

Enriched Network-aware Video Services over Internet Overlay Networks

www.envision-project.org



Deliverable D6.2

Evaluation Results of Content-aware Cross-layer Optimisations for Advanced Multimedia Applications

Public report, v2d, 28 February 2013

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 the results of the evaluation activities undertaken in the third year of the ENVISION project. The validity and the benefits of the cross-layer optimisation techniques developed in the project are assessed with prototype-based evaluations in realistic environments provided in the project testbeds and with simulations testing the scalability, stability and performance of particular optimisation techniques under various conditions. The CINA interface is proven to increase both network and overlay application performance.

© Copyright 2013 ENVISION Consortium

University College London, UK (UCL)

Alcatel-Lucent Deutschland AG, Germany (ALUD)

Université Bordeaux 1, France (LaBRI)

France Telecom Orange Labs, France (FT)

Telefónica Investigación y Desarrollo, 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 second and final 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 [D3.3]. The aim of WP6 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 evaluation activities documented here are the result of the work undertaken in the second half of the project. In this report we are providing an update on the testbed setup (T6.1), the specification of the functional validation and performance analysis tests (T6.2) and the results captured through the experimentation with the ENVISION components (T6.3).

The project has setup four testbeds presented in chapter 2. The FT testbed (section 2.1) is considered as the main ENVISION testbed, allowing for evaluation and demonstration of the integrated prototypes and most of the developed technologies. It consists of a number of wired network segments located in Lannion and a wireless 3G network segment located in Paris, modelling in total three different ISPs and using replicas of the network elements used in FT's operational network. The other testbeds are built by ALUD, TID and LiveU and serve different purposes typically specialised to a particular technology under evaluation.

The tests undertaken in the project and their corresponding results are documented in chapter 3. An overview of the evaluation objectives sets the requirements for each test suite, followed by the description of the particular evaluation environment and associated scenarios for each technique under evaluation.

Prototype-based experimentation is documented in section 3.1 and a great part focuses on evaluating the CINA interface. Tests are undertaken in the context of a live streaming application to evaluate the impact of using static and dynamic costs under normal and congested network conditions. Some of the logs obtained from these tests are used to reconstruct video clips that are then subject to user evaluation. Multicast, caching and high capacity node network services are evaluated for live and interactive media applications, also integrating the use of multi-link enabled peers. Tests are undertaken in the context of a CDN infrastructure, and also using the CINA interface as an integral part of the content-aware network functions developed by the COMET project.

Section 3.2 elaborates on the results from simulations and theoretical studies, complementing the prototype-based experimentation with results from diverse, large scale scenarios of complex application setups, network topologies and traffic conditions. These results cover several cross-layer optimisation techniques including the calculation and consolidation of costs reflecting the ISP preferences, the calculation of costs reflecting load-balancing rather than localisation objectives, the selection of high capacity nodes satisfying application-layer delay constraints and minimising the associated costs, and finally the overlay topology construction and data request scheduling with the objective of achieving higher and smoother video stream quality.

Chapter 4 concludes this document, highlighting some of the most outstanding results of the project.

TABLE OF CONTENTS

EXECUTIVE SUMMARY	2
TABLE OF CONTENTS	3
LIST OF FIGURES	5
1. INTRODUCTION	7
2. UPDATE ON THE TESTBEDS	8
2.1 FT Testbed	8
2.2 ALUD Testbed	9
2.2.1 Testbed Specification.....	9
2.3 TID Testbed.....	10
2.3.1 Testbed Specification.....	10
2.4 LiveU Testbed	11
2.4.1 Testbed Specification.....	11
3. EVALUATION RESULTS	13
3.1 Prototype Evaluation	13
3.1.1 <i>Live Content Distribution and Content Adaptation, Objective Evaluation</i>	13
3.1.1.1 Objectives and Evaluation Requirements.....	13
3.1.1.2 Evaluation Scenarios and Metrics.....	13
3.1.1.3 Evaluation Results	15
3.1.1.4 Conclusions.....	19
3.1.2 <i>Live Content Distribution and Content Adaptation, User Evaluation</i>	20
3.1.2.1 Objectives and Evaluation Requirements.....	20
3.1.2.2 Evaluation Scenarios and Metrics.....	20
3.1.2.3 Evaluation Results	21
3.1.2.4 Conclusions.....	22
3.1.3 <i>ENVISION-COMET Integration</i>	22
3.1.3.1 Objectives and Evaluation Requirements.....	23
3.1.3.2 Evaluation Scenario	23
3.1.3.3 Evaluation Results	26
3.1.3.4 Conclusions.....	28
3.1.4 <i>Content Distribution via IVCD and Multi-link enabled peers</i>	29
3.1.4.1 Objectives and Evaluation Requirements.....	29
3.1.4.2 Evaluation Environment	29
3.1.4.3 Performance Metrics.....	30
3.1.4.4 Controlled and Uncontrolled Variables	30
3.1.4.5 Evaluation Scenarios.....	31
3.1.4.6 Evaluation Results	32
3.1.4.7 Conclusions.....	39
3.1.5 <i>Multicast Network Service</i>	39
3.1.5.1 Objectives and Evaluation Requirements.....	39
3.1.5.2 Evaluation Scenarios and Metrics.....	40
3.1.5.3 Evaluation Results	42
3.1.5.4 Conclusions.....	46
3.1.6 <i>Caching</i>	46
3.1.6.1 Objectives and Evaluation Requirements.....	46
3.1.6.2 Evaluation Scenarios & Metrics	46
3.1.6.3 Test platform	47
3.1.6.4 Evaluation Results	47
3.1.6.5 Conclusions.....	53
3.1.7 <i>CDN Request Routing</i>	53
3.1.7.1 Evaluation Scenarios & Metrics	53
3.1.7.2 Evaluation Results	55
3.1.7.3 Conclusions.....	59
3.2 Simulations and Theoretical Analysis	60

3.2.1	<i>ISP Preferences</i>	60
3.2.1.1	Objectives and Evaluation Requirements.....	60
3.2.1.2	Conclusions.....	60
3.2.2	<i>ISP Preference Consolidation</i>	61
3.2.2.1	Objectives and Evaluation Requirements.....	61
3.2.2.2	Evaluation Scenarios & Metrics	61
3.2.2.3	Evaluation Results	63
3.2.2.4	Conclusions.....	67
3.2.3	<i>Distributed Data Management System</i>	67
3.2.3.1	Objectives and Evaluation Requirements.....	67
3.2.3.2	Conclusions.....	67
3.2.4	<i>Distribution Tree Optimisation</i>	68
3.2.4.1	Objectives and Evaluation Requirements.....	68
3.2.4.2	Conclusions.....	68
3.2.5	<i>Content Adaptation</i>	68
3.2.5.1	Smoothing	68
3.2.5.2	Conclusions.....	73
3.2.6	<i>Layered-video Quality Bottleneck in P2P overlays</i>	74
3.2.6.1	Objectives.....	74
3.2.6.2	Evaluation Scenarios & Metrics	74
3.2.6.3	Conclusions.....	75
3.2.7	<i>CINA for CDN</i>	75
3.2.7.1	Conclusion	76
3.2.8	<i>TCP-friendly Rate Control and FEC over Multiple Links</i>	77
3.2.8.1	Objectives and Evaluation Requirements.....	77
3.2.8.2	Conclusions.....	77
4.	CONCLUSIONS	78
	REFERENCES	80

LIST OF FIGURES

Figure 1: FT Testbed	8
Figure 2: ALU Testbed configuration	9
Figure 3: Simplified CDN workflow.....	11
Figure 4: ML Laboratory testbed.....	12
Figure 5: Live Content Distribution – Overlay Topology Examples	16
Figure 6: Live Content Distribution – Missed Frames Summary Results.....	18
Figure 7: Live Content Distribution – Stream Quality Summary Results.....	18
Figure 8: Live Content Distribution – Stream Liveness Summary Results.....	19
Figure 9: Live Content Distribution – Network Load Summary Results	19
Figure 10: Mean Opinion Score – No delay, no loss.....	21
Figure 11: Mean Opinion Score – 10% errors, no delay.....	22
Figure 12: Mean Opinion Score – 500ms delay, no errors.....	22
Figure 13: Mean Opinion Score – 1000ms delay, no errors.....	22
Figure 14: COMET-ENVISION Layout.....	24
Figure 15: Behaviour without CINA Activated.....	25
Figure 16: Behaviour with CINA Activated	26
Figure 17: Ranking Assignment with CINA deactivated	26
Figure 18: Traffic in CAFEs with CINA deactivated	27
Figure 19: Network MAP retrieved from CINA Server	27
Figure 20: Ranking Assignment with CINA activated	28
Figure 21: Traffic in CAFEs with CINA activated	28
Figure 22: IVCD functional evaluation integrating three consumer peers, the source peer and LiveU equipment providing HD video content upload.....	32
Figure 23: On-line visualisation of distribution tree topology for live HD-video distribution.	33
Figure 24: Evaluation of Direct link versus regular internet link	33
Figure 25: Packet loss as a function of time for the regular link.....	34
Figure 26: Packet loss as a function of time for the direct link.....	34
Figure 27: Packet delays for the regular link.....	35
Figure 28: Packet delays for the direct link	35
Figure 29: Comparing the PSNR of two different types of movies - one with relatively high motion and one with low motion under different resolutions for a range of bitrate values.....	36
Figure 30: PSNR as a function of time for a periodically changing channel bit rate at several resolutions. The dashed black line shows the maximal PSNR, obtained when the resolution is adaptively changed according to the channel bit rate.....	36

Figure 31: PSNR as a function of time for a Gaussian randomly changing channel bit rate at several resolutions. The dashed black line shows the maximal PSNR, obtained when the resolution is adaptively changed according to the channel bit rate..... 37

Figure 32: Video bit rate vs. time, with resolution switching points, 1st algorithm implementation .. 38

Figure 33: Video bit rate vs. time, with resolution switching points, 2nd algorithm implementation. 38

Figure 34: Video bit rate vs. time, with resolution switching points, 3rd algorithm implementation . 39

Figure 35: Tracker CLI to switch a set of peers to multicast 40

Figure 36: Peer receiving traffic in unicast..... 40

Figure 37: Peer receiving traffic in multicast..... 41

Figure 38: Multicast activation delay 43

Figure 39: traffic rate on the peering link 43

Figure 40: content flows in Test1 (unicast only) 44

Figure 41: content flows in Test1 (with multicast)..... 45

Figure 42: total bandwidth usage in ISP1..... 45

Figure 43: Volume saving with caching..... 51

Figure 44: CDN testing scenario 54

Figure 45: response time with optimized topology 56

Figure 46: response time with random topology..... 56

Figure 47: system throughput with optimized topology 57

Figure 48: system throughput with random topology 57

Figure 49: buffering time (CDF) with optimized topology..... 58

Figure 50: buffering time (CDF) with random topology 58

Figure 51: underflow rate with random topology..... 59

Figure 52: Topology Construction Scenarios..... 63

Figure 53: Simulation results for $s = \{.1,.5,1,1.5\}$ 65

Figure 54: Overlay-wide simulation results for $.1 \leq s \leq 1.5$ 66

Figure 55: Bandwidth variation 69

Figure 56: Row stream 69

Figure 57: Amplitude reduction 69

Figure 58: Frequency reduction (prefetching-based) 69

Figure 59: Frequency reduction (mean-based)..... 69

Figure 60: 1st scenario analysis..... 70

Figure 61: Impact of smoothing window size on number of layers changes..... 72

Figure 62: Impact of smoothing window size on number of unused chunks 72

Figure 63: Impact of smoothing window size on number of longest sequence 73

Figure 64: Number of Layer Changes 73

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 system requirements.

This deliverable documents the results of this evaluation. In chapter 2 we describe the infrastructure and the configurations established to enable experimentation in the ENVISION testbeds. Chapter 3 documents the project test specification and presents the corresponding results. Finally, chapter 4 summarises the main evaluation results.

2. UPDATE ON THE TESTBEDS

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

Only few updates have been made to the FT testbed described in [D6.1].

The FT testbed relies on an experimental network in Orange Labs, shared between several Orange projects, representative of a commercial network. The network comprises a rich set of network equipment such as DSLAM, BRAS, core and edge routers, wireless access points and 3G equipment, spread over several locations.

The Envision testbed emulates three different ISPs. Two ISPs are divided in different regions (named PIDs, to be compliant to the ALTO terminology). ISP1 has 2 PIDs located in Lannion (the 2 on the right-hand side in Figure 1) and 1 located in Paris (compared to [D6.1] the 3G access in PID3 has been replaced by a broadband wireline access).

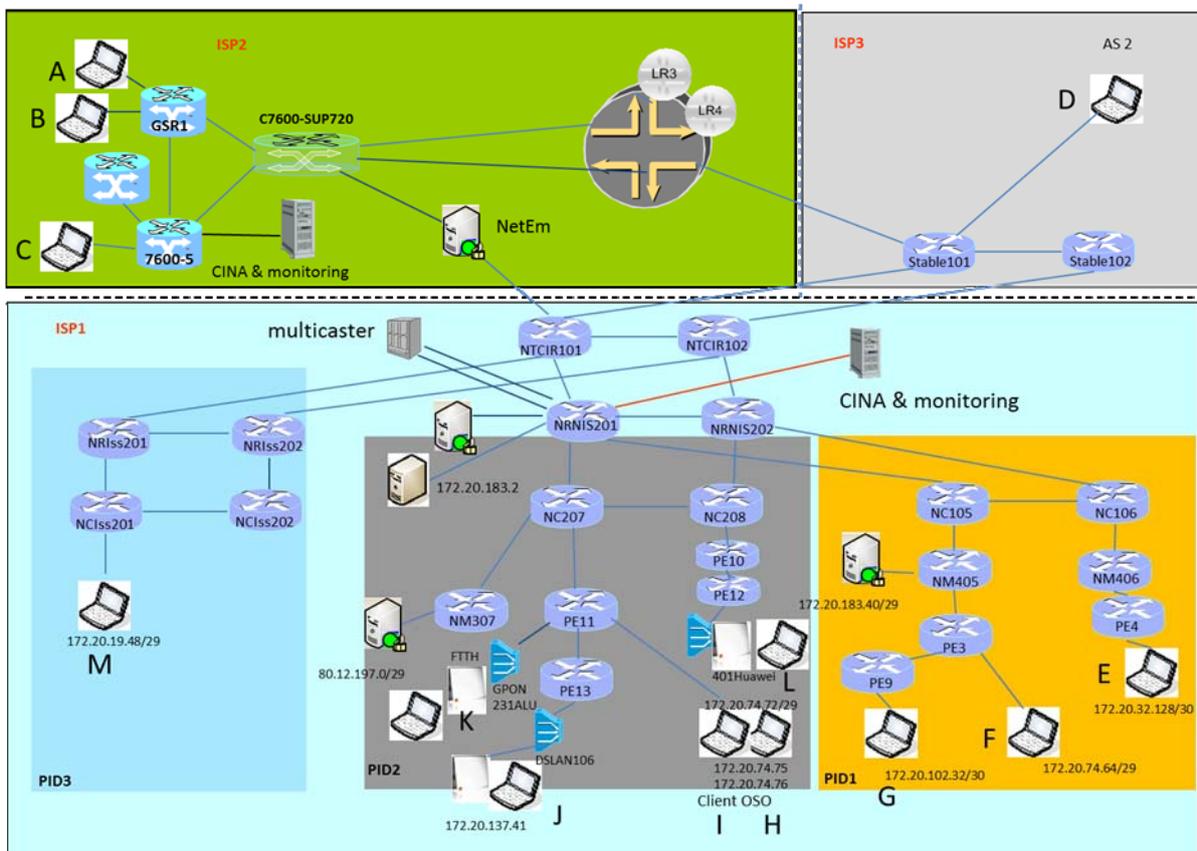


Figure 1: FT Testbed

Two CINA servers and the associated monitoring tools have been deployed, one in ISP1 and the other in ISP2. A multicaster and some network caches have been deployed in ISP1. Priorities (lower cost) are defined so that clients in Lannion preferentially connect to other clients in Lannion. For cross-ISP

traffic, a lower cost is defined 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 involve increased delay between ISP2 and ISP1 such as testing the delay metric. The Linux function NetEm is used to simulate delay (and packet loss) on the peering link between the two ISPs.

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 that has performed with accordance 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.

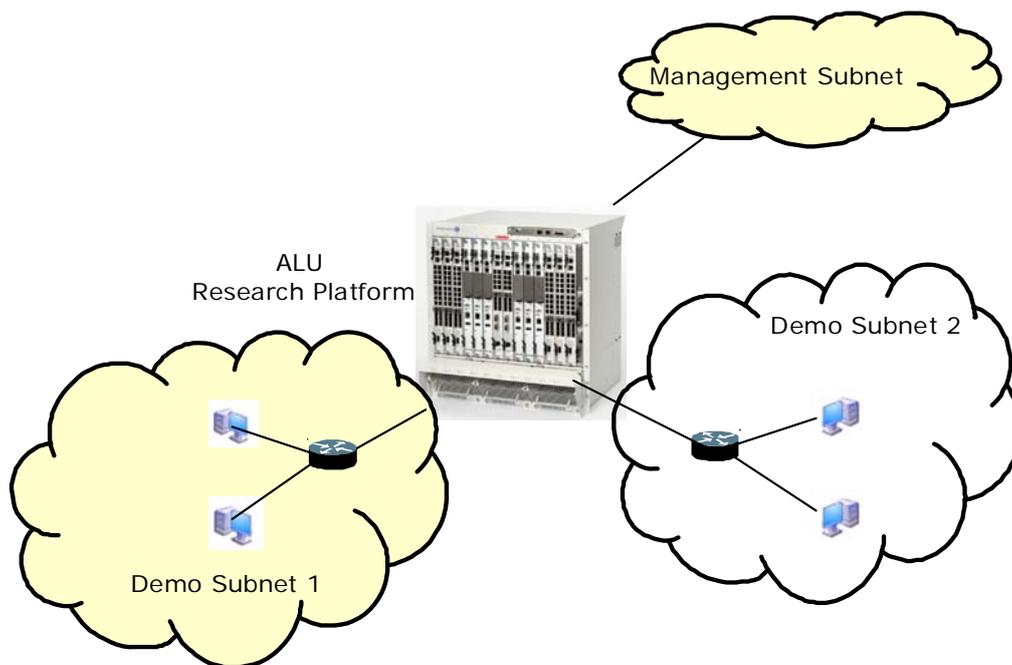


Figure 2: ALU Testbed configuration

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.

2.3 TID Testbed

2.3.1 Testbed Specification

It is first necessary to know about the main components of the CDN. This previous knowledge is helping us in understanding the test bed and the way ENVISION technology can 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 3 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 were 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.

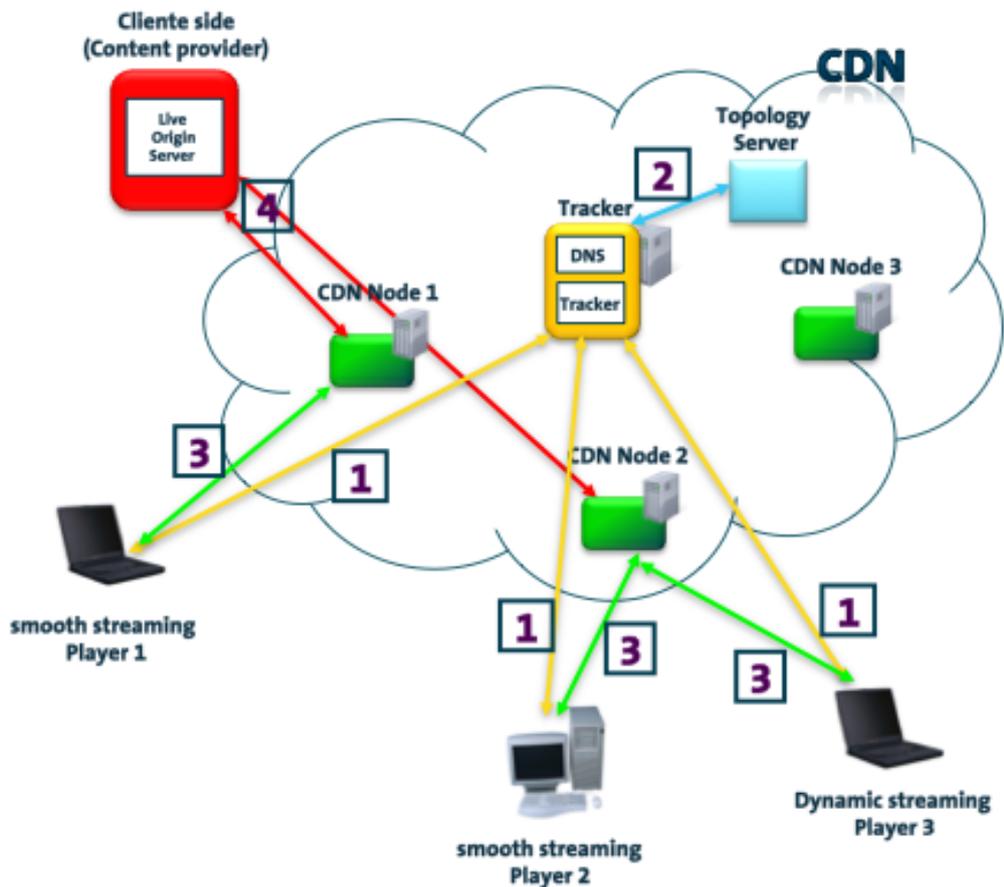


Figure 3: Simplified CDN workflow

2.4 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.

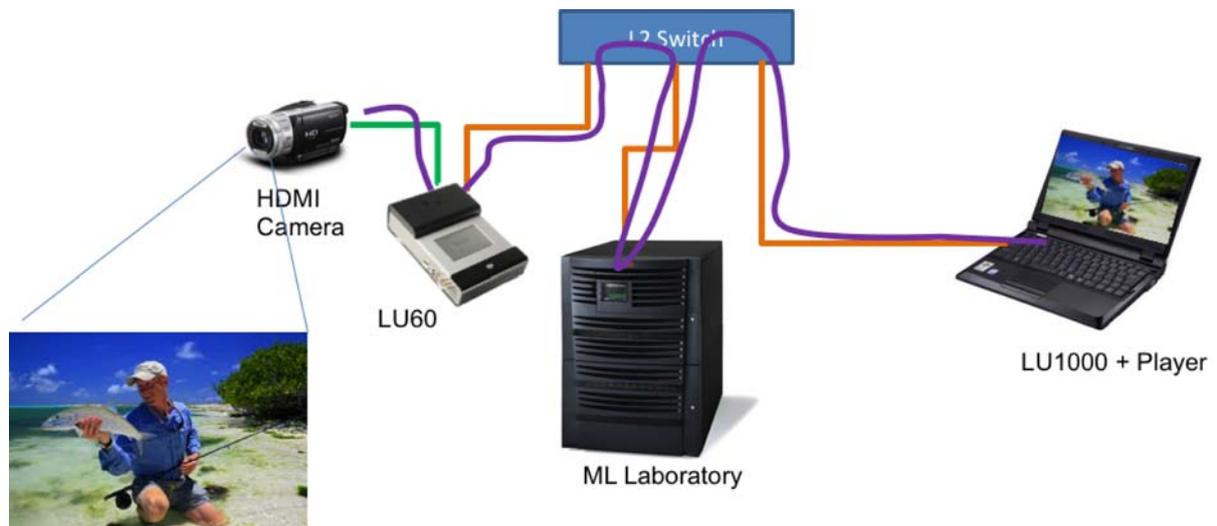


Figure 4: ML Laboratory testbed

3. EVALUATION RESULTS

3.1 Prototype Evaluation

3.1.1 Live Content Distribution and Content Adaptation, Objective Evaluation

The specifications of this system are provided in detail in section 4 of [D4.2] and section 4 of [D5.3].

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 static routing costs and dynamic network performance costs.
- Functional validation of the content adaptation function for the selection of SVC layers according to the achieved throughput over all overlay links.
- In the scale permissible in the testbed, evaluate the improvement in application performance and network utilisation achieved with the use of CINA static and dynamic costs. Generate the clips for the subjective user evaluation of the content adaptation function (see section 3.1.2).

3.1.1.2 Evaluation Scenarios and Metrics

3.1.1.2.1 Evaluation Environment

These tests are performed in the FT testbed (see section 2.1).

The software prototype integrating live content distribution and content adaptation functions is deployed in eleven terminals in total, seven in ISP1 (E, F, G, H, J, K, L), three in ISP2 (A, B, C) and one in ISP3 (D). Each peer reports logs including information on the frame number and the SVC layer that was successfully received on time for playout, shifts in the playout point with changes in network delay or changes in the liveness of the sender peers, etc. An overlay tracker maintaining the list of peers participating in the application overlay is deployed in ISP1. The tracker reports logs including information about the active overlay connections used to actually transmit data, and the stream liveness for each peer. This is calculated based on updates the peers send to the tracker every five seconds, reporting their current playout point. The difference between a peer's playout point and the playout point reported by the stream source peer is the peer's liveness. Each run for a given configuration setting lasts five minutes.

Peer A in ISP2 is always used as the content source peer. Each peer is configured to provide a certain upload capacity to the overlay. This configured upload capacity is taken into account when establishing overlay connections but it is not enforced at the data transmission scheduling phase. Two values are used for upload capacity, 0 denoting leeching peers and 1.5 Mbps for all non leeching peers. The file that was used to emulate a live stream is encoded to a stream with 662 kbps bitrate in total, split over 25 SVC layers.

The overlay tracker implements the CINA client and is interacting with the CINA servers provided for each ISP domain. The CINA static costs are configured based on the network routing costs and are

lower for the connections between ISP2 and ISP1 than for the connections between ISP2 and ISP3 or ISP3 and ISP1. The CINA dynamic costs reflect the delay measured in the network. Each ISP reports the delay within its domain and the overlay tracker adds all the costs to provide the dynamic cost consolidated view. When no congestion is present, these end-to-end costs are also lower for the connections between ISP2 and ISP1 than between ISP2 and ISP3 or ISP3 and ISP1. The dynamic costs, however, are dynamically updated when congestion is introduced at the inter-domain link using the Network Emulator tool on the link connecting ISP2 and ISP1.

3.1.1.2.2 Performance Metrics

- *stream liveness*: The time difference between the playout point at a consumer and the time a chunk is produced at the source.
- *missed frames*: The percentage of frames that were not received or were received after their playout point over the total number of frames.
- *stream quality*: The spatial, temporal and SNR resolution of the stream (i.e. the SVC layer) a user receives for each frame.
- *network load*: The average total (incoming and outgoing) traffic rate in kbps observed over the ISP2 to ISP1 inter-domain link.

3.1.1.2.3 Controlled and Uncontrolled Variables

Controlled variables:

- *use of CINA static costs*: The overlay topology construction algorithm may establish overlay connections randomly (*Random* policy), or taking into account the CINA static costs associated with the topology and remaining unchanged in the event of congestion (*RoutingCost*, *RoutingCostLastHop* policies).
- *use of CINA dynamic costs*: The overlay content adaptation functions may respond to increased delay or decreased throughput caused in the event of congestion over an overlay link taking into account the CINA dynamic costs (*Latency* and *LatencyLastHop* policies).
- *increasing overlay versus network performance*: The overlay topology construction algorithm may establish overlay connections using two different criteria: a) increase the overlay performance by connecting more peers to remote peers with good liveness (*RoutingCost* and *Latency* policies), or b) increase the network performance by reducing the number of connections to remote peers over paths with high CINA costs (*RoutingCostLastHop* and *LatencyLastHop* policies).

Uncontrolled variables:

- *overlay link network delay*: The delay between any two overlay nodes in the testbed can vary. An additional delay of 500 milliseconds is introduced for specific connections to emulate network congestion in order to evaluate the benefits achieved using the CINA dynamic costs.
- *overlay link throughput*: The throughput achieved between any two overlay nodes in the testbed may vary. An artificial loss rate of 5% is introduced for specific connections to emulate network congestion in order to evaluate the benefits achieved using CINA dynamic costs.

3.1.1.2.4 Evaluation Scenarios

The live content distribution and content adaptation evaluation scenarios are captured in Table 1. For the objective evaluation results presented in this section, peers H has been inactive, while peers J, K, and L were configured as leechers.

Description	Metrics	Controlled Variables	Uncontrolled Variables
Performance Enhancement at Normal Network Conditions: Evaluate the impact on performance with and without the use of CINA costs; Evaluate the impact on overlay versus network performance using different policies.	stream liveness, stream quality and missed frames, network load	<i>Random</i> policy (without CINA costs) <i>Latency</i> and <i>RoutingCost</i> policies (with CINA costs, increase overlay performance) <i>LatencyLastHop</i> and <i>RoutingCostLastHop</i> policies (with CINA costs, increase network performance)	No delay and no loss induced at inter-domain link.
Performance Enhancement at the Event of Delays over a Congested Link: Evaluate the impact on performance with and without the use of CINA costs; Evaluate the impact on overlay versus network performance using different policies.	as above	as above	No loss and 500 milliseconds induced at inter-domain link.

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

3.1.1.3 Evaluation Results

3.1.1.3.1 Basic Operation

This section elaborates on the basic operation of the system, in order to provide the necessary background for the analysis of the performance results.

Figure 5 shows the overlay topologies produced using random and static CINA costs under normal network conditions. These topologies show the active overlay links, i.e. the connections used to transmit data as opposed to just connections that were established but were not chosen to carry data traffic by the data request algorithms at the peers.

As the CINA cost between ISP2 and ISP3 is higher than the cost between ISP2 and ISP1, peers in ISP2 including the source peer A will never prefer to connect to peer D in ISP3 when the CINA costs are taking into consideration (b, c), which is not the case for the random policy depicted in (a).

The effect of the CINA costs in the topology is more pronounced when the *RoutingCostLastHop* policy is used. With this policy, each peer selects its senders taking into account only the CINA cost of the connection between them (last hop), as opposed to the corresponding *RoutingCost* policy where

the cumulative CINA costs across the full path from the peer to the stream source over a sender is taken into account.

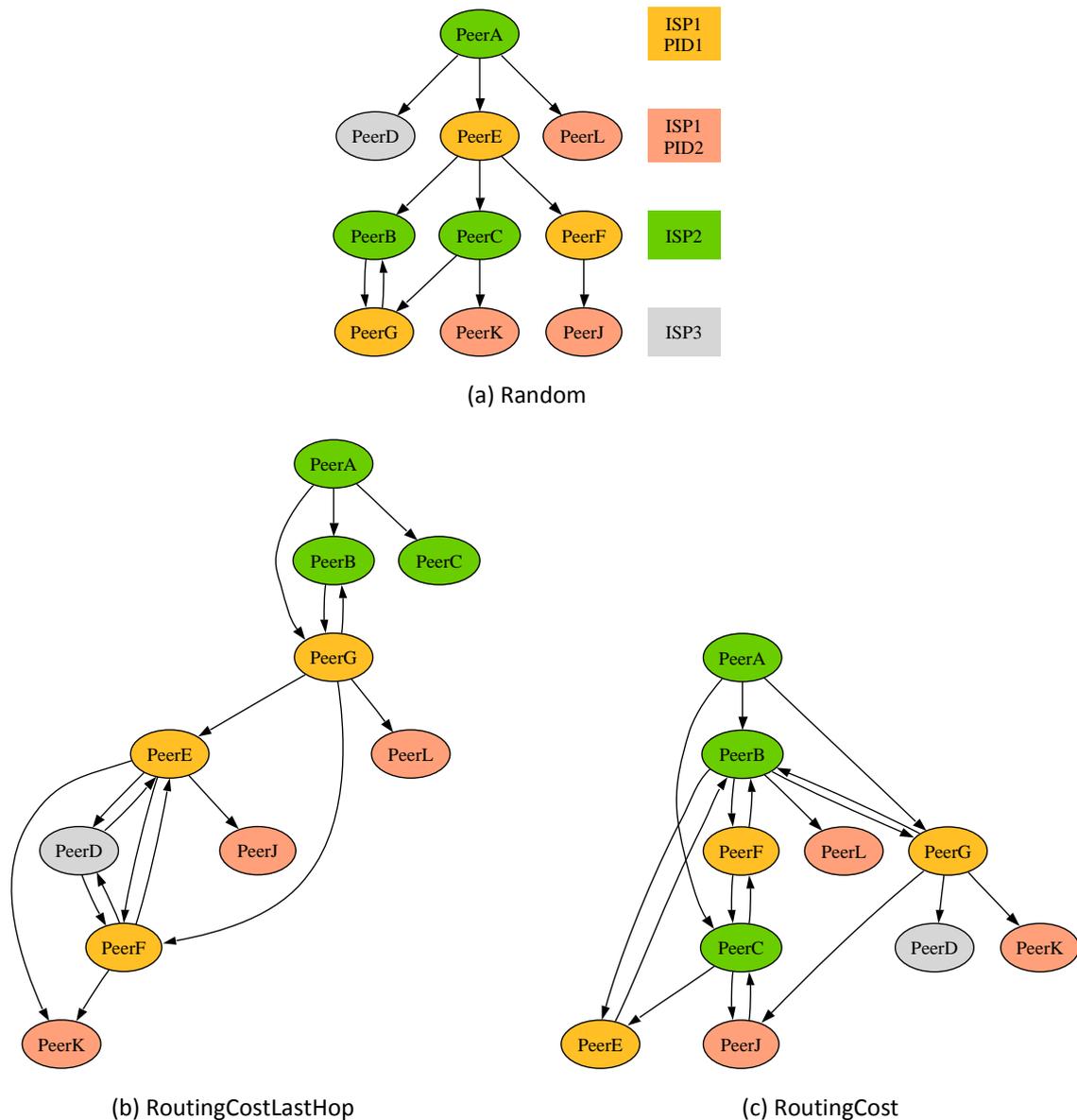


Figure 5: Live Content Distribution – Overlay Topology Examples

With RoutingCostLastHop the connections between ISP2 and ISP1 are reduced to the ones required to feed a single peer in ISP1 (peer G) with the entire stream. Once one peer in ISP1 has the entire stream, then all other peers will prefer to connect to this peer over the very low cost intra-domain connections rather than going over the higher cost inter-domain connection to ISP2. In contrast, the RoutingCost policy may rank higher connections with high cost over connections with lower cost when the corresponding neighbour peers are "closer" to the content source, i.e. connected to the content source over fewer and/or less costly connections. Indeed, the RoutingCost topology includes as many connections to the peers in ISP2 (peers B and C close to the content source peer A), that are feasible given their upload capacity. Only when the capacity of these more live peers is exhausted, new receiving peers will attempt to connect to peers further down the overlay content distribution paths.

The motivation behind introducing the distinction between the last hop and the full path policies was to experiment with two extreme policies, one where the network performance is the dominant optimisation criterion, and another where the overlay performance is optimised instead. The resulting overlay and network performance trade-offs with these two policy flavours are discussed in the following sections.

3.1.1.3.2 Performance Under Normal Network Conditions

The overlay performance results can be seen in Figure 6, Figure 7 and Figure 8 depicting the missed frames, stream quality (number of SVC layer) and the playout lag mean values for each frame received across all the peers under different overlay topology creation policies.

The stream quality appears to be independent of the particular topology in these experiments. This is hardly surprising given the absence of any congestion in the network. The missed frames percentage presents some differences between different policies, with the *Random* topology to get the highest percentage, followed by the *Latency* and *RoutingCost* policies. Higher variance in chunk arrival times for these two policies can be justified by the fact that peers tend to have higher out-degrees serving more receivers which introduces additional delays in their transmission queues.

The stream liveness also varies for different topologies but only slightly, which can be explained given the small number of peers participating in the experiment and the negligible end-to-end network delays in the testbed. The policy producing the best performance is *RoutingCost*, closely followed by *Latency*. The similarity between dynamic and static costs in these experiments can be explained by the fact that at low utilisation the latency across the network approximates the propagation delay, which is determined by the number of hops in a path, which in turn determines the values of the static CINA routing costs.

It is expected for these two policies to outperform the others in the stream liveness metric as they are built to reduce the cumulative cost along the overlay distribution paths. However, as it can be seen in Figure 9, prioritising the overlay utility increases the inter-domain traffic compared to *LastHop* policies, which achieve lowering the load by 75% in the case of *RoutingCostLastHop* and 72% in the case of the *LatencyLastHop* tests. It is also worth noting that all the CINA policies reduce the inter-domain network load compared to the *Random* test, by 11%, 25%, 77% and 79% for *RoutingCost*, *Latency*, *RoutingCostLastHop* and *LatencyLastHop* respectively.

3.1.1.3.3 Performance at the Event of Congestion

This test suite evaluates the system performance under the event of congestion in the network, and in particular in the event of 500ms delay over the inter-domain link. Looking at Figure 9 it becomes apparent that the policies that use the dynamic CINA costs react to the increased delay by shifting traffic away from the congested inter-domain link.

The other most striking effect of congestion is the increased number of missed frames compared to normal conditions, which is particularly severe for *RoutingCost* which heavily uses the congested link to receive data from different senders. This effect is significantly reduced for its equivalent *RoutingCostLastHop* policy that reduces the number of connections over inter-domain links.

Overall, the best performing policy is *LatencyLastHop* which maintains high quality in terms of low number of missed frames and high quality level for the received ones, while also outperforms all other policies in terms of stream liveness by routing around the congested link for the established overlay connections. It is worth noting that the improvement in stream liveness even in this limited setup is 30% compared to the *Random* overlay topology.

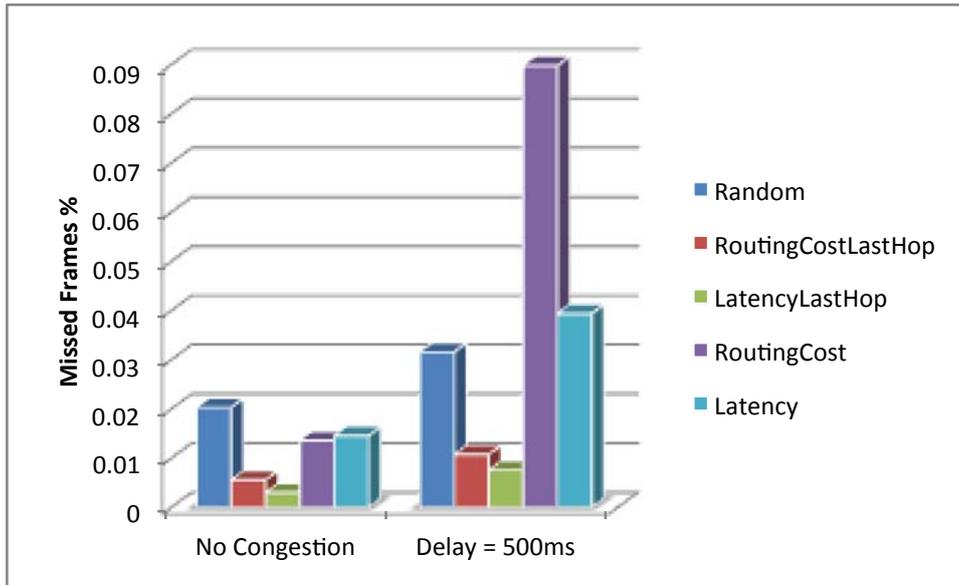


Figure 6: Live Content Distribution – Missed Frames Summary Results

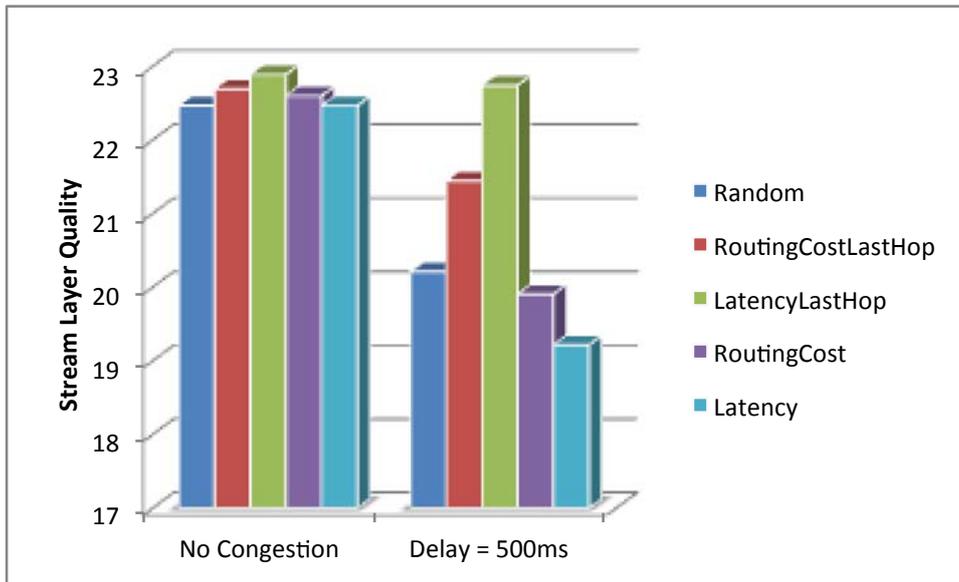


Figure 7: Live Content Distribution – Stream Quality Summary Results

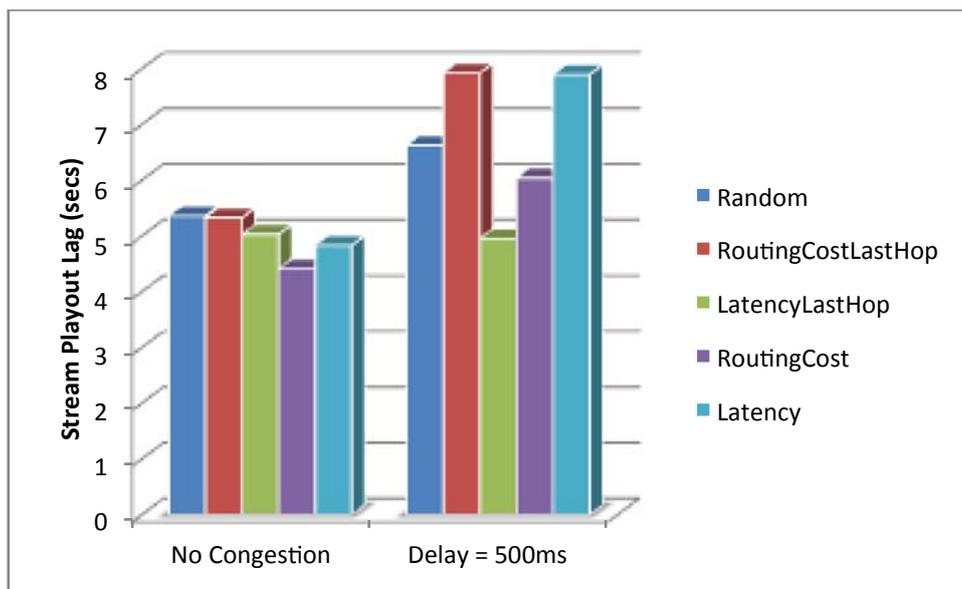


Figure 8: Live Content Distribution – Stream Liveness Summary Results

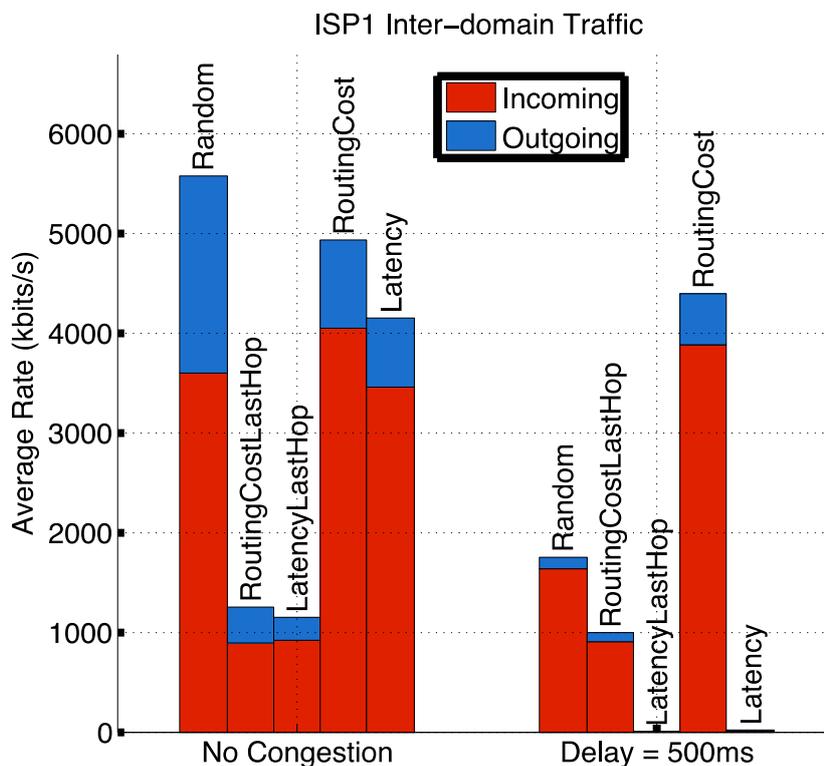


Figure 9: Live Content Distribution – Network Load Summary Results

3.1.1.4 Conclusions

A large set of prototype tests were performed in the FT testbed with a focus on evaluating in a realistic environment the benefits of the CINA interface from the perspective of both the network operators and the video streaming overlay applications. It has been shown that overlay topology construction policies that use CINA costs as their primary optimisation objective can reduce the load over inter-domain links for up to 79%, while at the same time improving the application performance comparing to random topologies. Further, at the event of congestion, policies that react to dynamic

CINA costs outperform random and static cost policies, significantly improving the stream quality and reducing the playout lag for live streaming applications.

3.1.2 Live Content Distribution and Content Adaptation, User Evaluation

The specifications of the user evaluation scenario can be found in detail in section 5 of [D5.3].

3.1.2.1 Objectives and Evaluation Requirements

- Subjective user evaluation of the content distribution and adaptation functions under different network conditions.
- Subjective user evaluation of the overlay construction mechanisms using CINA costs.
- Subjective user evaluation of the smoothing techniques

3.1.2.2 Evaluation Scenarios and Metrics

3.1.2.2.1 Evaluation environment

In order to permit subjective user evaluation, the streaming application was run under different network and overlay scenarios as described in section 3.1.1.2.4. For each network-overlay scenario logs are collected in order to reconstruct the viewed video clip at each receiving peer. New videos clips were also reconstructed using the proposed smoothing techniques a posteriori.

Evaluation was made using the Double Stimulus Impairment Scale (DSIS) method. In this method, the original video is shown for 10 seconds, then the evaluated video for the same 10 seconds. This process was repeated for a total duration of 30 seconds of the video. The scores were expressed over 10, with 0 being the worst score and 10 being the best score.

3.1.2.2.2 Network and Overlay Scenarios

The live content distribution and content adaptation evaluation scenarios are captured in Table 2. For the subjective evaluation results presented in this section, all eleven peers are active and they participate with the same upload capacity of 1.5Mbps.

Description	Metrics	Controlled Variables	Uncontrolled Variables
Performance Enhancement at Normal Network Conditions	Mean Opinion Score (MOS)	<i>Random, Latency, RoutingCost, LatencyLastHop and RoutingCostLastHop</i> policies	No delay and no loss induced at inter-domain link.
Performance Enhancement at the Event of Loss	Mean Opinion Score (MOS)	<i>Random and Latency</i> policies	No delay and 10% loss induced at inter-domain link.
Performance Enhancement at the Event of Delays	Mean Opinion Score (MOS)	<i>Random, Latency, RoutingCost, LatencyLastHop and RoutingCostLastHop</i> policies	No loss and 500 milliseconds delay induced at inter-domain

			link.
	Mean Opinion Score (MOS)	<i>Random</i> and <i>RoutingCost</i> policies	No loss and 1000 milliseconds delay induced at inter-domain link.

Table 2: Live Content Distribution and Content Adaptation User Evaluation Scenarios

3.1.2.3 Evaluation Results

3.1.2.3.1 Performance enhancements at normal network conditions

At normal network conditions, i.e., no delays and no loss in the network, all the evaluated videos were rated above average for all the metrics used for the overlay construction except those based on "Latency" and "LatencyLastHop". Also, for all the evaluated videos, smoothed videos achieved a higher MOS than videos without smoothing. Figure 10 shows the obtained results.

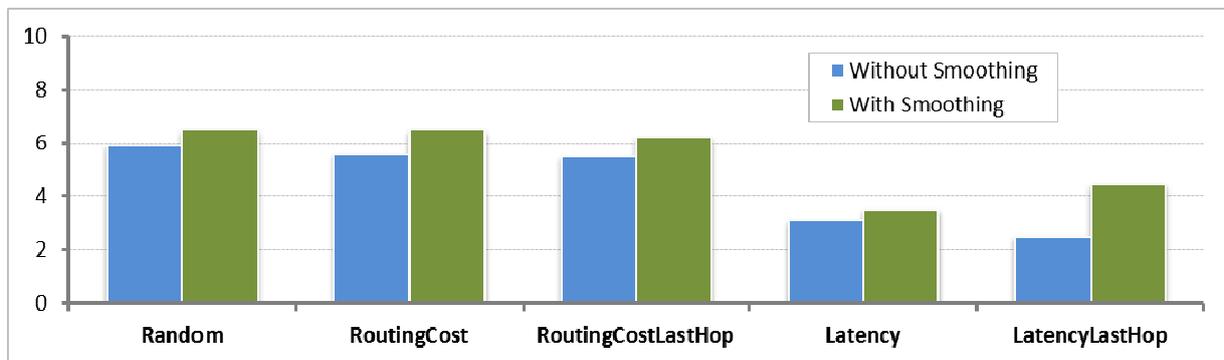


Figure 10: Mean Opinion Score – No delay, no loss

3.1.2.3.2 Performance enhancements at the Event of Loss

At the event of packet loss in the network, a good streaming experience (video quality) has been achieved for topologies constructed using "Random" and "Latency" metrics (> 6). However, we notice that random topology was more efficient in terms of MOS than topology built using "Latency" metric. Moreover, in this case, smoothed videos have not achieved a higher MOS than videos without smoothing. Figure 11 shows the obtained results.

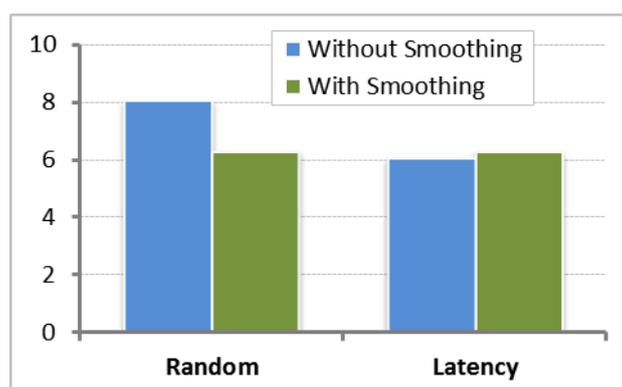


Figure 11: Mean Opinion Score – 10% errors, no delay

3.1.2.3.3 Performance enhancements at the Event of Delays

At the event of delays in the network, the resulting streaming experience was rated above average (>5) in the case of different topologies. Videos from topologies built based on "Latency" and "RoutingCost" achieve the highest MOS (>6). Moreover, in general, smoothed videos are rated slightly higher than videos without smoothing. Figure 12 and Figure 13 show the obtained results.

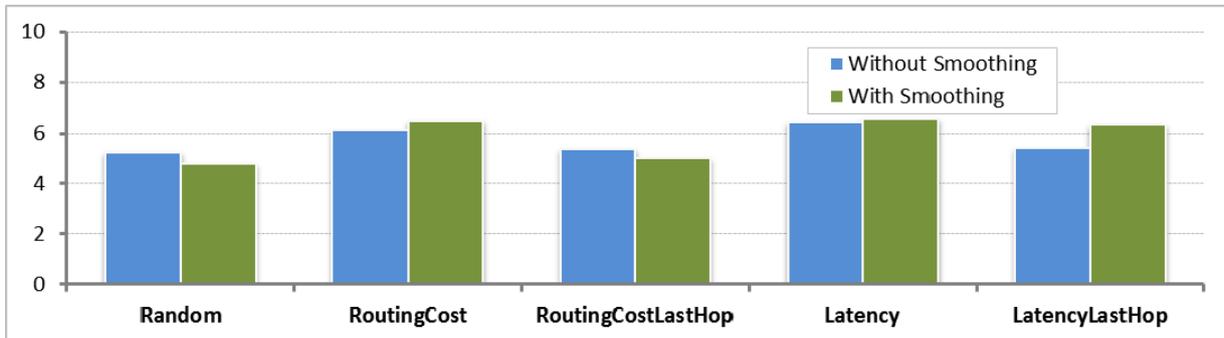


Figure 12: Mean Opinion Score – 500ms delay, no errors

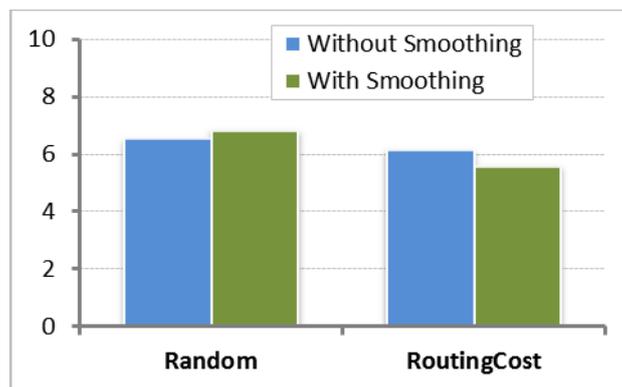


Figure 13: Mean Opinion Score – 1000ms delay, no errors

3.1.2.4 Conclusions

Each client peer receiving the live video stream distributed during the experiments in the FT testbed documented in the previous section (3.1.1) logged with a timestamp each video frame that was successfully received before the playout point deadline. The video clips were reconstructed from these logs using a) no other correction and b) the quality smoothing technique evaluated in detail in section 3.2.5. These clips were evaluated using the Double Stimulus Impairment Scale method by LaBRI students with some basic training in content adaptation. Overall, the Mean Opinion Score for the smoothed video clips was higher. Finally, the analysis of some of the low Mean Opinion Scores attributed to some clips has shown that even the slightest of quality fluctuations (a single missed frame) may have a big impact on the value of the Mean Opinion Scores (drop by two points).

3.1.3 ENVISION-COMET Integration

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.

3.1.3.1 Objectives and Evaluation Requirements

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 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 [D3.3] 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.

3.1.3.2 Evaluation Scenario

The basic Layout of COMET-ENVISION joint scenario is illustrated in Figure 14.

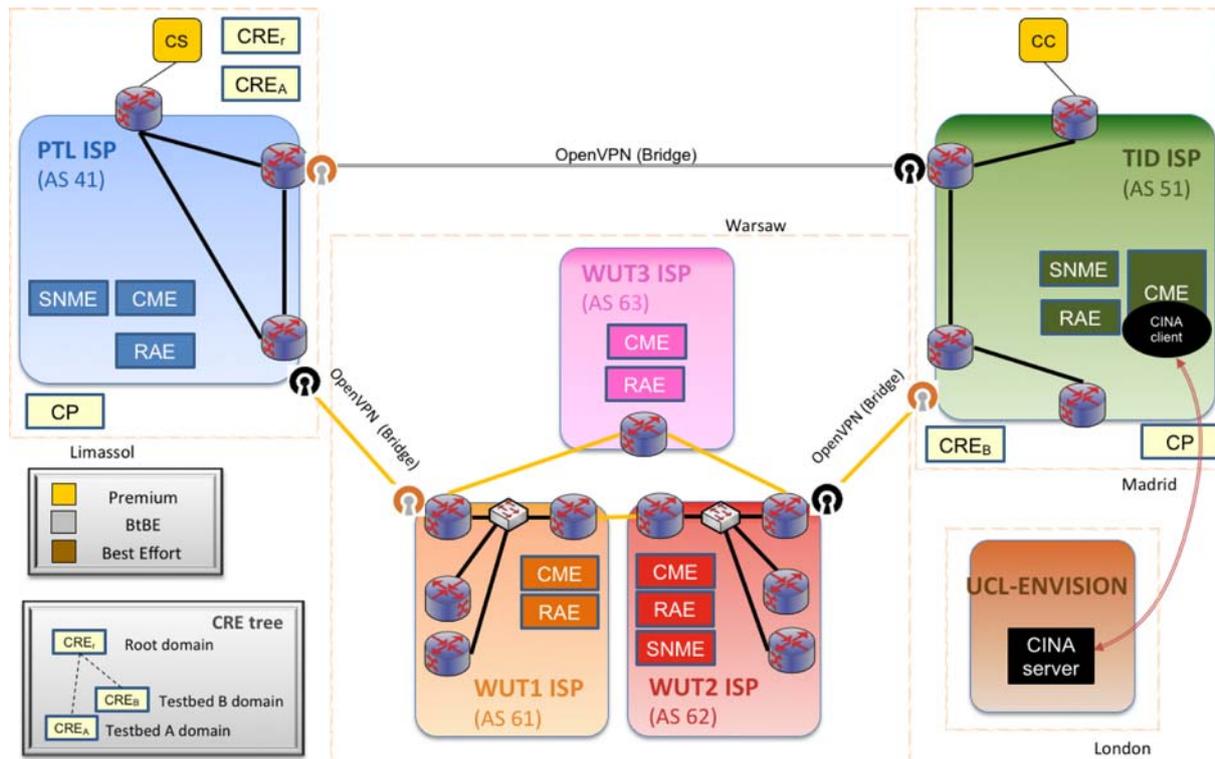


Figure 14: COMET-ENVISION Layout

A CINA server is deployed at UCL premises (London) and is accessible over a public IP address. This CINA server is configured to emulate a CINA server for the TID domain in the COMET federated testbed.

The COMET CME function has been enhanced with a CINA client that can map destination and path cost requests using the next-hop AS numbers to CINA PID cost map requests. This behaviour can be activated/deactivated through the CME Web Interface, thus enabling to compare the behaviour of the decision algorithm with CINA on and CINA off.

When a Content Consumer (CC) at TID requests a content, the CINA-enabled CME requests from CINA the cost of the incoming links from neighbouring ISP, but only for those included in the list of candidate paths for content retrieval. This cost information can then be fed in the CME decision algorithm, together with the other selection parameters including IPTD, IPLR, server Load, Bandwidth and Path Length, modifying the results of the server selection.

For demonstration purposes, the CINA enhanced CME has been only deployed at TID, so tests will involve a CC at TID requesting contents from Content Servers (CSs) located at other ISPs. Therefore, the cost information retrieved by CINA involves only the following links and traffic directions:

- From AS 41 to 51 (contents transmitted from PTL to TID) with an assigned cost of 12.5
- From AS 62 to 53 (contents transmitted from WUT to TID), with an assigned cost of 30

In this context, a greater value means a higher cost, so in general paths using the AS62-51 link will be assigned a worse ranking by the decision algorithm than those using the link AS41-51. The actual ranking can vary according to other configuration parameters of the decision algorithm, and in particular the aspiration level parameter, which defines the importance of a parameter in the computation of the path ranking.

To demonstrate how COMET and ENVISION can interwork, a use case has been defined to show how the decision algorithm can be swayed when the CINA is activated in CME and consequently the cost information for incoming links is fed into the decision algorithm.

For this use case, a CC is deployed at TID and a CS at PTL. When the Content Name is requested and CINA is deactivated, the path chosen by the decision algorithm will be selected solely on the basis of QoS parameters configured to make the long path crossing the WUT ISPs preferable.

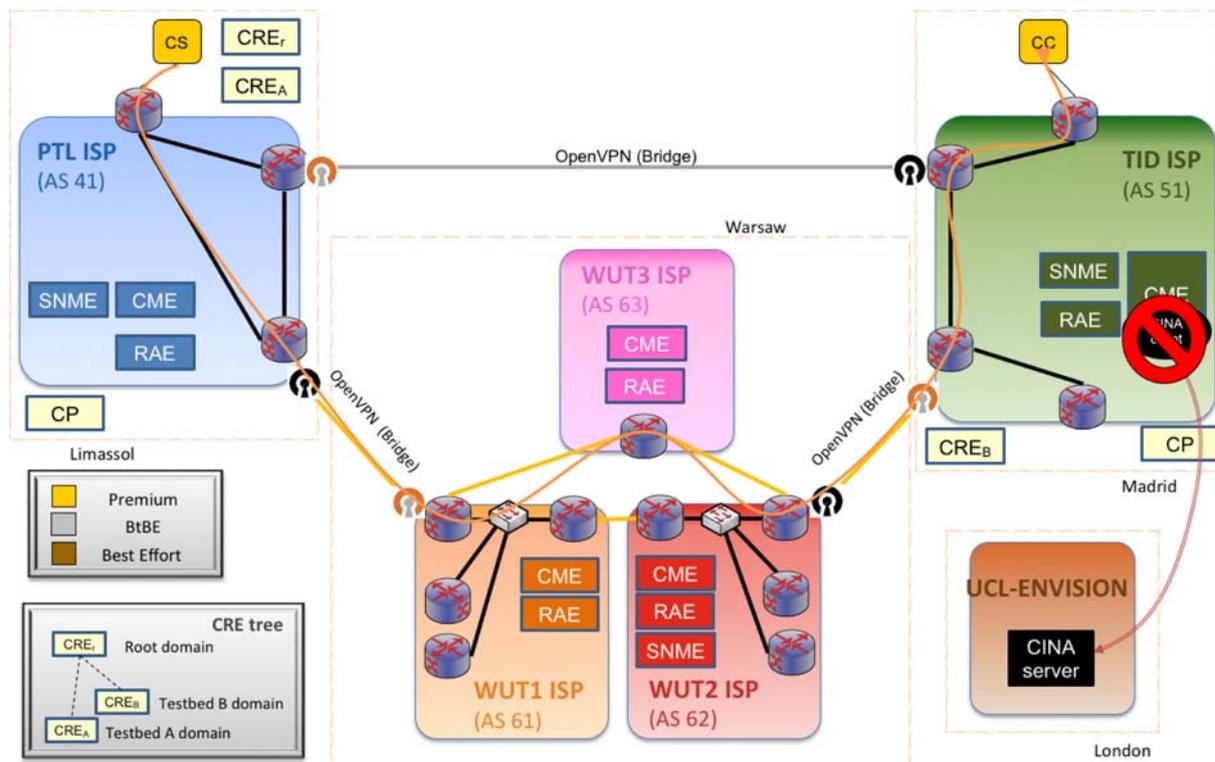


Figure 15: Behaviour without CINA Activated

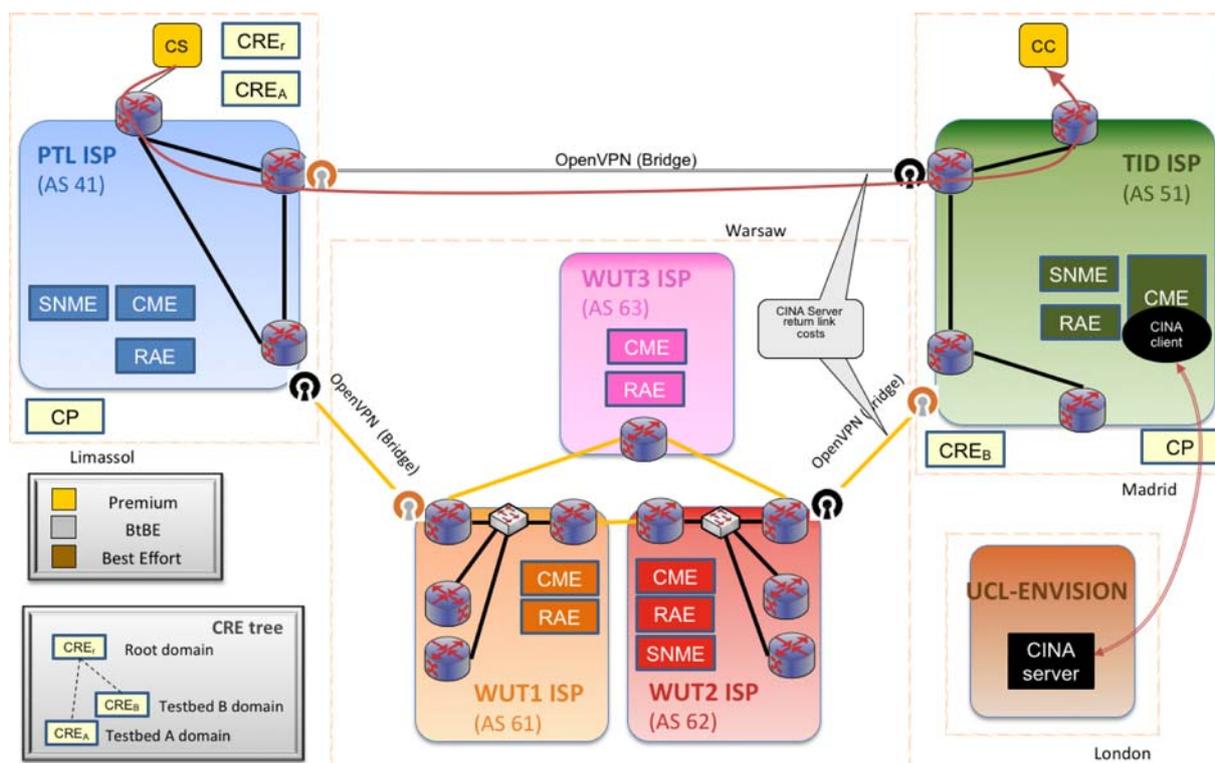


Figure 16: Behaviour with CINA Activated

For demonstrating the ENVISION-COMET interworking capabilities, CINA will be activated in TID CME and the following parameters will be set in the decision algorithm:

- Cost will set to strict mode, so it will reject any path not matching the QoS levels.
- Reservation Level will be set to 60, so the cost defined in the CINA server will be taken into consideration (maximum cost is 30 for TID links).
- The Aspiration Level will be set to 0,01, thus maximizing the contribution of this parameter to the ranking estimation.

So, when the Content Name is requested, COMET will retrieve the cost for the incoming links to TID (AS41 → AS51 and AS62 → AS51) and select the direct link between PTL and TID as the optimal option as illustrated in Figure 16.

Therefore, this use case demonstrates how the inclusion in COMET decision algorithm of the link costs provided by ENVISION can modify the final selected path.

3.1.3.3 Evaluation Results

This section presents the information collected to demonstrate the behaviour of the CME decision algorithm with CINA off or CINA on following the scenario described in the above section.

First, note that in the COMET federated testbed there are three paths from PTL to TID that can be used to download a content:

- AS41 → AS51 (Direct Link)
- AS41 → AS61 → AS62 → AS51 (WUT Path)
- AS41 → AS61 → AS63 → AS62 → AS51 (WUT Path)

With CINA deactivated AS41/AS51 will obtain the worst ranking (0.90707064) compared to the other two that will be assigned a value of (0.91277903). AS41/AS61/AS62/AS51 is eventually selected because it has fewer hops than AS41/AS61/AS63/AS62/AS51. The log from the decision process with the rank assignment is shown in Figure 17.

```

2012-11-21 16:45:11,786 DEBUG DecisionMakerImpl - sload, plength, bw, iptd, iplr, cina: 0.91277903, 79.999695, Infinity, 96.9
9981, 0.90707064, 1.010101
2012-11-21 16:45:11,786 DEBUG DecisionMakerImpl - Server 3101:0:0:0:0:0:3 and path [41, 51] rate=0.90707064
2012-11-21 16:45:11,786 DEBUG DecisionMakerImpl - sload, plength, bw, iptd, iplr, cina: 0.91277903, 49.99981, Infinity, 72.49
9855, 0.9979799, 1.010101
2012-11-21 16:45:11,786 DEBUG DecisionMakerImpl - Server 3101:0:0:0:0:0:3 and path [41, 61, 63, 62, 51] rate=0.91277903
2012-11-21 16:45:11,786 DEBUG DecisionMakerImpl - sload, plength, bw, iptd, iplr, cina: 0.91277903, 59.99977, Infinity, 73.49
985, 1.0020202, 1.010101
2012-11-21 16:45:11,786 DEBUG DecisionMakerImpl - Server 3101:0:0:0:0:0:3 and path [41, 61, 62, 51] rate=0.91277903
2012-11-21 16:45:11,786 DEBUG DecisionMakerUtil - Server 3101:0:0:0:0:0:3 rate=0.91277903
2012-11-21 16:45:11,786 DEBUG DecisionMakerUtil - Server 3101:0:0:0:0:0:3 rate=0.91277903
2012-11-21 16:45:11,787 DEBUG DecisionMakerUtil - Server 3101:0:0:0:0:0:3 rate=0.90707064
    
```

Figure 17: Ranking Assignment with CINA deactivated

By using the CAFE visualisation tool available in COMET, it can also be shown that the traffic is going through CAFEs in TID (TID link to PTL, TID link to WUT) and entering WUT through the border CAFE in AS 62 (WUT Link to TID), as depicted in Figure 18. When CINA is activated and content is retrieved, the cost for links entering TID from PTL and WUT is retrieved from the CINA server at UCL premises. This retrieval operation is shown in the extract of the CME log depicted in Figure 19.

So the paths will be complemented with the following cost information:

- AS41 → AS51 (Direct Link). Cost 12.5

- AS41 → AS61 → AS62 → AS51 (WUT Path). Cost 30
- AS41 → AS61 → AS63 → AS62 → AS51 (WUT Path). Cost 30.

In this case the ranking assigned to AS41/AS51 is 0.799633 as opposed to 0.45454544 for the other two paths. So the selected path with CINA activated is the direct one between PTL and TID. The CME log from the decision process with the rank assignment is shown in Figure 20.

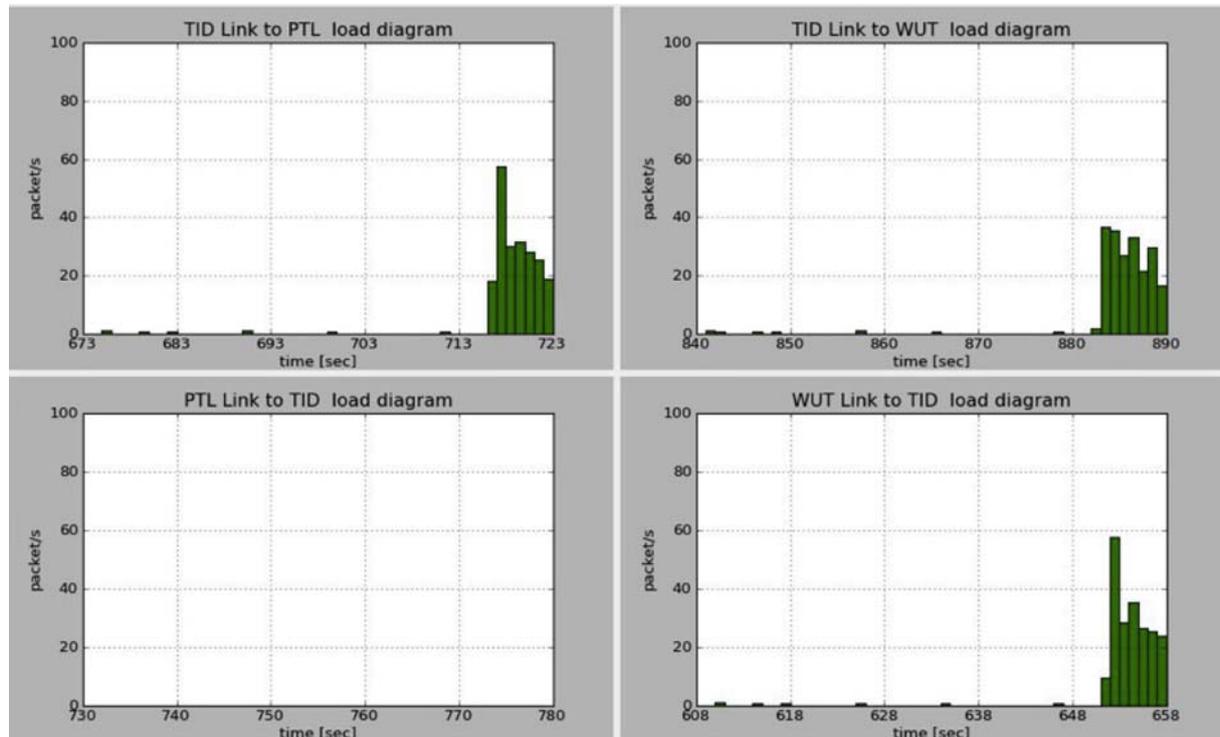


Figure 18: Traffic in CAFEs with CINA deactivated

```

2012-11-21 16:48:38,169 DEBUG PathDiscoveryImpl - Initializing CINA client with URI: http://128.40.39.246/cgi-bin.
2012-11-21 16:48:38,198 DEBUG header - >> "GET /cgi-bin/NetworkMap.cgi HTTP/1.1[\r][\n]"
2012-11-21 16:48:38,198 DEBUG header - >> "ACCEPT: application/cina-networkmap+json,application/cina-error+json[\r][\n]"
2012-11-21 16:48:38,198 DEBUG header - >> "User-Agent: Jakarta Commons-HttpClient/3.1[\r][\n]"
2012-11-21 16:48:38,198 DEBUG header - >> "Host: 128.40.39.246[\r][\n]"
2012-11-21 16:48:38,198 DEBUG header - >> "[\r][\n]"
2012-11-21 16:48:38,328 DEBUG header - << "HTTP/1.1 200 OK[\r][\n]"
2012-11-21 16:48:38,328 DEBUG header - << "HTTP/1.1 200 OK[\r][\n]"
2012-11-21 16:48:38,328 DEBUG header - << "Content-type: application/cina-networkmap+json[\r][\n]"
2012-11-21 16:48:38,328 DEBUG header - << "Content-length: 959[\r][\n]"
2012-11-21 16:48:38,329 DEBUG header - << "Date: Wed, 21 Nov 2012 13:25:56 GMT[\r][\n]"
2012-11-21 16:48:38,329 DEBUG header - << "Server: lighttpd/1.4.28[\r][\n]"
2012-11-21 16:48:38,329 DEBUG header - << "[\r][\n]"
2012-11-21 16:48:38,355 DEBUG content - << "{"data":{"map-vtag":"123456780","map":{"ASTID":{"ipv6":["2A02:9008:0:1911:0:50:56
A3:4F","2A02:9008:0:1919:0:50:56A3:5F","2A02:9008:0:1918:0:50:56A3:5D"],"ipv4":["10.95.51.3","10.95.51.81","10.95.51.85"]},"A
SPT":{"ipv6":["3105::6","2001:7f8:4::8a61:1","2001:7f8:4::3f89:2","2001:7f8:4::1a00:1"],"ipv4":["10.50.50.3","10.50.50.200","
10.50.50.240","10.50.50.250"]},"ASWUT2":{"ipv6":["2001:67c:24cc:31d1:0:0:ae:2","2001:67c:24cc:31d0::cafe:c","2001:67c:24cc:31
d0::cafe:a","2001:67c:24cc:31d0::cafe:a"],"ipv4":["10.203.62.228","10.203.61.247","10.203.62.246","10.203.62.249"]},"ASWUT3":
{"ipv6":["2001:67c:24cc:31e1:0:0:ae:3","2001:67c:24cc:31e0::cafe:9","2001:67c:24cc:31e0::cafe:9"],"ipv4":["10.203.63.228","10
.203.63.246","10.203.62.247"]},"ASWUT1":{"ipv6":["2001:67c:24cc:31c1:0:0:ae:1","2001:67c:24cc:31c0::cafe:f","2001:67c:24cc:31
c0::cafe:d","2001:67c:24cc:31c0::cafe:d"],"ipv4":["10.203.61.228","10.203.61.246","10.203.63.247","10.203.61.250"]}}},"meta":
{}}"
    
```

Figure 19: Network MAP retrieved from CINA Server

```

2012-11-21 16:48:39,258 DEBUG DecisionMakerImpl - sload, plength, bw, iptd, iplr, cina: 0.91277903, 79.999695, Infinity, 96.9
9981, 0.90707064, 0.7996633
2012-11-21 16:48:39,258 DEBUG DecisionMakerImpl - Server 3101:0:0:0:0:3 and path [41, 51] rate=0.7996633
2012-11-21 16:48:39,258 DEBUG DecisionMakerImpl - sload, plength, bw, iptd, iplr, cina: 0.91277903, 59.99977, Infinity, 73.49
985, 1.0020202, 0.45454544
2012-11-21 16:48:39,258 DEBUG DecisionMakerImpl - Server 3101:0:0:0:0:3 and path [41, 61, 62, 51] rate=0.45454544
2012-11-21 16:48:39,258 DEBUG DecisionMakerImpl - sload, plength, bw, iptd, iplr, cina: 0.91277903, 49.99981, Infinity, 72.49
9855, 0.9979799, 0.45454544
2012-11-21 16:48:39,259 DEBUG DecisionMakerImpl - Server 3101:0:0:0:0:3 and path [41, 61, 63, 62, 51] rate=0.45454544
2012-11-21 16:48:39,259 DEBUG DecisionMakerUtil - Server 3101:0:0:0:0:3 rate=0.7996633
2012-11-21 16:48:39,259 DEBUG DecisionMakerUtil - Server 3101:0:0:0:0:3 rate=0.45454544
2012-11-21 16:48:39,259 DEBUG DecisionMakerUtil - Server 3101:0:0:0:0:3 rate=0.45454544
    
```

Figure 20: Ranking Assignment with CINA activated

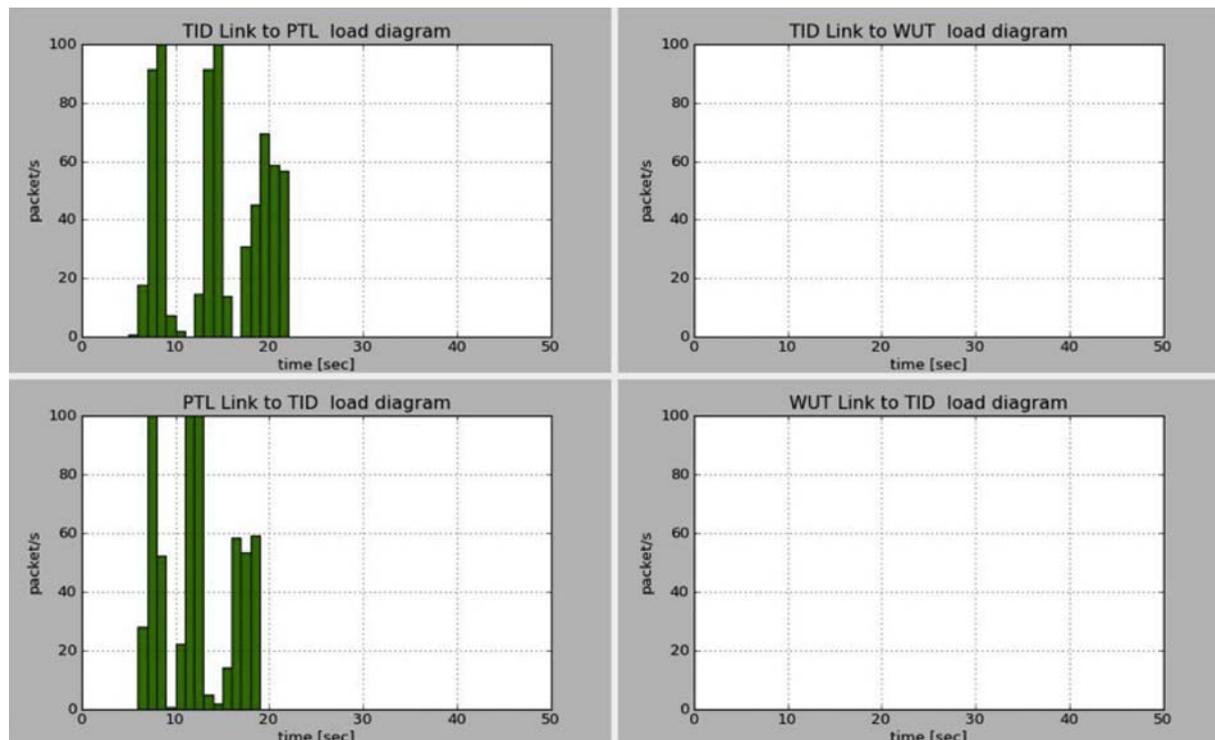


Figure 21: Traffic in CAFEs with CINA activated

By using the CAFE visualisation tool, it can also be shown that the traffic is going through CAFEs in TID (TID link to PTL only) and entering PTL through the border CAFE in AS 41 (PTL link to TID), as depicted in Figure 21.

Thus, it is shown how the path selection varies following the CINA costs, allowing thus for more sophisticated server and path selection policies in COMET taking into account the ISP preferences set by ENVISION based on different optimisation criteria, including for example time and space traffic shifting to lower ISP transit costs.

3.1.3.4 Conclusions

The use of CINA in COMET allows for taking into account network performance and ISP preference information in a consistent way when the optimisation logic is embedded into the network and when it resides outside the network into third-party overlay applications. The CINA interface has been integrated with the content-aware path selection functionality developed in COMET and evaluated using the COMET federated testbed interconnecting three European sites. The CINA interface provided additional costs for the path selection in COMET, resulting in the successful diversion of traffic from the path using the Warsaw University site towards the Primetel site in Cyprus.

3.1.4 Content Distribution via IVCD and Multi-link enabled peers

These tests are performed in the FT testbed 2.1, ALUD testbed 2.2, and LiveU testbed 2.4.

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 PSNR by electing the best resolution in AVC or SVC and an analysis of the impact between the wireless access network and the internet domain connectivity for determine the influence it has on proper selection of the best Multi-link Server (MLS) out of several using the CINA services. Further the IVCD system functionality is demonstrated including the invocation of 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, In theory the MLEP shall invoke network queries from the different CINA servers, based on the retrieved information a selection of MLS, and links priorities are performed, however in reality there was a single MLS in the evaluation. 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 3 of [D4.3].

The specifications of the High-Capacity Node service are provided in detail in section 4.3 of [D3.3].

3.1.4.1 Objectives and Evaluation Requirements

- Demonstrate the functionality of the peer-to-peer video streaming system and its ability to integrate CINA services such as HCN.
- Demonstrate the functionality of the tree topology creation algorithm in the tree management module.
- 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.
- Evaluate the QoE improvement at the source by providing adaptive mechanisms
- Evaluate the most critical network segment, weather it is on the wireless networks or on the internet domain.

3.1.4.2 Evaluation Environment

These tests have been performed in FT testbed, ALUD testbed, and LiveU labs, see section 2 for more information on the testbeds. The evaluation has been carried out in steps by ALUD and LiveU and has been be integrated into FT testbed for evaluation and demonstration of the technology during the ENVISION plenary meeting in Lannion. Several integration options have been identified and analysed to enable the functional testing and for the integration of LiveU and ALUD contributions, including the invocation of CINA services such has the HCN in a live video content distribution setting. Special emphasis was taken in the evaluation environment on the capability to handle live HD video content while dynamically modifying the distribution tree topology to enable online distribution tree optimisation (see section 3.2.4).

3.1.4.3 Performance Metrics

- *Latency*: The time difference between the playout point and the time a chunk is produced at the source.
- *Stream quality*: The PSNR 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, and whether the internet domain path is of great influence on section the MLS.

3.1.4.4 Controlled and Uncontrolled Variables

IVCD demonstration scenarios including the invocation of the high-capacity node:

Controlled variable:

- number and degree of nodes

Uncontrolled variable:

- distribution tree topology

MLEP uplink scenarios, selection of the MLS and improving the stream quality:

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.

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.

3.1.4.5 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	<ul style="list-style-type: none"> Source Peer is initialised to stream from a live HD video source Consumer Peers join the tree topology and start displaying the live video content The overlay topology is visualised
Demonstrate the functionality of the High Capacity Node network service	functionality of the instantiation of the network service functionality of the integration into the overlay topology	<ul style="list-style-type: none"> Source peer is initialised manual instantiation and configuration of HCN Tree topology is reorganised accordingly Visualisation shows new topology with HCN

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	<ul style="list-style-type: none"> <i>node out degree</i> <i>number of nodes</i> <i>network and cost map</i> 	<ul style="list-style-type: none"> tree topology
MLEP uplink, selection of the MLS and improving the stream quality	stream liveness, throughput, Quality (in PSNR)	<ul style="list-style-type: none"> <i>Number of domains:</i> The maximum number of domains the MLEP is connected to. <i>Number of MLS (LU1000):</i> <i>Number of HCNs:</i> The number of HCN participates in the evaluation scenario, could be one per domain or less, with a minimum of 1 per all. 	<ul style="list-style-type: none"> <i>Resolution of the encoded stream (number or pixels)</i> <i>Generated video data rate</i> <i>FEC overhead</i> <i>The bitrates per link</i>

		<ul style="list-style-type: none"> • <i>Latency between peers</i> • Latency of links • Loss of links • Rate of links 	
--	--	--	--

Table 4: IVCD Integrated Test Scenarios

3.1.4.6 Evaluation Results

The functional evaluation of the IVCD demonstration scenario including the invocation of the high-capacity node has been performed employing a distribution tree topology with a number of consumer Peers join the tree topology and displaying the live HD video content provided by the LiveU equipment.

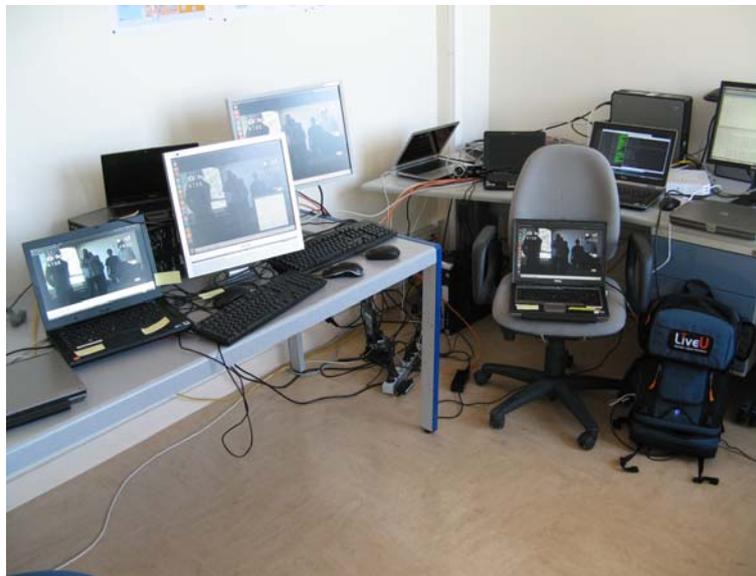


Figure 22: IVCD functional evaluation integrating three consumer peers, the source peer and LiveU equipment providing HD video content upload.

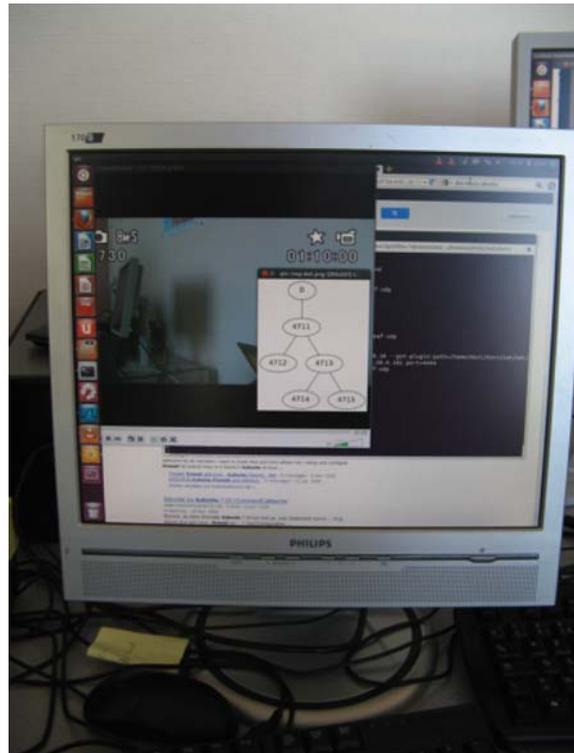


Figure 23: On-line visualisation of distribution tree topology for live HD-video distribution.

3.1.4.6.1 Direct link to the operator’s network

In this test scenario we aim to see where is the min network segment that influences on our performance. We have compared two cases, one is when the uplink transmitter is connected directly to the mobile operator’s network via dedicated private lines, and the other is just using the normal connectivity of the ISP to the internet. The direct link represents the best QoS possible.

We performed these tests together with one of Israel’s largest operators, installing a direct LAN connection to the operator’s inner network. Then, we ran the following tests using our uplink transmitter, and assessed the resulting BW, delays and losses, results are provided in Figure 24.

As can be seen in Figure 24 to Figure 28 - using the direct link improved the delays a little but had no major impact on the bandwidth and losses.

Therefore, our current conclusion is that selection of the MLS has little impact on the performance and the real performance limiter resides in the wireless network part, thus, we can select the MLS based on the CINA server recommendation without penalty in performance.

ABR	200	400	500	600	900	1000	1100	1300
direct average of bandwidth	420.2035	629.6942	772.3076	738.5136	943.0709	1128.546	661.4669	576.0195
regular average of bandwidth	436.1537	247.4189	727.5835	536.2543	566.8769	558.8496	571.4012	555.0408
direct average of delay	68.81024	1071.943	304.9888	1981.234	1496.777	757.1836	1906.521	2277.764
regular average of delay	169.0845	1277.674	1003.68	2463.511	2160.348	2136.224	2413.542	2543.43

Figure 24: Evaluation of Direct link versus regular internet link

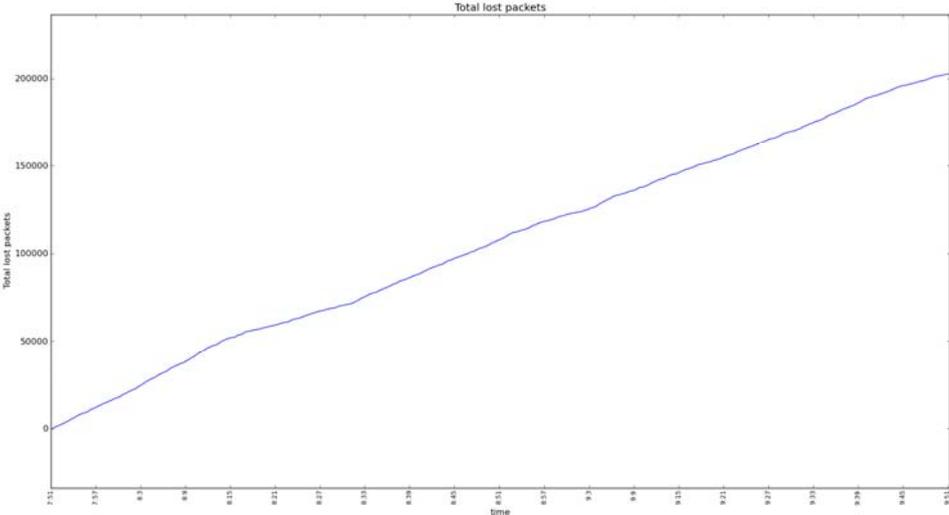


Figure 25: Packet loss as a function of time for the regular link

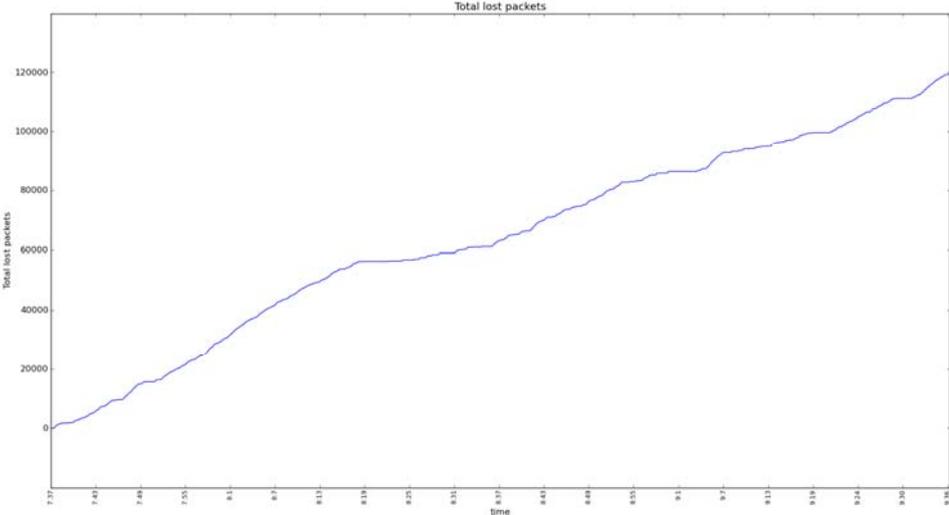


Figure 26: Packet loss as a function of time for the direct link

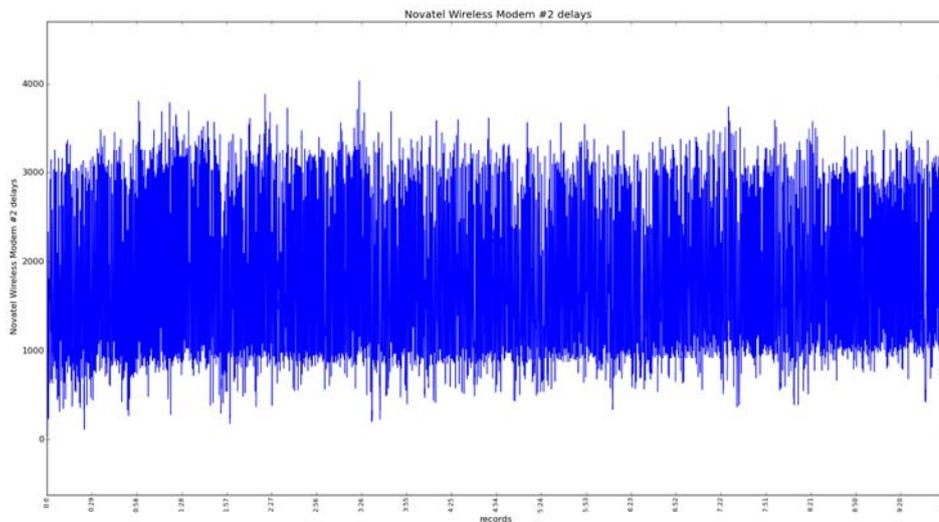


Figure 27: Packet delays for the regular link

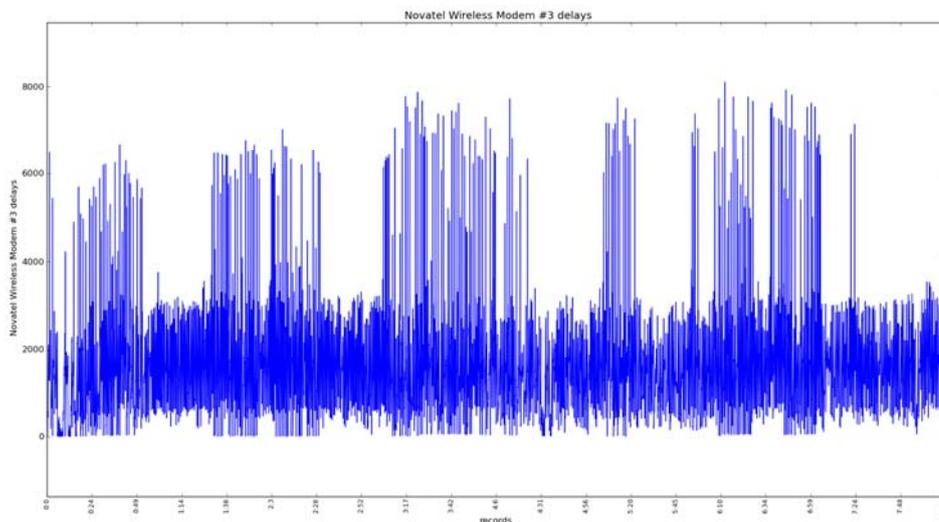


Figure 28: Packet delays for the direct link

3.1.4.6.2 QOE Assessments – Automatic resolution changes

In this section we are evaluating the transmitted video stream resolution. This applies to AVC and SVC streams and is determined using content generation and adaptation techniques defined in [D5.1, D5.2, D5.3]. The QoE of a video stream depends on the available bandwidth and on the video resolution. High resolution video streaming requires large bandwidths. Trying to transmit a high resolution video over a channel with relatively low bandwidth results in lower QoE compared to selecting smaller resolution. Figure 29 below shows that the PSNR metric for video quality can be improved at low bitrate if the video is downscaled prior to encoding and then upscaled back after the decoding. This procedure leaves more bits per macroblock on the encoding process, which compensates for the quality loss in the down scale process. On the other hand, high bit-rates enable the encoder to produce best quality when working at input resolution.

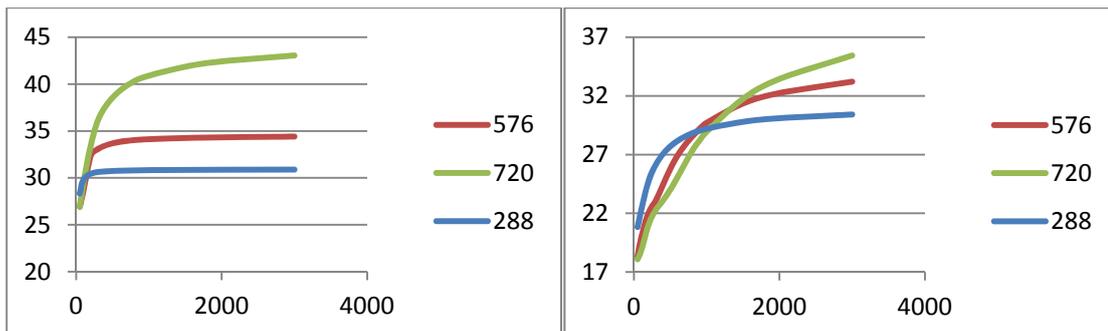


Figure 29: Comparing the PSNR of two different types of movies - one with relatively high motion and one with low motion under different resolutions for a range of bitrate values

Our assessments first included building a database of different videos, and calculating the PSNR for different bitrate values and different resolutions (As seen in Figure 29).

Based on this database, we were able to test two possible channel behaviours, and estimate the optimal resolution for each movie under various bitrate conditions.

The following figures show the PSNR for a “high-motion” video, under different resolutions. Each figure presents a different channel behaviour. Figure 30 presents a channel with alternating high bitrate (2000kbps) and low bitrate (400kbps) every few seconds. Figure 31 presents a random channel, where the bitrate has a Gaussian distribution with an average bitrate of 1200kbps.

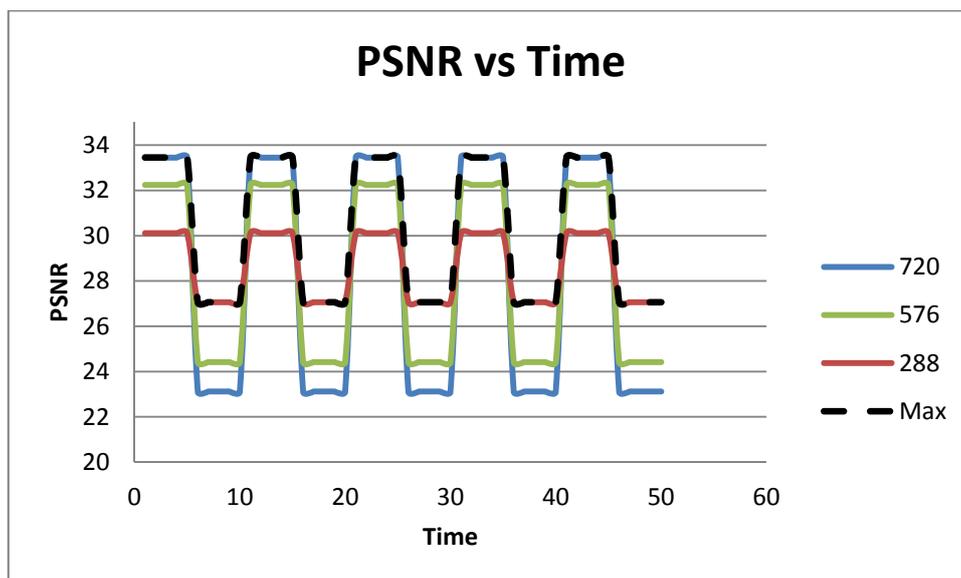


Figure 30: PSNR as a function of time for a periodically changing channel bit rate at several resolutions. The dashed black line shows the maximal PSNR, obtained when the resolution is adaptively changed according to the channel bit rate.

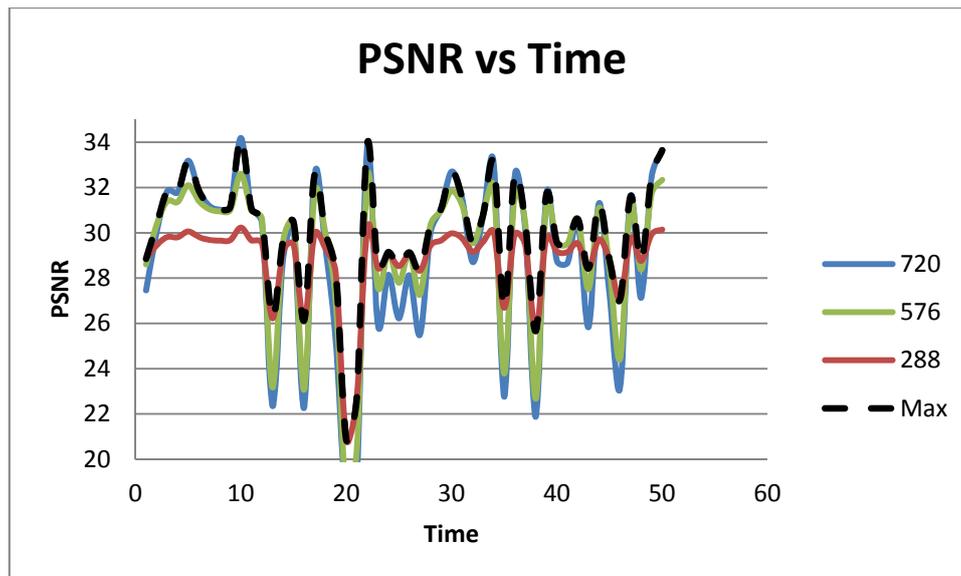


Figure 31: PSNR as a function of time for a Gaussian randomly changing channel bit rate at several resolutions. The dashed black line shows the maximal PSNR, obtained when the resolution is adaptively changed according to the channel bit rate

When looking at Figure 30 and Figure 31 we can see that the optimal PSNR is achieved when combining the different resolutions graphs (Using the MAX values at each point). On the average we find that the overall improvement of the PSNR is in the order of a 1.4 DB with respect to the Fix 288, and of 1.0 DB from the 576 and 720 resolutions, this results however does not represent the actual QoE improvement which is potentially much higher, when eliminating blockiness and frame dropping which were not considered in this test of PSNR. In LiveU systems, where the worst case scenario is of high interest, the increase in DB for isolated cases in PSNR stands on approximately 3-4 DB, this could be seen in Figure 30.

Therefore, the algorithm which was developed dynamically determines the optimal resolution switching points, i.e. when the transmitted video resolution should be changed to maximize QoE. The algorithm was also assessed using LiveU's LU60 unit, on a single channel with controllable bandwidth. The following figures which describe the testing phase in the algorithm development show how when the modem was given a limitation of a lower bandwidth, the resolution dropped, and when the bandwidth was then restored, a higher resolution again reappeared.

The following three figures demonstrate the evolution of the algorithm:

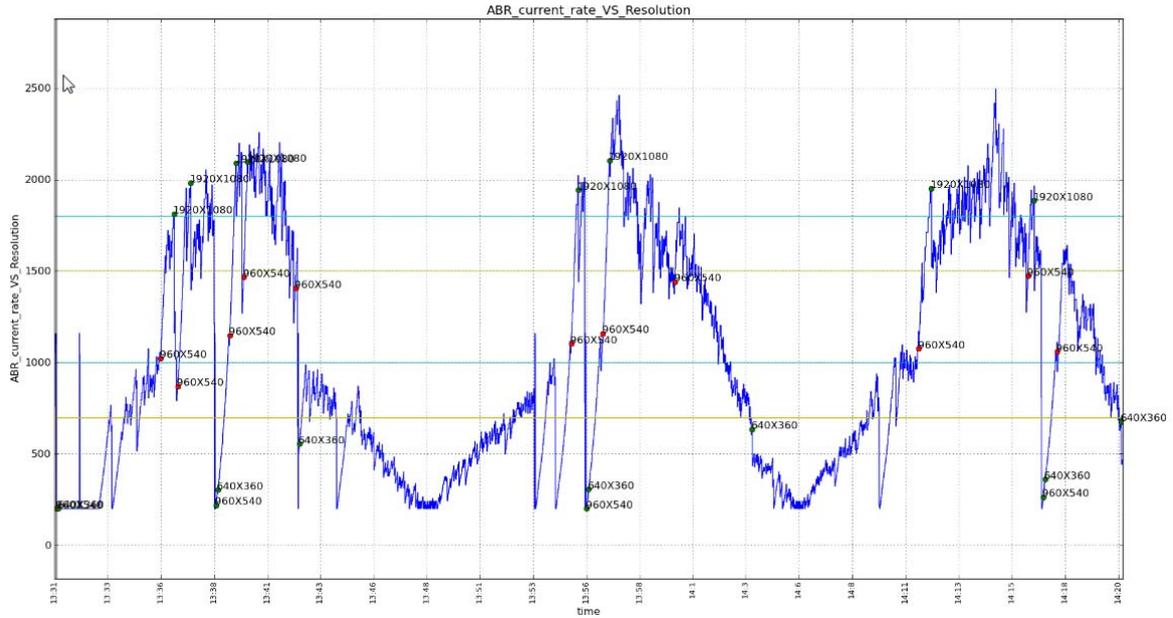


Figure 32: Video bit rate vs. time, with resolution switching points, 1st algorithm implementation

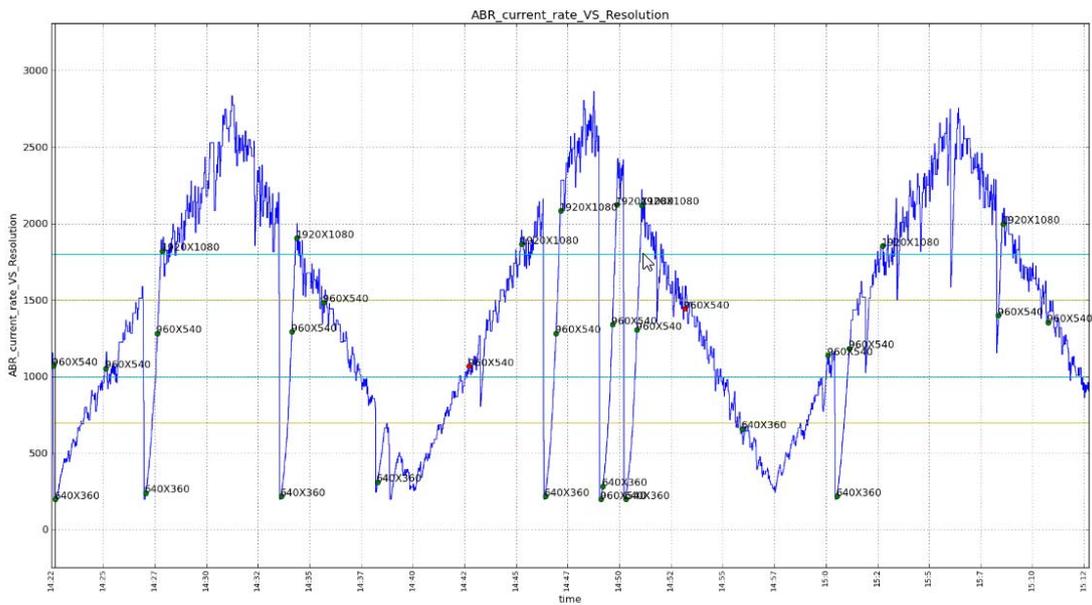


Figure 33: Video bit rate vs. time, with resolution switching points, 2nd algorithm implementation

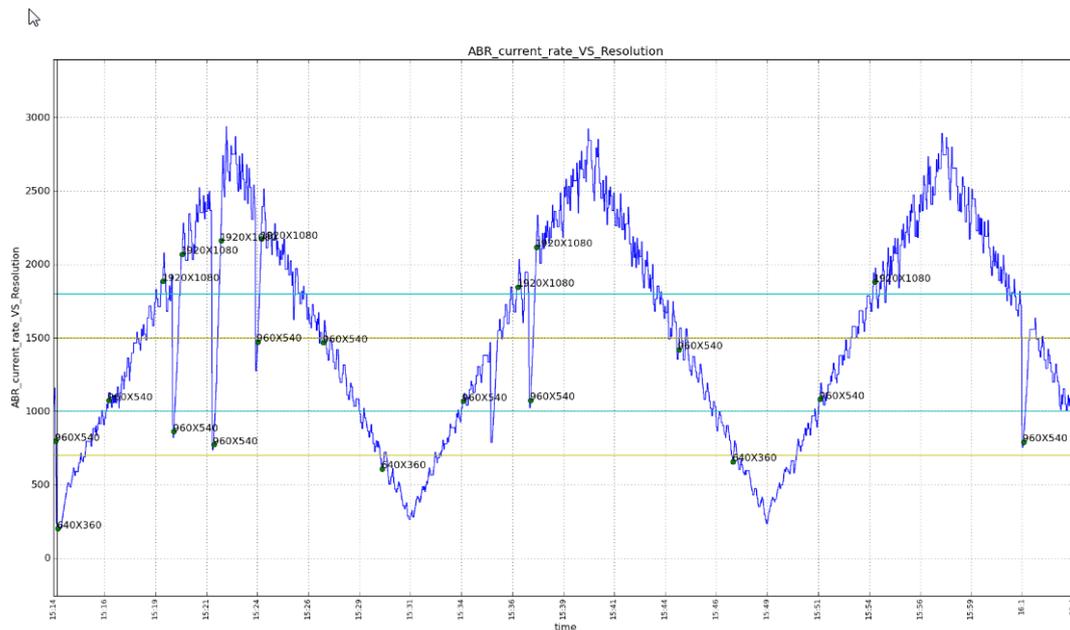


Figure 34: Video bit rate vs. time, with resolution switching points, 3rd algorithm implementation

3.1.4.7 Conclusions

The tests in this section involved two independent components working together under the integrated scenario of interactive video streaming applications: the High Capacity Node (HCN) CINA service and the Multi-Link Enabled Peer (MLEP) encoding and streaming functions. The HCN has been tested in the ALUD testbed and its correct invocation through CINA and operation when integrated with the interactive video streaming has been validated.

The MLEP has been tested in a stand-alone scenario using two types of connections: the standard service offered by a mobile operator and another where dedicated private lines are used by the operator in the fixed part of its network. The increase in performance was deemed very small to justify the associated cost. Further, a study has been undertaken showing the benefits from the encoding algorithm that dynamically determines the resolution switching points for a content source operating over a wireless connection of variable performance. The gains in PSNR vary from 1 to 1.4 and 4 DB, depending on the original resolution of the video. Finally, the MLEP was integrated and tested with the ENVISION interactive video overlay application in the FT testbed.

3.1.5 Multicast Network Service

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

3.1.5.1 Objectives and Evaluation Requirements

- Demonstrate the feasibility of an open and on-demand 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 on inter and intra-domain link load.

3.1.5.2 Evaluation Scenarios and Metrics

3.1.5.2.1 Evaluation Environment

These tests are performed in the FT testbed (see section 2.1).

The software prototype used for these tests is the same as the one described in section 3.1.1.

The software integrates the possibility for the tracker to initiate a switching to a reception in multicast for some lechers for a specific live content.

The parameters required to use the multicast service (source sending the content, peers having to switch to multicast, SVC layers concerned) can be specified via a command line on the tracker, see below.

```
[Tracker] Announce received from 172.20.74.66:2000
[Tracker] 172.20.74.66:2000 want to be connected with 172.20.32.130:2000,2002
multicast start T
  sender      : P1
  destinations: P2 P3 P4 P5
  layers      : 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
Contact CINAServer to allocate a multicaster ? (y/n) : y
Command Sent to Source 172.20.6.178:2000-2002 <172.20.6.178:2000-2002>
Command Sent to Destination 172.20.102.34:2000-2002 <172.20.102.34:2000-2002>
Command Sent to Destination 172.20.32.130:2000-2002 <172.20.32.130:2000-2002>
Command Sent to Destination 172.20.19.50:2000-2002 <172.20.19.50:2000-2002>
Command Sent to Destination 172.20.74.66:2000-2002 <172.20.74.66:2000-2002>
>>
```

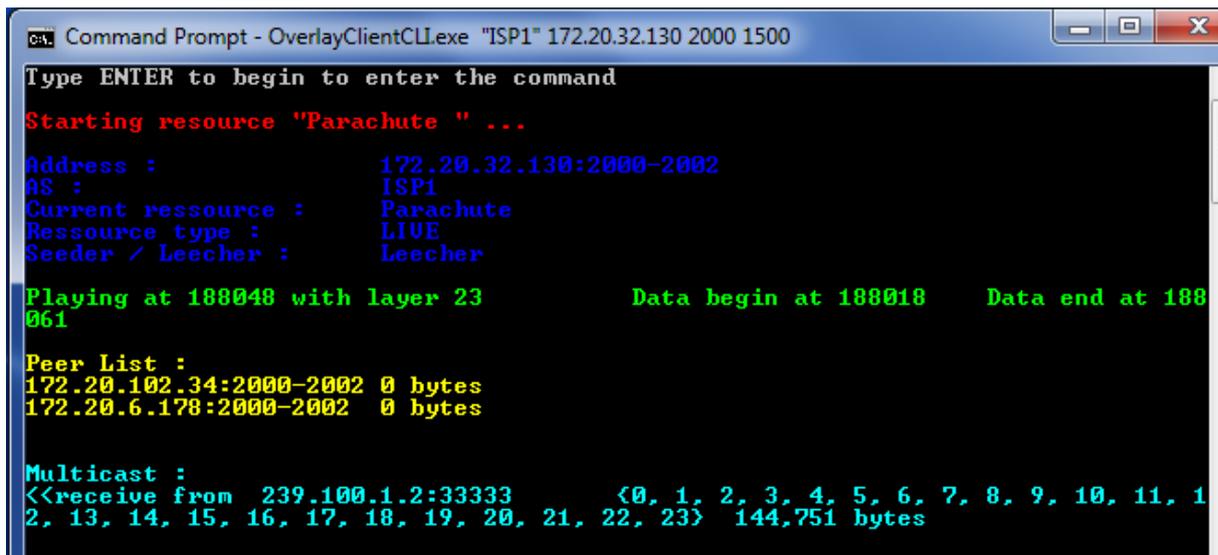
Figure 35: Tracker CLI to switch a set of peers to multicast

```
cmd: Command Prompt - OverlayClientCL.exe "ISP1" 172.20.32.130 2000 1500
Type ENTER to begin to enter the command
Starting resource "Parachute " ...
Address : 172.20.32.130:2000-2002
AS : ISP1
Current ressource : Parachute
Ressource type : LIVE
Seeder / Leecher : Leecher

Playing at 187862 with layer 23 Data begin at 187832 Data end at 187864
Peer List :
172.20.102.34:2000-2002 0 bytes
172.20.6.178:2000-2002 61,747 bytes

Multicast :
```

Figure 36: Peer receiving traffic in unicast



```

Command Prompt - OverlayClientCL.exe "ISP1" 172.20.32.130 2000 1500
Type ENTER to begin to enter the command
Starting resource "Parachute " ...
Address :          172.20.32.130:2000-2002
AS :              ISP1
Current resource : Parachute
Resource type :   LIUE
Seeder / Leecher : Leecher

Playing at 188048 with layer 23          Data begin at 188018      Data end at 188061

Peer List :
172.20.102.34:2000-2002 0 bytes
172.20.6.178:2000-2002 0 bytes

Multicast :
<<receive from 239.100.1.2:33333      {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23} 144,751 bytes
    
```

Figure 37: Peer receiving traffic in multicast

Then the tracker sends a request to the CINA server to use the multicast service, and, if the request is accepted, receives the information required to use the service. The server then notifies a seeder (the source) to start streaming the content towards the multicaster, and notifies the leechers that they must switch to a multicast reception for the indicated SVC layers.

The following figures show the leecher CLI before and after the reception of the notification to switch to multicast.

The software is deployed on ten terminals. Four terminals in ISP1 are able to receive content in multicast: E, F, G and M. Please note that because not all the terminals in ISP1 are able to receive multicast (because of OS versions), the evaluation results do not illustrate fully the interest of the multicast service.

Peer A in ISP2 is always used as the content source peer, and as the peer sending the flow to multicast to the multicast service in ISP1.

The file that was used to emulate a live stream is encoded to a stream with 662 kbps bitrate in total, split over 24 SVC layers.

3.1.5.2.2 Performance Metrics

- *Inter-network load*: The average total (incoming and outgoing) traffic rate in kbps observed over the ISP2 to ISP1 inter-domain link.
- *Intra-network load*: The total bandwidth required to distribute a specific live content to N users, considering all the used links.

3.1.5.2.3 Controlled and Uncontrolled Variables

Controlled variables:

- *multicast source selection*
- *multicast data selection*
- *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)

3.1.5.2.4 Evaluation Scenarios

Scenario	Metrics	Description
Time to activate the multicast service through CINA	Time between the CINA request, the CINA response and the first packet received in multicast	The tracker and a leecher are run on the same terminal (F). A seeder is run on another terminal (). The content distribution starts in unicast, and then a command on the tracker triggers a request to the CINA server in ISP1 to use the multicast service, and notifications are sent to the seeder and the leechers to switch to multicast.
Impact on network load	induced network load	Several leechers subscribe to the same content in full quality in unicast. Then some of them in ISP1 are switched to multicast. The bandwidth used on each link and the total bandwidth in the simulated peering link between ISP3 and ISP1 are compared in both situations.

Table 5: Multicast evaluation scenarios

3.1.5.3 Evaluation Results

3.1.5.3.1 Time to activate the multicast service through CINA

As it can be shown in Figure 38, the time to receive a positive response to a CINA multicast request in ISP1 from terminal F is, in normal network conditions, between 200 and 250 ms (219ms in the figure, between 1 and 2), and the first multicast UDP packet is received about 300ms after the CINA response (line 3).

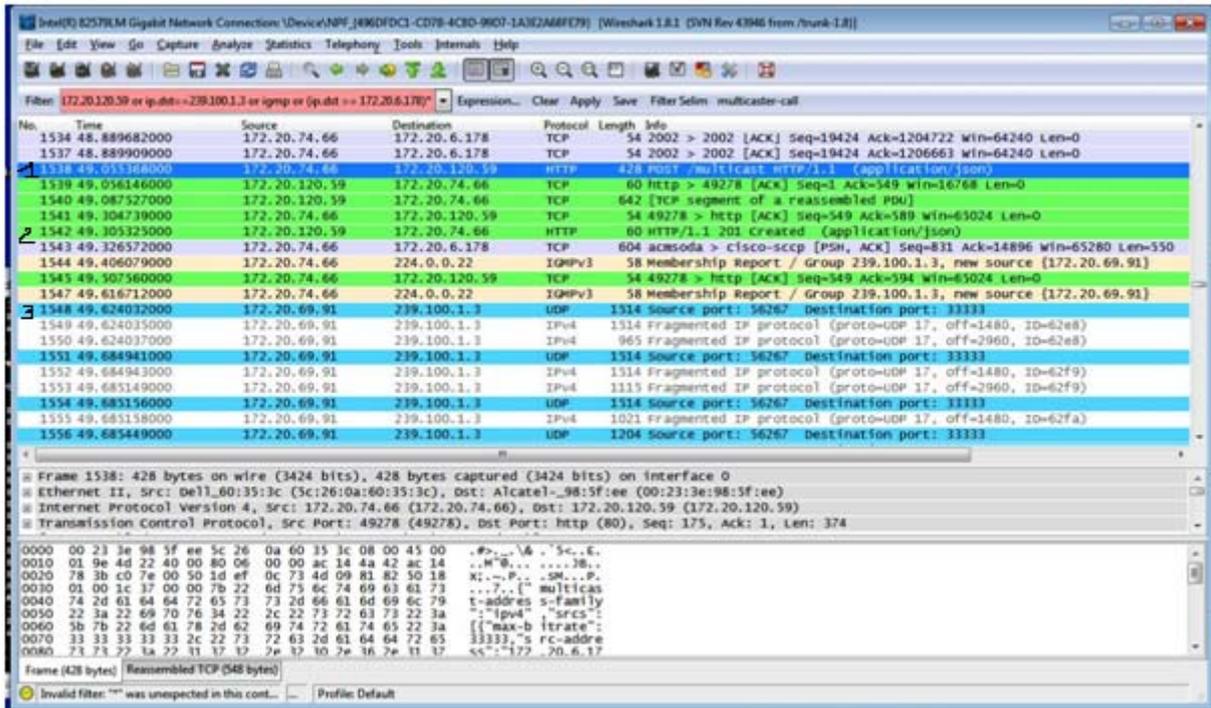


Figure 38: Multicast activation delay

3.1.5.3.2 Impact on network load

Impact on inter-network load:

As it can be seen on Figure 39, with a constant number of peers and a constant content quality level, the use of the multicast in the ISP1 network decreases the total traffic on the peering link between ISP2 and ISP1, of respectively of 11.4%, 12.2%, 45%, 3.6% and 2% in the five tests.

An interesting thing is that it is mainly the ISP1 incoming traffic which is reduced (of 25% in Test1, 35% in Test2, 43% in Test3, 27% in Test4 and 22% in Test5), whereas the outgoing traffic towards ISP2 tends to increase (20% in Test1, 142% in Test2, 100% in Test4, 184% in Test5).

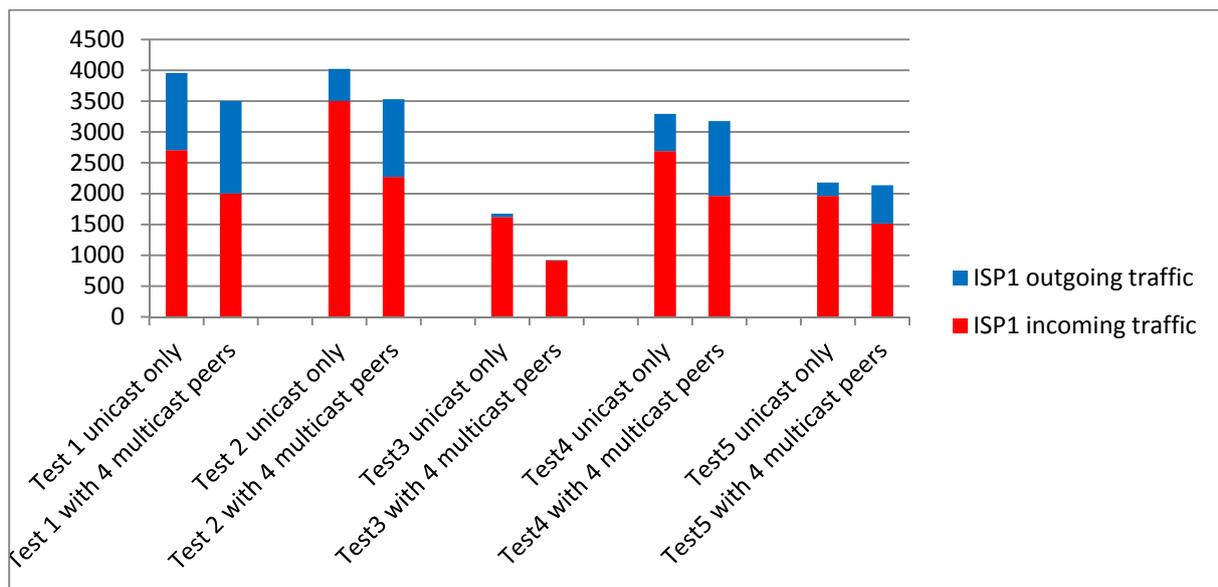


Figure 39: traffic rate on the peering link

Impact on intra-network load:

Using the multicast service globally improves the total bandwidth usage in ISP1. To compare the bandwidth used, we add the number of times the content flow crosses each link in ISP1.

As an example, Figure 40 and Figure 41 show the content flows in Test1, respectively with and without the use of the multicast service in ISP1. Red arrows represent incoming traffic; blue arrows represent outgoing traffic, grey arrows represent internal traffic and green arrows represent the multicast traffic. In this favourable example, where the locality is poor in unicast, the intra-network traffic of ISP1 is reduced of about 25.6%, from 36.5 times the content on the set of links, to 27.16 times.

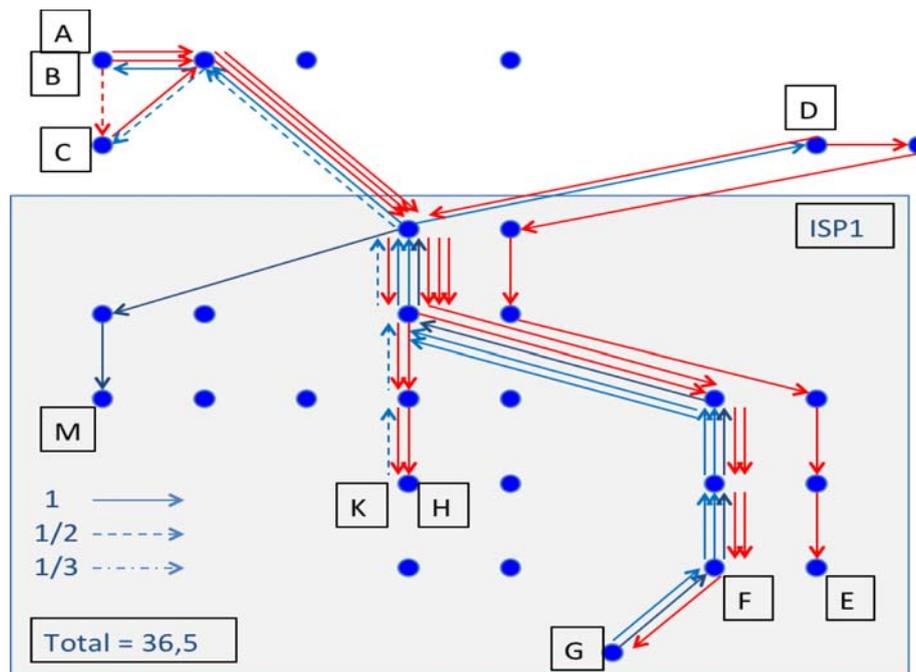


Figure 40: content flows in Test1 (unicast only)

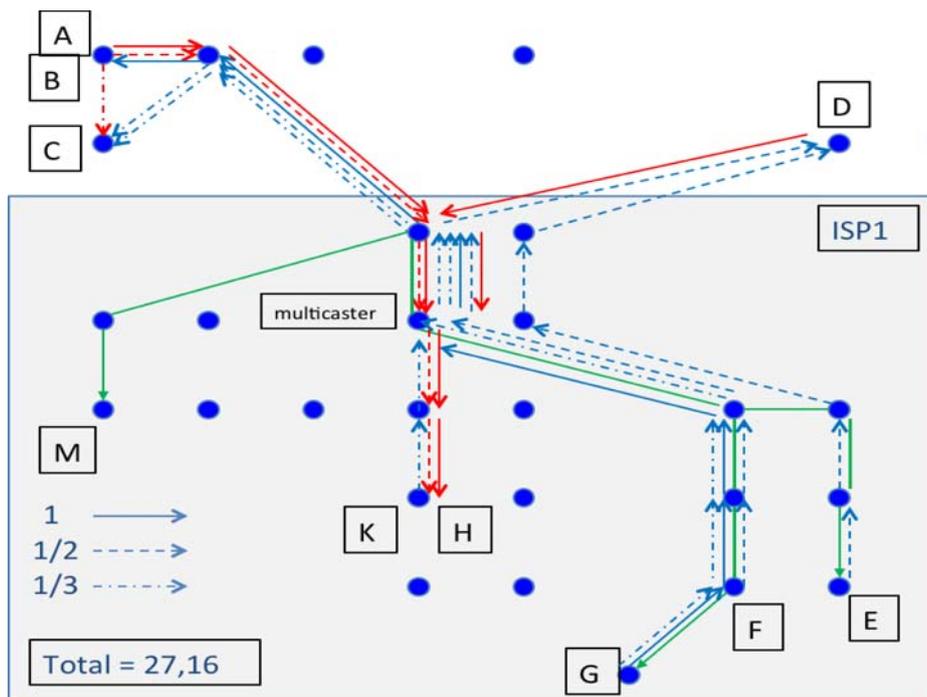


Figure 41: content flows in Test1 (with multicast)

Figure 42 shows the bandwidth usage for the five tests. As it can be seen for Test2 and Test5, the bandwidth usage is practically the same (-4.40% and +4.10%) but in these cases the total bandwidth amount in the unicast configuration is low (the optimal case in our multicast configuration corresponds to a total bandwidth usage of 14, with 11 due to the multicast flow and 3 to provide content to K and H in unicast). The bandwidth usage is well improved for Test1, Test3 and Test4 (-36.50%, -32.40% and -28.40%)

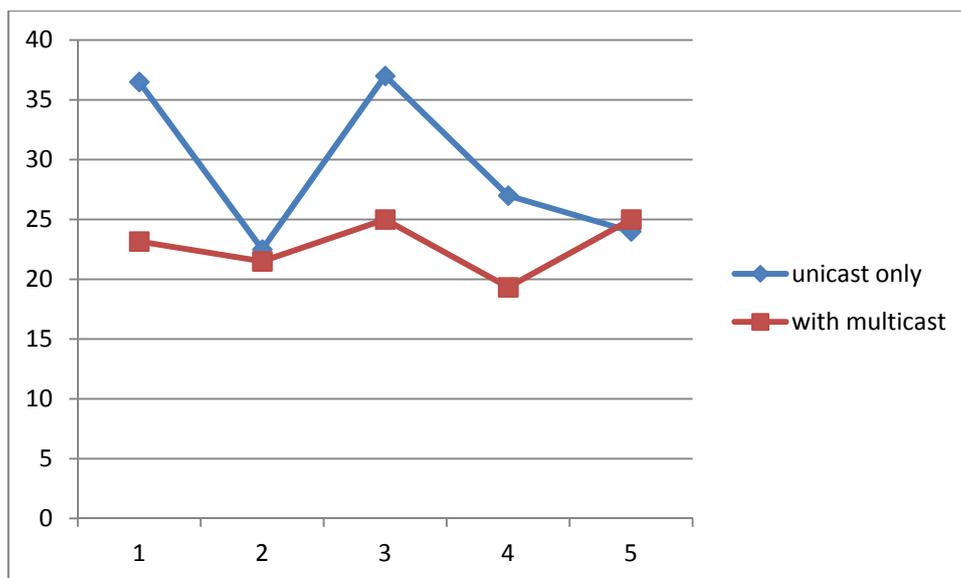


Figure 42: total bandwidth usage in ISP1

3.1.5.4 Conclusions

The multicast network service has been tested in the FT testbed. The service activation time was under 250ms proving the efficiency of the design and the prototype implementation of the CINA interface and the multicaster components. For the ISP, the motivation for deploying multicast is the traffic saving that can be achieved a) on the inter-domain links as the stream needs to be downloaded only once, and b) on the intra-domain links as the stream is distributed along the multicast tree instead of using a mesh of unicast connections crossing the same physical link multiple times. The tests were undertaken with a very limited set of peers (four in total) switching to receiving the stream over multicast. Even in this small scale scenario, the savings were up to 45% and 36% for the inter- and intra-domain traffic respectively.

3.1.6 Caching

3.1.6.1 Objectives and Evaluation Requirements

The objectives of testing cache function in ENVISION are:

- Evaluation of bandwidth saving for ISP hosting caches inside its network
- Evaluation of bandwidth improvement of Quality of Experience for customers

For the test, we use the FT platform including 3 different ASes representing ISPs. The AS1 hosts 3 caching servers. Each of them is capable to start automatically a certain number of ENVISION clients per content uploaded by a seeder located anywhere in the network.

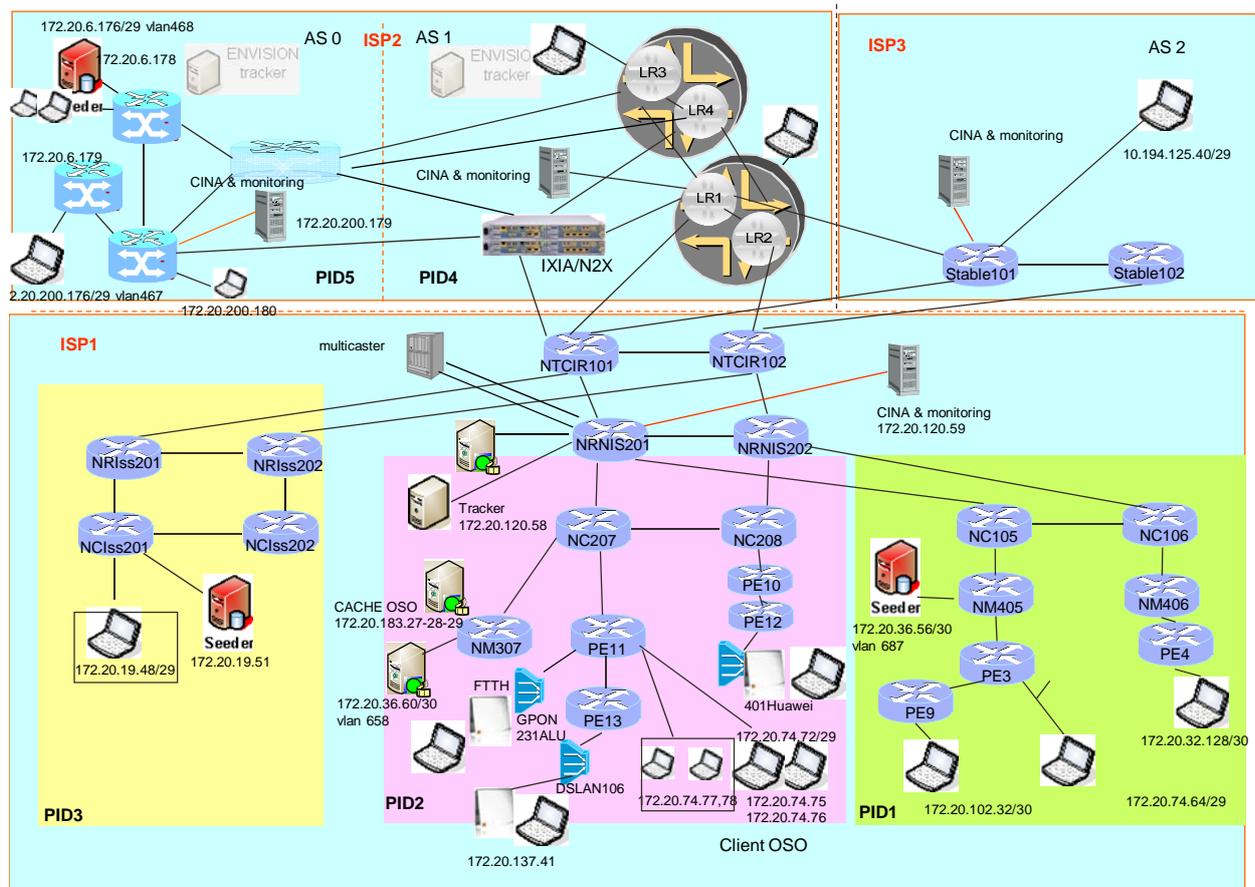
Tests are made with desktop or laptop terminals connected in the 3 ASes.

3.1.6.2 Evaluation Scenarios & Metrics

The test platform is part of a real and complete network, hosting several concurrent experimentation activities in Orange Labs. So, it is impossible to measure the part of traffic generated by the ENVISION tests. So, to measure the bandwidth saving, we cannot see it on router interfaces but we consider logs and NIC on each ENVISION client (origin seeder, seeders in caches, leechers).

QoE is measured on leecher peers which download contents from different seeders, without and with congestion on links. The measurement is based on the difference of download speed.

3.1.6.3 Test platform



The diagram above represents the network and ENVISION elements connected on it.

Caches are integrated in a DELL server which integrates virtual machines based on Windows7. These virtual machines host the ENVISION client software produced by LaBRI with different added functions produced by UCL.

3.1.6.4 Evaluation Results

3.1.6.4.1 Ingestion mode

Tests were made with different VoD but also with live contents, using the ENVISION client developed by UCL/LaBRI. The following paragraphs describe VoD tests but they are also valid for live contents.

3.1.6.4.2 Launching the ingestion

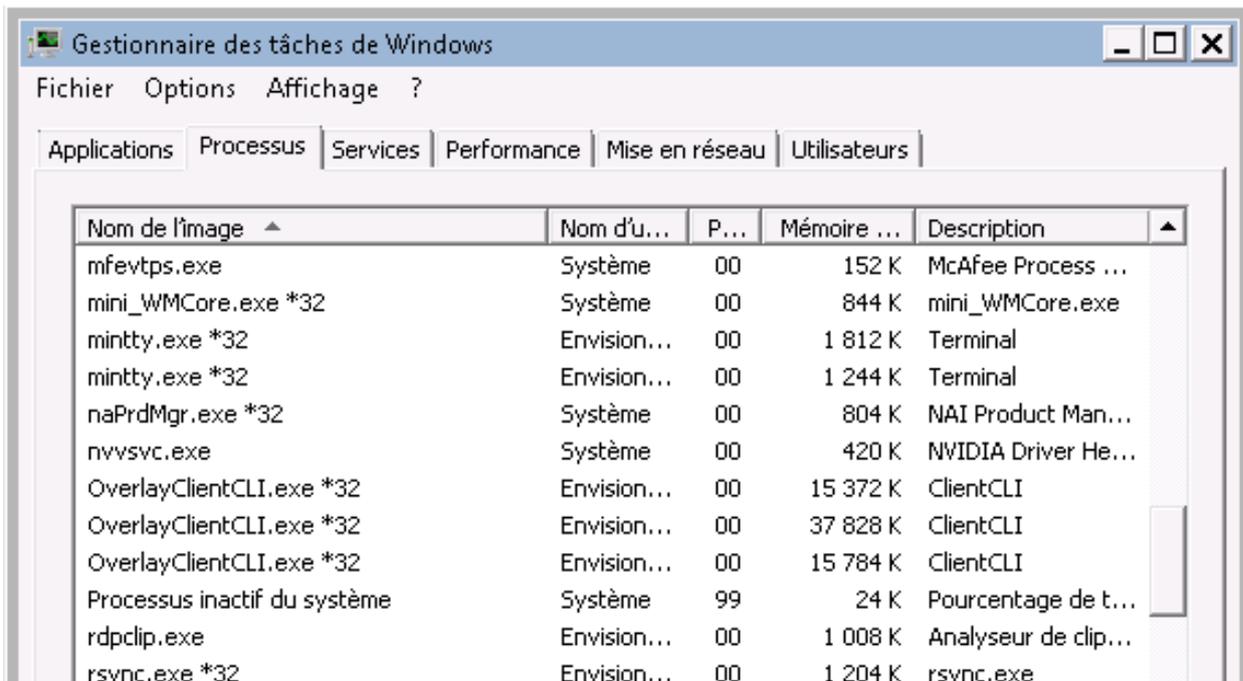
The seeder is on ISP2 network.

There are 3 VoD in seeding mode from the seeder.

Caches are in service and detect new torrents downloaded from Portal to Cache Controller through CINA Server.

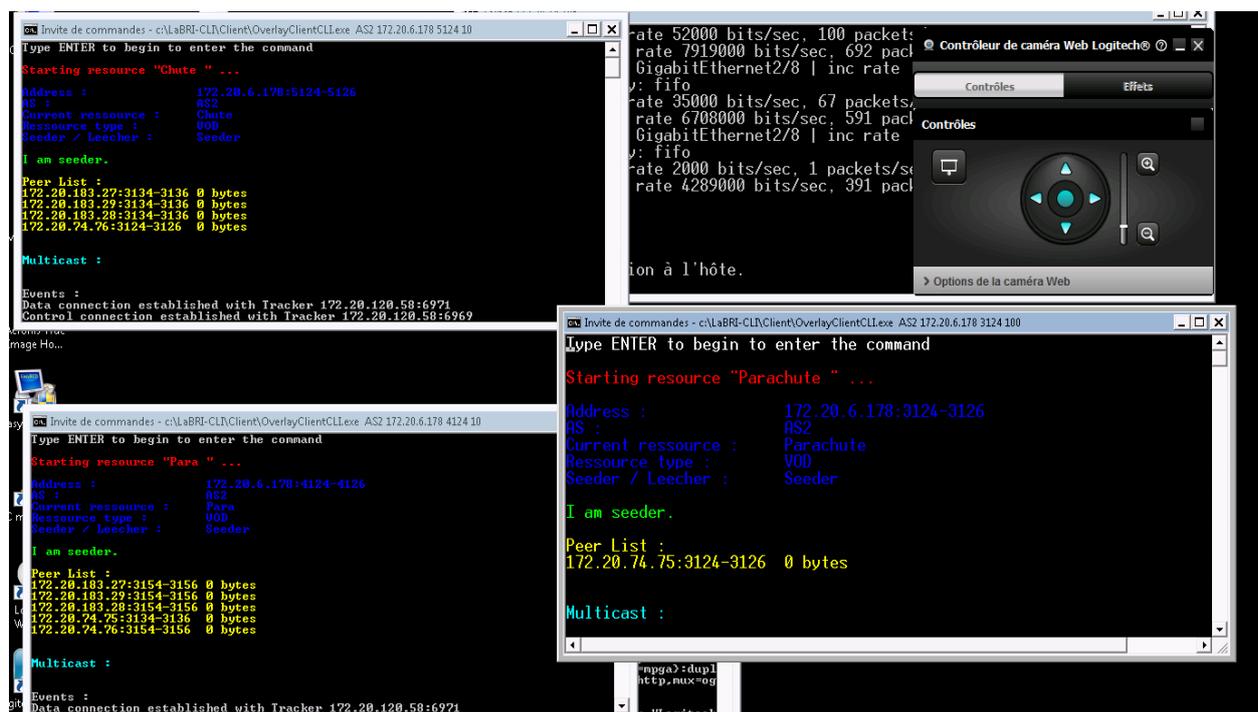
Cache Software launches one process "Leecher" for each new torrent.

Results: 3 OverlayClientCli.exe per cache



3.1.6.4.3 Result on Seeder

The next screenshot presents the seeder and the 3 connected caches as regular peers for the 3 VoDs.



3.1.6.4.4 Result on tracker

Tracker is able to show the result for each ENVISION client.

Next screenshot presents results for each cache and each VOD.

On tracker:

```
C:\WINDOWS\system32\cmd.exe - OverlayTrackerCLI.exe 172.20.120.58 6969 TestbedMap.cfg
overlay showall
Name PID IP:Port Resource Type Liveness/Download Ratio
P1 2 172.20.6.178:3124-3126 <172.20.6.178:3124-3126> Parachute UOD 1
P2 2 172.20.6.178:3134-3136 <172.20.6.178:3134-3136> Para UOD 1
P3 2 172.20.6.178:3144-3146 <172.20.6.178:3144-3146> Chute UOD 1
P4 1 172.20.183.27:3134-3136 <172.20.183.27:3134-3136> Chute UOD 1
P5 1 172.20.183.27:3154-3156 <172.20.183.27:3154-3156> Para UOD 1
P6 1 172.20.183.27:3184-3186 <172.20.183.27:3184-3186> Parachute UOD 0.0982359278211689
P7 1 172.20.183.28:3134-3136 <172.20.183.28:3134-3136> Chute UOD 0.0621128467546458
P8 1 172.20.183.28:3154-3156 <172.20.183.28:3154-3156> Para UOD 0.231063156477242
P9 1 172.20.183.28:3184-3186 <172.20.183.28:3184-3186> Parachute UOD 0.164405467277134

>> overlay showall
Name PID IP:Port Resource Type Liveness/Download Ratio
P1 2 172.20.6.178:3124-3126 <172.20.6.178:3124-3126> Parachute UOD 1
P2 2 172.20.6.178:3134-3136 <172.20.6.178:3134-3136> Para UOD 1
P3 2 172.20.6.178:3144-3146 <172.20.6.178:3144-3146> Chute UOD 1
P4 1 172.20.183.27:3134-3136 <172.20.183.27:3134-3136> Chute UOD 1
P5 1 172.20.183.27:3154-3156 <172.20.183.27:3154-3156> Para UOD 1
P6 1 172.20.183.27:3184-3186 <172.20.183.27:3184-3186> Parachute UOD 0.108116751952599
P7 1 172.20.183.28:3134-3136 <172.20.183.28:3134-3136> Chute UOD 0.0621128467546458
P8 1 172.20.183.28:3154-3156 <172.20.183.28:3154-3156> Para UOD 0.266209938055481
P9 1 172.20.183.28:3184-3186 <172.20.183.28:3184-3186> Parachute UOD 0.199552248855373

>> overlay showall
Name PID IP:Port Resource Type Liveness/Download Ratio
P1 2 172.20.6.178:3124-3126 <172.20.6.178:3124-3126> Parachute UOD 1
P2 2 172.20.6.178:3134-3136 <172.20.6.178:3134-3136> Para UOD 1
P3 2 172.20.6.178:3144-3146 <172.20.6.178:3144-3146> Chute UOD 1
P4 1 172.20.183.27:3134-3136 <172.20.183.27:3134-3136> Chute UOD 1
P5 1 172.20.183.27:3154-3156 <172.20.183.27:3154-3156> Para UOD 1
P6 1 172.20.183.27:3184-3186 <172.20.183.27:3184-3186> Parachute UOD 0.457968623754377
P7 1 172.20.183.28:3134-3136 <172.20.183.28:3134-3136> Chute UOD 0.0621128467546458
P8 1 172.20.183.28:3154-3156 <172.20.183.28:3154-3156> Para UOD 1
P9 1 172.20.183.28:3184-3186 <172.20.183.28:3184-3186> Parachute UOD 1

>> overlay showall
Name PID IP:Port Resource Type Liveness/Download Ratio
P1 2 172.20.6.178:3124-3126 <172.20.6.178:3124-3126> Parachute UOD 1
P2 2 172.20.6.178:3134-3136 <172.20.6.178:3134-3136> Para UOD 1
P3 2 172.20.6.178:3144-3146 <172.20.6.178:3144-3146> Chute UOD 1
P4 1 172.20.183.27:3134-3136 <172.20.183.27:3134-3136> Chute UOD 1
P5 1 172.20.183.27:3154-3156 <172.20.183.27:3154-3156> Para UOD 1
P6 1 172.20.183.27:3184-3186 <172.20.183.27:3184-3186> Parachute UOD 0.484564368435228
P7 1 172.20.183.28:3134-3136 <172.20.183.28:3134-3136> Chute UOD 0.0621128467546458
P8 1 172.20.183.28:3154-3156 <172.20.183.28:3154-3156> Para UOD 1
P9 1 172.20.183.28:3184-3186 <172.20.183.28:3184-3186> Parachute UOD 1

>> overlay showall
Name PID IP:Port Resource Type Liveness/Download Ratio
P1 2 172.20.6.178:3124-3126 <172.20.6.178:3124-3126> Parachute UOD 1
P2 2 172.20.6.178:3134-3136 <172.20.6.178:3134-3136> Para UOD 1
P3 2 172.20.6.178:3144-3146 <172.20.6.178:3144-3146> Chute UOD 1
P4 1 172.20.183.27:3134-3136 <172.20.183.27:3134-3136> Chute UOD 1
P5 1 172.20.183.27:3154-3156 <172.20.183.27:3154-3156> Para UOD 1
P6 1 172.20.183.27:3184-3186 <172.20.183.27:3184-3186> Parachute UOD 0.499427686506868
P7 1 172.20.183.28:3134-3136 <172.20.183.28:3134-3136> Chute UOD 0.0621128467546458
P8 1 172.20.183.28:3154-3156 <172.20.183.28:3154-3156> Para UOD 1
P9 1 172.20.183.28:3184-3186 <172.20.183.28:3184-3186> Parachute UOD 1

>> overlay showall
Name PID IP:Port Resource Type Liveness/Download Ratio
P1 2 172.20.6.178:3124-3126 <172.20.6.178:3124-3126> Parachute UOD 1
P2 2 172.20.6.178:3134-3136 <172.20.6.178:3134-3136> Para UOD 1
P3 2 172.20.6.178:3144-3146 <172.20.6.178:3144-3146> Chute UOD 1
P4 1 172.20.183.27:3134-3136 <172.20.183.27:3134-3136> Chute UOD 1
P5 1 172.20.183.27:3154-3156 <172.20.183.27:3154-3156> Para UOD 1
P6 1 172.20.183.27:3184-3186 <172.20.183.27:3184-3186> Parachute UOD 0.518381362779424
P7 1 172.20.183.28:3134-3136 <172.20.183.28:3134-3136> Chute UOD 0.0621128467546458
P8 1 172.20.183.28:3154-3156 <172.20.183.28:3154-3156> Para UOD 1
P9 1 172.20.183.28:3184-3186 <172.20.183.28:3184-3186> Parachute UOD 1

>>
```

After few minutes

```
>> overlay showall
Name PID IP:Port Resource Type Liveness/Download Ratio
P1 2 172.20.6.178:3124-3126 <172.20.6.178:3124-3126> Parachute UOD 1
P2 2 172.20.6.178:3134-3136 <172.20.6.178:3134-3136> Para UOD 1
P3 2 172.20.6.178:3144-3146 <172.20.6.178:3144-3146> Chute UOD 1
P4 1 172.20.183.27:3134-3136 <172.20.183.27:3134-3136> Chute UOD 1
P5 1 172.20.183.27:3154-3156 <172.20.183.27:3154-3156> Para UOD 1
P6 1 172.20.183.27:3184-3186 <172.20.183.27:3184-3186> Parachute UOD 0.980894828979262
P7 1 172.20.183.28:3134-3136 <172.20.183.28:3134-3136> Chute UOD 0.0621128467546458
P8 1 172.20.183.28:3154-3156 <172.20.183.28:3154-3156> Para UOD 1
P9 1 172.20.183.28:3184-3186 <172.20.183.28:3184-3186> Parachute UOD 1
```

The 3 VODs are stored inside caches. Now caches are considered as seeders for ENVISION overlay network.

3.1.6.4.5 Logs on caches

Caches present log files to record any issues during ENVISION client software. A sized file equal to "0" is the sign of no detected issue.

```

Envision-FT@L-M2ICDS-14-VM /cygdrive/c/ENVISION/Torrent/log/7453c4d3-6721-437f-a1d4-e8b6d62b28e7
$ ls -ltr
+ ls -ltr
total 34
-----+ 1 Envision-FT None      0  7 sept. 14:36 7453c4d3-6721-437f-a1d4-e8b6d62b28e7.log
-----+ 1 Envision-FT None      71  7 sept. 14:36 UselessData.log
-----+ 1 Envision-FT None     158  7 sept. 14:36 Neighbor.log
-----+ 1 Envision-FT None    21338  7 sept. 14:41 ViewedLayer.log

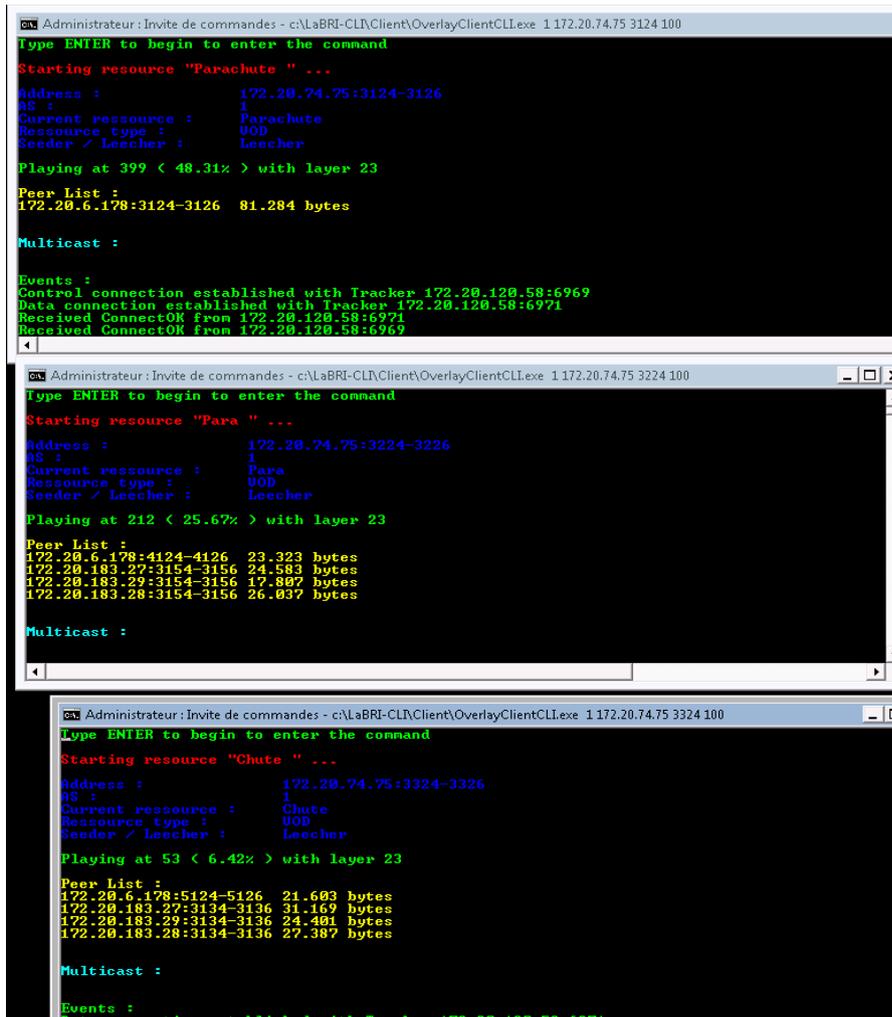
Envision-FT@L-M2ICDS-14-VM /cygdrive/c/ENVISION/Torrent/log/7453c4d3-6721-437f-a1d4-e8b6d62b28e7
$ tail -f 7453c4d3-6721-437f-a1d4-e8b6d62b28e7.log
+ tail -f 7453c4d3-6721-437f-a1d4-e8b6d62b28e7.log
    
```

3.1.6.4.6 Delivery mode

Once caches are seeders, they work as regular peers and upload VoDs to other leechers which can appear on ENVISION overlay network.

3.1.6.4.6.1 Single delivery

Next screenshot presents the leecher getting part of VoDs from 1 to 3 caches.



3.1.6.4.6.2 Delivery and network issues

Network presents some issues between the different ISP, for example links are busy, cut, or undersized. Tests must be shown the ISP cache impacts to deliver contents to other ENVISION clients depending signalling given by ISP to CINA server.

3.1.6.4.7 Caches attempts: bandwidth saving and QoE to end-users

In the different tests, upper traffic to external ISP clients is not taken in account. ENVISION clients as leechers are considered only for ISP1 (integrating cache service).

3.1.6.4.7.1 Bandwidth saving

Test is based on Parachute_VoD.svc which has a length of 22830669 bytes. Leecher receives segments from each available seeder. Cache is configured with 10 clients per torrent. Seeder origin is in AS2 and named OS (origin seeder).

3.1.6.4.7.2 Gain on ISP network using cache

Following number of ENVISION clients integrated in cache, we got estimation of traffic generated by cache.

Without any issue on interconnection between ISPs, leecher (configured in default mode) receives equal traffic from each seeder even if seeders are closed and have most important network resources than others.

The figure below represents the part of video volume saved by the cache with 10 inside clients-Seeders and with 1 to 6 external seeders (in ISP1, ISP2 or ISP3).

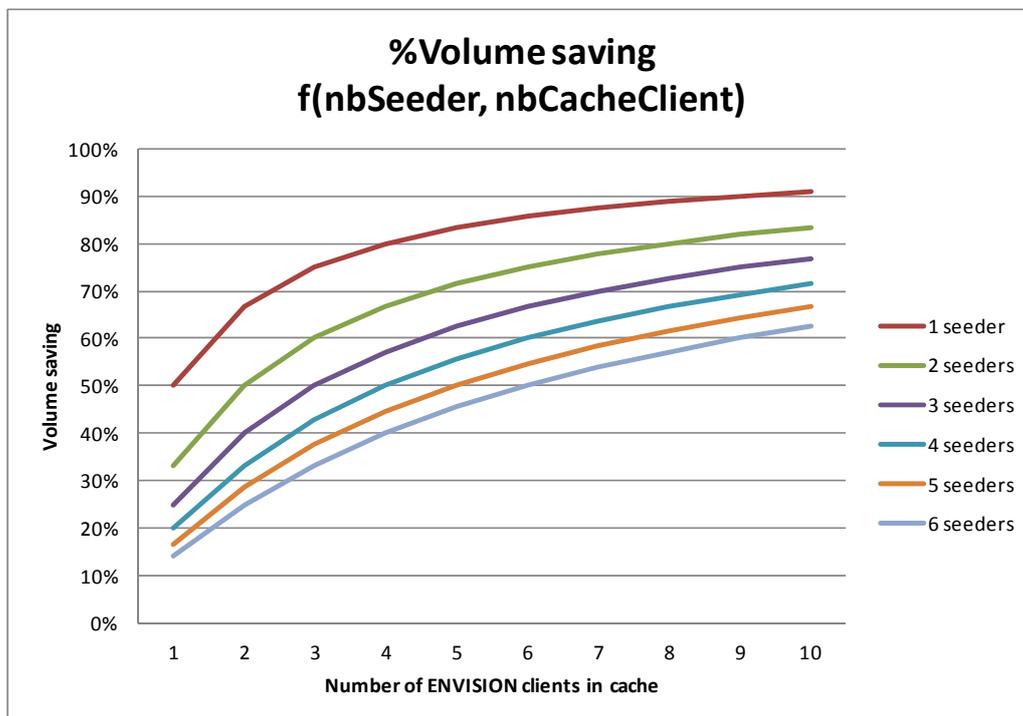


Figure 43: Volume saving with caching

We can conclude that a cache is able to save 90% of content volumes if storage is able to support the amount of contents and if the number of ENVISION clients per VoD (or torrent) is about 10.

If there are many external seeders, the saving is decreased. So caching is interesting if the P4P is also integrated to this technique.

3.1.6.4.7.3 Gain on ISP network including cache

Following leecher location, gain is measured with cache in ISP1 (PID2 zone).

In the experimentation, cache is located at NC207 level (PID2 zone). All clients connected to this router directly download VoD from cache and gain is max on the ISP1 network.

If we consider clients on PID1 or PID3 zone, gain is equal but not on complete path like the first case.

Seeder AS	Leecher location	Gain upper NC207		Gain upper NR201		Gain upper NT201	
		Volume	BW	Volume	BW	Volume	BW
OS	AS1						
OS	AS1 PID2	20 MB	150 kbps				
OS	AS1 PID1			20 MB	150 kbps		
OS	AS1 PID3					20 MB	150 kbps

We can conclude that cache location is a choice between saving in network and cost of cache solution: bandwidth saving is most important when caches are closed to end-users but necessarily in bigger number so with a more expensive cost, may be compensated by a lower cost of equipment.

3.1.6.4.8 QoE for end-users

The quality of experience is measured comparing download time without and with cache presence.

End-user leecher is monitored to know the delay time to get VoD from ENVISION overlay network.

We consider ratio between the 2 DL speeds.

3.1.6.4.8.1 DL speed without cache

In a first test, seeder is in AS2 and we consider a unique leecher in AS1. Cache service is not available.

3.1.6.4.8.2 DL speed with cache

In a second test, cache service is available in AS1. Cache ingests VoD.

The leecher receives chunks from seeder and cache.

In this test, cache is considered as unique Envision client.

Then we add to cache multiple ENVISION client for the same VoD.

For each test context, we measure the new DL speed on leecher.

3.1.6.4.8.3 Measurements

Tests	Seeder	AS Seeder	Leecher	AS Leecher	DL speed	Notes
Reference	1 (OS)	AS2	1	AS1	4'45"	Reference
T1	OS + 2 caches	AS2 + AS1	1	AS1	4'45"	DL equality between seeders

T2	OS + 8 caches	AS2 + AS1	1	AS1	4'45"	DL equality between seeders
T3	OS + 8 caches	AS2 + AS1	1	AS1	4'45"	Link between AS1 and AS2 is in congestion
T4	OS + 2x10 caches	AS2 + AS1	1	AS1	4'45"	

All tests give same value for download times. This is due to LaBRI client which downloads VoD as streaming so with VoD bitrate.

Confirmation is made with T3 test where link between the origin seeder in AS2 and AS1 is in congestion. The traffic before congestion is equalized between all seeders (OS and caches) and, during congestion, is equalized between seeders in AS1 (clients/seeders in caches), OS giving nothing in this case.

3.1.6.5 Conclusions

Similarly to multicast, the motivation for offering a caching network service through CINA lies primarily with the traffic saving an ISP can extract from localising the demand for content within its own network. The saving increases with the storage and bandwidth resources dedicated to the distribution of a particular stream. In the tests undertaken in the FT testbed, the caching resources are modelled as independent seeders of equal upload capacity instantiated at a particular domain. The saving on reducing traffic is substantial (50%) already with the introduction of a single seeder and can grow up to 90% when 10 seeders are added. This saving depends on the ratio of upload to download bandwidth among the existing seeders and leechers in the overlay; the higher the ratio the smaller the impact of adding more seeders by the ISP. This can be seen using an overlay configuration with 6 seeders; the gain by adding another 10 seeders now drop from 90% to ~60%. With this in mind, the allocation of ISP caching resources to the distribution of particular streams should take into account the ratio of the overlay's upload resources to download demand.

3.1.7 CDN Request Routing

3.1.7.1 Evaluation Scenarios & Metrics

The main testing scenario is shown in Figure 44, where we can identify the following elements:

- The *DNS* is the frontend system for finding content in the CDN from a user's perspective. It transforms and relays users requests to the *Tracker*, the system that knows what and where is the content currently managed by the CDN, and performs *consistent hashing* operations for locating the closest nodes with the help of the *Topology Server*, the system where topology information is stored.
- The *CDN nodes* keep the content we will serve to users. In our tests, they can be seen as *intelligent* web servers that stream content to users by using many different streaming protocols. For our tests, we have selected *Smooth Streaming*, a popular streaming solution provided by Microsoft.
- The *users simulator* is our own proprietary software for stress testing and report generation. The simulator performs multiple downloads in parallel, generating video download requests to the CDN, and then emulating regular users by following the same patterns users do when

watching online videos (i.e., skipping and jumping in the video playback) ¹. The simulator finally generates some reports where we can see the results.

CDN nodes are high performance machines: *blade* servers with 15000 rpms disk, connected with, at least, 4Gbps links. So, in order to keep these valuable resources available for real users, we have restricted the group of nodes used in our tests to the machines listed in Table 6. By doing this, our downloads have not interfered with the regular CDN operations.

The aim of these tests is to verify that the topology information is essential for the operation of the CDN. So we have compared the results obtained in following testing scenarios:

- In the first test, we have used random topology information. Our *topology server* has been loaded with random information (i.e., a cost matrix with random values), so the *CDN tracker* does not know the optimal nodes for serving our video content to users.
- In a second test, we have repeated the first scenario but we have loaded the *topology server* with the regular topology information used in the CDN, using an optimized costs matrix. We will show that this scenario produces an optimal result for users.

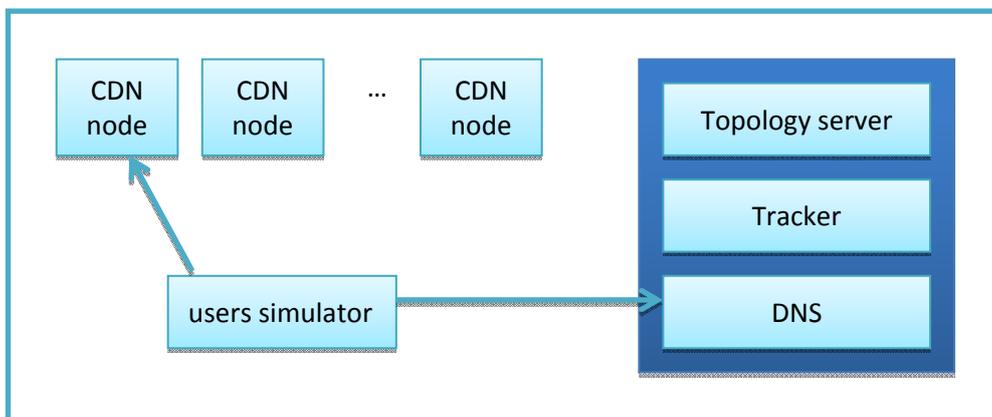


Figure 44: CDN testing scenario

IP	Location
81.45.8.5	Madrid
81.45.8.6	Madrid
81.45.8.7	Madrid
81.45.8.37	Barcelona
81.45.8.38	Barcelona
81.45.8.39	Barcelona
94.142.115.146	London
94.142.115.150	London
94.142.115.154	Frankfurt

¹ Kangaroo: Video Seeking in P2P Systems (http://static.usenix.org/event/iptps09/tech/full_papers/yang/yang_html/)

94.142.115.158	Frankfurt
94.142.115.178	Paris
94.142.115.182	Paris
94.142.115.218	Paris
66.201.186.110	Washington
66.201.186.114	Washington
66.201.186.118	Washington
94.142.115.186	Prague
94.142.115.42	Prague

Table 6: CDN nodes

In both tests, our *users simulator* has performed the downloads in parallel from a machine located in Spain, requesting a list of *URLs* that correspond to short (5 minutes) videos, encoded with 2Mbps bitrate and served with the *Smooth Streaming* protocol. As we do not want to stress the CDN nodes, new downloads will be started at a rate of one download per second. And, in order to increase the accuracy of our simulation, the *users simulator* selects *URLs* from the list following the *Zipf* distribution, a mathematical model frequently used for representing video popularity on Internet where 20% of the videos count for 80% of the requests.

We have used several metrics for evaluating the quality of service from a user perspective.

- *Response time*: the time elapsed from the request starts until the user receives the first video data.
- *Buffering time and underflow rate*: the *buffering time* is the amount of data, in seconds, the user's player is keeping in the buffer. *Underflow* will happen when the buffering time drops to 0, resulting in video playback errors.
- *Global throughput*: the sum of the throughput obtained by all the downloads.

In the following section, we will show the reports for these metrics that our *users simulator* has generated. We have discarded the first two runs of the simulator as we want to be sure that CDN nodes have "*pre-load*" the video content in their disks, and this could lead to some instability in the results.

3.1.7.2 Evaluation Results

The first difference between optimized and random topology is found in the response times from the CDN nodes. We show the results obtained with the optimized topology in Figure 45, where we can see that more than 50% of the connections are established in less than 220 milliseconds.

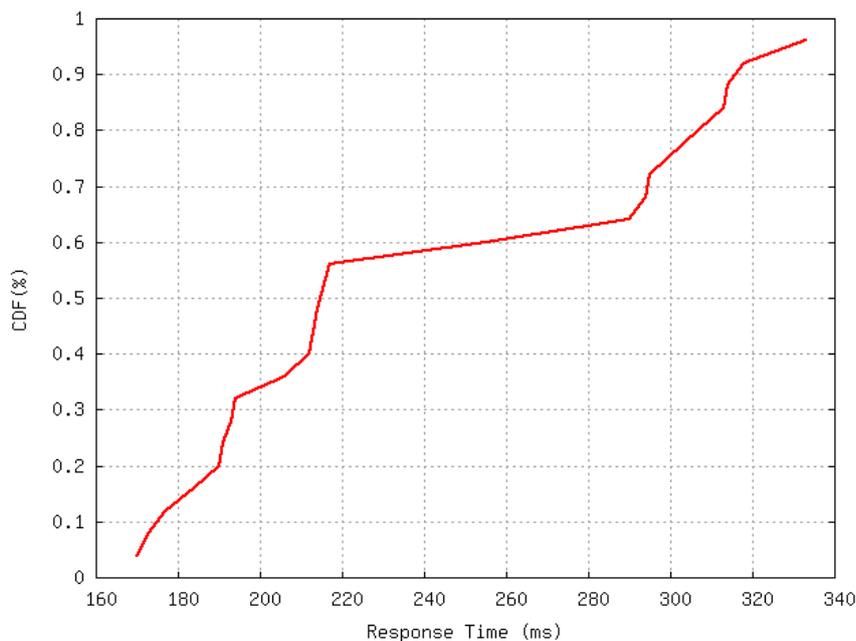


Figure 45: response time with optimized topology

In contrast, Figure 46 shows the results obtained with the random topology information, where the average response times are slightly higher, around 250 milliseconds. This matches the expected behaviour, as most of the connections will be established to nodes in Europe, and we can expect latency in the range from 10 to 30 milliseconds.

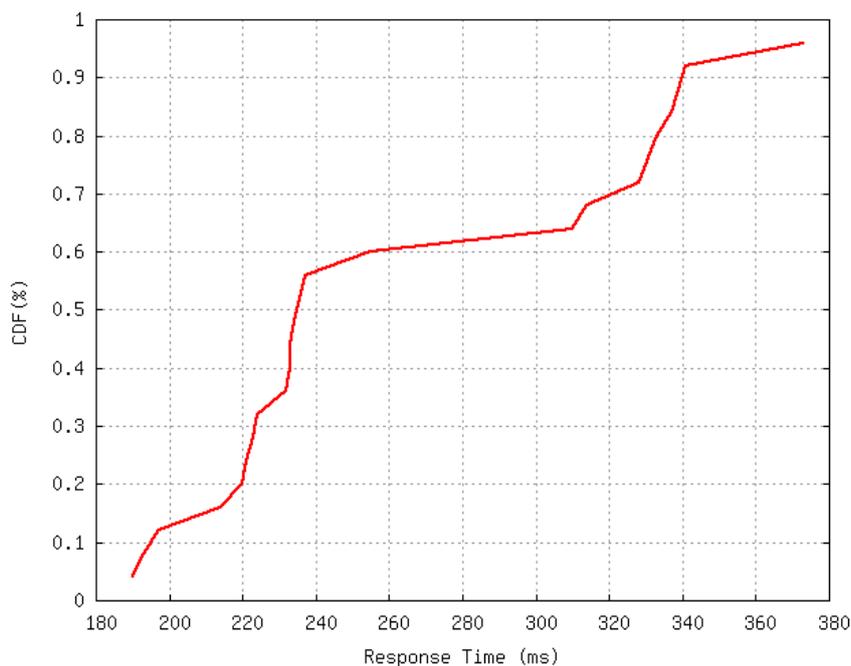


Figure 46: response time with random topology

The locality is also shown in the system throughput obtained in both scenarios. Figure 47 shows the results obtained when using optimized topology information. The average system throughput is about 50Mbps, the amount of bandwidth required by 25 downloads, each one with 2Mbps. The CDN nodes used for performing these downloads are all located in Spain, as the *Tracker* has returned the optimal nodes for the location of our *users simulator*. Throughput is steady, and there are no

significant spikes or valleys that could be seen as a problem in the bandwidth available. There are no visible bandwidth restrictions in the downloads, and the content is obtained at the proper speed from the CDN servers.

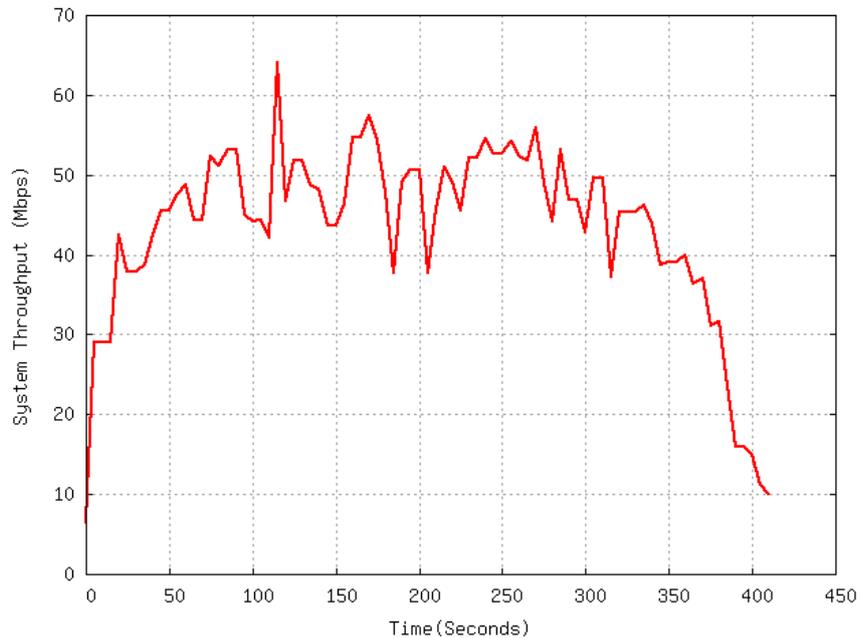


Figure 47: system throughput with optimized topology

In contrast, Figure 48 shows the throughput obtained when no topology information is used by the CDN. The average global throughput is around 40Mbps, 10Mbps less than the required 50Mbps, and the video quality will suffer some problems because of this bandwidth deficit. Downloads also last more than the required 300 seconds, going beyond 400 seconds.

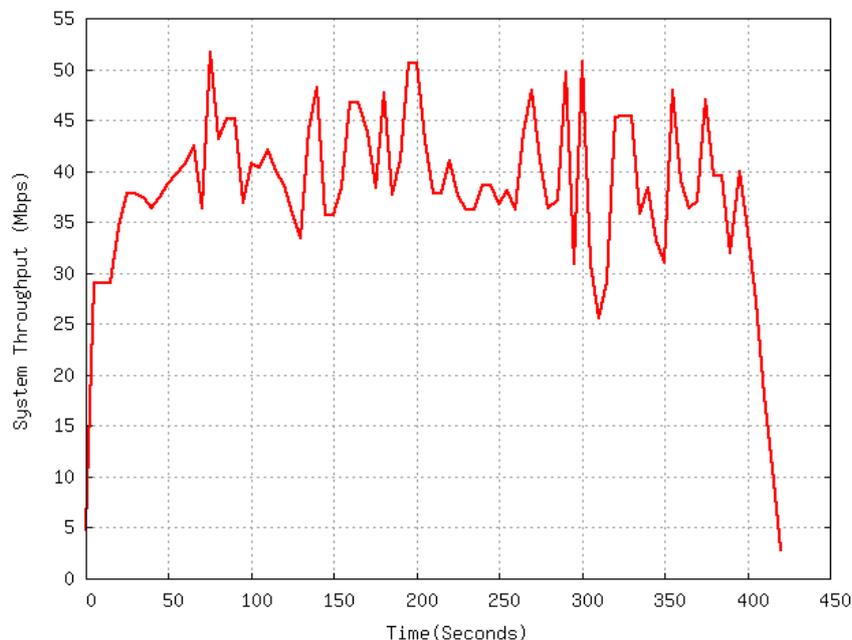


Figure 48: system throughput with random topology

The immediate consequence of the system throughput is the buffering time in the users players. In Figure 49 we see how the users keep usually at least 1 second of buffering in their players when

downloading content from servers in Spain, more than enough for keeping a smooth playback for this kind of video.

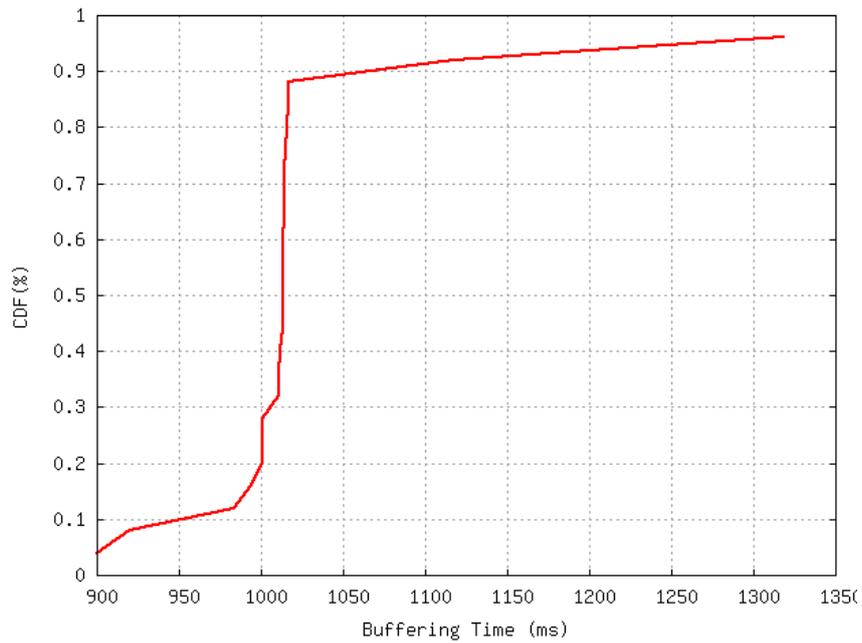


Figure 49: buffering time (CDF) with optimized topology

In contrast, downloading from a group of suboptimal CDN nodes do not produce the same results. We can see the results in Figure 50, where we did use the random topology information. The standard deviation for the buffering time is clearly wider and, as many connections must have been running low on buffering, some downloads have had underflow problems. This has probably led to frequent playback interruptions due to an insufficient data flow.

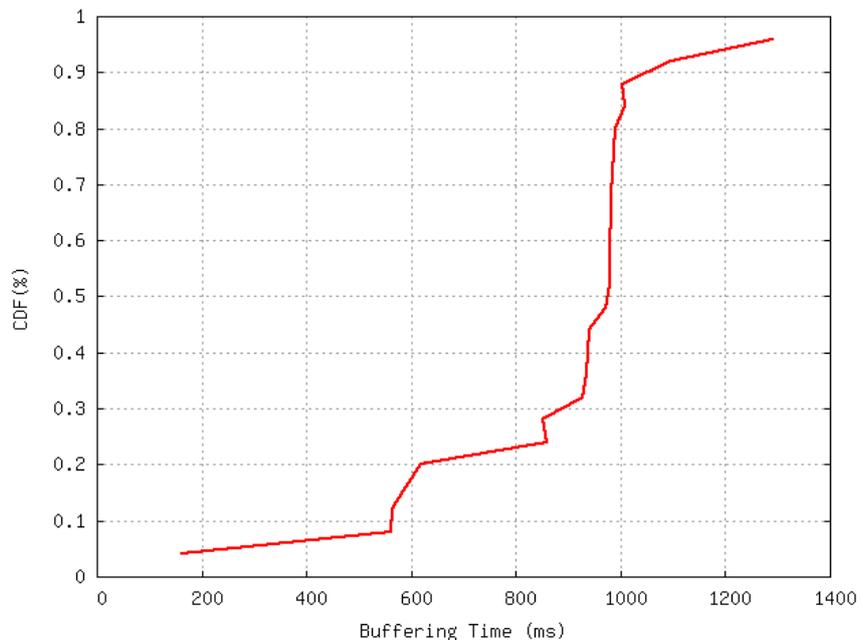


Figure 50: buffering time (CDF) with random topology

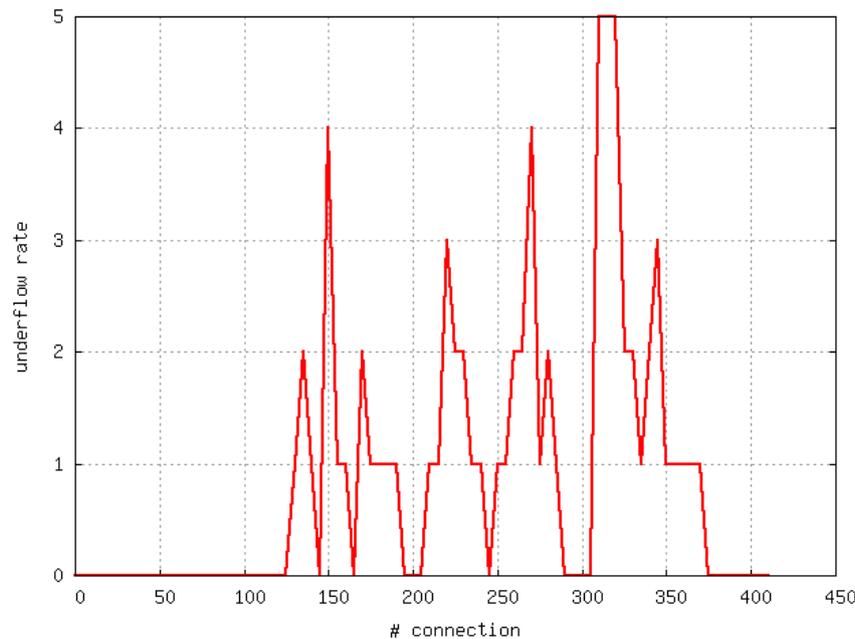


Figure 51: underflow rate with random topology

We can confirm this in Figure 51, where we can see the underflow rate with random topology information. When using optimal topology information, the number of underflow events was zero, meaning that there was not a single download where the amount of data buffered in the player dropped to zero. But with random topology information, we see a constant number of connections in underflow state. We have seen up to 5 underflow events at the same time, representing 5 players that stop because of insufficient data. This represents an important quality degradation for users, as they have to wait until the playout buffer is filled again with data and they can continue the playback.

In conclusion, optimal topology information is essential for the proper operation of the CDN, with evident results both for users, with less bandwidth and higher latencies, and for CDN owners, with higher operational costs.

3.1.7.3 Conclusions

This section investigates the benefits a CDN operator can gain from the use of CINA costs representing locality to perform the routing of a request to the best available replica. The evaluation is performed in TID's distributed testbed with 3 CDN nodes located in the US and 15 nodes across Europe. The particular metrics considered capture the impact on the performance perceived by the customer and include the application response time, the buffering time, the underflow rate and the global throughput. Response times are improved from 250ms to 220ms with the selection of local CDN nodes for each emulated user request. The impact on throughput is more dramatic, with a drop from 50Mbps to 40Mbps for the sum of all the content downloads, which in turn impacts the amount of data in the buffer causing buffer underflows that translate to player freezing and re-buffering events for a significant number of connections.

3.2 Simulations and Theoretical Analysis

3.2.1 ISP Preferences²

3.2.1.1 Objectives and Evaluation Requirements

The objective of the ISP preferences work is to create a method by which customer facing ISPs can move traffic in both space (by shifting it between transit links) and in time (by shifting it to different hours of the day) in order to reduce their transit bills. By taking advantage of the fact that transit is typically billed at the 95th percentile, ISPs can save money by "flattening" their traffic profile, moving traffic from the peak to the off peak. If different transit links are billed at different levels or if they have their peaks at different hours of the day then shifting traffic between links can also reduce bills.

The method proposed here enabling customer ISPs to guide traffic to different links or different times of day (in line with customer preferences) needs to satisfy the following twofold objective:

- 1) Reduce the transit bills,
- 2) Reallocate traffic in a "stable" manner (traffic is not continually shifted and then shifted back between links).

To meet the first part, it is simply necessary to model the reduction in price for the customer ISP. As ISP pricing is typically a closely kept secret, it is necessary to evaluate the model within a variety of pricing scenarios. To meet the second part, the concept of equilibrium is introduced. Equilibrium in this case is a measure of how far the assigned traffic is from an assignment which minimises the price it experiences. The distance from equilibrium for a given flow f_i (where i specifies both the time and the link to which the flow is assigned) is given by $f_i(c_i - \min_j c_j)$ where c_i is the "cost" of the time/link i and the sum is over all possible j that this flow could choose between. Note that the costs c_i and c_j are functions of the assigned flow pattern. Generally speaking as more flow is assigned to i then the cost rises. At equilibrium, this expression is zero for all flows – that is, flows are assigned to minimal cost times/links, as defined above.

3.2.1.2 Conclusions

This section evaluates a technique for encouraging applications to change their selection of servers and schedule their data transmission at different times in order to reduce the ISP transit costs by shifting traffic to less expensive paths and lowering their peak-time utilisation. Results show remarkable potential cost improvement with 15% cost reduction in the most conservative cases where 20% of users comply to shift their traffic in time or in space.

This research showed the feasibility for significant savings for ISPs by performing traffic balancing on the basis of 95-th percentile costs. However, for this balancing to become feasible in practice, network user software should be made aware of the ISP cost preferences. This can be achieved using the CINA interface, which then becomes central to cost-saving traffic engineering. However, our research also suggested that one of the most important topics for further research is the provision of incentives for users; a higher proportion of users opting in to perform time and space shifting leads to uniformly higher savings. This means that relevant studies in behavioural economics should be made for specific ISP billing and incentives scenarios.

² The main technical contributions of the ISP preferences work - the problem formulation, algorithm design and simulations results - have been suppressed from the public version of this deliverable as the content is currently under review for publication.

3.2.2 ISP Preference Consolidation

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

3.2.2.1 Objectives and Evaluation Requirements

We will focus our evaluation on the degree to which the overlay behaviour induced by a given consolidation strategy aligns with the preferences of ISPs as expressed by their costs $c(l, m)$. We will start by defining a set of measures that will allow us to assess the performance of our proposed preference consolidation strategies.

3.2.2.2 Evaluation Scenarios & Metrics

3.2.2.2.1 Performance Metrics

The first measure that we define is the *full Inter-PID preference cost* $f(l, m)$, that models the preference cost imposed upon PID l by all overlay links between l and m . Formally,

$$f(l, m) = k \left(N(l)p(l, m) + N(m)p(m, l) \right) c(l, m), \quad (1)$$

where $l \in \mathcal{P}$ denotes a *local* PID that has a number of overlay links with PID $m \in \mathcal{P}$, $N(l)$ the population of l , k the number of neighbours of each overlay node, and $p(l, m)$ the probability of a node in l choosing a neighbour in m .

To model the preference cost that PID l incurs from overlay links both originating and terminating on its internal nodes, we simply aggregate $f(l, m)$ over m . This allows us to define the *ISP Cost* $\phi(i)$, that models the preference cost that ISP i incurs from overlay links both originating and terminating on its internal PIDs. Formally, we have that

$$\phi(i) = \sum_{l \in \mathcal{P}_i} \sum_{n \in \mathcal{P}} f(l, n). \quad (2)$$

The expressions that we have just defined for $f(l, m)$ and $\phi(i)$ will allow us to estimate the overall preference cost that our proposed consolidation strategies have on the ISPs providing connectivity services to the overlay. We now provide a measure that directly assess the degree to which the outcome of a consolidation algorithm matches the preferences of each given ISP as expressed in their advertised OIC costs. To achieve this, we define the *preference concordance* $\delta(l)$, a measure for the *cosine similarity* between the full Inter-PID cost $f(l, m)$ and the generic preference cost $c(l, m)$ for a given PID l . We define $\delta(l)$ as

$$\delta(l) = \frac{\sum_{n \in \mathcal{P}} f(l, n) c(l, n)}{\sqrt{\sum_{n \in \mathcal{P}} f(l, n)^2} \sqrt{\sum_{n \in \mathcal{P}} c(l, n)^2}}. \quad (3)$$

Conceptually, $\delta(l)$ measures the cosine of the angle subtended between $f(l, m)$ and $c(l, m)$ if l is fixed and both are treated as vectors. Hence, if for PID l the full Inter-PID costs $f(l, m)$ induced by the topology construction function $p(l, m)$ have a similar structure as the ISP preference costs $c(l, m)$, we have that $\delta(l) \approx 1$ and the concordance is high. Conversely, if the structures of $f(l, m)$ and $c(l, m)$ are dissimilar, the costs induced by $p(l, m)$ are very different from the $c(l, m)$ expressed by the ISP, $\delta(l) = 0$ and the concordance is low.

3.2.2.2.2 Evaluation Scenarios

We use a simulation-based approach to evaluate a series of scenarios. In order to directly compare the results between simulation runs, the number of ISPs, PIDs and nodes was kept constant for all runs. In particular, all our simulation runs consider an overlay with $N_p = 10^6$ nodes, $N_p = 10^3$ PIDs, $N_I = 10^2$ ISPs, and $k = 10$ outgoing overlay links per node. However, within these constraints, we did vary the internal structure of the overlay by varying both the number of PIDs per ISP and the number of nodes per PID. In both cases, due to its simplicity and flexibility we used a Zipf distribution to assign sizes to ISPs and PIDs by their rank. This means that the j -th biggest PID will have $N(j) = N_p z(j; s, N_p)$ nodes, where $z(j; s, N)$ is the Zipf density

$$z(j; s, N) = \frac{j^{-s}}{\sum_{n=1}^N n^{-s}}. \quad (4)$$

Similarly, the j -th biggest ISP will have $N_p z(j; s, N_I)$ PIDs. By changing the characteristic exponent s it is possible to explore different degrees of variability in the sizes of ISPs and PIDs. If s is small, $z(j; s, N)$ will be very similar for all ranks, and we have a situation where all ISPs have approximately the same number of PIDs, and where most PIDs have approximately the same number of nodes. Conversely, for larger s , $z(j; s, N)$ will decrease quickly with rank, leading to ISPs with widely varying numbers of PIDs and PIDs with widely varying numbers of nodes. We do not claim that this is a realistic model for a specific class of overlays. Rather, we focus our work on the analysis of the inherent properties of consolidation strategies, and use (4) as a controlled mechanism to generate a diverse set of overlay topologies for simulation.

In this section we present results for a number of values of s in the interval $[.1, 1.5]$. Results for the values $s = \{.1, .5, 1, 1.5\}$ are shown in Figure 58. The overlay structures induced by these values go from an almost homogeneous scenario in which there are approximately 10 PIDs per ISP, 10^3 nodes per PID, and 10^4 nodes per ISP (when $s = .1$) to a very heterogeneous scenario where these values range over 2 to 4 orders of magnitude across all ISPs and PIDs (for $s = 1.5$). Hence, the results presented are representative of a very wide variation on the ISP and PID structure of a large application-layer overlay. For the **Low Cost** consolidated topology construction strategy, we used

$$q = \frac{1}{10} N_p.$$

With regards to the preference costs $c(l, m)$, we stress that they are not meant to be representative of actual interdomain billing costs. Instead, they are meant to convey the *relative* preferences of ISPs regarding which egress links to use. Since ISPs could potentially advertise their preference costs over widely different ranges, we assume that the overlay normalises them before processing so that $\sum_{n \in P} c(l, n) = S_c$; this facilitates preference cost comparisons between PIDs and eliminates the problem of some PIDs reporting arbitrarily high costs. Given the lack of asymmetric preference cost datasets, we decided to generate $c(l, m)$ artificially by uniformly drawing values from the interval $[0, S_c]$ subject to the normalisation restriction; the resulting CDF is shown in Figure 58. In order for preference cost comparisons to be valid, the consolidated values $\kappa(l, m)$ were also normalised so that $\sum_{n \in P} \kappa(l, n) = S_c$. For these tests, we used $S_c = 10^3$.

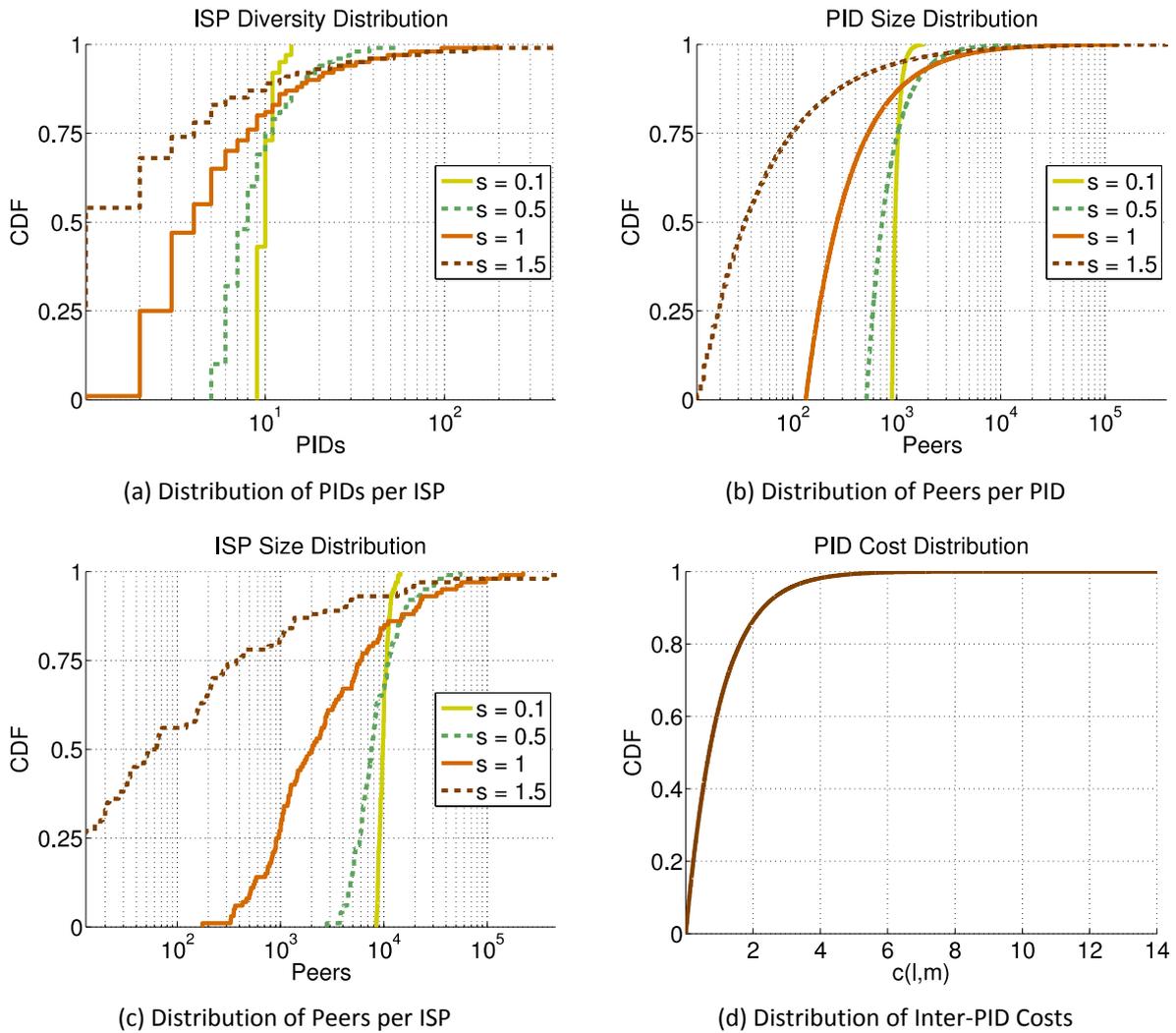


Figure 52: Topology Construction Scenarios

In order to quantify overlay-wide performance, we define two measures: the *total normalised overlay cost* Φ and the *total normalised preference concordance* Δ . These are designed to be overlay-wide versions of (2) and (3), and are defined as

$$\Phi = \frac{N_p}{N_p k S_c} \sum_{i \in I} \phi(i), \quad \Delta = \frac{1}{N_p} \sum_{l \in P} \delta(l).$$

Intuitively, Φ represents the overlay-wide average preference cost per overlay link once it has been normalised by $\frac{S_c}{N_p}$, the average $c(l,m)$ associated with any PID l ; Δ represents the average preference concordance over all PIDs in all ISPs.

3.2.2.3 Evaluation Results

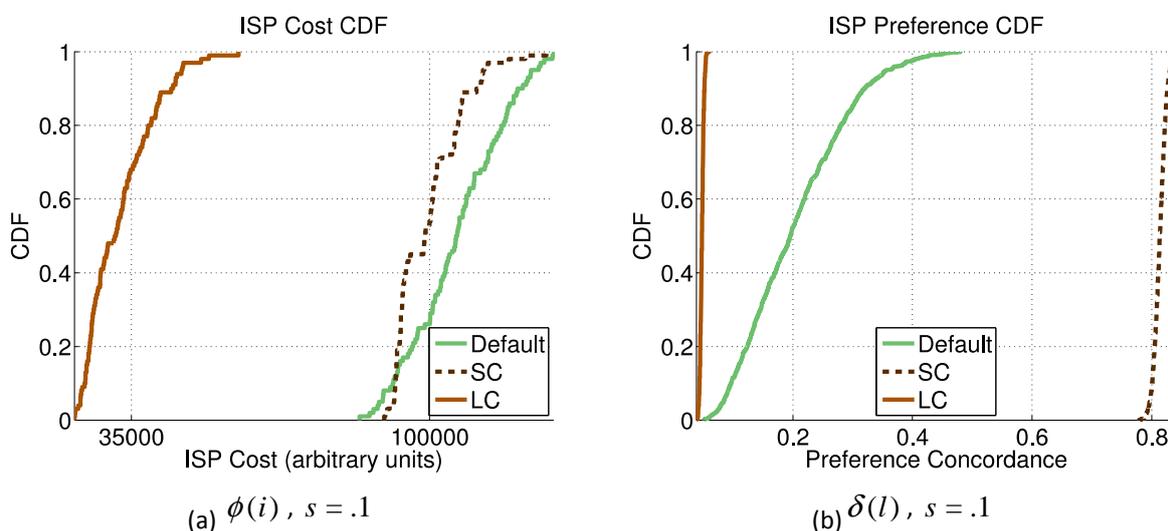
The result of four sample simulation runs for $s = \{.1, .5, 1, 1.5\}$ are shown in Figure 59. Each one of the curves in each graph correspond to one of the three consolidated topology construction strategies presented. In Figure 59a, we see that for very homogeneous overlay structures ($s = .1$), the **Default** consolidation strategy produces consistently higher preference costs $\phi(i)$ for many ISPs when compared with both **SC** and **LC**, with **LC** producing the lowest preference cost. However, as

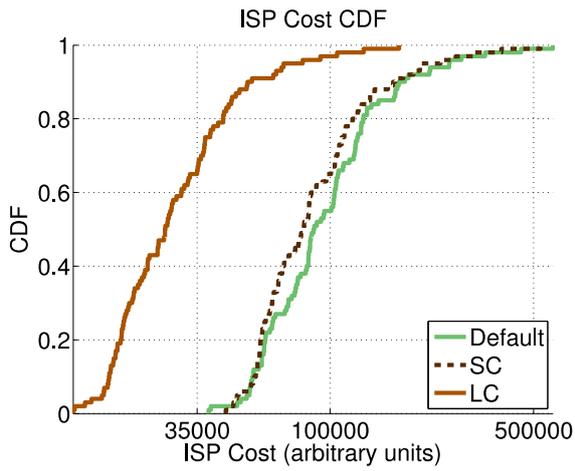
shown in Figure 59b, **LC** also produces overlays with very low concordance values $\delta(l)$, which may be unattractive for ISPs as they greatly deviate from their preferences $c(l,m)$. For these homogeneous overlays, **SC** not only achieves slightly lower preference cost, but also a greatly improved concordance of approximately .8. The **Default** consolidation strategy achieves a maximum concordance of around 0.4, and has a significant spread, indicating that the degree of alignment between $c(l,m)$ and $f(l,m)$ varies greatly between PIDs.

As overlay heterogeneity increases to $s = .5$ in Figure 59c and $s = 1$ in Figure 59e, we see that the preference cost benefits provided by **LC** are reduced, since its $\phi(i)$ CDF extends further to the right. In addition, the CDFs for **SC** and **Default** increase their variability, including more ISPs with both smaller and larger $\phi(i)$. As a result of the increasing number of smaller ISPs, this manifests itself as a *shifting* of the CDFs towards the left. However, since a small number of much bigger ISPs are added as well, the expectation of the cost distributions exhibit limited change. Regarding the concordance $\delta(l)$, the CDF of **SC** is quickly shifted to the left first with $s = .5$ (Figure 59d) and further still with $s = 1$ (Figure 59f), implying that as overlay heterogeneity increases the preference alignment of **SC** with $c(l,m)$ is quickly reduced for many PIDs. The concordance curve for **Default** is also shifted to the left, but at a much slower rate. The **LC** concordance experiences little change.

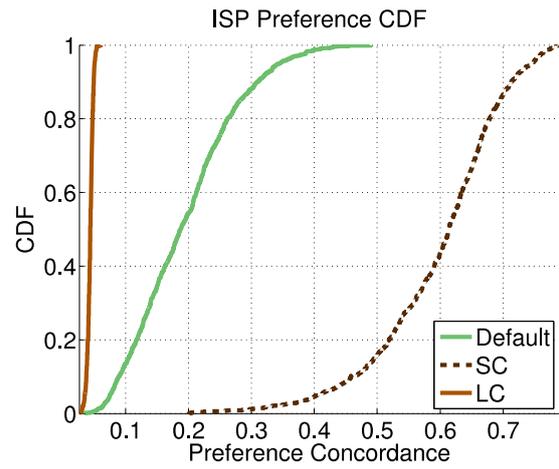
For the most heterogeneous case studied ($s = 1.5$), Figure 59g shows little evidence for a change in the CDF of $\phi(i)$ for **LC**, **SC** or **Default**. However, the CDF of $\delta(l)$ continues shifting to the right, with **SC** and **Default** becoming indistinguishable.

In order to track changes in Φ and Δ more accurately, we performed : 36000 simulation runs for a sequence of values of $s \in [.1, 1.5]$. The results of these simulations are presented in Figure 60. Each data point in Figure 60a and Figure 60b represents : 3000 simulation runs, with the error bars marking the 10-th and 90-th percentiles of the Φ and Δ distributions respectively.

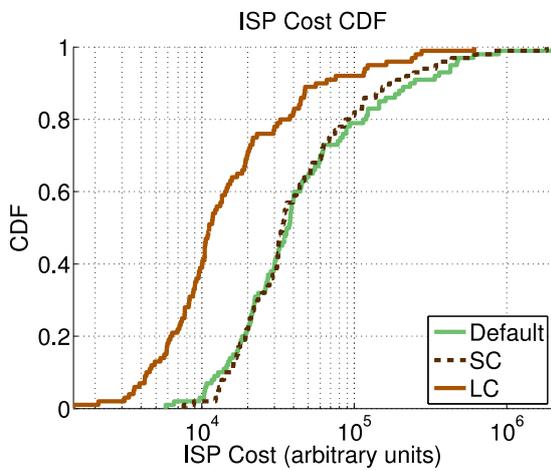




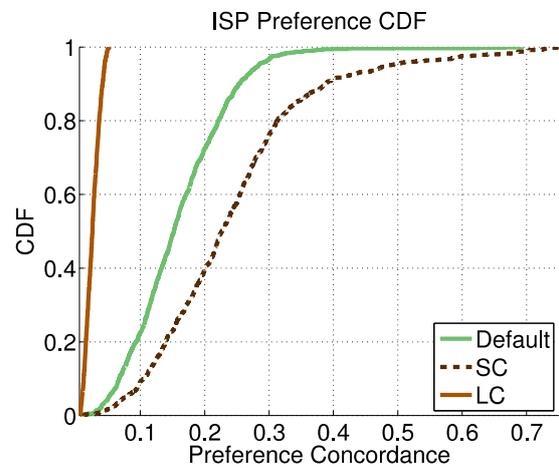
(c) $\phi(i), s = .5$



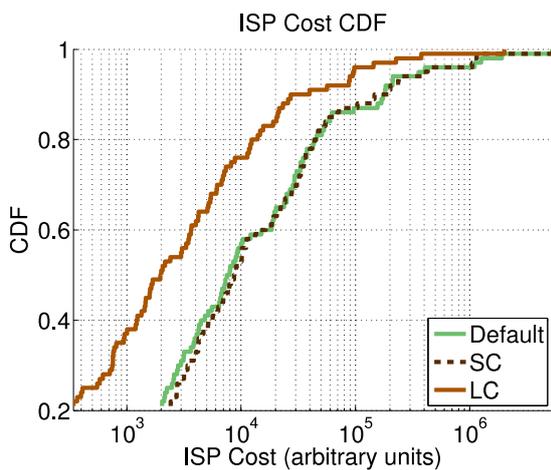
(d) $\delta(l), s = .5$



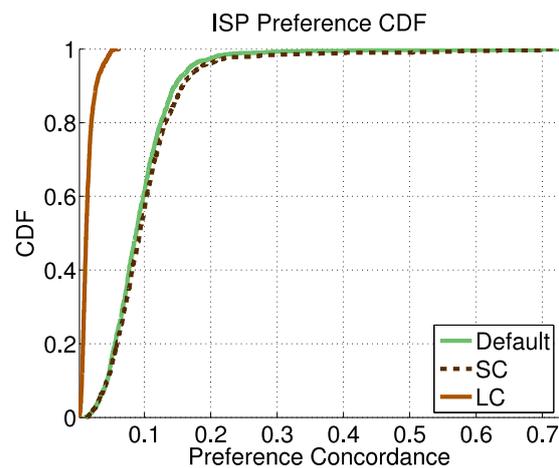
(e) $\phi(i), s = 1$



(f) $\delta(l), s = 1$



(g) $\phi(i), s = 1.5$



(h) $\delta(l), s = 1.5$

Figure 53: Simulation results for $s = \{.1, .5, 1, 1.5\}$

The general conclusions derived from the $\phi(i)$ and $\delta(l)$ CDFs in Figure 59 are confirmed by this analysis. First, we note that as s increases, Φ for the **Default** and **LC** consolidation policies remains

essentially unchanged in the mean, but exhibits increased variability. For **SC**, both Φ and the variability of the distribution increase slowly with s , with these increases becoming much more visible for $[1 \leq s \leq 1.5]$.

Finally, regarding Δ we see that for small values of s the **SC** policy has vastly higher concordances than those of either **LC** or **Default**. However, the expected Δ of **SC** drops very quickly as s increases, approaching that of **Default** for $s \geq 1.2$. This drop in concordance can also be seen for **Default** and **LC**, but these two cases it is much less pronounced.

The main conclusion that can be drawn from these results is that, for *all* values of s considered (i.e. ISP and PID size distributions), **SC** provides higher concordance Δ at a lower preference cost Φ when compared with **Default**. As shown in Figure 59, these properties are also present in terms of the CDFs of both $\phi(i)$ and $\delta(l)$, with the $\delta(l)$ CDF of **SC** generally to the right of that of **Default**, and with the $\phi(i)$ CDF of **Default** generally to the right of that of **SC**. Hence, both when considering preference cost or concordance, **SC** is more compatible with the preferences of ISPs than **Default**. However, as shown in Figure 60b this is critically dependent on the structural properties of the overlay, with **SC** performing much better for overlays with more homogeneous ISP and PID structures. A more detailed analysis of this issue, based on both overlay measurements and appropriate modelling based on the Internet AS topology, is left for future work.

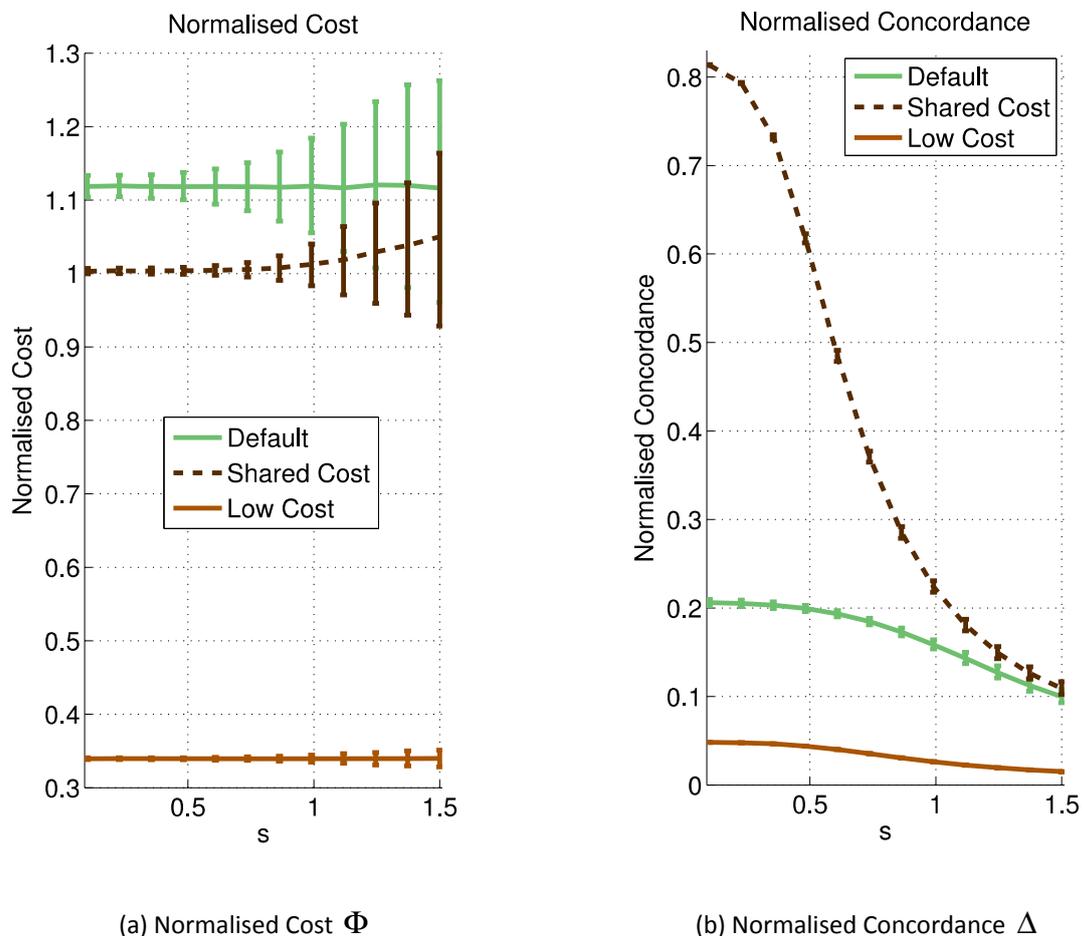


Figure 54: Overlay-wide simulation results for $.1 \leq s \leq 1.5$

3.2.2.4 Conclusions

This section elaborates on the related application-layer problem of consolidation preferences from different ISPs. Two methods are evaluated both resulting in improvements in total ISP cost savings when considering the preferences of all the ISPs as opposed to only the ISP in one end of an overlay connection. These improvements can be as high as 70% when fairness between different ISPs is not considered important.

The main finding of this section, that certain ISP preference consolidation mechanisms achieve better trade-offs between disjoint sets of ISP preferences, suggests that preference consolidation may be of great importance when considering the widespread deployment of the CINA interface. Moreover, it showed that for uncorrelated, uniform preferences, *averaging* (an operation explicitly seeking to achieve a trade-off between ISP preferences) achieves a better alignment with the preferences of all ISPs when compared to policies leading to unilateral action (such as minimising cost). Hence, practical deployments of CINA consolidation explicitly based on achieving trade-offs may be easier to implement in practice. In cases where ISP preferences are correlated with the network service quality experienced by the overlay, further work is needed to elucidate the best trade-offs.

3.2.3 Distributed Data Management System³

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

3.2.3.1 Objectives and Evaluation Requirements

- Performance evaluation of the n-casting application overlay under various conditions and different popularity distributions for the resource groups:
 - Due to its impact on building high-performance overlays, the *accuracy* of message delivery is one of the most important performance parameters of the n-casting system. Informally, this is a measure of how far away from the optimum is the geographic locality performance of the n-casting system.
 - Study the speed of query resolution, which is one of the main motivations behind the design choice of distributing the membership indexes using locality rather than the group key space; with area-specific indexes, queries for groups with local members can be resolved locally, traversing shorter distances and therefore incurring shorter response times.
- 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 connections and the frequency of updates in the system.

3.2.3.2 Conclusions

This section elaborates on the evaluation of a distributed data management infrastructure built to enable n-casting request forwarding, used in distributed applications to discover a number *n* of resources that participate in the distribution of a particular video stream and are closer to the node issuing the request. The system is evaluated using a dataset of about 450K nodes with large geographical coverage. The system is proved to be very accurate in discovering the closest resources, with a small message overhead scaling well with the number of nodes and high resource availability churn.

³ The main technical contributions of the Distributed Data Management System work - the problem formulation, algorithm design and simulations results - have been suppressed from the public version of this deliverable as the content is currently under review for publication.

This section showed that scalable application-layer n-casting systems with high accuracy and bounded query response time are feasible, and that real-time performance information, when used as an input to host clustering algorithms, can be the basis for innovative network services. However, it also illustrated some of the complexities that can be expected from a practical implementation. Examples of these challenges include message overheads, scalable address aggregation and Internet-wide measurement of n-casting metrics. This challenges notwithstanding, the usefulness of an n-casting system makes the presented design a valuable first step.

3.2.4 Distribution Tree Optimisation⁴

The objective is to evaluate the performance of the Distribution Tree Optimisation algorithm as presented in [D4.3] and understand the importance of the placement of the nodes and the impact of the underlying network topology.

3.2.4.1 Objectives and Evaluation Requirements

We use simulations on a variety of network topologies and sets of facility candidates. For constructing the topologies we used available tools such as the Rocketfuel POP-level topologies available from [RFUE01]. For cost modelling we adopted a model based on Amazon EC2 pricing scheme [AMAZ01] for the parameterisation of internet data transfer and service node instantiation cost.

3.2.4.2 Conclusions

This section looks into minimising the cost for interactive video applications with an upper bound in application layer delays. Such applications can operate at arbitrary locations without relying on fixed infrastructure, by mobilising CINA High Capacity Node (HCN) resources. The developed optimisation algorithm is evaluated using a cost model for the CINA services derived based on the Amazon EC2 pricing scheme. The results show that the use of HCNs significantly improve the experienced application delay and that the optimisation algorithm achieves a reduction of the operator costs from 0.42\$/hour/peer to 0.21\$/hour/peer for an overlay topology of 200 peers.

3.2.5 Content Adaptation⁵

3.2.5.1 Smoothing

In this section, we present the performance evaluation of our proposed playout smoothing mechanisms for layered streaming.

We first need to define the relevant metrics that reflect the key features of perceived video quality. We focus mainly on three important points: layers changes, longest sequence and the unused chunks.

Layer changes: Represents the number of layers changes during the studied video stream.

Longest sequence: Represents the size of the longest smoothed sequence of the stream (sequence of the same quality level).

Unused chunks: Represents the number of chunks that are sacrificed (not played) in each smoothing window, in order to ensure the smoothness.

⁴ The main technical contributions of the Distribution Tree Optimisation work - the problem formulation, algorithm design and simulations results - have been suppressed from the public version of this deliverable as the content is currently under review for publication.

⁵ Parts of this section covering the results of the bandwidth allocation algorithms have been suppressed from the public version of this deliverable as the content is currently under review for publication.

Three scenarios are considered in this study. First we present a real scenario of smoothing and we show the difference in terms of layers change between the proposed smoothing mechanisms comparing to the row stream. In second time, we are interested in the impact of the smoothing window size on the performance of the proposed smoothing algorithms. Finally we compare the frequency reduction algorithm with the Layer2P2 in terms of number of layers changes.

3.2.5.1.1 Scenario 1

In order to evaluate the three smoothing strategies, namely amplitude reduction and frequency reduction (prefetching-based and mean-base) described in D5.3, we consider a scenario where the video stream is composed of 8 layers (each layer is streamed at 100 kbps) under bandwidth variation (from 100 kbps to 900 kbps). The duration of the video is 400 seconds. We fixed first the smoothing window size at 15 seconds, and we study the behaviour of the three proposed algorithms.

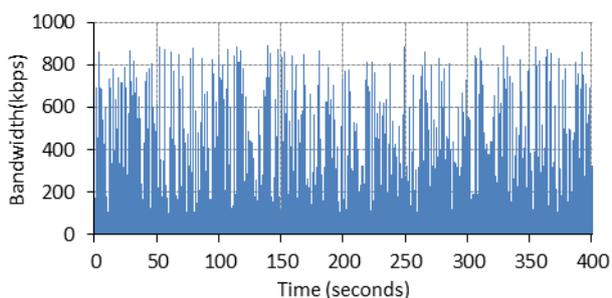


Figure 55: Bandwidth variation

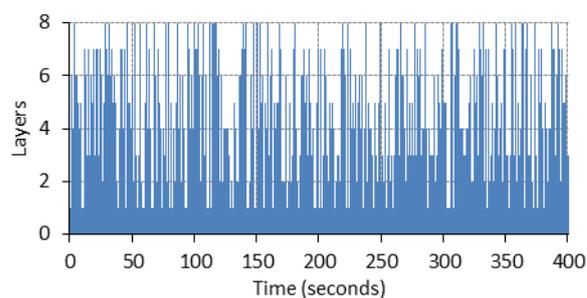


Figure 56: Row stream

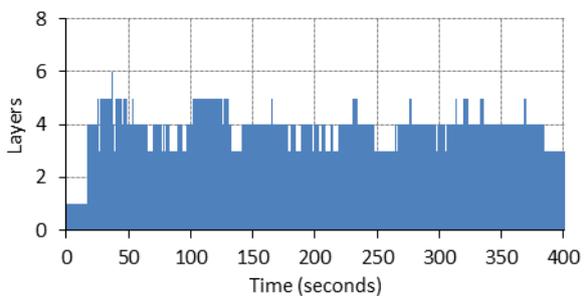


Figure 57: Amplitude reduction

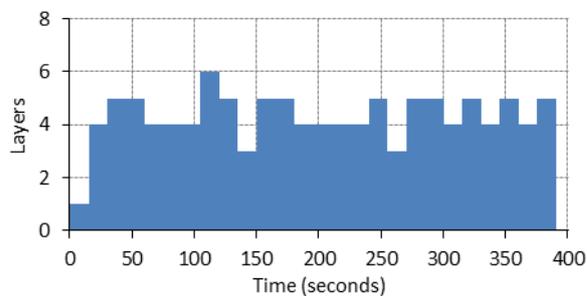


Figure 58: Frequency reduction (prefetching-based)

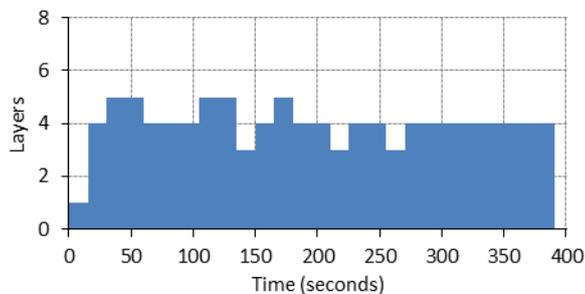


Figure 59: Frequency reduction (mean-based)

Figure 76 plots the aggregated download bandwidth in the receiver peer. Figure 77 represents the layers changes without any smoothing action on the stream. In this figure we note that the quality

level fluctuates with the fluctuation of the aggregated upload bandwidth of the peer. In Figure 78 we apply the amplitude reduction algorithm on the row stream. In this case we note that the quality level fluctuation is drastically reduced, and the user can enjoy a relative stable video quality comparing to row stream. Indeed, this figure shows that the quality level change does not exceed 1 level in most cases. However, they could be a fluctuation of the quality within a short time period (from second 200 to 250 for example). In Figure 79, we apply the prefetching-based frequency reduction algorithm on the same row stream. We note that the quality level is stable at least for a smoothing window, however we note some big quality level jumps in some cases (jump of 2 layers from 149s to 150s for example). The stream is smoother in the case of mean-based frequency reduction strategy, where we note the stability of the quality level for several smoothing windows (for example from 270s to 380s).

In Figure 81 we study the performance of the three proposed smoothing algorithms regarding the parameters defined above, under the conditions of the scenario 1 (smoothing window size = 15).

In terms of layers change, the results obtained in Figure 81 confirm the observations in Figure 77, Figure 78, Figure 79 and Figure 80. The frequency reduction algorithm (both prefetching-based and mean-based) presents the best performance in terms of number of layer changes. It diminishes the row stream layers changes about 20 times and it performs $\frac{1}{4}$ of layer changes of amplitude reduction algorithm.

Secondly in terms of unused chunks, naturally the not-exploited chunks are null in the case of the row stream, since the quality level follows the bandwidth fluctuation. However we note an important portion of unused chunks in the case of frequency reduction algorithms compared to the amplitude reduction, because the formers tries to smooth the stream along a larger portion of time (at least a smoothing window), however the amplitude reduction smooth the stream locally.

As expected, regarding the longest sequence in the stream, we note that the frequency reduction algorithms outperform the amplitude reduction. This can be explained by the fact that the frequency reduction algorithms try to keep a stable quality level for a least one smoothing window. In the contrary of the amplitude reduction that tries to reduce the jump of quality from a time slot to another.

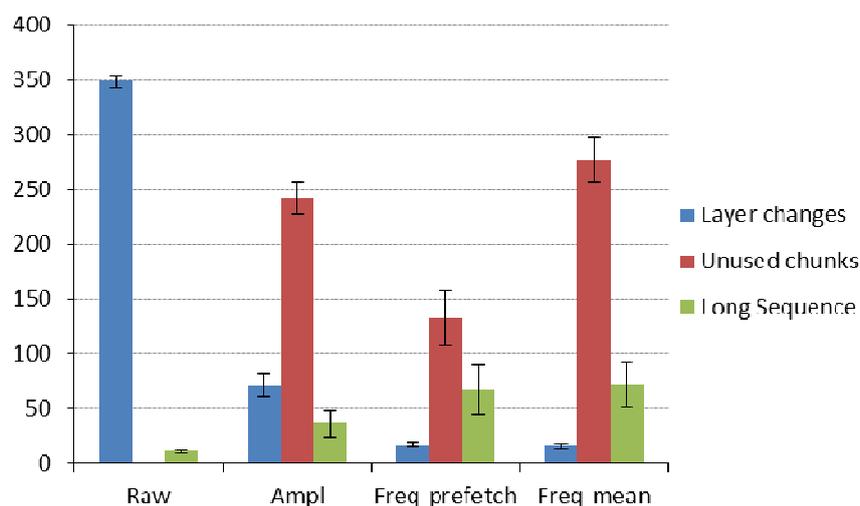


Figure 60: 1st scenario analysis

3.2.5.1.2 Scenario 2: Smoothing window size impact

In this scenario, we are interested in the impact of the smoothing window size on the performance of amplitude and frequency reduction algorithms. For that reason we performed several tests while varying the size of the smoothing window from 5 to 30 seconds.

3.2.5.1.2.1 Impact of smoothing window size on number of layers changes

Figure 82 shows the evolution of the number of layers changes while varying the smoothing window size for the amplitude reduction and frequency reduction algorithms. We note that the general trend is the decrease of the number of layers changes with the increase of the smoothing window size. It is obviously explained in the case of frequency reduction algorithm, since this later maintain the same quality level within at least smoothing window. The largest the smoothing window is, the lowest the global number of layers changes is. The number of layers changes in the case the amplitude reduction algorithm is inversely proportional to the smoothing window size, too. This can be explained by the fact that the amplitude reduction algorithm tries to reduce the quality level jump regarding the average quality level experienced in the previous smoothing window. A large smoothing window, means that the algorithms converge to same quality level (the mean of the smoothing window) for a long period of time. Thus, number of layers change is reduced.

3.2.5.1.2.2 Impact of smoothing window size on number of unused chunks

We note in Figure 83 that the number of unused chunks is higher in the case of the frequency reduction algorithms compared to the amplitude reduction. In addition, the general trend is the increase of the number of unused chunks with the expansion of the smoothing window for the frequency reduction scenario. Indeed, a large smoothing windows leads to a smoothing action over a large interval of time, and consequently a large number of scarified chunks. We can see this through the following example: assume a smoothing window of 10 seconds, and the number of chunks (of 1 second each) received within the smoothing window is 19 chunks. The maximum smoothed quality level possible is layer 0. The number of unused chunks in this case is 9 chunks. However, for the same amount of received chunks (19 chunks) in the same period of time (10 seconds) with a smoothing window of 5 seconds (2 smoothing windows), we can have a first window smoothed at layer=1 (layer 0 + layer 1) and the second window smoothed at layer 0. In this case the number of unused chunks is 4.

3.2.5.1.2.3 Impact of smoothing window size on longest sequence

We are interested in the evolution of the largest sequence for each smoothing algorithm (Figure 84). We note that this parameter is in accordance with the size of the smoothing window in the case of the frequency reduction algorithms since they act on a smoothing window, the larger it is, the longer the sequence is. However the amplitude reduction is not very impacted by the smoothing window size.

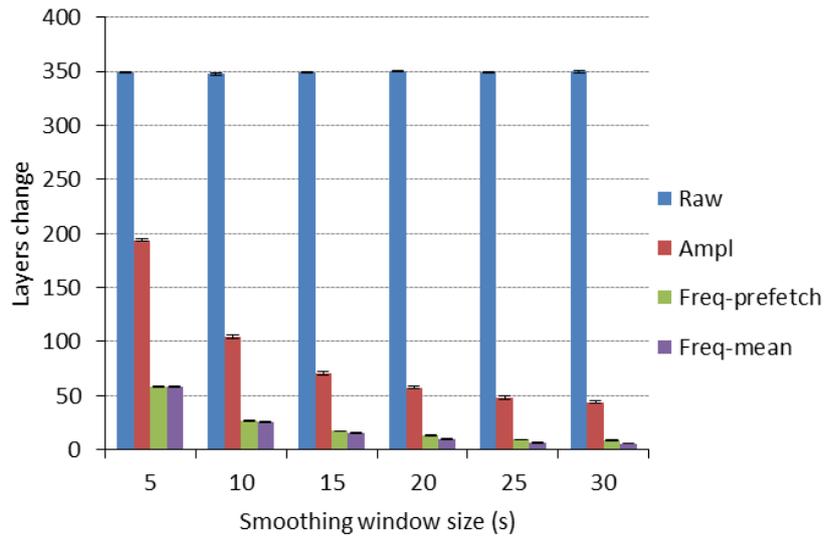


Figure 61: Impact of smoothing window size on number of layers changes

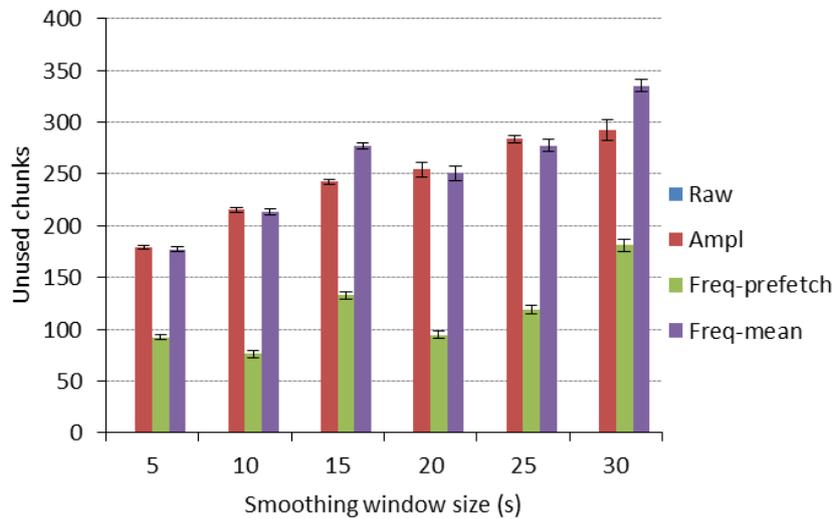


Figure 62: Impact of smoothing window size on number of unused chunks

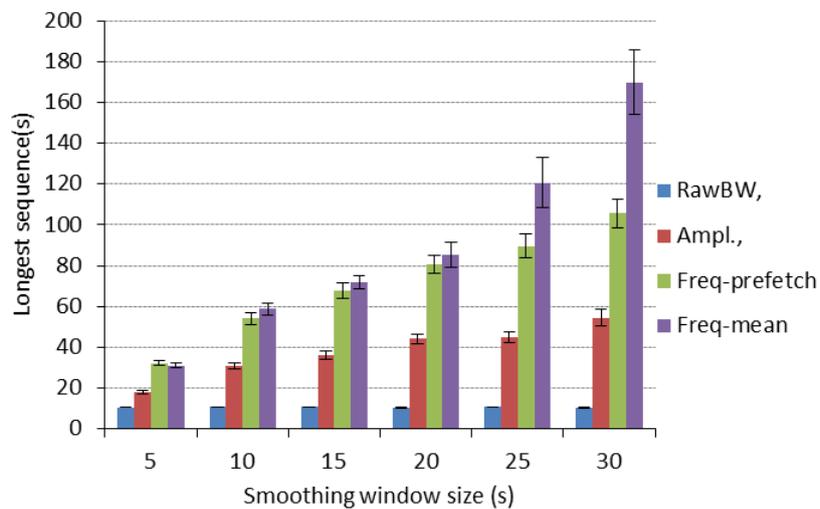


Figure 63: Impact of smoothing window size on number of longest sequence

3.2.5.1.3 Scenario 3: Comparison with layerP2P

Figure 85 shows the number of layer changes at different intervals for both mechanisms: LayerP2P and the frequency reduction mechanism (prefetching-based). The interval on x-axis represents the length 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. The lower number of layer changes in the proposed mechanism is due to the smoothing window, during which we aim to keep the same quality level. As a result, the frequency of changes in quality level is minimized. Comparatively, LayerP2P system has higher number of quality changes due to the absence of an efficient smoothing mechanism.

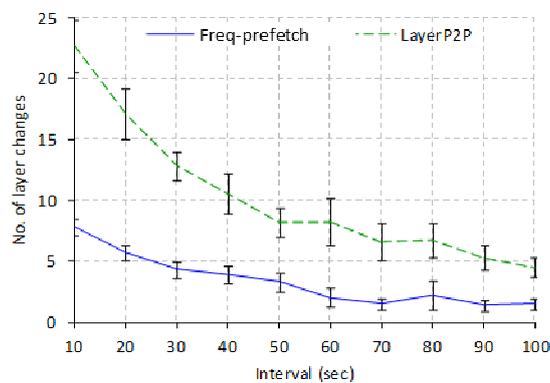


Figure 64: Number of Layer Changes

3.2.5.2 Conclusions

Content adaptation techniques have been developed for push- and pull-based approaches. A push-based scenario is considered in section 3.1.4 focusing in particular in source nodes uploading live video streams over wireless connections. In pull-based scenarios, the receiver is responsible for sensing the network conditions and accordingly adjust its data request rate. The mechanism proposed in this section improves this process by smoothing the transition between different qualities in the presence of frequent and high variations in network performance. The developed policies are proven to reduce the number of changes in the played SVC layer and to outperform alternative methods proposed in the literature.

3.2.6 Layered-video Quality Bottleneck in P2P overlays⁶

In this section we will study through simulation the performance of proposed techniques for overlay construction in order to avoid the problem of quality bottleneck in P2P layered streaming systems as described in [D5.3].

3.2.6.1 Objectives

Evaluate the performances of overlay construction techniques to achieve wide distribution of layered video content in overlay networks subject to high churn and heterogeneity in terms of network capabilities. Toward this objective, we evaluate the viewed layer by each peer during its session and to show the efficiency of our overlay construction techniques, we measure: (1) the bandwidth utilisation of the downlink in order to prove that our proposed techniques ensure the best quality according to its downlink capacity, and (2) the bandwidth utilisation of the uplink in order to prove that we ensure that each peer will contribute

3.2.6.2 Evaluation Scenarios & Metrics

The performance evaluation of our techniques is carried for mesh-based P2P networks. Network model is detailed in deliverable [D5.3].

In order to prove the effectiveness of our overlay formation technique (Probabilistic overlay formation: PRB) to avoid the quality bottleneck problem, we compared our technique against two other techniques:

- *Random overlay formation (RND)*: in this technique, a joining peer is connected randomly to K peers that are already in the overlay.
- *Scale free overlay formation (SFR)*: in this technique, a joining peer is connected to the K best serving peers at its time of arrival. The serving capability of a peer is defined by the ratio of its uplink capacity to its out degree (i.e. the number of peers receiving data from this peer).

For the different techniques, we have measured the following metrics:

- *Average viewed layer*: this metric shows the average viewed video layer by all the peers composing the overlay.
- *Viewed layer distribution*: this metric gives a more accurate view of the layer distribution in the network since it shows the proportion of peers receiving each layer.
- *Download bandwidth utilisation*: this metric shows the effective usage of the downlink capacity during the peer session.
- *Upload bandwidth utilisation*: this metric shows the effective usage of the uplink capacity during the peer session.
- *Overlay graph properties represented by the average in-degree and the average out-degree of the peers*.

The different parameters of the simulations are highlighted in Table 9.

⁶ The main technical contributions of the work on Layered-video Quality Bottleneck in P2P overlays - the problem formulation, algorithm design and simulations results - have been suppressed from the public version of this deliverable as the content is currently under review for publication.

Parameter	Value
<i>Peer lifetime</i>	Expo($\lambda = 1/300 \text{ s}^{-1}$)
<i>Peer arrival</i>	Expo($\mu=1/3 \text{ s}^{-1}$)
<i>Simulation duration</i>	5 hours
<i>Video stream</i>	1 video source : 5 Layers $\{L_0, L_1, L_2, L_3, L_4\}$ Overall bitrate : 3000 Kbps $\{L_0\} = 1000 \text{ Kbps}$ $\{L_0, L_1\} = 1800 \text{ Kbps}$ $\{L_0, L_1, L_2\} = 2400 \text{ Kbps}$ $\{L_0, L_1, L_2, L_3\} = 2800 \text{ Kbps}$ $\{L_0, L_1, L_2, L_3, L_4\} = 3000 \text{ Kbps}$
<i>Download bandwidth</i>	$\{1.0 \text{ Mbps}, 1.8 \text{ Mbps}, 2.4 \text{ Mbps}, 2.8 \text{ Mbps}, 3 \text{ Mbps}\} = \{20\%, 20\%, 20\%, 20\%, 20\%\}$
<i>Upload bandwidth</i>	$\{1.20 \text{ Mbps}, 2.16 \text{ Mbps}, 2.88 \text{ Mbps}, 3.36 \text{ Mbps}, 3.60 \text{ Mbps}\} = \{20\%, 20\%, 20\%, 20\%, 20\%\}$

Table 7: Simulation parameters

3.2.6.3 Conclusions

One issue that is particular to large-scale peer-to-peer video streaming applications is quality bottleneck, occurring when peers that need lower quality are connected closer to the source reducing the availability of higher quality layers further down the content distribution paths. This section evaluates an overlay topology construction technique for mitigating this problem.

3.2.7 CINA for CDN⁷

The Internet is witnessing a massive growth of traffic generated by web services and multimedia applications. It is moving from being a simple monolithic data service network to a pervasive and multi-service network where content streaming encounters a great success. Considerable evidence supports the claim that most of the IP network traffic is video. Recent traffic studies [APY12] has shed some light on the impact of media streaming services on networks. Netflix, a video streaming company, and YouTube, both relying on Content Delivery Networks (CDNs), account alone for more than 35% of the traffic in peak hours in North America. We reasonably expect these services to continue their growing with more subscribers, higher definition streams, richer content libraries and better device support. For Network Operators (NOs) who provide IP networks, these trends present important challenges arising from issues related to the management of growing traffic demand while maintaining appropriate quality of service. CDN service providers are also concerned by the problem as the QoE delivered is tightly related to the quality of content delivery, subject to network fluctuations. But rather than digging into solitary responses to the problem on each side, a new trend focuses on developing communication interfaces between NOs and overlay applications in order to leverage the assets of each with respect to the constraints of the other. With CDNs, content is replicated within surrogates located close to users, resulting in fast, reliable web and content delivery services for the users. These services require an optimum redirection of their clients, distributed across wide-area networks to appropriate surrogate server sites. The process of server

⁷ The main technical contributions of the CINA for CDN work - the problem formulation, algorithm design and simulations results - have been suppressed from the public version of this deliverable as the content is currently under review for publication.

selection could integrate Quality-of-Service policies based on predefined information and metrics provided by NOs. The latter are expected to witness improvements on the utilization of their resources. CDN operators benefit from enhanced Quality of Experience (QoE) perceived by the application users. The CDN use-case started to raise the interest of the IETF Application-Layer Traffic Optimization (ALTO) working group [APY12]. The proposed draft [NJV12] depicts the ecosystem of the interactions between CDNs, networks and end-users. It describes the integration of the ALTO framework into CDN processes. However, to the best of our knowledge, no previous work proposed an in-depth analysis of the utilization of ALTO by CDN providers. We are interested on comparing classes of server selection policies based on information provided by NOs through ALTO. ALTO may carry different types of metrics, exposing abstract or real values such as routing costs, delay or bandwidth. We therefore propose to analyse the technical challenges and the performances the different classes of metrics ALTO can provide. Our contributions in this study are to analyse the relevance of the classes of metrics they are expected to provide in the context of CDN media services, to capitalize on network assets to propose a novel approach providing adequate information with the profile of CDN streaming services and to expose results of extensive simulations demonstrating significant potential for performance improvements of resources utilization and perceived quality.

3.2.7.1 Conclusion

In this section, we addressed the traffic implosion issues raised by the growing trend of CDN-based content streaming services. We introduced and compared different server selection approaches for optimizing the traffic generated by CDN streaming applications. We developed an analytic model for analysing the performances of the different techniques and we demonstrated that an exchange of information based on the delivery cost metrics we propose provides better network performances through an improved load balancing of the CDN traffic, a remarkable network utilization gain compared to network-agnostic techniques and a better service response time and resource availability for the end-users. We provided recommendations on the utility and inconveniences of the different approaches. In the future, we intend to work on the inconveniences of the cost map updates to provide a more practical and efficient approach for server selection based on cross-layer mutual information exchange.

While the use of locality and its implication on network latency has been studied extensively as an option for determining CINA and ALTO costs, very little has been done to consider alternatives using path utilisation, a metric closely linked with throughput that is critical to many applications including streaming and file transfer over CDNs. This section elaborates on the calculation of CINA costs based on path utilisation, and evaluates the performance gains in the context of a CDN request routing function, showing improvement in link utilisation of up to 45% compared to random request routing policies. These overall gains are comparable to the ones achieved with locality-based costs, but in addition spread evenly across links, leading to more load-balanced conditions in the network.

3.2.8 TCP-friendly Rate Control and FEC over Multiple Links⁸

3.2.8.1 Objectives and Evaluation Requirements

- Explore the benefits of the centralised version of the cooperative rate controller that takes per path utility functions into account
- Evaluate the performance of the decentralised version of the cooperative rate controller
- Evaluate the time efficiency of the implementation of the decentralised version of the cooperative rate controller and in particular the developed convex optimisation solver
- Evaluate the benefits of the cooperative rate controller for a streaming application with deadlines

3.2.8.2 Conclusions

In this section we evaluate how applications not only adapt to the performance they receive from the network, but rather they seek to increase performance by transmitting data over multiple paths. A TCP-friendly rate controller is designed that takes into account an application-specific utility function that is diverse enough to accommodate applications with strict transmission deadlines like live streaming. A convex optimisation solver developed to enable the online deployment of the rate controller is proved to operate within acceptable time constraints. The performance of the rate controller outperforms Multipath TCP and is found to increase receiver goodput by up to 2.5 times for a particular application with deadline constraint of 100ms.

Although TCP has remained the transport protocol of choice for decades, the rise of real time traffic has meant that serious consideration has been given to alternative transport mechanisms. By incorporating multipath capabilities, retransmission-free error recovery, per-sub-flow utility functions and TCP-compatible fairness, the protocol presented in this section is a flexible alternative. Given the lightweight computations needed to solve its main optimisation problem, it is well-suited for deployment in mobile devices. Of course, the same closed loop control architecture can be applied to other resource allocation problems; we leave such further exploitation of these results for future work.

⁸ The main technical contributions of the TCP-friendly Rate Control and FEC over Multiple Links work - the problem formulation, algorithm design and simulations results - have been suppressed from the public version of this deliverable as the content is currently under review for publication.

4. CONCLUSIONS

This document presents the evaluation scenarios and results of the cross-layer optimisation mechanisms developed in the ENVISION project, including in-depth evaluation of individual components through simulation experiments and prototype evaluation of subsystems for different media applications.

A large set of prototype tests were performed in the FT testbed with a focus on evaluating in a realistic environment the benefits of the CINA interface from the perspective of both the network operators and the video streaming overlay applications. The FT testbed provides a powerful experimentation platform with a sufficiently complex topology modelling three different ISPs, a variety of access technologies and experimentation tools for introducing artificial traffic latency and loss. A number of CINA services have been tested using the FT testbed, ranging from providing dynamic CINA cost information reflecting the network conditions, to the activation of network services like multicast and caching.

It has been shown (section 3.1.1) that overlay topology construction policies that use CINA costs as their primary optimisation objective can reduce the load over inter-domain links for up to 79%, while at the same time improving the application performance comparing to random topologies. Further, at the event of congestion, policies that react to dynamic CINA costs outperform random and static cost policies, significantly improving the stream quality and reducing the playout lag for live streaming applications. The validity of the approach has been also established in the context of CDN-based overlay applications using the TID testbed (section 3.1.7). Random selection of the CDN node to serve a particular user request results in slower response times, decreased throughput for the streaming sessions and as a consequence buffer underflow events. The CINA interface has also been integrated with the content-aware path selection functionality developed in COMET (section 3.1.3). The use of CINA in COMET allows for taking into account network performance and ISP preference information in a consistent way when the optimisation logic is embedded into the network and when it resides outside the network into third-party overlay applications.

Further benefits are extracted for both the network and the overlay applications with the use of CINA network services. Multicast (section 3.1.5) has a notable impact in reducing the network load by 36% in the best cases, even in a limited scenario with only four peers switching from overlay unicast to network multicast streaming. Similarly, the use of the CINA caching network service achieves a 90% saving in traffic volume in the best cases (section 3.1.6). Functional validation tests have also proven the feasibility of using High Capacity Nodes as a CINA network service enabling the operation of interactive multi-participant applications without pre-established service infrastructure (section 3.1.4).

In addition to the above prototype-based evaluation, several simulation studies have looked into scalability, stability and large-scale performance aspects for the optimisation techniques developed in the project. Section 3.2.1 evaluates a technique for encouraging applications to change their selection of servers and schedule their data transmission at different times in order to reduce the ISP transit costs by shifting traffic to less expensive paths and lowering their peak-time utilisation. Results show remarkable potential cost improvement with 15% cost reduction in the most conservative cases where 20% of users comply to shift their traffic in time or in space. Section 3.2.2 elaborates on the related application-layer problem of consolidation preferences from different ISPs. Two methods are evaluated both resulting in improvements in total ISP cost savings when considering the preferences of all the ISPs as opposed to only the ISP in one end of an overlay connection. These improvements can be as high as 70% when fairness between different ISPs is not considered important.

While the use of locality and its implication on network latency has been studied extensively as an option for determining CINA and ALTO costs, very little has been done to consider alternatives using

path utilisation, a metric closely linked with throughput that is critical to many applications including streaming and file transfer over CDNs. Section 3.2.7 elaborates on the calculation of CINA costs based on path utilisation, and evaluates the performance gains in the context of a CDN request routing function, showing improvement in link utilisation of up to 45% compared to random request routing policies. These overall gains are comparable to the ones achieved with locality-based costs, but in addition spread evenly across links, leading to more load-balanced conditions in the network.

Section 3.2.4 looks into minimising the cost for interactive video applications with an upper bound in application layer delays. Such applications can operate at arbitrary locations without relying on fixed infrastructure, by mobilising CINA High Capacity Node (HCN) resources. The developed optimisation algorithm is evaluated using a cost model for the CINA services derived based on the Amazon EC2 pricing scheme. The results show that the use of HCNs significantly improve the experienced application delay and that the optimisation algorithm achieves a reduction of the operator costs from 0.42\$/hour/peer to 0.21\$/hour/peer for an overlay topology of 200 peers.

ENVISION also developed application layer techniques focusing on content adaptation, the improvement of goodput using multiple paths, and the efficient management of overlay resource information at large scale.

Content adaptation techniques have been developed for push- and pull-based approaches. A push-based scenario is considered in section 3.1.4 focusing in particular in source nodes uploading live video streams over wireless connections. In pull-based scenarios, the receiver is responsible for sensing the network conditions and accordingly adjust its data request rate. The mechanism proposed in 3.2.5 improves this process by smoothing the transition between different qualities in the presence of frequent and high variations in network performance. The developed policies are proven to reduce the number of changes in the played SVC layer and to outperform alternative methods proposed in the literature. Applied to logs taken from the live streaming testbed tests, smoothing optimisation has also been subjected to user evaluation (see section 3.1.2). The users watched the video clips reconstructed with and without smoothing and provided higher scores for the smoothed video clips in the majority of the cases. One issue that is particular to large-scale peer-to-peer video streaming applications is quality bottleneck, occurring when peers that need lower quality are connected closer to the source reducing the availability of higher quality layers further down the content distribution paths. Section 3.2.6 evaluates an overlay topology construction technique for mitigating this problem.

In section 3.2.8 a different approach is adopted, whereby the application does not only adapt to the performance it receives from the network, but rather it seeks to increase it by transmitting data over multiple paths. A TCP-friendly rate controller is designed that takes into account an application-specific utility function that is diverse enough to accommodate applications with strict transmission deadlines like live streaming. A convex optimisation solver developed to enable the online deployment of the rate controller is proved to operate within acceptable time constraints. The performance of the rate controller outperforms Multipath TCP and is found to increase receiver goodput by up to 2.5 times for a particular application with deadline constraint of 100ms.

Finally, section 3.2.3 elaborates on the evaluation of a distributed data management infrastructure built to enable n-casting request forwarding, used in distributed applications to discover a number n of resources that participate in the distribution of a particular video stream and are closer to the node issuing the request. The system is evaluated using a dataset of about 450K nodes with large geographical coverage. The system is proved to be very accurate in discovering the closest resources, with a small message overhead scaling well with the number of nodes and high resource availability churn.

REFERENCES

- [AMAZ01] aws.amazon.com/ec2/pricing/
- [APY12] R. Alimi, R. Penno, and R. Y. Yang, "Alto protocol," IETF, draft 11, 2012.
- [BCM75] F. Baskett, K. M. Chandy, R. R. Muntz, and F. G. Palacios, "Open, closed, and mixed networks of queues with different classes of customers," *J. ACM*, vol. 22, no. 2, pp. 248–260, Apr. 1975. [Online]. Available: <http://doi.acm.org/10.1145/321879.321887>
- [BJZ09] Y. Bai, B. Jia, J. Zhang, and Q. Pu, "An efficient load balancing technology in cdn," in *Fuzzy Systems and Knowledge Discovery, 2009. FSKD '09. Sixth International Conference on*, vol. 7, aug. 2009, pp. 510–514.
- [BP08] N. Ball and P. Pietzuch, "Distributed content delivery using load-aware network coordinates," in *Proceedings of the 2008 ACM CoNEXT Conference*, ser. CoNEXT '08. New York, NY, USA: ACM, 2008, pp. 77:1–77:6. [Online]. Available: <http://doi.acm.org/10.1145/1544012.1544089>
- [CE10] N. Carlsson and D. L. Eager, "Server selection in large-scale video-on-demand systems," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 6, no. 1, pp. 1:1–1:26, Feb. 2010. [Online]. Available: <http://doi.acm.org/10.1145/1671954.1671955>
- [CRC+08] Cha, Meeyoung and Rodriguez, Pablo and Crowcroft, Jon and Moon, Sue and Amatriain, Xavier, *Watching Television Over an IP Network*, *Proceedings of ACM IMC*, pp. 71-84, 2008.
- [D3.3] ENVISION deliverable D3.3, Final Specification of the ENVISION Interface, Network Monitoring and Network Optimisation Functions, June 2012, FP7 ICT ENVISION project, www.envision-project.org
- [D4.2] ENVISION deliverable D4.2, Refined Specification of Consolidated Overlay View, Data Management Infrastructure, Resource Optimisation and Content Distribution Functions, December 2011, FP7 ICT ENVISION project, www.envision-project.org
- [D4.3] ENVISION deliverable D4.3, Final Specification of Consolidated Overlay View, Data Management Infrastructure, Resource Optimisation and Content Distribution Functions, June 2012, FP7 ICT ENVISION project, www.envision-project.org
- [D5.3] Final Specification of Metadata Management, Dynamic Content Generation and Adaptation, Adaptation and Caching Node Functions, June 2012, FP7 ICT ENVISION project, www.envision-project.org
- [FPS12] B. Frank, I. Poese, G. Smaragdakis, S. Uhlig, and A. Feldmann, "Content-aware traffic engineering," in *Proceedings of the 12th ACM SIGMETRICS/PERFORMANCE joint international conference on Measurement and Modeling of Computer Systems*, ser. SIGMETRICS '12. New York, NY, USA: ACM, 2012, pp. 413–414. [Online]. Available: <http://doi.acm.org/10.1145/2254756.2254819>
- [GK77] M. Gerla and L. Kleinrock, "On the topological design of distributed computer networks," *Communications, IEEE Transactions on*, vol. 25, pp. 48–60, 1977.

- [GLP09] C. Griffiths, J. Livingood, L. Popkin, R. Woundy, and Y. Yang, "Comcast's isp experiences in a proactive network provider participation for p2p (p4p) technical trial," IETF, RFC 5632, 2009.
- [L02] E. Liu, "A hybrid queueing model for fast broadband networking simulation," Ph.D. dissertation, Queen Mary, University of London, 2002.
- [L61] J. D. C. Little, "A proof for the queueing formula: $L=w$," Operations Research, vol. 9, pp. 383–387, 1961.
- [LJ12] K. Lee and G. Jian, "Alto and decade service trial within china telecom draft-lee-alto-chinatelecom-trial-04," IETF, Draft, 2012.
- [LMI09] C. Labovitz, D. McPherson, S. Iekel-Johnson, J. Oberheide, F. Jahanian, and M. Karir, "Atlas internet observatory 2009," NANOG, Annual Report. [Online]. Available: <http://www.nanog.org/meetings/nanog47/abstracts.php?pt=MTQ1MyZuYW5vZzQ3I\&nm=nanog47>
- [LW81] S. S. Lam and J. W. Wang, "Queueing network models of packet switching networks," Austin, TX, USA, Tech. Rep., 1981.
- [MDJ11] B. Meskill, A. Davy, and B. Jennings, "Server selection and admission control for ip-based video on demand using available bandwidth estimation," in Local Computer Networks (LCN), 2011 IEEE 36th Conference on, oct. 2011, pp. 255–258.
- [N98] P. Nain, "Basic elements of queueing theory - application to the modelling of computer systems," 1998. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.128.15&rep=rep1&type=ps>
- [NETW01] <https://networkx.lanl.gov/>
- [N JW12] B. Niven-Jenkins, G. Watson, N. Bitar, J. Medved, and S. Previdi, "Use cases for alto within cdns," IETF, draft 03, 2012.
- [QUE76] Queueing Systems, Computer Applications. Wiley, 1976, vol. 2, no. ISBN: 978-0-471-49111-8.
- [RFUE01] <http://www.cs.washington.edu/research/networking/rocketfuel/>
- [RFUE02] <http://www.stanford.edu/class/msande91si/www-spr04/readings/week1/rocketfuel.pdf>
- [S11] Sandvine, "Global internet phenomena spotlight - netflix rising," Sandvine Incorporated, Tech. Rep., 2011. [Online]. Available: <http://www.sandvine.com/downloads/documents/05-17-2011-phenomena/Sandvine%20Global%20Internet%20Phenomena%20Spotlight%20-%20Netflix%20Rising.pdf>
- [TANG02] H. Tangmunarunkit, R. Govindan, S. Jamin, S. Shenker, and W. Willinger. Network topology generators: Degree-based vs. structural. In ACM SIGCOMM, August 2002.
- [V12] V. Valancius. Personal Communication, 2012.