanism has fewer layer changes due to the smoothing mechanism which adjusts the quality level for each smoothing window. As a result, the frequency of changes in quality level is minimised. Comparatively, LayerP2P system has higher number of quality changes due to the absence of an efficient smoothing mechanism.

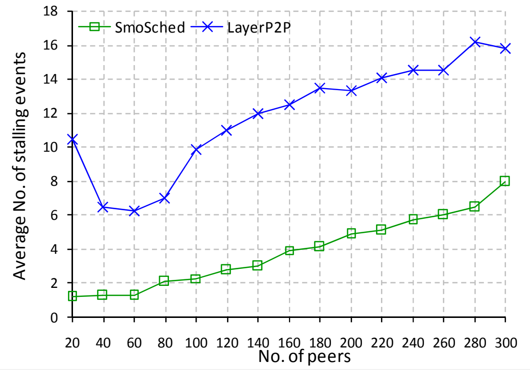

**Figure 10: Average Number of Stalling Events**

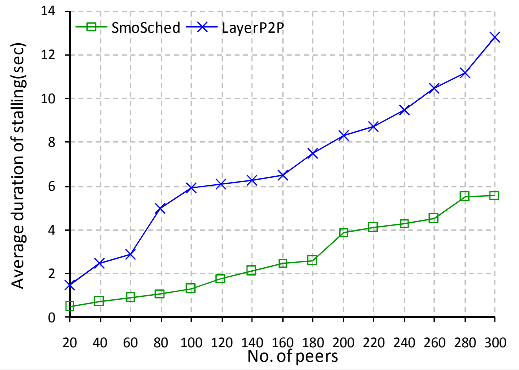

**Figure 11: Average Duration of Stalling Events**

Figure 10 shows the average number of stalling events for both mechanisms. The stalling of video stream occurs due to the unavailability of selected layer chunks at that particular time period. The maximum number of stalling event in our case was not more than 6. However, there is a huge variation of stalling events in case of LayerP2P. Moreover the average duration of stalling event is lower in our proposed mechanism as shown in Figure 11.

### 4.2.1.1.2 Scheduling mechanism

We use in our simulation a simple algorithm for overlay construction: each node randomly selects its neighbours so that a random graph is constructed. The overlay is composed of 500 nodes and each node has n neighbours. A node estimates the bandwidth that each of its neighbours is willing to provide based on the traffic history. We consider three categories of peers having different upload and download capacities, as described in section 3.2.8.5.

Peers exchange periodically their buffers maps (section 4.2.1 of D5.2) and run the proposed scheduling algorithm to decide which chunk to request from which peer in order to optimise the bandwidth utilisation in the network and reduce the useless chunk ratio and maximise the delivery ratio of the system. We compare the performance of our mechanism with three scheduling methods, namely Random strategy (RND), Local Rarest First (LRF) and Round Robin (RR).

***Bandwidth utilisation:*** We observe in Figure 12 that our proposed mechanism outperforms the three other scheduling schemes and ensures more than 90% bandwidth utilisation. In comparison, LRF only allows a maximum bandwidth utilisation of 88%; RND shows the worst performance, with a maximum utilisation of only 71%. On the other hand, we note that the bandwidth utilisation for all the schemes increases as the number of neighbours increases. This effect is more pronounced for the first 15 neighbours of each peer. This is easily explained by the concomitant increase in chunk availability that it produces. Conversely, a more stable behaviour is observed when the number of neighbours for a peer is greater than 15. We conclude that, for the considered overlay, the average of 15 neighbours per peer is enough to ensure adequate chunk availability.

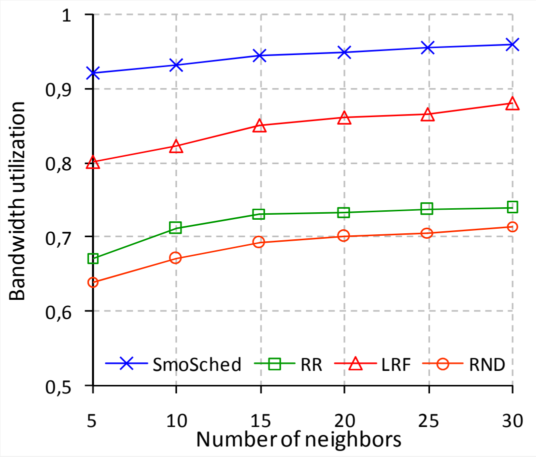

**Figure 12: Bandwidth Utilisation vs. Number of Neighbours.**

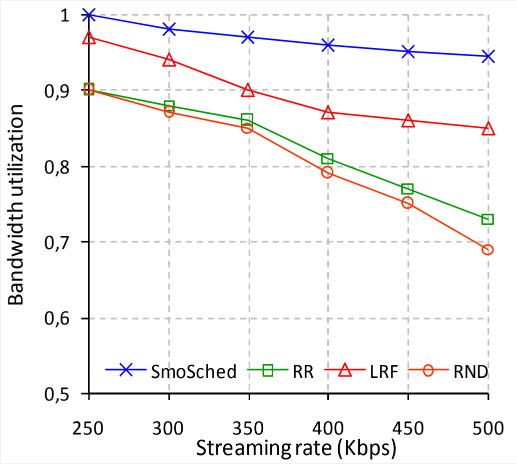

**Figure 13: Bandwidth Utilisation vs. Streaming Rate**

In Figure 13, we note the general decrease in bandwidth utilisation with increase in streaming rate. This can be explained by the fact that, when the streaming rate is high, a missed chunk has a smaller chance to be rescheduled before its playback deadline. Nevertheless, our proposed mechanism outperforms the three other mechanisms, and allows a gain up to 25%. This is due to our assignment mechanism, which takes better advantage of the available bandwidth capacity.

***Useless Chunk Ratio:*** In Figure 14 and Figure 15, we evaluate the useless chunk ratio according to both the neighbour density and the streaming rate. In Figure 14, it can be observed that the useless chunk ratio decreases with an increasing number of neighbours. This decrease can be explained by a higher probability of getting the requested chunks from the large pool of neighbouring peers. However, because our algorithm is engineered to request the right chunk from the right neighbour, it achieves a much lower useless chunk ratio when compared with competing scheduling mechanisms.

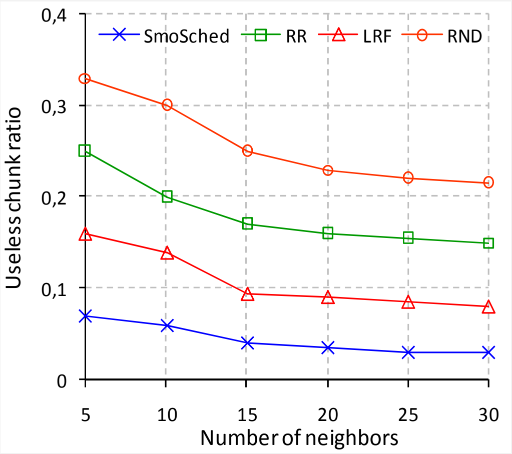

**Figure 14: Useless Chunk Ratio vs. Number of**

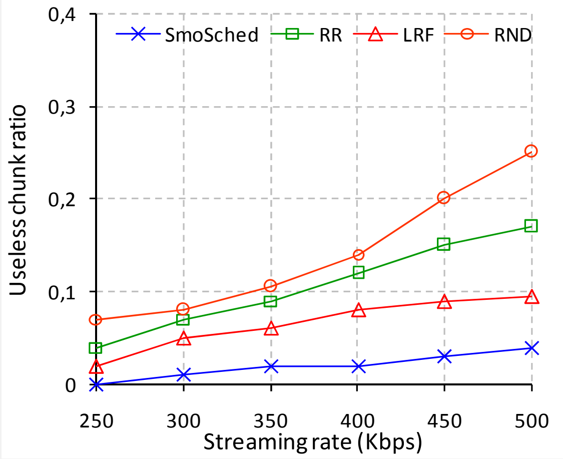

**Figure 15: Useless Chunk Ration vs. Streaming**

**Neighbours.**

**Rate.**

In Figure 15, we evaluate the useless chunks ratio as a function of streaming rate. It is found that, in general, the useless chunk ratio increases with an increase in the streaming rate. However, our proposed mechanism still outperforms the existing systems.

***Delivery Ratio:*** Figure 16 describes the delivery ratio for each enhancement layer. We encode the video into 12 layers and set the rate of each layer at 100 Kbps. First, we note that our proposed mechanism operates well: for the lower quality layers, the delivery ratio is approximately 1, and never drops below 0.9 even for the highest quality layers. Although for lower layer RR has very good performance, its performance decreases rapidly for higher layers and, for the highest quality layer, it is the worst scheduling policy. Due to this effect, the LRF strategy has higher delivery ratio than the RR strategy for high quality layers, but even then its performance is inferior to our proposed algorithm. Finally, the random strategy has the poorest performance. As shown in Figure 16, our proposed mechanism outperforms other strategies with a gain of 10%-50% in most layers.

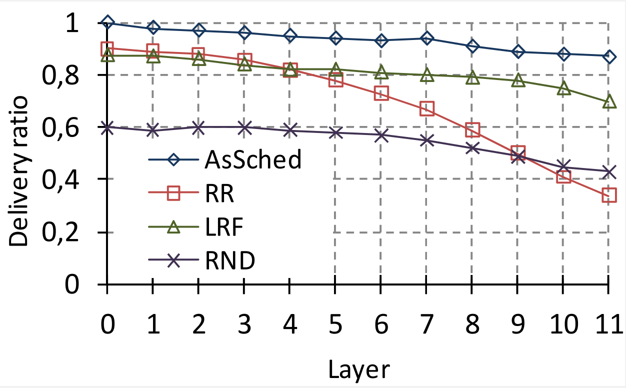

**Figure 16: Delivery Ratio in Case of Video Encoded into 12 Layers.**

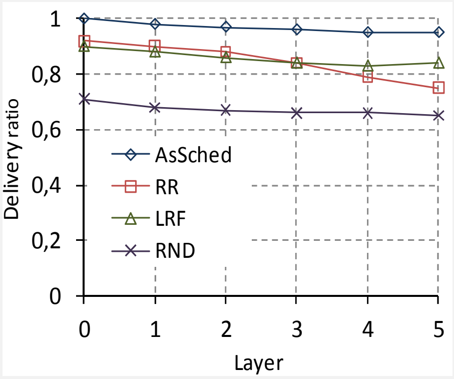

**Figure 17: Delivery Ration in Case of Video Encoded into 6 Layers.**

In order to show the importance of varying number of layers, we present an alternative scenario in which the stream is encoded only into 6 layers. In Figure 17, we note that the delivery ratio of each layer is similar to that one achieved in the 12 layer encoding scenario. Our mechanism is still the best among all the three others methods. However, we note that the delivery ratio of all the methods is marginally higher than in the case of 12 layers. This is due to the fact that encoding the video into six layers allows peers to allocate all their bandwidth to lower layers, whereas in the first case, some capacity is allocated to higher layers.

# 5. CONCLUSIONS

This document describes some of the WP6 activities during M18-M24, the document contains the specifications of the testbeds that will be used in the project for evaluation and experimentations. We believe that the four testbeds identified and prepared are fulfilling the requirements for evaluations and experimentation in the project. The FT testbed was chosen to be the primary ENVISION testbed, and hopefully at the final stage most of the technologies would be deployed and evaluate in this testbed. Additional testbeds that have been defined are TID testbed that will be used for optimisation of live content distribution, ALUD testbed for evaluation of the IVCD and the HCN and the LiveU testbed will be used to evaluate the MLEP technologies for video boosting over the uplink.

The progresses we have made in the evaluation specifications is reported and includes a set of prototype evaluation scenarios and a set of simulations/theoretical analysis that complement one another. The initial specifications provide a good and comprehensive environment for validating the requirements. Each of the identified prototypes and work is detailed described including the relevant parameters, the scenario and more. These evaluation specifications are defined in section 3.

Finally, we have provided some of our initial experimental results; these are samples of results we have achieved and many more results are expected in the following months. For the SVC content adaptation simulation an evaluation of the main smoothness metrics of our system was achieved for layer changes and stalling events. In the prototype evaluation we have provided some initial results of content monitoring and generation of video streams.

# REFERENCES

[AFS07] Aggarwal, V., Feldmann, A., Scheideler, C.: Can ISPs and P2P users cooperate for improved performance? SIGCOMM Comput. Commun. Rev. 37, 29–40 (July 2007)

[BCC+06] Bindal, R., Cao, P., Chan, W., Medved, J., Suwala, G., Bates, T., Zhang, A.: Improving traffic locality in BitTorrent via biased neighbour selection. In: Proc. of ICDCS '06. p. 66. Washington, DC, USA (2006)

[BLD10] Blond, S.L., Legout, A., Dabbousa, W.: Pushing bittorrent locality to the limit. Comput. Netw. 55(3) (2010)

[BRCA09] Fabrìcio Benevenuto, Tiago Rodrigues, Meeyoung Cha, and Virgìlio Almeida. Characterizing User Behavior in Online Social Networks. In IMC, 2009.

[CDZ97] K. Calvert, M.B. Doar, and E.W. Zegura. "Modeling Internet Topology". IEEE Communications Magazine, June 1997.

[CLIP01] Clip2, "The Annotated Gnutella Protocol Specification v0.4" (2001). Available: http://rfc-gnutella.sourceforge.net/developer/stable/index.html

[COHE08] Bram Cohen, "The BitTorrent Protocol Specification v.11031" (2008). Available: www.bittorrent.org/beps/bep_0003.html

[CLY+11] R. Cuevas, N. Laoutaris, X. Yang, G. Siganos, P. Rodriguez, "Deep Diving into BitTorrent Locality", IEEE INFOCOM'11.

[CMT3.2] W. K. Chai, et. al., "D3.2: Final specification of mechanisms, protocols and algorithms for the content mediation system", Dec. 2011, http://www.comet-project.org.

[CMT6.1] W. K. Chai, et. al., "D6.1: Demonstration Scenarios and Test Plan", Dec. 2011, http://www.comet-project.org.

[D3.2] ENVISION Deliverable D3.2, "Refined Specification of the ENVISION Interface, Network Monitoring and Network Optimisation Functions," January 2012.

[DLL+11] Dai, J., Li, B., Liu, F., Li, B., Jin, H.: On the efficiency of collaborative caching in isp-aware p2p networks. In: Proc. of INFOCOM. pp. 1224–1232. IEEE (2011)

[EYR11] Vijay Erramilli, Xiaoyuan Yang, and Pablo Rodriguez. Explore what-if scenarios with SONG: Social Network Write Generator. http://arxiv.org/abs/1102.0699, 2011.

[GSG02] K. P. Gummadi, S. Saroiu, and S. D. Gribble. King: estimating latency between arbitrary internet end hosts. In Proc. of SIGCOMM, volume 32, pages 11-11, 2002.

[GWH07] Scott A. Golder, Dennis M. Wilkinson, and Bernardo A. Huberman. Rhythms of Social Interaction: Messaging Within a Massive Online Network. In C&T, 2007.

[KLPM10] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is Twitter, a Social Network or a News Media? In WWW, 2010.

[LBSB09] Cristian Lumezanu, Randy Baden, Neil Spring, and Bobby Bhattacharjee. Triangle inequality variations in the internet. In Proc. of IMC '09, pages 177-183, New York, NY, USA, 2009. ACM.

[LL08] D. Leonard, and D. Loguinov, "Turbo King: Framework for Large-Scale Internet Delay Measurements," IEEE INFOCOM, Apr. 2008.

[LLS07] C. Lumezanu, D. Levin, and N. Spring. Peerwise discovery and negotiation of faster paths. In Proc. Workshop on Hot Topics in Networks (HotNets), 2007.

[LLSB08] Dave Levin, Katrina LaCurts, Neil Spring, and Bobby Bhattacharjee. BitTorrent is an auction: analyzing and improving BitTorrent's incentives. SIGCOMM Comput. Commun. Rev., 38(4):243-254, 2008.

[LNNK05] David Liben-Nowell, Jasmine Novak, Ravi Kumar, Prabhakar Raghavan, and Andrew Tomkins. Geographic Routing in Social Networks. Proceedings of the National Academy of Sciences, 102:11623–11628, 2005.

[MIP+06] Harsha V. Madhyastha, Tomas Isdal, Michael Piatek, Colin Dixon, Thomas Anderson, Arvind Krishnamurthy, and Arun Venkataramani. iPlane: An Information Plane for Distributed Services. In Proc. of ACM OSDI, pages 367-380, 2006.

[MLMB01] Alberto Medina, Anukool Lakhina, Ibrahim Matta, and John Byers. 2001. Brite: Universal Topology Generation from a User's Perspective. Technical Report. Boston University, Boston, MA, USA.

[PER09] Josep M. Pujol, Vijay Erramilli, and Pablo Rodriguez. Divide and Conquer: Partitioning Online Social Networks. http://arxiv.org/abs/0905.4918, 2009.

[RBP+11] Costin Raiciu, Sebastien Barre, Christopher Pluntke, Adam Greenhalgh, Damon Wischik, Mark Handley, Improving Datacenter Performance and Robustness with Multipath TCP, SIGCOMM, 2011.

[SFKW09] Fabian Schneider, Anja Feldmann, Balachander Krishnamurthy, and Walter Willinger. Understanding Online Social Network Usage from a Network Perspective. In IMC, 2009.

[Sin84] R. W. Sinnott. Virtues of the Haversine. Sky and Telescope, 68:159, 1984.

[TODH63] Isaac Todhunter, “Spherical Trigonometry: For the Use of Colleges and Schools”. Macmillan & Co., 1863

[WEIB51] Weibull, W., "A statistical distribution function of wide applicability", J. Appl. Mech.-Trans. ASME 18 (3): 293–297, 1951.

[ZLMZ05] Beichuan Zhang, Raymond Liu, Daniel Massey, and Lixia Zhang. Collecting the internet as-level topology. SIGCOMM Comput. Commun. Rev., 35(1):53-61, 2005.