Enriched Network-aware Video Services over Internet Overlay Networks

www.envision-project.org

Deliverable D5.2

Refined Specification of Metadata Management, Dynamic Content Generation and Adaptation, Adaptation and Caching Node Functions

Public report, Version 1.0, 6 February 2012

Authors

UCL David Griffin, Eleni Mykoniati, Miguel Rio, Raul Landa, Christopher Pluntke, Richard Clegg

ALUD -

LaBRI Toufik Ahmed, Damien Magoni, Abbas Bradai, Ubaid Abassi, Samir Medjiah

FT Bertrand Mathieu

TID -

LIVEU Noam Amram

Reviewers Klaus Satzke, Nico Schwan, Richard Clegg, David Griffin

Abstract: This deliverable presents work achieved in workpackage 5 (WP5) during the second year of the ENVISION project. It provides a novel audio/video content adaptation techniques, in order to ensure smooth content delivery as well as to guarantee an efficient content scheduling, by selecting the most relevant peer to provide the appropriate pieces of content (chunks), while trying to optimize and aggregate chunks requests. Then, we investigate some candidate techniques to incentivise peers to contribute their resource in the adaptation process. In addition, this deliverable provides techniques for the delivery of high–QoS multimedia content, such as a new TCP-friendly rate control mechanism combined with forwarding error correction and a new short-term cooperative caching mechanism. Finally, we tackle the problem of using multiple links in ENVISION to optimize the distribution of the content over multiple wireless channels.

Keywords: Metadata, AV Content Generation, AV Content Adaptation, SVC, Error Resilience, Forward Error Correction, Caching, Multilink Delivery.

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

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

EXECUTIVE SUMMARY

This deliverable describes proposed mechanisms, techniques and algorithms for content generation and adaptation that support the delivery of multimedia content to a large number of end-users. It also concentrated upon the use of SVC in P2P network supported by ENVISION components. This deliverable presents the following contributions:

- A refinement of metadata classes defined in D5.1 and a detailed specification of enriched content metadata and profile management functions;
- ENVISION media description in pull-based P2P networks, especially the stream map generation and buffer map exchange;
- A system for scheduling of smooth layered stream in P2P networks. It aims to smooth the layered stream by selecting the appropriate layers and the appropriate peers to provide these layers;
- A study of candidate mechanisms which give ENVISION peers incentives for contributing their CPU, memory and bandwidth capacity to provide adaptation services;
- A study of TCP-friendly rate control mechanisms for non-TCP traffic using multiple end-to-end paths with different characteristics simultaneously combined with error resilient transmission mechanisms.
- A study of cooperative caching technique, a short term caching solution in which peers store the stream (or portion of the stream) to serve the other peers in the overlay.
- An approach for the smooth quality generation of SVC content and SVC transport mechanism in cross scheduling and protection manner over wireless multiple channels.

LIST OF FIGURES

Figure 1: Illustration of optimal resolution per bitrate 7
Figure 2: Standard SVC generation 7
Figure 3: Smoothed SVC generation 7
Figure 4: 3GPP MPD Structure 8
Figure 5: Example of an RDX File Structure for SVC Streams 11
Figure 6: Example of a Buffer Map 11
Figure 7: Example of an Adapted SVC Stream 12
Figure 8: Example of Live SVC Streaming in Pull-Based Delivery Mode 12
Figure 9: Workflow of Scheduling for Smooth Layered Streaming 14
Figure 10: Harmony Search Algorithm for the SSP Problem 18
Figure 11: Diversity “Distance” between a Layer l* and the other Layers in the Scalable Stream 19
Figure 12: Diversity “Distance” for all Layers in a 10-Layer Stream 19
Figure 13: Reliability Indicator for a 15-Layer Stream 20
Figure 14: Example of Unfulfilled Requests 21
Figure 15: Removing Segment Redundancy 27
Figure 16: Cooperative Caching Algorithm 28
Figure 17: MLEP functional model 29
Figure 18: Delay of two cellular links 31
Figure 19: Monitoring Module design 31
Figure 20: Example of single 3.5G modem uplink bandwidth fluctuations 32
Figure 21: Cellular Bonding - transmission over multiple networks 33
Figure 22: illustration of probability density functions for the delay of multiple modems 34

TABLE OF CONTENTS

EXECUTIVE SUMMARY **2**

LIST OF FIGURES **3**

TABLE OF CONTENTS **4**

1. INTRODUCTION **5**

2. METADATA **5**

3. CONTENT GENERATION **6**

- 3.1 Smoothed SVC Generation 6
- 3.2 Stream Map Generation and Buffer Map Exchange 7
  - *3.2.1 Related Work 8*
    - 3.2.1.1 Media Description Elements 8
    - 3.2.1.2 Manifest File Exchange in HTTP-Streaming Technologies 9
  - *3.2.2 ENVISION Media Description in Pull-Based P2P Networks 10*
    - 3.2.2.1 Resource Description 11
    - 3.2.2.2 Example of Live SVC Buffer Map Exchange Techniques 12
  - *3.2.3 Conclusion 13*

4. CONTENT ADAPTATION **13**

- 4.1 Scheduling for Smooth Layered Streaming 13
  - *4.1.1 Smoothing Function 14*
    - 4.1.1.1 Related Work 14
  - *4.1.2 Scheduling Function 15*
    - 4.1.2.1 Related Work 15
    - 4.1.2.2 Request Aggregation in SVC Pull-Based Streaming Systems 16
    - 4.1.2.3 Conclusion 21
- 4.2 Incentive Mechanism for Adaptation 22
  - *4.2.1 Market-based Systems in the Research Literature 22*
  - *4.2.2 Systems Based on Exchange/Barter Economies 23*
  - *4.2.3 Systems Based on Auctions 24*
  - *4.2.4 Adaptation Incentives in ENVISION 26*

5. ERROR RESILIENT TRANSMISSION **26**

- 5.1 TCP-Friendly Multi-Link Rate Control and Forward Error Correction 26

6. SHORT TERM CACHING **26**

7. SVC LIVE VIDEO GENERATION AND DISTRIBUTION OVER MULTIPLE WIRELESS CHANNELS **28**

- 7.1 Introduction 28
- 7.2 System Overview 29
  - *7.2.1 User Preferences 30*
    - 7.2.1.1 Continuity vs. Sharpness 30
    - 7.2.1.2 Delay 30
    - 7.2.1.3 Stream Stability 30
  - *7.2.2 Resource Data Management 30*
    - 7.2.2.1 Performance monitoring module 31
  - *7.2.3 Content generation & Adaptation 31*
- 7.3 SVC Transport over multiple channels 32
  - *7.3.1 The Bonding Challenge 32*
  - *7.3.2 The Bonding Architecture 33*
  - *7.3.3 Scheduling and FEC Algorithms over multi-link 33*

8. CONCLUSION **34**

9. REFERENCES **35**

1. INTRODUCTION

This deliverable presents the refined specification of work performed in workpackage 5. Thus, in order to ensure efficient end-to-end content delivery, the content must be generated, encoded, transmitted, adapted and displayed in appropriate manner.

Toward this end, this deliverable summarises the ENVISION metadata and its management process as presented in D5.1 [D5.1] and finalized in I5.1 [I5.1]. We discuss then the problems of layered video streaming in ENVISION. Indeed, layered video streaming in peer-to-peer (P2P) networks has drawn great interest, since it can not only accommodate large numbers of users, but also handle peer heterogeneity. However, there's still a lack of comprehensive studies on chunk scheduling for the smooth playout of layered streams in P2P networks. In these situations, a playout smoothing mechanism can be used to ensure the uniform delivery of the layered stream. This can be achieved by reducing the quality changes that the stream undergoes when adapting to changing network conditions. Thus, we propose a playout smoothing mechanism for layered P2P streaming. The proposed mechanism relies on a novel scheduling algorithm that enables each peer to select appropriate stream layers, along with appropriate peers to provide them. In addition to playout smoothing, the presented mechanism also makes efficient use of network resources and provides high system throughput. An evaluation of the performance of the mechanism provided in D6.1 [D6.1], demonstrates that the proposed mechanism provides a significant improvement in the received video quality in terms of lowering the number of layer changes and useless chunks while improving bandwidth utilization.

Another important issue we investigated is related to incentive mechanisms for peer to contribute CPU, memory and bandwidth for providing content adaptation. We distinguish three main incentive mechanisms: market-based, exchange/barter economies based, and auctions based mechanisms. We believe that the studied mechanisms are well equipped to deal with requirements of ENVISION.

After being generated and eventually adapted, the content should be transported in an efficient manner. In this deliverable, we explore the question of how to design a TCP-friendly rate controller for non-TCP traffic using multiple end-to-end paths with different characteristics simultaneously.

Finally, we explore the benefit of using multiple links to distribute the content. In this perspective we investigate, especially, allocation of the SVC data and its transport over the multiple channels. This allows the source peer to protect and distribute the SVC content in cross scheduling and protection manner over the multiple channels.

2. METADATA

ENVISION metadata has an essential role in describing principal aspects of video content management from content generation to consumption. This metadata should also capture the state of all the components involved in the content distribution chain: user information and preferences, usage history, presentation preferences, device and codec capabilities, adaptation capabilities, type and description of services, network description, etc. It allows the creation of context-based multimedia services that maintain an acceptable level of QoE/QoS for the end-consumer.

In D5.1 [D5.1], we presented an initial specification of ENVISION metadata and grouped them into the following 7 classes:

- End user metadata for describing the end user profile (such as the user, user preferences and media usage history, etc.)
- Terminal capabilities metadata describing technical properties of the terminal both for source and destination users (such as codec and display capabilities, etc.)
- Content description metadata to characterise the content exchanged between users (such as AV characteristics, spatio-temporal context of the content, Intellectual property etc.)

- Network metadata to describe the parameters of the network (such as network capabilities and conditions, available bandwidth, etc.)
- Metadata for service management (such as service type, service cost, etc.)
- Session description metadata (such as session start time, number of active sessions, etc.)
- Peer metadata to describe the overlay functionalities of a peer (such as adaptation, caching capabilities, etc.)

We went further in I5.1 [I5.1] to refine these metadata classes, and provide a detailed specification of enriched content metadata and profile management functions.

As described above, the first element to consider in the end-to-end delivery chain is related to content generation. Thus, the next section describes the main component needed for live and on-demand content delivery over P2P network in ENVISION.

3. CONTENT GENERATION

The SVC standard allows the creation of scalable enhancement layers with certain dependencies between them. The main advantages of SVC are the degree of freedom provided by the layered structure which allows better distribution mechanisms and protection to maximise the QoE. The main drawbacks of SVC are its processing power requirements and overhead compared to AVC, which makes it less attractive for many existing implementations. These drawbacks are becoming less and less relevant as the SVC implementations are improving and the processing power of devices is increasing. SVC generates enhancements layers which add quality to the base layer. However, when the number of layers delivered to the client is changed, it creates an undesirable sharp quality change. Thus, we adopt in ENVISION the use of adaptable SVC stream generation, as described in the sub-section 3.1.

After being encoded, content needs to be described (part of content description metadata known as StreamMap) in order to allow its efficient delivery especially in the case of pull-based content delivery architecture (HTTP Streaming, pull-based P2P …). Based on this description, a receiver can orchestrate the acquisition of the desired data elements in time. This content description is detailed in sub-section 3.2.

3.1 Smoothed SVC Generation

The main idea of the proposed SVC generation adopted in ENVISION refers to generating a smoothed SVC stream from the source to the network for further distribution by changing the layers attributes dynamically, rather than changing the number of layers.

In Figure 1, we first illustrate the optimal resolution for a given bitrate for AVC stream, the blue line represents the perceived quality of experience versus the bitrate for a lowest resolution, for example QCIF, and the green line represents the QoE for higher resolution let say CIF resolution and so on. The illustration shows that the optimal QoE for different bitrates is achieved by a different resolution. This also gives rise to additional potential algorithm studied. However in that approach there is limited scheduling and protection mechanisms due to the lack of layers.

In Figure 2 we illustrate how a fixed layer attribute algorithm uses a varying number of layers to adapt the stream quality to the available bitrate. As shown, a small change in the available bitrate may cause delivery of an additional layer which than causes to a step change to the perceived quality. For a large dynamic range of available bitrates this approach will could cause the generation of many layers to avoid large fluctuation in quality. This however causes larger overhead and computation power.

Alternatively, in Figure 3 we illustrate the smoothness concept which maintains a fixed or small number of layers throughout the entire bitrate dynamic range by modifying the layers themselves.

This allows a generation of smooth stream in the sense that a small change in bitrate does not result in sharp change in quality, rather than a smooth quality change achieved by slightly changing of the target layers attributes. This allows an “AVC-like” quality of experience while maintaining SVC benefits in unequal protection and scheduling approaches that can meet with rapid changes in the throughput.

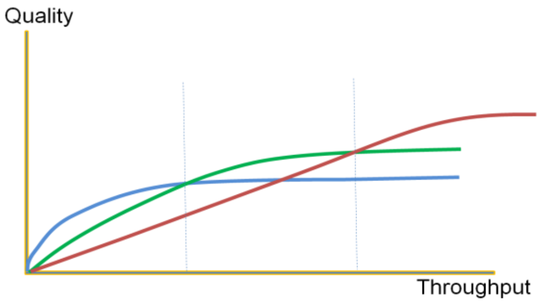


Figure 1: Illustration of optimal resolution per bitrate

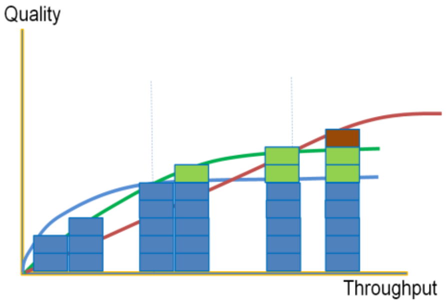


Figure 2: Standard SVC generation

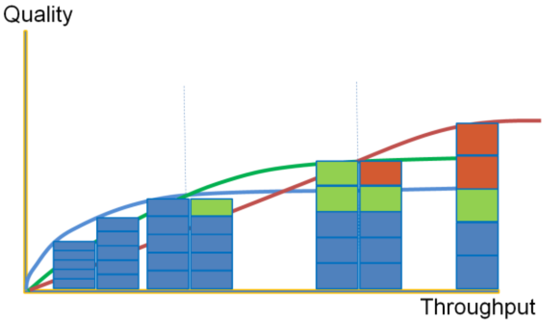


Figure 3: Smoothed SVC generation

The creation of the layers attribute takes into account the user preferences and the estimations of expected and worst case channel conditions. Thus, the dynamic target layer encoding attributes must satisfy these settings to allow best perceived QoE over the varying conditions. The smoothing layers algorithm is a subject for further implementation tests to optimise parameters.

3.2 Stream Map Generation and Buffer Map Exchange

In a cooperative overlay networks, and in the case of adaptable content distribution, peers may have different data elements of the same content. The peers need then to periodically advertise the available data pieces among them to allow the other peers requesting the data elements that they need.

In the following sub-sections, we present a brief state of the art regarding the media description in existing HTTP-streaming technologies. Then, we present the adopted approach for media description in ENVISION and we introduce the problem of media advertisement in pull-based Live-SVC streaming scenario.

3.2.1 Related Work

Pull-based video delivery represents the core of the new emerging standard related to HTTP streaming with or without dynamic adaptation due to the inherent two characteristics of such systems: (1) HTTP protocol (request / response) and, (2) Client/Server architectures with support to the stream switching mechanisms.

In existing HTTP streaming technologies and standards, the HTTP Client is supposed to have access to metadata that describe the video stream. In the 3GPP standard this is called the Media Presentation Description (MPD) [PM11] [Zo10] [Wu10]. An MPD includes sufficient information to provide a streaming service to the client by allowing him to sequentially download media data from an HTTP server and to render the included media appropriately.

3.2.1.1 Media Description Elements

A media presentation is a structured collection of data that is accessible to the HTTP Client, which is described in a MPD. As shown in Figure 4, a media presentation consists of:

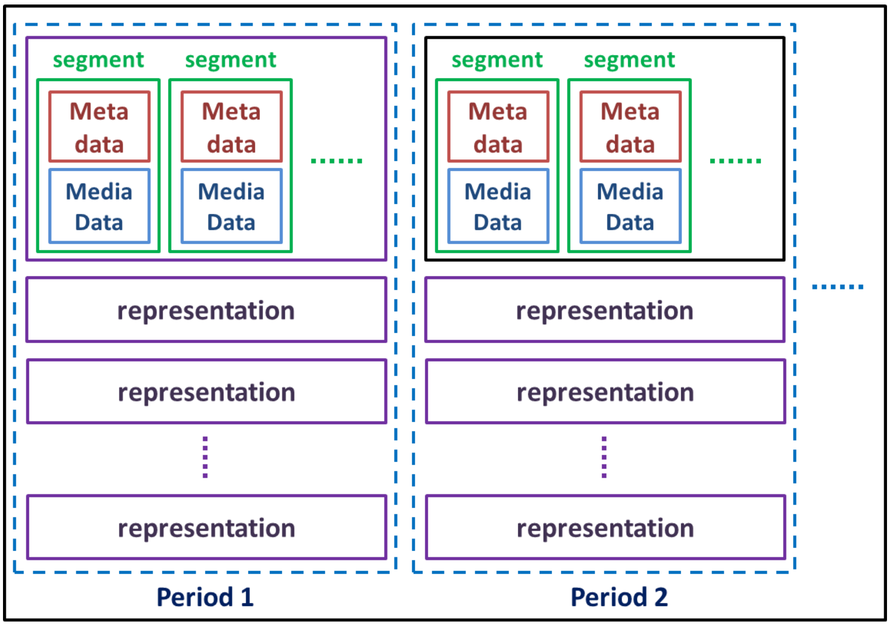


Figure 4: 3GPP MPD Structure

- A sequence of Periods.
- Each Period contains one or more Representations of the same media (resolution, bitrate, framerate, language, etc.).
- Each Representation is composed of one or more segments.

Each segment contains media data and/or metadata to decode. It is the smallest unit that can be uniquely referenced by an http-URL. Each segment has a start time relative to the start time of the period to allow the client to download specific segments.

The logic structure of media presentation, as described above, is actually described as the data structure (e.g. xml schema) in an MPD file. This file is required by the client in order to form specific URIs to access segments.

Along this logic structure, MPD includes a set of attributes:

a) "type" attribute: it indicates whether the media is VOD or Live.

b) "availabilityStartTime" attribute: it indicates the media presentation start time (0 if Type=VoD).

c) "duration" attribute: length of the media presentation. For Live media, the sum of "duration" and "availabilityStart" specifies the end time of the media. If "duration" is not provided, then the MPD does not describe an entire media presentation and the MPD may be updated during live presentation.

d) "minimumUpdatePeriodMPD" attribute: it indicates minimum update interval.

e) "timeShiftBufferDepth" attribute: it indicates the duration of time shifting buffer maintained at the server for live presentation.

f) "minBufferTime" attribute: it indicates the minimum buffer time for the stream.

g) Multiple "Period" elements: A "Period" element contains the following attributes and elements:

    1. "start" attribute: the start time of this period.
    2. Multiple "Representation" elements: describe a representation with different bit-rate, resolution, language, etc. A "Representation" element contains the following important attributes and elements:
        i. "bandwidth" attribute: maximum bit-rate of the representation averaged over any interval of "minBufferTime" duration.
        ii. "qualityRanking" attribute: the quality ranking of the representation.
        iii. "SegmentInfo" element: describes all segments in a representation. Each "SegmentInfo" element permits generating a list of Media Segment URLs. A "SegmentInfo" element contains information related to (1) the duration of the segment, (2) an initialization URL, and (3) a URL template to allow the construction of the following URLs.

The client is responsible for updating its MPD. It derives the update time from the sum of the time of his last requested update and the *"minimumUpdatePeriodMPD"* attribute.

3.2.1.2 Manifest File Exchange in HTTP-Streaming Technologies

It can be observed that 3GPP, OIPF, MS Smooth Streaming, Adobe Dynamic Streaming and Apple HTTP Live Streaming all follow a similar design scope (see Table 1), which is:

- The Streaming server utilises a stream encoder/segmenter to write the video content into a sequence of small files and in multiple versions, and produces a manifest file to describe these media files.
- The HTTP client firstly obtains the manifest file, and then constructs a series of URIs pointing to the media files.
- Based on the condition of client (e.g. network, device type, etc.), the client chooses to request certain media file using HTTP request with the corresponding URI. In Live streaming, the client is responsible of updating this manifest file periodically during the session.

Upon receiving the HTTP request, the HTTP server sends the media file corresponding to the URI.

	Media File	Manifest File
3GPP/OIPF [3GPP10] [OIPF10]	*.3GP File	*.3GP File
Microsoft HTTP Smooth Streaming[Ms09]	*.ismv / *.isma Files	*.ism / *.ismc File
Adobe Dynamic HTTP Streaming [Ad10]	*.F4F File	*.F4M File
Apple HTTP Adaptive Streaming [App09]	*.ts File	*.M3U8 File

Table 1: Media Description in HTTP Streaming Technologies

It is important to notice that these emerging technologies commonly implement the video source as first generating the content and then describing it through manifest files in order to make the client aware of the available content and its possible representations. We can also notice that layered video content such SVC is not part of any of these technologies. The particularity of layers dependency in SVC and the fact that different layers (i.e. representation in 3GPP vocabulary) share some data elements, does not respond to the media description needs in ENVISION. Thus, a new media descriptor is needed.

HTTP streaming is based on one-to-one communications (server to client), and the server is dimensioned to serve a fixed number of representations and to multiple clients, the video segments in all representations are always available at the server. This is not the case in today's P2P communications where the buffer map of relay peers differ based on their history of download/playback during the session. This leads to the fact that a requesting peer has to be aware of the content availability in the neighbourhood.

In order to achieve an efficient data scheduling, we must satisfy the following objectives:

1) Efficient media description that includes layered content such as SVC and its particularities,
2) Smart advertisement mechanisms to enable efficient sharing of data elements among the peers,
3) Minimum communication overhead while exchanging media description and data availability information.

3.2.2 ENVISION Media Description in Pull-Based P2P Networks

In video streaming over peer-to-peer networks, a consumer peer may be interested to know, before starting the session, general information about the stream such its title, duration, spatial resolution, etc. This information describes the overall stream in general and it is what we have called *Content Metadata*.

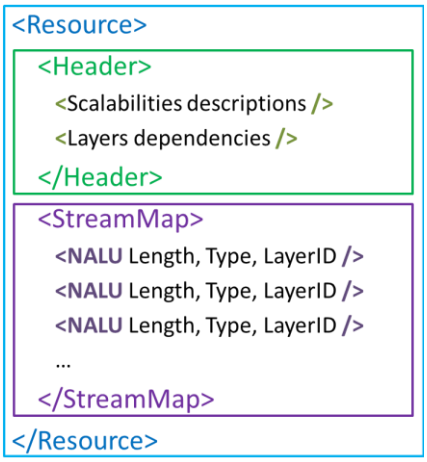
When delivering data chunks to the peers, especially in pull-based delivery systems, some information regarding the composition of the binary stream needs to be provided in order to orchestrate chunk scheduling. This information includes a description about the exchanged chunks such as their formation, size and composition. This is what we will call the *Stream Map*.

A Stream Map is thus a data structure describing the different data chunks that compose the video content. This data structure may be simple or more complex.

In the case of non-layered video, for example, a stream map for an MPEG2-TS video may include just two indications: (1) the size of the MPEG2-TS packets (188 bytes) and, (2) the number of these packets. Respectively, the stream map may be more detailed in the case of layered video codecs such as H.264 AVC and SVC. In this case, NALUs with their different characteristics (size, type, etc.) need to be supplied in order to achieve an efficient data scheduling and to enable content adaptation in the fly (i.e. SVC layer adding/dropping or AVC frames dropping).

3.2.2.1 *Resource Description*

In order to achieve the cited objectives, we propose to embed the both information in a single manifest file that accompanies each video stream to be consumed and delivered through the overlay network. Thus, the manifest file is composed of two separate parts: (1) General information, and (2) the Stream Map. An example of the manifest file structure is shown in Figure 5.



```
<Resource>
  <Header>
    <Scalabilities descriptions />
    <Layers dependencies />
  </Header>
  <StreamMap>
    <NALU Length, Type, LayerID />
    <NALU Length, Type, LayerID />
    <NALU Length, Type, LayerID />
    …
  </StreamMap>
</Resource>
```

Figure 5: Example of an RDX File Structure for SVC Streams

When comparing VoD and live content, the content and the generation process of this manifest file is different.

VoD content delivery: For VoD service, the manifest file includes all the information described earlier. The general information is frozen and the stream map can describe in detail all the data pieces composing the video stream.

The manifest file, so formed, can be transmitted at the beginning of the session; the scheduling of the different data pieces can then be optimally achieved.

Live content delivery: For Live content delivery, the manifest file, as transmitted at the beginning of the session, can only include general information about the video stream containing such non evolving information as title, codec, etc. or the most recent dynamic information like spatio-temporal context (data and time, geo-location, etc.). The stream map can only be transmitted incrementally/periodically as the content is being generated at the source.

In pull-based P2P networks, each peer participating in the distribution of a video stream needs to advertise the available data. This information is maintained in a data structure called the Buffer Map. Unlike the stream map, the buffer map maintains the presence state of the data piece available at each peer side. As shown in Figure 6, a buffer map is generally a bitfield structure where each bit indicates whether the peer already has the corresponding data piece (=1) or not (=0). The peers exchange their buffer map or parts of it in order to make the other peers aware of the data availability in the neighbourhood.

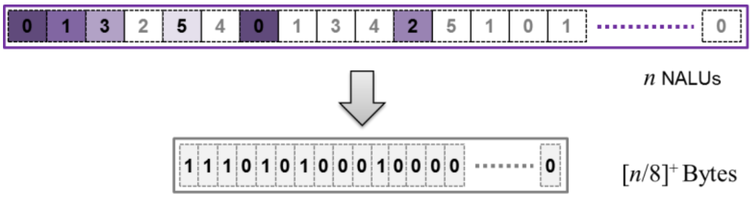


Figure 6: Example of a Buffer Map

For VoD service, the buffer map can be exchanged among the peers at the beginning of the session, and then be updated partially and periodically. However, for Live services, the content of the exchanged buffer map must be carefully decided.

3.2.2.2 Example of Live SVC Buffer Map Exchange Techniques

SVC content offers the flexibility of online stream adaptation by switching between scales (a version of the stream: <Resolution, Framerate, SNR>) during playback. In practice, this change of scale is allowed at special NALUs (NALU of type IDR: Instantaneous Decoding Refresh). Two successive IDR NALUs delimit a portion of the stream where the decoded/viewed layer is constant. An example is shown in Figure 7.

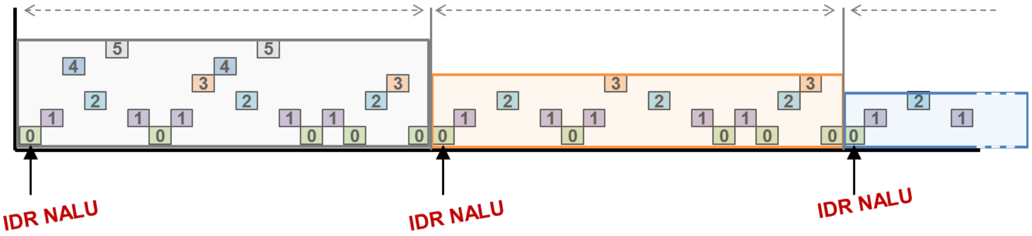


Figure 7: Example of an Adapted SVC Stream

In non-layered video codecs and in pull-based live streaming, peers exchange mainly information related to the liveness of their current buffer (actually they exchange the delay from the live instant).

But, in SVC live streaming; peers need to exchange their history of playback in terms of viewed layers, and thus the exact composition of their buffer.

Considering the example shown in Figure 8, the video source is streaming a live SVC video composed of a base layer (0) and three enhancement layers (1, 2, 3). A consumer peer (Peer A) is pulling the content from the source and another consumer peer (Peer B) is pulling the same video from Peer A.

In this example, and due to network conditions or obtained information from CINA interface, Peer A has decoded/viewed the first 4 chunks at the scales [3, 2, 0, 1] respectively.

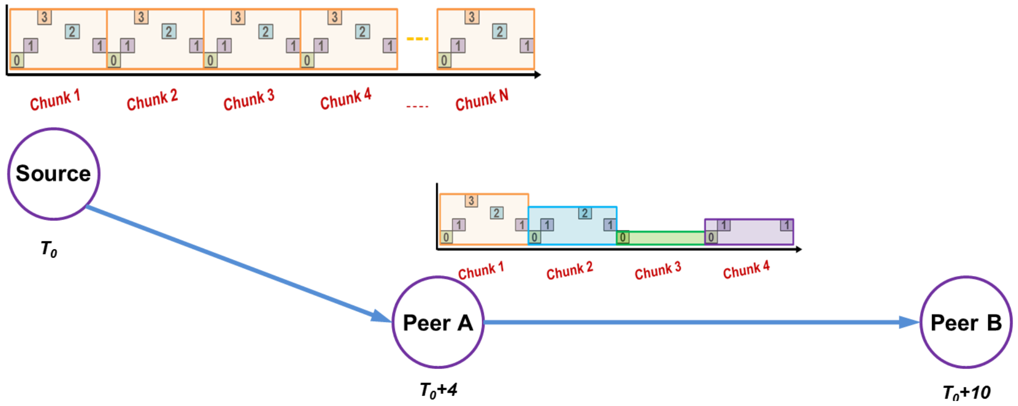


Figure 8: Example of Live SVC Streaming in Pull-Based Delivery Mode

At this instant, Peer B has requested the buffer map of Peer A. Peer A need to supply sufficient information about its liveness and its buffer map in order to let Peer B efficiently schedule the data pieces in its neighbourhood.

Peer A can then transmit the stream map up to the 4 chunks, along the corresponding buffer map.

*[Last_Chunk_ID=**4**, [Layer=**0**, Type=**IDR**, Present=**1**], [Layer=**0**, Type=**VCL**, Present=**1**], … [Layer=**0**, Type=**VCL**, Present=**0**]]*

In order to reduce the transmission overhead, resulting from the exchange of partial stream and buffer maps, and in order to gain more liveness, we can suppose that the source can encode the live video following a constant pattern for each chunk. Thus, this pattern can be exchanged only once at the beginning of the session, and peers exchange only information about the liveness and the playback history.

For example, Peer A can transmit:

- The last available chunk ID, along the lower layer viewed so far: [Chunk_ID=4, Lower_Layer=0].
- The Last available chunk ID, along the history of viewed layers [Chunk_ID=4, History=3, 2, 0, 1]

Based on the stream map and the buffer map knowledge, Peer B will schedule the desired data pieces accordingly.

3.2.3 Conclusion

Since layered content (such as SVC) is not part of any media description technique used in existing technologies, we introduced the ENVISION media description that describes content by both its general information and information related to its bitstream composition.

In cooperative overlay network, peers exchange data elements among them. Since the content can be adapted (i.e. switching from one representation to another one) at anytime during the playout session, peers have different buffers composition. Thus, data element availability information needs to be exchanged in an efficient way in terms of communication overhead and delay especially in the case of live services where content is generated, described and consumed at the same time scale.

This information (the description of the media and the availability of the data) will permit a receiver to adapt and schedule the download of the desired data elements in an efficient way. The adaptation mechanisms are described in the next section.

4. CONTENT ADAPTATION

4.1 Scheduling for Smooth Layered Streaming

Layered video streaming in peer-to-peer (P2P) networks has attracted interest as it can accommodate a large number of users with heterogeneous capabilities. However, there remains a lack of comprehensive studies on smoothing and scheduling for layered streaming in P2P networks. The smoothing mechanism ensures the smooth delivery of layered streams, by reducing the changes from higher quality to very low quality. The previous works in this area focused on maximising the throughput and minimising the delay, without considering the smoothing of layers and its consequences on the scheduling mechanism for acquiring these layers.

There are several complexities involved in layered streaming over P2P networks due to bandwidth fluctuation and peer's unreliability. In this case, the correct decision regarding the selection and the scheduling of the layers is very crucial: how many layers to request, in which order and from which peer?

In this section, we propose a system for scheduling of a smooth layered stream in P2P networks. The proposed system aims to smooth the layered stream by selecting the appropriate layers and the appropriate peers to provide these layers. The proposed smoothing mechanism is a part of the Adaptation Decision Function (ADF), while the peer selection mechanism refers to the Adaptation Execution Function (AEF). Both functions have been described in detail in D5.1 [D5.1].

We assume a mesh-based pull approach in which the receiver peer requests the content from different neighbour peers. Figure 9 depicts the basic architecture of proposed scheduling for smooth layered streaming system. It is composed of two main modules: the smoothing module and the scheduling module.

The smoothing is achieved by selecting the appropriate layers according to the available bandwidth at the receiver peer. For that purpose, a bandwidth estimation algorithm is used for estimating the available bandwidth for the next time period. The initial quality smoothing module is invoked only once and is responsible for the selection of appropriate layers at the beginning of the session while the run time quality smoothing module is used to adjust the layers according to the variation in bandwidth.

Once the selection of appropriate layers for the next time period has been carried out, the scheduling module is responsible for requesting the content to achieve the selected quality level. This module decides the chunks priority according to their playback deadline and layer dependency. The output of this module is a chunk to peer assignment matrix as shown in Figure 9. The ultimate goal is to request the appropriate chunks from appropriate neighbours in order to minimise the useless chunk ratio and maximise the utilisation of the available bandwidth capacity. The following sub-sections describe both modules in detail.

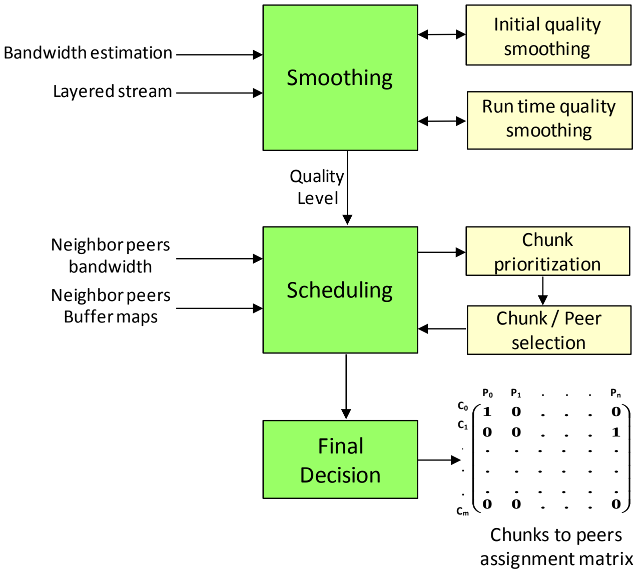


Figure 9: Workflow of Scheduling for Smooth Layered Streaming

4.1.1 Smoothing Function

4.1.1.1 Related Work

To handle the changing networks dynamics, several layered streaming systems has been proposed [ZXZY09][DCX07][SR00][RO03][TCX09][NLE10]. Saparilla and Ross [SR00] were among the earliest and they investigated the two-layer coded video streaming over the Internet. The dynamic allocations of the available bandwidth to the two layers are studied in order to minimise the impact of client starvation. However, the selection of appropriate layers under the varying network conditions was not discussed.

PALS proposed by Rejaie et al. [RO03] focused on another dimension of the layered streaming problem. It is a receiver driven P2P video streaming system with quality-adaptive playback of layered video. The system provides an adaptive streaming mechanism from multiple senders to a single receiver in P2P networks. It enables a receiver peer to orchestrate quality adaptive streaming of a single layer encoded video stream from multiple congestion controlled senders, and is able to support a spectrum of non-interactive streaming applications. However, PALS did not consider the smoothing problem due to bandwidth variation. Moreover, the assignment of the chunks to appropriate peers is also not discussed.

Recently, the authors in [NLE10] focused on combining the benefits of network coding with layered streaming. The objective was to design a practical adaptive P2P streaming protocol, by taking advantage of network coding and layered streaming to mitigate the inherent challenges in unstructured layered P2P streaming. The work focuses on the average quality satisfaction of the peer by considering only the maximum quality level achieved. The proposed protocol doesn't consider the degradation in user's quality of experience (QoE) due to variation in quality levels. In [HGL10] authors proposed P2P layered streaming designs which included a taxation mechanism, including the layer subscription strategy and mesh topology adaptation. The taxation mechanism is devised to manage the trade-off between social welfare and individual peers' welfare. Although, the work considers the strategy for layer selection but it mainly focus on fairness in P2P systems. The challenges of incorporating layered streaming in a P2P network such as quality degradation, and smoothing are not considered.

In [NLE10], the authors demonstrate the importance of neighbour selection in layered streaming and identify unique challenges of neighbour selection to the system performance. The authors proposed a new neighbour selection technique that can offer good performance while keeping the scalability of the mesh overlay under network fluctuations. The core of the technique is a pre-emption rule that allows a higher-capacity peer to replace a lower-capacity peer to be a neighbour of another peer with a certain probability. The work mainly focuses on achieving the high quality level for the peer by providing the set of neighbours having higher perceived quality.

4.1.2 Scheduling Function

In the context of P2P pull-based video streaming, selecting the appropriate peers to provide the appropriate chunks is a complicated task since chunks received after their playback deadline are not played and considered as useless chunks. Moreover, in the context of layered video, the task becomes more and more complicated, where an additional constraint should be taken into consideration, namely the layers' dependency. Hence, in layered video coding, video is encoded into a base layer and several enhancement layers, where a higher layer can be decoded only if all related lower layers are available. This is what we call the *layers' dependency* [D5.1].

4.1.2.1 Related Work

In the literature, some considerable importance has been given to scheduling in P2P networks. Most of these works are based on empirical studies which include pure random strategy [PA05], Local Rarest First (LRF) strategy [AR05] and Round Robin strategy (RR) [ZLY05]. Apart from empirical studies, some work exploits the concept of queuing theories for scheduling [GLL08]. The algorithm proposed in [RR06] minimises the base layer losses, but it assumes base layer rate equal to the enhancement one. This model of video is rather ideal and can be approximated only by fine grained scalability (FGS). Furthermore, a few theoretical studies tackle the optimal stream scheduling. Most of these works are under restrictive hypothesis or computationally expensive. In [BSWG07] a scheduler has been proposed to maximise the video quality by prioritising the most important chunks. This strategy is particularly suited for push-based, tree-structured overlay.

The scheduling mechanism proposed in [XSG08] is able to save base layer losses to the detriment of the enhancement one, but it has not been evaluated in terms of reconstructed video quality. The authors in [ZWLZG09] propose an optimal scheduling strategy to minimise the overall video distortion, but the approach is strongly related to the Multiple Description (MD) coding, which is less efficient compared with layered coding [FCPK04].

Recently, the authors in [ZXZY09] discussed the scheduling problem in data-driven streaming system. They define a utility for each chunk as a function of the rarity, which is the number of potential senders of this chunk, and the urgency, which is the time difference between the current time and the deadline of this chunk. They, transform the chunk scheduling problem into a min-cost flow problem. This algorithm however, is computationally expensive and may not be feasible for live video streaming systems because the rarity policy cannot be applied to live video streaming where the content are delivered in real-time having certain playback deadline.

In the following sub-sections we present a new analytical model and its corresponding algorithms to deal with the chunks scheduling problem in pull-based P2P video streaming, both in case of layered and non-layered video streaming. First, we propose a chunks prioritisation strategy in order to represent the urgency of chunks and its layers dependency. Then, we model the problem as an assignment problem for the cases of layered and non-layered streaming and we propose new algorithms to resolve it in order to fully take advantage of bandwidth capacity of the network and to meet the availability of chunks in neighbourhood. Then, we study the problem of chunks requests aggregation and request load balancing among the senders in SVC streaming.

4.1.2.2 Request Aggregation in SVC Pull-Based Streaming Systems

An overlay swarm is constructed around an SVC stream taking into account network resource optimization that can be obtained from CINA. The overlay may be composed of different peers with different profiles (terminal/network capabilities, user preferences, etc.). These peers are interested in different stream scalabilities (i.e. different versions). For example, a mobile device is interested in a lower spatial resolution, while a connected TV-set is interested in higher spatial resolution. Thus, the requested video content differs from one peer to another one.

Stream Map: A stream map is constructed upon the SVC stream in order to describe its binary stream and structure. The stream map contains all the necessary information about all the NALUs that form the bitstream. Each piece π_j is described by its offset o_j in the overall stream, its size s_j, and the index of the layer l_j it belongs to (i.e. $\pi_j = (o_j, s_j, l_j)$).

Peer Set: In this architecture, each peer maintains a set of peers $\mathbb{P} = \{p_1, p_2, \ldots, p_N\}$ serving all or parts of the video stream. Each peer p_i is associated with information used to estimate its reliability r_i to deliver the requested data pieces. This information includes: (1) the similarity of its buffer map with the local one, (2) the ratio of satisfied requests so far and (3) information about its uplink capacity.

Piece-Map: When a peer connects to the overlay to begin consuming the SVC video content, it exchanges a buffer map that indicates the availability of the pieces described in the stream map. Considering a stream that contains K pieces, then the buffer map is as follows:

$$\{b_j, 1 \le j \le K\}; b_j = \begin{cases} 1 & \text{if the piece is avaialable} \\ 0 & \text{otherwise} \end{cases}$$

All the received buffer maps from the peers that belong to the same swarm will form a matrix $(b_{ij})_{1 \le j \le K}^{1 \le i \le N}$ (where b_{ij} indicates the availability of the piece π_j at the peer p_i) that we call the *Piece Map*.

The Piece Map differs from the stream map. The former describes the availability of data pieces (NALUs) among the peers for the current streaming session while the latter describes the stream organisation in terms of number of layers, index of each NALU, size of NALUs, etc.

The main goal of our approach is to achieve efficient requests in the network. By efficient requests, we mean: (1) requesting the important pieces (i.e. pieces with a lower layer index) from reliable peers, (2) reducing the overhead introduced by multiple-piece requests, and (3) load balancing the requests of different layers from different peers.

The SVC Streaming over P2P problem (SSP Problem):

Given a set of pieces to request from other peers with multi-pieces requests, the SSP problem is to *"pack"* these pieces into a minimum number of multi-piece requests while optimising the following metrics: (1) the reliability of all the requests, (2) the overhead induced by the multi-piece requests, and (3) the load balancing of requests among the senders.

Theorem: The SVC streaming over P2P problem (SSP Problem) is NP-Hard.

Proof: First we show the problem is NP. The best set of requests can be found by going through all the possible requests sets, considering only the feasible ones, and assessing the "*goodness*" of each set in order to choose the best among them in a polynomial time. Next we show the problem is NP-Hard. The *Bin Packing Problem (BP)* can be reduced to the SSP problem. Given a set of items $\mathbb{I} = \{i_1, i_2, \dots, i_N\}$ of different packing cost c_i, (volume, size, weight…) and an infinite set[1] of bins $\mathbb{B} = \{b_1, b_2, \dots\}$ of capacity C each, a BP formulation is to pack all the items in a way that minimises the number of used bins. The SSP problem can be seen as a constrained variant of the classical BP problem. Indeed, the SSP formulation is to *"pack"* pieces π_j of different sizes s_j into a minimum set of requests while aiming at the same time to optimise other objectives (i.e. R: *Reliability*, LB: *Load Balancing* and O: *Overhead*). The problem instance reduction is in fact done by considering an overloaded version of the objective function. In the classical formulation of the BP problem, the objective function used to evaluate an *"assignment"* $\mathcal{A}$ is the number of used bins $\mathbb{U}$ (i.e. $\mathcal{F}(\mathcal{A}) = |\mathbb{U}|$), while in the SSP problem, the objective function considers other parameters in addition to the number of *"bins" (i.e. requests in the SSP problem)*, in other words $\mathcal{F}(\mathcal{A}) = (|\mathbb{U}|, R, LB, O)$. Above, we showed that BP can be reduced to our problem. Therefore, the SSP problem is also NP-Hard.

Harmony Search Heuristic

Given the *NP-Hardness* of the SSP problem, we propose a heuristic based on Harmony Search (HS) [GKL01]**.** The original HS heuristic refers to a metaphor of musicians in an orchestra playing notes together in an attempt to find the most harmonic combination. The Harmony Search meta-heuristic exhibits the following properties:

- HS can consider both continuous and discrete objective functions.
- HS can handle discrete as well as continuous variables.
- HS is free from divergence and may escape local optima.

Along these properties, the harmony search heuristic is known to be a quick optimisation technique in terms of necessary iterations to converge to the optimum. Figure 10 shows the harmony search heuristic applied to the SSP problem. Harmony Search algorithm is known to be free from parameter setting problem. In fact, the default values [GKL01] are often of good choice.

```
Harmony_Search_for_SSP (SSP Data)
SSP inputs:
    PieceSet: a set of data pieces to request (i.e. NALs)
    π.Peers / π in PieceSet: a set of peers having piece π
    Piece-Map: a matrix for the availability of the different pieces at the different peers.
HS inputs (default values):
    HMS: Harmony Memory Size (default : 20)
    hmcr: Harmony Memory Choosing Rate (default: 0.80)
    par: Pitch Adjustment Rate (default : 0.20)
    Termination Criterion: (default: 10 iterations)

    1. Generate the set HarmonyMemory of size S:
       foreach solution s in HarmonyMemory do
           foreach piece π in s.Pieces do
```

---

[1] Actually, the cardinality of the subset of $\mathbb{B}$ that will be used to pack the items in $\mathbb{I}$, is bounded to the number of items in $\mathbb{I}$, i.e., $|\mathbb{B}| \leq |\mathbb{I}|$**.** In other words, for the worst case, each item is packed into a different bin.

```
        Assign piece $\pi$ to randomly chosen peer p in the set $\pi$.Peers.
    done
done
```

2. Harmony Search Algorithm:

```
repeat :
    Generate a new solution $s^*$ :
    foreach piece $\pi$ in PieceSet do
        with probability hmcr do
            Assign piece $\pi$ to the same peer p in a randomly chosen solution s from HarmonyMemory.
            with probability par do
                Assign piece $\pi$ to a peer $p^*$ "similar" to peer p within the set $\pi$.Peers
            done
        done
        with probability 1–hmcr do
            Assign piece $\pi$ to randomly chosen peer p in the set $\pi$.Peers.
        done
    done
    Compare the new solution $s^*$ to the worst solution $s^\infty$ in HarmonyMemory:
    if (solution $s^*$ "is better than" solution $s^\infty$) then
        Replace solution $s^*$ by solution $s^\infty$ in the set HarmonyMemory.
    endif
until (termination criterion is satisfied).
```

Figure 10: Harmony Search Algorithm for the SSP Problem

Solution Coding

A solution to the SSP problem is a set of requests. Each request is a collection of information including the peer to which it is sent, and the different pieces to request. To apply the harmony search algorithm to the SSP problem, we need a vector representation of the solution. Thus, we introduce the following definitions:

Given a set of K pieces to request, $S = [s_1, s_2, \dots, s_K]$ is a solution, where $s_i$represents the peer from which the piece π_i will be requested. Since all the peers do not have the same pieces, each $s_i$takes a value in a different set rather than the global peers set *P*. We denote this set as $\pi_i.Peers$. It represents the peers that have the piece π_i.

Based on a solution vector S, the algorithm will construct the requests with respect to the size constraint C (for example, $\mathrm{C} = 16\mathrm{KB}$ is the optimal response size in BitTorrent Protocol as described in [MLL08][BIT10]).

Solution Fitness

In order to measure the fitness of a SSP solution *f(s),* we compute the following criteria:

Request Diversity:

First, the diversity of a single request is computed by the following equation

$$d(Req_i) = \sum_{\pi_j \in \text{Req}_i} \left(l_* - l_j\right)^2 / \left(\alpha L - l_j\right)\,, \tag{11}$$

where l_* is the dominant layer index in the request Req_i (for example, l_* related to the request Req[1,2,3,2,2,3,2] is l_*=2). L is the highest layer index of the SVC stream and α is a constant set to avoid a zero dominator (any α≠1). This indicator computes the “distance” between a layer index of a piece $l_i$and the dominant layer index l_* within a particular request, in a way that a larger difference will result in a larger “distance”. Moreover, this distance is not the same for two layer indexes l_i and l_j surrounding l_* (i.e. $l_i < l_* < l_j$).

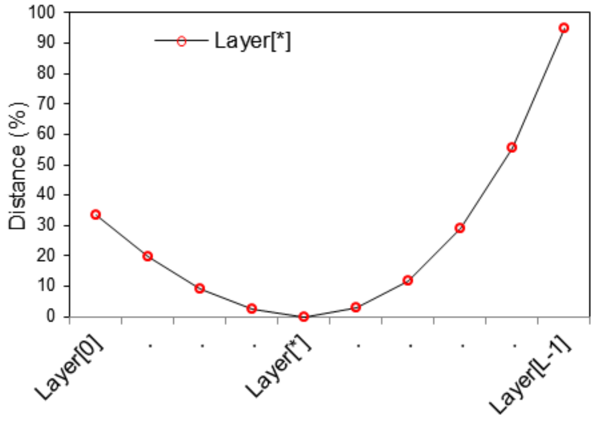


Figure 11: Diversity “Distance” between a Layer l* and the other Layers in the Scalable Stream

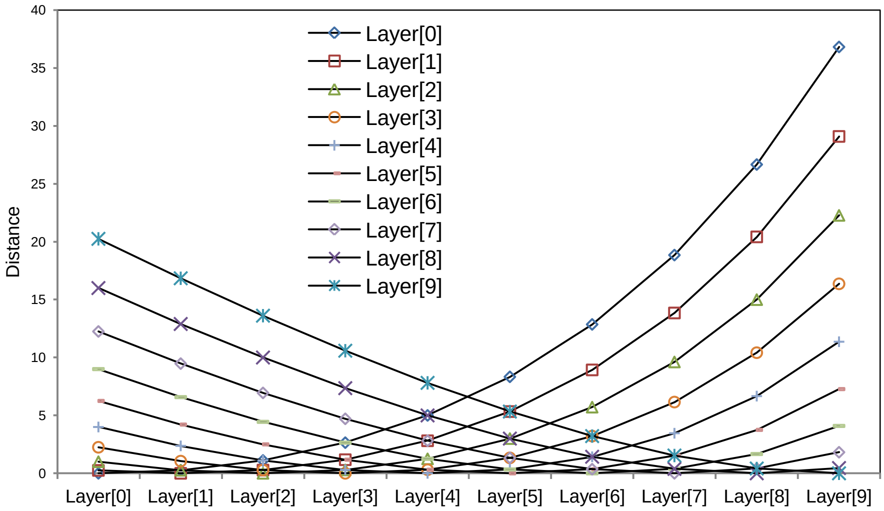


Figure 12: Diversity “Distance” for all Layers in a 10-Layer Stream

We defined this indicator in a way to tolerate diversity with lower layer indexes rather than with higher layer indexes. A plot of this indicator is given in Figure 11.

Figure 12 shows this indicator regarding a scalable stream composed of 10 layers. The diversity indicator has the following properties:

- The diversity between l_* and itself is null.
- The diversity is proportional to the lag between l_* and a layer index l_i.
- The diversity is less important with lower layers than with higher layers.

While the first two properties are obvious, the third property is driven by the fact that a peer having pieces of layer l_* is more likely to have lower layers ($l < l_*$) than higher layers ($l > l_*$).

Second, the overall diversity of a SSP solution $D(s)$ is defined as follows:

$$D(s) = \sum_{\boldsymbol{Req}_i \in s} (Req_i) \qquad (12)$$

Requests Reliability

The reliability of a request depends inherently on the reliability of the peer to which it is sent and the priority of the layers being requested in this request. The reliability of a request is computed by equation 13:

$$\rho(\mathrm{Req}_i) = \overline{l_*} \times r_i \qquad (13)$$

Where $\overline{l_*}$ is the priority of the dominant layer index in a request. In order to have a monotone increasing function, the priority of a layer l is $\bar{l} = L - l$ (i.e. lower layers are given higher priority values, and higher layer are given lower priority values).

Clearly, the reliability of a request is proportional to the priority of the dominant layer in the request and the reliability of the peer to which this request is sent. A plot of ρ is given in Figure 13 for a scalable stream composed of 15 layers and peer reliability ranging from 0% to 100%. This figure shows that in the ρ function, more importance is given to the layer priority than to the reliability of the peer. This is because the lower layers in a SVC stream are essential for its successful decoding.

The overall reliability of a SSP solution $R(s)$ is defined as follows:

$$R(s) = \sum_{Req_i \in s} \rho(Req_i) \qquad (14)$$

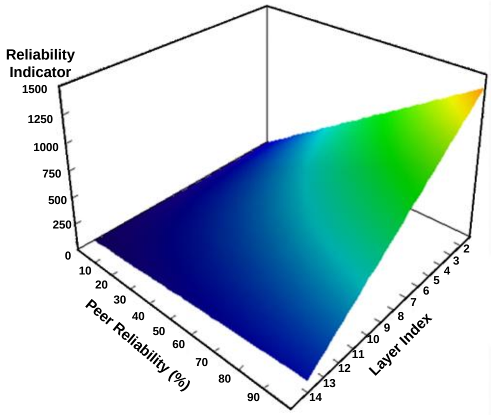


Figure 13: Reliability Indicator for a 15-Layer Stream

Requests Overhead

In our architecture, we have defined a BitTorrent-like protocol for data exchange between the peers. Each data piece, when sent to a peer, introduces an overhead (piece index, bloc index, offset) while the overhead of a request is defined by the request index and the data size. Thus, the overhead of a request depends on the number of pieces sent in the response, since all these pieces will share the same request header.

$$\eta(s) = \frac{1}{N} \sum_{Req_i \in s} |P_i| \qquad (15)$$

To model the overhead, we have also considered the ratio of response filling to the optimal response size C.

Figure 14: Example of Unfulfilled Requests

Thus, a response to a request that minimises the overhead will contain a great number of pieces while filling the response to its optimal size C.

$$w(Req_i) = C - \sum_{\pi_j \in Req_i} s_j \qquad (16)$$

So the overall overhead for a SSP solution is defined as follows:

$$W(s) = \sum_{Req_i \in s} w(Req_i) \qquad (17)$$

Having the entire indicators (i.e. the criteria to optimise), we build a simple objective function that is a linear combination of all the standardised indicators: request count, request diversity, request reliability and request overhead ($\hat{N}$, $\hat{D}$, $\hat{R}$ and $\hat{W}$):

$$f(s) = \alpha_1.\hat{N} + \alpha_2.\hat{D} + \alpha_3.\hat{R} + \alpha_4.\hat{W} \qquad (18)$$

In this objective function, we have modelled the above objectives (small number of requests, request composition in terms of layers, reliability of peers and communication overhead). By expressing the overall objective by the mean of a linear expression, we can tune the requesting mechanism by adopting different values of the coefficients α_i.

In our mechanism, we have considered a simple parameter configuration where all the coefficients are equal to 1.

With this requesting approach, a peer can form efficient requests dispatched to the connected peers in a way to:

- load balance the download from all the peers,
- request the important data pieces from the reliable peers,
- reduce the communications overhead.

4.1.2.3 Conclusion

In this sub-section, we tackled the optimal scheduling problem in pull-based real-time streaming systems in multilayer streaming scenarios. We modelled the problem as a Generalised Assignment Problem and we proposed a heuristic to resolve it. Then, we adapted the solution to non-layered streaming and we modelled it as m-cardinality assignment problem and we proposed a new solution for this problem. Then we proposed a new mechanism, based on the Harmony search algorithm, in order to aggregate chunks requests and ensure chunks load balancing among the senders.

4.2 Incentive Mechanism for Adaptation

In this section we address the issue of mechanisms which incentivise peers to contribute CPU, memory and bandwidth to provide adaptation services. We do not assume that nodes will simply follow a specified protocol. Instead, we consider overlay peers as rational entities whose behaviour is susceptible to modelling and the provision of incentives. In this regard, the current state of the art is sufficient for the purposes of ENVISION. In particular, to model adaptation incentives we tap into the rich body of work on the provision of incentives for CPU processing power. Two examples of this are CompuP2P [GSS06] and Spawn [WHH+92], which use open markets to provide incentives for the contribution of CPU and bandwidth resources. As is the case in many other proposals, these two rely on virtual economies in which peer wealth can later be converted into increased QoE.

The provision of contribution incentives for CPU and bandwidth can be more generally understood as belonging to the class of incentive problems in which peer resources are exposed as public goods in peer-to-peer overlays. This has led researchers to propose a suite of solutions based at least partially on standard results from economics regarding market failure and the Tragedy of the Commons [GR04]. In particular, the idea that any consumption of peer-to-peer resources by a peer has a value, and can be therefore imbued with a price, has been studied in great detail. In this formalism, peers are provided with an incentive to devote resources to the peer-to-peer system (acting as servers) because they will be paid by consuming peers (acting as clients). This payment can be either in actual national currencies or in intra-system currency exchangeable by electronic goods or services. In any event, the currency units are assumed to have an intrinsic value, and game theoretical formalisms rely on the conversion of peer utility in currency terms for the determination of appropriate payments (see, for instance, [GLBM01]). There are a number of drawbacks for purely monetary systems, such as the need for a micropayment infrastructure, central usage accounting and contract dispute resolution, trust anchoring and delegation, economic cycle control (e.g. inflation and deflation avoidance), valuation and pricing, and communication and computation overheads for market convergence. These problems notwithstanding, market systems have been repeatedly proposed in order to solve the incentives and resource allocation problem in a scalable fashion. In [FNSY96] use economic models to coordinate the activities of strategic agents. A general framework is presented and applied to load balancing, resource allocation, flow control, data management and QoS enabled queuing. Another example is [HT95], where the authors propose to include in the header of Layer 2 frames a field that describes the value of the frame, expressed in some fraction of an actual legal tender currency. When these packets are transmitted through inter-operator boundaries, the costs are accumulated and cleared periodically by using conventional electronic funds transfer. Thus, the authors propose the idea of micro-payment based inter-domain billing, with reputation systems used for risk minimisation.

4.2.1 Market-based Systems in the Research Literature

Another currency-based system is KARMA [VCS03]. In KARMA, each participant is assigned a single scalar, called its karma that is, at the same time, a currency amount and a reputation rating. Whenever a peer contributes resources, its karma is increased (the peer is paid), and when it consumes resources its karma is reduced (the peer pays). The karma of a peer is controlled by a bank set elected through Pastry [RD01]. Each one of the member peers of the bank set of another given peer maintains its balance and its latest transactions. If no peers left the system, the amount of karma in the system would remain constant: KARMA does not allow peers to create wealth. Thus, when new peers join the peer-to-peer system and old peers leave, the amount of karma decreases, increasing the risk of starvation. To prevent this, there are periodic, network-wide signalling episodes where each peer learns the total amount of karma and the total number of users in the system, and adjusts the karma accounts that they control to compensate for any surplus or missing karma. This is, from a practical perspective, of limited efficacy: any peer-to-peer operation that requires network-wide coordination can be very difficult to maintain successfully in practice.

Offline Karma [GH05] extends KARMA by relaxing its online transaction verification requirement on the bank set of each peer, therefore enabling offline transactions. Double spending of currency is no longer preventable, and thus the system is aimed at fraud detection and accountability. Offline Karma uses partial hash collisions to limit the rate of currency minting. Currency in this system is transferable by nested digital signing, in exactly the same way as iWat [Izu01, Sai06] and SHARP [FCC+03, CFV03]. However, Offline Karma limits the growth in coin sizes (which was first encountered in the context of digital cash schemes [CP92]) by forcing each coin to be “re-minted” periodically during the lifetime of the coin. When a coin is re-minted double spending can be detected and its associated history, in the form of its nested signature sequence, removed (this might be beneficial for privacy reasons, as well). The peers responsible for re-minting are elected in a similar fashion to the bank set in KARMA: by proximity in the ID space of a DHT.

Another similar system, (PPay) [YGM03], is a micro-payment scheme optimised for peer-to-peer applications. PPay seeks to eliminate the central bank bottleneck of common micro-payment systems by delegating some of its functions to the peers. Each peer purchases raw coins from a central bank and then is able to re-assign them to other users by digitally signing them. However, the coin growth problem of WAT tickets is avoided by implementing an online coin reassignment protocol, where the original holder of the raw coin is required to reassign the coin after each transaction. This means that online bank functions are taken by the peers themselves, and problems may arise if they are not continuously available to manage their reassigned coins. To this end, a downtime protocol is used. As payments are made offline, coin fraud cannot be prevented, but it can be detected and controlled. The system scalability is improved by using soft credit windows, which essentially implements micro-payment accounts for transaction amounts smaller than the minimum value of a coin.

One of the problems of such market-based approaches is price determination, which can be addressed in various ways. For instance, [WL05] proposes a market-driven bandwidth allocation procedure for overlay networks with strategic peers. Peers trade bandwidth units (in the form of one-hop flows) for currency units with peers in their neighbourhood. Bandwidth providing and consuming peers and are chosen so that a utility function that considers upload utilisation, download utilisation, paid currency costs and earned currency income is maximised. Pricing is determined by using reinforcement learning techniques, so that the price charged for each unit of bandwidth maximises utility in the light of past system performance. The system uses the Receiver Only Packet Pair bandwidth estimation method [LB99] to determine bottleneck bandwidth and thus feed this into the local utility functions.

4.2.2 Systems Based on Exchange/Barter Economies

The design of contribution accounting mechanisms can benefit from the analysis of bartering schemes for small communities, such local virtual currencies [Izu01, Sol96] or peer-to-peer lending [CGL09, Kla08]. The WAT system was conceived and launched in 2000, and it enables Japanese farmers to issue self-certified acknowledgements of personal debt (IOUs) that can be transferred along chains of trust, in such a way that there is no superior administration group controlling the relationships between any two ordinary participants [Lie04]. Thus, in a way, the WAT system is a conversion mechanism between capital in currency form and social capital. In practice, the WAT system works by providing enhanced bartering possibilities for the users of the system, which is now spreading rapidly and widely in Japan (due to its government-independent nature, it is difficult to ascertain exactly what its scale of use is, and how many participants are involved in it). Its users tout it as an effective means of payment, which in addition creates and strengthens cycles of trust among groups of participants.

The operation of the system is simple. Any person can issue a WAT ticket, which becomes active by being signed. The ticket represents a commitment on the part of the issuer to present to the bearer, at redemption, certain goods (or to provide a given service). Each recipient of the ticket may pass it on to other economic actors by signing the ticket with his/her personal signature. Therefore, the ticket becomes a record of all the transactions that it has supported, and by navigating social trust chains, it transforms reputation and trustworthiness (social capital) into economic solvency.

In [AL04] the WAT system is complemented by integrating self-guarantees, in the form of legal tender currency. Thus, economic value on a given WAT system can be supplemented if necessary by tying its value to other currency systems (maybe even other complementary currency systems). This means that different currency systems can inter-operate with each other in a peer-to-peer fashion, by converting their monetary value into trust (operationally, this is similar to the currency exchange procedures for the Lightweight Currency Protocol [TR04]). Like KARMA (and conventional digital transferable cash [CP92]), iWat, the digital implementation of the WAT system [Sai06], has the property that as a WAT ticket circulates the economy, it grows in size, as each participant peer adds its own digital signature. Another very similar system is SHARP [FCC+03, CFV03], an architecture for a generalised computational economy based on pair-wise barter chains. It uses probabilistic assurance over resources2 and self-signed identities, and resource entitlement tickets that can be delegated through nested signing (this is identical to [Sai06] and [Izu01]).

Another way of allowing peers to create their own currency is by defining pair-wise currencies: Each peer will have a currency to transact with every other peer. In this case, all transaction state information can be kept by the two interested parties (as opposed to (Izu01, FCC+03, CFV03, Sai06) where state is recorded in the tickets/coins themselves), and the system demands in terms of cryptographic protection are alleviated. One such system is SWIFT [TPM04], where each peer maintains an account for each one of its market peers. Trading on these pair-wise currencies amounts to subtracting or adding to the other peer's account. By explicitly constructing trading strategies that take into account altruism and risk preferences, and that track the contributions of each one of the peers to others, argue in [TPM04] that participation in the system is advantageous for strategic peers, and that moral hazard is avoided.

Many of the trade mechanisms that have been detailed so far fail to differentiate between the actual tradeable resources of the peer (e.g. bandwidth) and the content that it possesses (e.g. media fragments). In [GEv07], propose the formation of alliance groups to allow peers to contribute upload bandwidth to other peers even if they do not possess any content that the other peer might be interested in. Basically, the authors modify the BitTorrent protocol so that peers aggregate in trusted groups where peers download fragments on behalf of each other. Each peer in the group then becomes, for each particular swarm, either a collector that is interested in participating in the swarm, or part of a set of allies that are not interested in the swarm but help the collector to achieve high download rates by downloading fragments themselves (as instructed by the collector) and relaying them, in the expectation of reciprocity at a later time. This can be achieved because the allies may have content that the collector does not have and which may be of interest to the peers that have the content that the collector seeks. They name their approach Amortised Tit-for-Tat, because the allies collect fragments on behalf the collector only out of a reciprocal altruism expectation. Since this reciprocative altruism expectation is not managed by their protocol, assume that alliance groups are formed only within a high-trust social network (e.g. between friends and relatives). Another strength of [GEv07] is that, by eliminating the assumption that sharing in peer-to-peer networks is done in terms of mutually useful content, it decouples the problem of content search, distribution and replication from the incentives problem. In this case, peers use their resources to download content for which they have no interest, in order to “store” this contribution and use it later to download different content.

4.2.3 Systems Based on Auctions

One of the problems that currency-based systems have is the pricing of peer resources. A simple way to achieve this with no central control, and if the peers can calculate their own valuations for resources, is through auctions. An example of this is CompuP2P [GSS06], a system that implements an open market for peer resources quantised into different markets. Each market is managed by a particular peer through a Chord [MKKB01] DHT. Pricing is arrived at by using iterated single-item, sealed-bid, second price auctions (*Vickrey* auctions [Vic61]) as an information-revelation mechanism, and the system is designed to be robust in the presence of self-interested decisions of resource providers and users.

Another auction-based system is Spawn [WHH+92], where a distributed CPU resource allocation problem is solved by using an open market. Money in this virtual economy becomes an abstract form of priority, so that better funded processes can obtain correspondingly better access to the computing infrastructure than others. The system also uses iterated Vickrey auctions, and each of the economic actors taking part in the economy for CPU cycles maintains a resource manager that manages auctions and assigns CPU time slices accordingly.

The previous two examples of auction-based systems used only single-item auctions. This is representative of much of the work in the literature - few studies use full combinatorial auctions, since they can be very computationally intensive and involve large delays [CSS06]. Sometimes, simplifications are made to make combinatorial auctions tractable. In [FLZ05], each participant allocates its finite budget to bid on a given resource set, and receives a proportion of each resource commensurate with the proportion of its bid with the bids of other participants; this same technique was later on proposed by [LLSB08] as a replacement for the unchoking policy of BitTorrent. The Nash equilibria of this game are analysed, and it is unsurprisingly found that they depend on the precise structure of the utility functions for the participants. However, for practical simulation cases, it is found that the Nash equilibria attained have efficiencies close to the social optimum. Additionally, the equilibria are shown to exhibit good utility uniformity (the difference between the minimum and the maximum utilities experienced among the users is low) and envy freeness [Var73] (the utility that a participant enjoys given its outcome is close to the utility that it could have enjoyed had he received the outcome of any other participant). Another auction-based resource management implementation is Mirage [CBA+05], where combinatorial auctions using a centralised virtual currency environment are used for sensor network test-bed resource allocation.

Some peer-to-peer mechanisms use auction-based techniques, even without explicitly stating so. One example of this is GNUNet [Gro03], that uses iterated single-item, sealed-bid, first price auctions [MM05] to resolve resource contention (only the incentives aspects of the protocol are detailed in [Gro03]; see [BG03] for the anonymity related protocol aspects). Essentially, GNUNet implements a trust-based economy, where each peer maintains a trust rating for every peer it interacts with. If a peer takes the role of a client, it is making use of overlay network resources and is expected to lose trust in the eyes of other peers, whereas a peer taking the role of a server is donating its resources to the peer-to-peer system and is expected to gain trust in the eyes of other peers. When a peer requests a protocol action that implies a cost (such as forwarding a given message), it associates with the request a number (called the priority) that measures the importance that the sender places in its request being satisfied. The priority becomes, essentially, its bid.

If the priority of the request is high enough to be allocated service over its competing requests, the serving peer decreases its trust on the requester by an amount equal to the request priority (if the peer is insufficiently trusted by the serving peer to completely perform this subtraction, its request is treated as having priority equal to zero). When the requester receives the response, it will increase its trust in the responder by an amount equal to the priority of the related request. In this way, a payment has been effected from the requester to the responder by only modifying local accounts. Therefore, each the trust that the system places on each peer is not under its control; it is under the control of the peers that interact with it. Additionally, given that trust itself encodes information regarding the best peers with which to carry out transactions, it is to the best advantage of every peer not to arbitrarily modify their trust in other peers. In GNUNet trust is not used to store value, nor to limit access to resources; it is used to prioritise in case of excessive demand. In case of resource contention, higher priority requests are served preferentially over lower priority requests, as long as their originating peers are sufficiently trusted to support the corresponding priority. By allowing more trustworthy peers to set higher priorities, the protocol rewards peers that have served urgent or important requests, and in so doing, increased the utility of the peer-to-peer system.

4.2.4 Adaptation Incentives in ENVISION

As briefly shown in this section, the state of the art in the provision for participation incentives is vast and already well equipped to deal with the requirements of ENVISION. Since media-aware adaptation is essentially a CPU-bound task that can sometimes be expedited by using specialised hardware or software (especially in the case of adaptation), those incentive mechanisms focusing on CPU are well equipped to deal with the requirements of ENVISION. In addition, given that different overlays have different requirements and tolerate different kinds of strategic peer behaviours, no single incentive mechanism can be adopted by ENVISION. Instead, each overlay will implement their own in a case-by-case basis. Decentralised systems such as those presented in this section are much better aligned with overlay applications that are not operated by a single authority and have no centralised coordination functionality. For other overlays which rely on support from infrastructure provider-provisioned nodes and already implement some centralised functionality, centralised incentives mechanisms are feasible, and these tend to be faster and more reliable.

5. ERROR RESILIENT TRANSMISSION

5.1 TCP-Friendly Multi-Link Rate Control and Forward Error Correction

In this section we explore the question of how to design a TCP-friendly rate controller for non-TCP traffic using multiple end-to-end paths with different characteristics simultaneously. Preliminary results of this work have been published in [PR11].

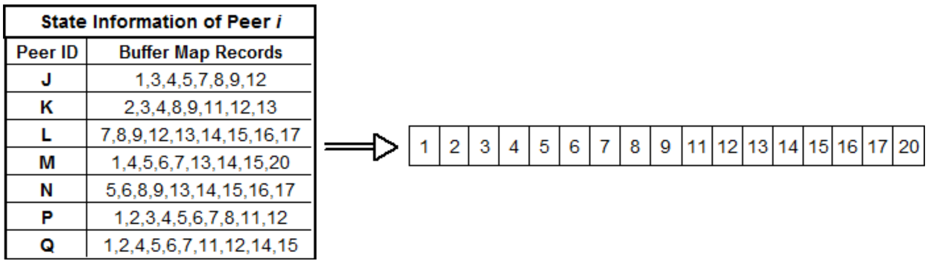
When designing a rate controller for non-TCP traffic, how the traffic shares bottleneck links with TCP traffic is used as a measure of fairness. TCP traffic adapts its sending rate according to the path characteristics it observes. This enables TCP flows to share bottleneck links fairly without direct knowledge about where bottlenecks are. Since most of Internet traffic is TCP, some kind of rate control is desirable for non-TCP traffic since overly aggressive flows can cause competing TCP flows to overly reduce their rates. We found that TCP-friendly rate control for unicast flows [FHPW00] does not directly extend to the multipath case. Furthermore, the TCP rate controller for multiple sub-flows which is currently being standardised by the IETF [FRHB11, RHW11, WRG11] does not cope well with some forms of non-TCP traffic, for example streaming traffic.

The second part of the problem is that there is more than one choice of rates that is considered TCP-friendly. TCP theory uses per-user utility functions that control the sharing properties of TCP while all sub-flows are judged according to their throughput. By introducing per-sub-flow utility functions, we can benefit from sub-flow specific features such as per-interface energy consumption or varying per-sub-flow goodput and select rates that maximise other objectives than throughput in heterogeneous systems. We present a rate controller that manages to take per-sub-flow utility functions into account. The controller is cooperative and makes all flows in the network communicate through the drop rates they observe per sub-flow. It guarantees an improvement in utility for each user in comparison to the best single path solution. At the same time it drives a network of flows towards the network wide global optimum.

6. SHORT TERM CACHING

Caching in the current Internet is mainly used for web traffic but its benefits and applicability to P2P networks have also started to draw research attention [DLLLJ11]. The caching mechanism presented in deliverable D3.2 represent an alternative way to unify the different desires of P2P users and network operators and, in particular, mitigating inter-ISP traffic costs while improving the service quality to P2P users. In this deliverable, we present the short term caching solution in which peers store the stream (or portion of the stream) to serve the other peers in the overlay. The main focus here is to support on-demand streaming applications in which content is pre-recorded and users can access different portions of streams at any point of time.

For short term caching in ENVISION we propose a new "*cooperative caching technique*". In this technique the peers organised in an overlay exchange buffer maps with the neighbouring peers. All the peers in the same neighbourhood exchange buffer map to obtain the information of available segments. Once the state information is collected from all peers (in same neighbourhood), each peer creates a table of available segments in that particular session. Figure 15 shows the state information table received by a particular peer *i*.



State Information of Peer *i*

Peer ID	Buffer Map Records
J	1,3,4,5,7,8,9,12
K	2,3,4,8,9,11,12,13
L	7,8,9,12,13,14,15,16,17
M	1,4,5,6,7,13,14,15,20
N	5,6,8,9,13,14,15,16,17
P	1,2,3,4,5,6,7,8,11,12
Q	1,2,4,5,6,7,11,12,14,15

1	2	3	4	5	6	7	8	9	11	12	13	14	15	16	17	20

Figure 15: Removing Segment Redundancy

Each peer performs the necessary computation to remove redundancy and creates a list of available non redundant segments in the session. The peer then requests a segment which is near the front of its "playhead" position. In the case of Figure 15, missing segments like 10, 18, and 19 would be requested from peers in other sessions depending on the current playhead position. Thus the segment request is broadcasted to other sessions. As a result, those rare segments are obtained from other sessions that did not exist in the current session. Later on, if a seek operation is carried out and the segment is available in the same session, it will take less time to acquire it from neighbour peers instead of server or far peers. If there is no response from shortcut neighbours, the desired segment is requested from the server as a last resort. The algorithm for cooperative caching is presented in Figure 16. It is important to note that each peer also prefetches the segments near to its playhead position as an *urgent downloading target.* In our case, each peer prefetches the next 20 seconds of video segments as an urgent downloading target. Apart from these segments, remaining segments are prefetched using cooperative caching techniques.

```
Segment Select ()     //Find the next segment to prefetch
{
  Find the segments $S_i$ that didn't exist in buffermap
        Return S;     //S is the desired segment;
}
Void Prefetch ()      // The function to do prefetching
  {
     While (prefetching set is not empty)
       {
        segment S = select ();
        Broadcast(S); //Broadcast the prefetching of segment
       If (segment S is cached by a peer in same session)

       //situation where same segment is also requested by some other peer
             {
              Download segment S;
              Remove the segment S from prefetching list;
             }
       else if (segment S is located on a remote peer P)
             {
              Connect with the peer P;
              Download Segment S;
              Remove segment S from the prefetching set;
             }
       else // when timeout expires
             {
             Send the segment request to server;
             Connect with server;
             Download segment S;
             Remove segment S from the prefetching set;
             }
       }}}
```

Figure 16: Cooperative Caching Algorithm

7. SVC LIVE VIDEO GENERATION AND DISTRIBUTION OVER MULTIPLE WIRELESS CHANNELS

7.1 Introduction

Today many types of smart phones have more than a single wireless interfaces such as 802.11, 3G, 4G and more, we refer to such devices as a Multi-Link Enabled Peer (MLEP) and describe in here how the MLEP using SVC encoding and its wireless interfaces would generate a smoothed and reliable video stream. The use of multiple interfaces is becoming more and more valuable as standards are becoming mature [MPTCP, 802.21], and existing devices supports multiple interfaces, recently is was announced that SKT is planning to launch 3G+Wi-Fi real multilink systems, and later LTE+Wi-Fi, for higher end-user capacity [SKT]

In live distribution the source has a significant role in the distribution chain and it influences the perceived quality for all the receivers. The source peer is connected to the camera and generating the compressed video in real time to be distributed over the wireless interfaces, from anywhere anytime, this is aligned with the recent trends of user generated content and personal broadcast. Major broadcasters such as CNN, AP, BBC, and more are increasing the use of cellular bonding technologies to deliver live news to the internet and to TV users, such services are not tolerant to low quality of experience and the system should be optimise for the worst case scenarios.

In P2P live distribution, the source might be a single wireless peer with limited bandwidth. In such a case it is a challenging task to provide continuous high quality to the receiving peers in the swarm while creating a swarm that can potentially feed many users. In that sense, duplicating delivered content to a number of peers from the source would decrease the quality of the video. In these cases, it would seem better to diversify by sending different data to different peers to maximise the multiplexing gain of the swarm and to inject the highest quality possible from the source to the wired network. The HCN suggested by the ENVISION project in D3.1, is a perfect candidate destination peer for the MLEP stream, while the MLEP makes the optimisation over the uplink direction, the HCN makes it over the downlink path, combining the two technologies provides a complementary solution as further details in ENVISION D6.1 [D6.1].

The SVC technology as described in detail in ENVISION D5.1 [D5.1] and D5.2 is a potentially good technology to overcome channel and wireless distribution path conditions as it provides the ability to drop higher layers either in the network or at the source buffers while preserving good stream quality. However in the suggested literature so far the mechanism and content generation assume the creation of fixed layers (i.e. resolution, frame rate…) followed by transport optimisation mechanism such as scheduling as detailed in chapter 4.1. This approach is most suitable for VoD style service where the chucks and the content encoding is done offline and the distribution is in real time, as a result of the predefined fixed layer approach, the stream suffers from sharp quality changes when the number of delivered layers is changed. The following subsections propose an algorithm which smoothes the SVC layer creation in order to avoid the step sharp changes of the perceived quality while adapting to channel conditions. Furthermore, in this work we describe a novel approach for improving the smoothness of quality changes by suggesting dynamic layer generation to meet the underlying network conditions and provide an adaptive streaming solution of SVC content, which include layers generation, protection and scheduling.

7.2 System Overview

In accordance with the functional model defined in ENVISION D4.1, the MLEP related technologies are described in Figure 17 below. The details of the SW design, the main interfaces, information flow and the algorithms are provided in the following subsections, and described here in a high-level manner.

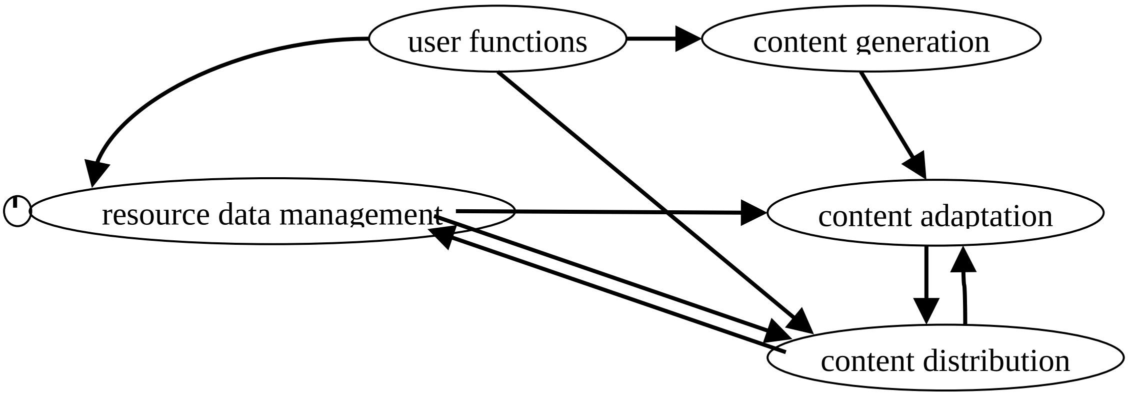


Figure 17: MLEP functional model

This functional model describes the main components of a source peer for generating live video streams, starting from the user preferences, which controls a set of attributes with accordance to the user preferences, such as controlling the attributes of the encoder or the distribution as further described in 7.2.1. The data and resource management is responsible for monitoring of the channels as well as the session attributes, such as the buffers state, and links momentary and statistics performance, based on the monitor process and the available predictions, the resource and data manager is responsible to assign the target transmission rate, video encoding rates and other attributes as further defined in 7.2.2. In particular, the resource data management specifies the expected “channel bit rate” and the “worst case expected bit rate”. The main target of the content generation and adaptation modules is to optimise the stream quality, to be as smooth as possible assuming the expected rates, while allowing high protection levels for and preparing for the worst case conditions. Thus the generated layers are protected by dynamic FEC to mitigate with the varying channel conditions and are scheduled over the multiple interfaces, taking into consideration links quality i.e. delay, jitter, loss, bandwidth and correlations as further details in section 7.4.

7.2.1 User Preferences

MLEP provides some freedom to the user to set preferences over the system performance as further detailed in the following sub-sections. This module exposes interfaces to the user in order to collect the following main attributes as described below.

7.2.1.1 Continuity vs. Sharpness

The trade-off between continuity and Sharpness is a parameter used by the encoder to handle the user preference. Generating content in a given average bitrate could be done in many ways. The encoder consists of look up tables for mapping between target bitrate and encoder attributes such as frame per seconds and quantisation factors that controls the SNR and more. The user based on his knowledge of the scene (i.e. football or static scene) can control this attribute to favour between SNR and continuity and thus to improve the overall QoE based on the user preference.

7.2.1.2 Delay

In order to utilise the available bandwidth to the maximum, in a varying channel conditions there is a trade-off between delay and the variable encoding rate. As much as the delay is reduced it is required to change the quality of the stream to follow the varying links conditions. The stability of the stream quality is thus determined by the delay. If the delay is large enough (more than the channel “delay spread”) the encoding quality could be fixed while transmission follow the available quality. The user thus can set the preference of delay Vs quality stability to control the system behaviour accordingly.

7.2.1.3 Stream Stability

The Stability of a stream could also be achieved at the expense of the throughput utilisation; a stream of lower rate (utilising let say 80% of the available bitrate) is more likely to maintain its stability than a stream that utilises the available bandwidth to the maximum. Thus the user can choose to reduce utilisation in order to achieve greater stability.

7.2.2 Resource Data Management

Resource data management is responsible for channel performance monitoring, session monitoring and predictions in order to allocate resources for the transmission as well as for the content generation. The prediction of expected and worst case scenario is a key issue in maintaining high QoE throughout the entire session, and mitigating effects due to network conditions. It allows content creation and adaptation based on these predictions, while “planning” for the worst case scenario. The prediction is based on both real-time short-term data and long-term offline data. In current 3G and 802.11 networks there is no guarantee of certain QoS for data delivery. Some examples of collected data are provided in Figure 18, which shows that the networks conditions such as delay are varying significant and are generating challenging conditions to meet. In D4.2 [D4.2] we elaborate more on the channel monitoring, on the offline database as well as on the session monitoring, which all provides the assignments of network resources to the scheduler and feeds the content generation and adaptation modules.

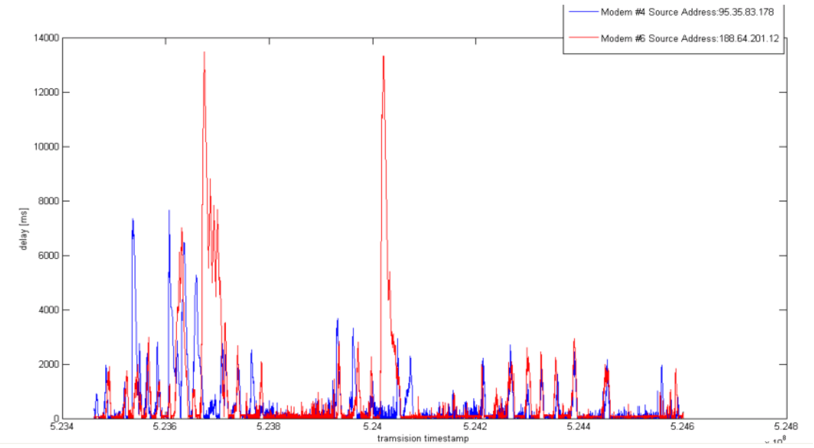


Figure 18: Delay of two cellular links

7.2.2.1 *Performance monitoring module*

The performance monitoring module has 3 main components as illustrated in Figure 19, the Statistics sub module, the History DB, and the Analysis and Output Decision which is the main module.

The History DB, is based on analysis of hundreds of events and log files, and it assist in the prediction of what could be the worst case condition given the momentary conditions, the momentary conditions contain short term statistics which include the correlations between modems for a number of attributes like the delay, throughput and loss, the Decision module rely on both short term and offline data, as well as environmental attributes like the operator, number of modems ad more to generate the output values which are used in both the content generation, adaptation and distribution as further discussed below in section 7.2.3 and 7.3.3.

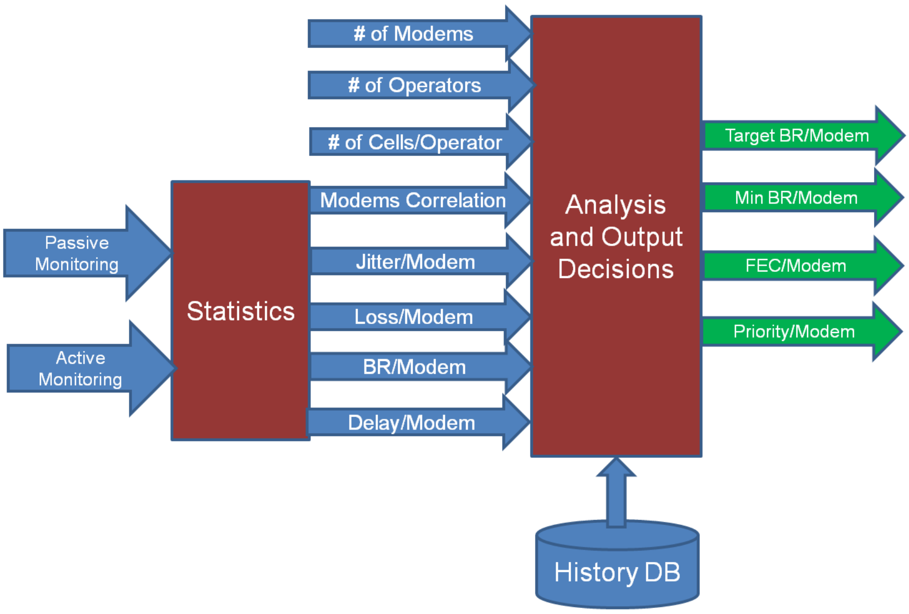


Figure 19: Monitoring Module design

7.2.3 Content generation & Adaptation

The content generation and adaptation module is responsible for creating the SVC/AVC stream with the target bitrate retrieved from the monitoring module in the resource data management 7.2.2 as specified in section 3.1. It receives both the expected bitrate as well as the worst case bit rate and generates the stream accordingly to allow the protection of the stream to meet the expected conditions. The main algorithm is described in detail in section 7.3.

In the following sub-sections we focus on the live SVC transport over multiple channels.

7.3 SVC Transport over multiple channels

The transport of SVC layers over multiple channels in the form of point-to-point has not been intensively studied. The benefit of using SVC in a point-to-point manner and its relation to multiple interfaces is the subject of this chapter. It is provided here to complement the SVC layer generation. It has been shown that, in P2P systems, using SVC and MDC (multiple descriptor coding) provides reliability in case where peer drops the stream as described in ENVISION D5.1 [D5.1]. It is assumed that all the peers hold the same information and can mutually share a codeword by using a simple scheme to deliver the elements. In the case of multiple interfaces for point-to-point as we describe in this section, we claim that we can achieve a better distribution of the codewords and allocation of the SVC data over the multiple channels since the source and destinations are collaboratively using the links and thus holding the knowledge of all paths correlations at both sides and avoiding a simple scheme in a more optimised manner. This allows the source peer to protect and distribute the SVC content in cross scheduling and protection manner over the multiple channels as described in the following subsections.

7.3.1 The Bonding Challenge

While high-quality video experience relies on smooth and uninterrupted video delivery, cellular links are inherently unstable and fluctuate continuously**.** Transmitting video over such a link may result in black screens, video breaks, pixelisation, jitters, audio problems, lip synchs etc., even from a stationary location and over 4G. Parameters that impact the experience and can change over the course of a few to tens of milliseconds include: uplink bandwidth, uplink latency, loss rate or all of them together.

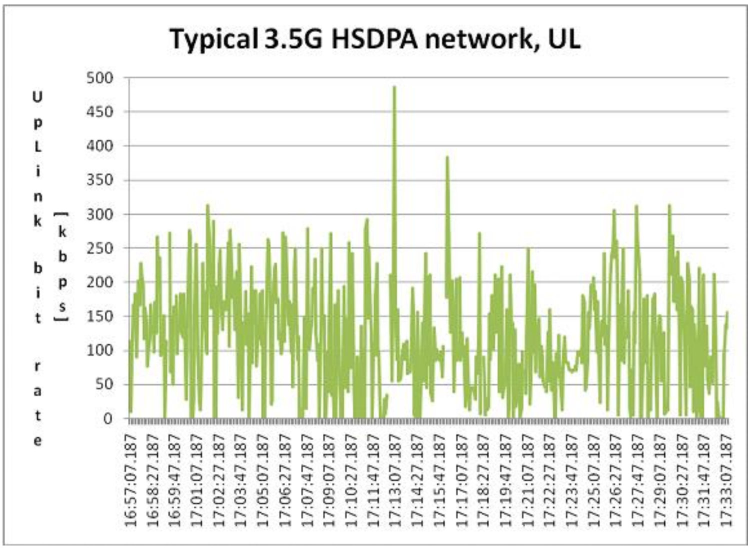


Figure 20: Example of single 3.5G modem uplink bandwidth fluctuations

As a result, there is an inherent gap between the desired quality of experience and cellular technology capabilities. Cellular-bonding bridges this gap.

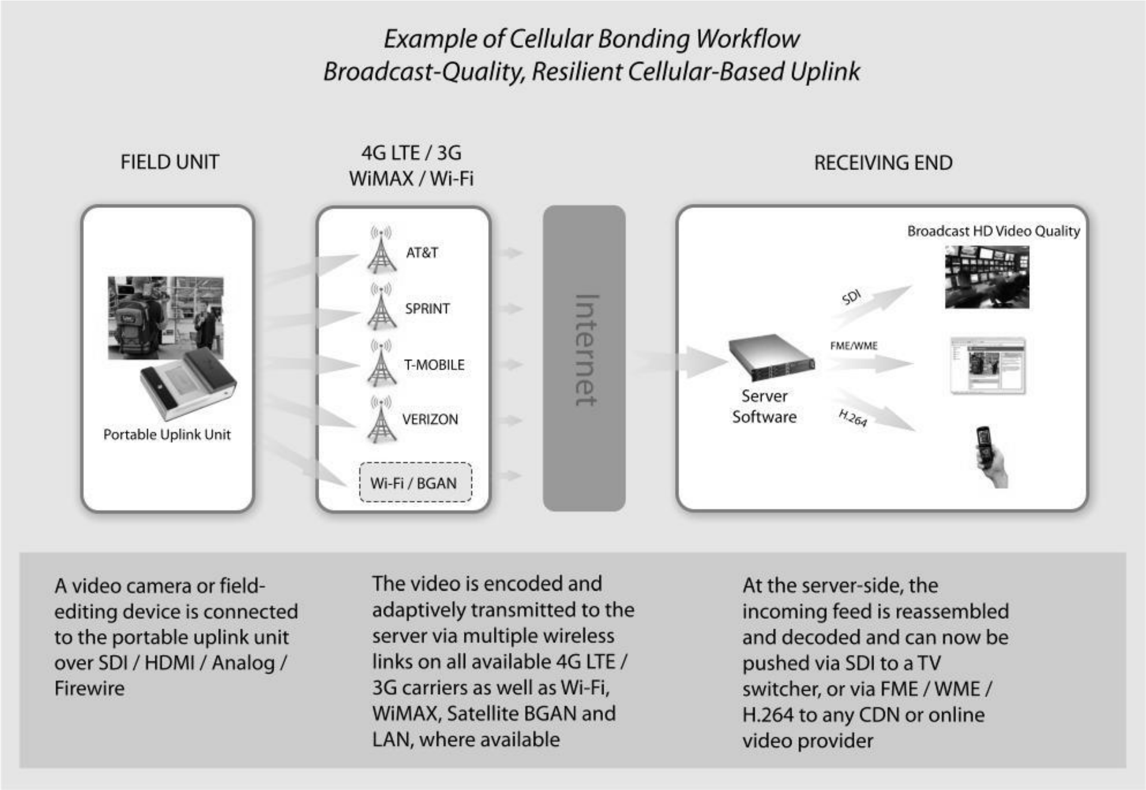
The main idea of multiple interface scheduling is that each channel has its own statistics for bitrate, delay, loss and jitter. Looking at the group of channels as an aggregated link combining the above statistics is a valid option. However, it is sub-optimal as the freedom to choose which packet goes through which modem is not being used, as a result the most urgent packet may be delivered over the most congested path. A better approach is to specifically choose which packet at the sending buffer goes through which of the interfaces in order to optimise a criteria such as minimising the delay or maximising the arrival probability. For a SVC stream which has layers with a different priorities, matching the packets to the path can reduce the delay or alternatively per a given delay increase the transmit buffer size, and thus allow better scheduler smoothness, and FEC level of protection as elaborated in section 7.4.3.

7.3.2 The Bonding Architecture

Cellular bonding involves taking multiple cellular modems, compressing the video (usually H.264), and transmitting the packets over each of the modems as described in 7.3.3. Instead of relying on a single unreliable link representing a single point of failure, the risks while achieving a higher performance are minimised.

Bonding solutions differ according to how much they reduce the risks and on how much they increase the performance.

To achieve this, the solution has to continuously, and in real-time, monitor all available links and understand how it can make optimal use of them, now and in the near future. It also has to dynamically adapt the video encoder according to the momentary total available bandwidth in all links, compensate for, and recover from, any losses, and smartly interacts with the operator to best suit its needs. On the receiver side, which can be anywhere in the world, software installed on any Internet-connected PC receives those multiple packet streams and reconstructs the video.



A video camera or field-editing device is connected to the portable uplink unit over SDI / HDMI / Analog / Firewire

The video is encoded and adaptively transmitted to the server via multiple wireless links on all available 4G LTE / 3G carriers as well as Wi-Fi, WiMAX, Satellite BGAN and LAN, where available

At the server-side, the incoming feed is reassembled and decoded and can now be pushed via SDI to a TV switcher, or via FME / WME / H.264 to any CDN or online video provider

Figure 21: Cellular Bonding - transmission over multiple networks

7.3.3 Scheduling and FEC Algorithms over multi-link

The generated SVC content has different importance to the perceived QoE. The basic layer is the most important layer to the perceived QoE, and thus it should be the most protected one. Protection does not necessarily mean more FEC overhead. Mechanisms for sending those packets over more reliable links are considered which provide improved robustness against burst loss conditions. Experience shows that the most severe phenomena are mitigated with a burst loss where the entire data over a certain period of time is considered as loss data, either because of a sudden increase of the delay of as a result of mobility to 2G or a temporally loss of connectivity in 3G/4G networks. This phenomenon is unexpected and occurs immediately, not allowing the content to be adapted to the event at the time of the event.

To meet with modem/path errors or loss phenomena's the content should be protected in a crosslink manner allowing a drop of one or more links while preserving the more important layers. Thus the group of available links is no longer considered as a single channel with the aggregated characteristics of all links, alternatively each of the links based on the monitoring are graded and evaluated for his jitter, delay, throughput, loss and cross correlation with all the other links.

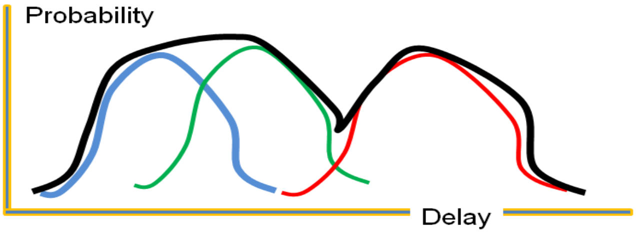


Figure 22: illustration of probability density functions for the delay of multiple modems

This is illustrated in Figure 22. The black line represents the (non normalised) aggregated probability density function for all three links while the red, green and blue lines represents the PDF of each link separately. These types of statistics are done for the Delay, loss, jitter, bitrate, by the data resource management module. The challenge in scheduling and FEC over those links is to maximise the QoE perceived, taking as input the SVC stream with different priorities, and different characteristics such as layer bitrate. Different optimisation criteria results in different scheduling:

- Delay optimisation: in that case the SVC base layer should be delivered over the blue modem, and should be protected with FEC accordingly to the predicted loss behaviour of the modem.
- Resiliency optimisation: in that case the base layer should be delivered over channels which are more secure, moreover, since, in our experience the major events of loss can occur suddenly for a specific modem, this would be in a number of situations, for instance toggling between 2G, and 3G networks, or congestions at the cell level, or sudden loss of signal. To mitigate with that, the data of the base layer should be spread over as many links as possible satisfying the delay criteria, the amount of redundancy should take into consideration a drop of modem or more.
- Smoothness optimisation: mapping of the data over the links in a smart manner may reduce the delay or alternatively increase the transmission buffer, by mapping of early arrived data (oldest in the buffer) to the lowest delay link and the late arrived data (newest in the buffer) to the highest delay link will reduce the maximum delay, as it avoids sending more urgent data over high delay links, this allows an increase to the transmission buffer which provides better smoothness of the scheduling as detailed in section 4.1.

8. CONCLUSION

In this deliverable, we presented the work performed in content generation and adaptation during the second year of the ENVISION project. A special attention has been given to the content adaptation. We investigated especially the smoothness of the stream and how to minimise the number of layer changes under varying network conditions, while achieving a high layer delivery ratio. In order to attain this goal, we proposed two novel contributions. The first one aims to reduce the layers amplitude variation in layered P2P streaming, while the seconds aims to reduce the layers frequency variation. We investigated then a hybrid approach which aggregates the benefits of the two previous proposed mechanisms. These contributions are accepted to be published in ICC'12 [AA12] proceeding. We tackled, then, the scheduling problem in pull-based real-time streaming systems in multilayer streaming scenarios, and we proposed mechanism to efficiently request the appropriate layers from the proper peers. For that purpose, we modelled the problem as a Generalised Assignment Problem and we proposed a heuristic to resolve it. Then, we adapted the proposed solution to the non-layered streaming since, in ENVISION, we consider peers with no layers stream encoding/decoding capabilities, too. This contribution is also accepted to be published in ICC'12 proceeding [BA12].

Once the scheduling mechanism is decided, we went further to optimize and aggregate the chunks requests and ensure chunks requests load balancing among the senders, and we propose a mechanism based on the harmony search algorithm, to achieve this goal. This contribution is published in CAMAD'11 proceeding [MAEG11]

In order to motivate peers to contribute CPU, memory and bandwidth to provide adaptation services, we drew up a state of the art of existing incentive mechanisms. We distinguish three main incentive mechanisms: market-based, exchange/barter economies based mechanisms and auctions based mechanisms. We consider that the studied mechanisms are well equipped to deal with requirements of ENVISION.

With regards to the stream transport in ENVISION and error resilient transmission mechanisms adopted in ENVISION, we presented a new TCP-friendly rate controller for non-TCP traffic using multiple end-to-end paths with different characteristics simultaneously.

Finally, we investigated the benefit of using multiple links for content distribution. In this perspective we investigate, especially, allocation of the SVC data over different layers and its transport over the multiple channels. This allows the source peer to protect and distribute the SVC content in cross scheduling and protection manner over the multiple channels.

9. REFERENCES

3GPP10	3GPP, "Transparent end-to-end Packet-switched Streaming Service (PSS) - Protocols and codecs (Release 9)", March 2010.
802.21	http://ieee802.org/21/
AA12	U. Abassi and Toufik Ahmed, "Smooth Delivery of Layered Video Stream in P2P Networks", accepted in IEEE ICC'12
Ad10	Adobe, "Using ADOBE HTTP DYNAMIC STREAMING", 2010.
AL04	M. Ardron and B. Lietaer, Complementary currency innovation: Self-guarantee in peer-to-peer currencies, International Journal of Community Currency Research 10 (2004), 1-7.
App09	Apple, "HTTP Live Streaming Overview", November 2009.
AR05	V. Agarwal and R. Rejaie, "Adaptive Multi-Source Streaming in Heterogeneous Peer-to-Peer Networks," Proc. Multimedia Computing and Networking (MMCN '05), Jan. 2005.
BA12	A. Bradai and T. Ahmed, On the Optimal Scheduling in Pull-based Real-Time P2P Streaming Systems: Layered and Non-Layered Streaming, accepted in IEEE ICC'12
BG03	K. Bennett and C. Grothoff, GAP - Practical Anonymous Networking, Designing Privacy Enhancing Technologies, Springer-Verlag, 2003, pp. 141-160.
BIT10	BitTorrent Protocol Specification (V1.0): http://wiki.theory.org/BitTorrentSpecification, 2010.
BSWG07	P. Baccichet, T. Schierl, T. Wieg, and B. Girod, "Low-delay Peer-to-Peer Streaming using Scalable Video Coding," in In Proc. of International Conference on Packet Video, 2007.

CBA+05 B. N. Chun, P. Buonadonna, A. AuYoung, C. Ng, D. C. Parkes, J. Shneidman, A. C. Snoeren, and A. Vahdat, Mirage: A Microeconomic Resource Allocation System for SensorNet Testbeds, Proc. of EmNetsII, 2005.

CFV03 B. Chun, Y. Fu, and A. Vahdat, Bootstrapping a distributed computational economy, Workshop on Economics of Peer-to-Peer Systems, 2003.

CGL09 N. Chen, A. Ghosh, and N. Lambert, Social lending, Proceedings of the ACM Conference on Electronic Commerce (EC'09), 2009.

CP92 D. Chaum and T. P. Pedersen, Transferred cash grows in size., EUROCRYPT, 1992, pp. 390-407.

CSS06 P. Cramton, Y. Shoham, and R. Steinberg, Combinatorial auctions, MIT Press, 2006.

D4.2 ENVISION deliverable D4.2, Refined Specification of Consolidated Overlay View, Data Management Infrastructure, Resource Optimisation and Content Distribution Functions January 2012, FP7 ICT ENVISION project, http://www.envision-project.org

D5.1 ENVISION deliverable D5.1, Initial Specification of Metadata Management, Dynamic Content Generation and Adaptation, January 2011, FP7 ICT ENVISION project, http://www.envision-project.org

D6.1 ENVISION deliverable D6.1, Initial Testbed Description and Preliminary Evaluation Results of Content-aware Cross-layer Optimisations for Advanced Multimedia Applications, January 2012, FP7 ICT ENVISION project, http://www.envision-project.org

DCX07 L. Dai, Y. Cui, and Yuan. Xue, “Maximizing Throughput in Layered Peer-to-peer Streaming”, in Proc. IEEE ICC, 2007.

DLLLJ11 J. Dai, B. Li, F. Liu, B. Li, and H. Jin. On the Efficiency of Collaborative Caching in ISP-aware P2P Networks. In Proc. of INFOCOM, Shanghai, China, Apr. 10–15, 2011.

FCC+03 Y. Fu, J. Chase, B. Chun, S. Schwab, and A. Vahdat, SHARP: an architecture for secure resource peering, Proc. of SOSP, 2003.

FCPK04 F. Fitzek, B. Can, R. Prasard, and M. Katz, Overhead and Quality Measurements for Multiple Description Coding for Video Services, in In Proc. of International Symposium on Wireless Personal Multimedia Communications, Sep. 2004.

FHPW00 S. Floyd, M. Handley, J. Padhye, and J. Widmer, Equation-based Congestion Control for Unicast Applications. In Proceedings of SIGCOMM, pages 43–56, New York, NY, USA, 2000. ACM.

FLZ05 M. Feldman, K. Lai, and L. Zhang, A Price-Anticipating Resource Allocation Mechanism for Distributed Shared Clusters, Proc. ACM EC, June 2005.

FNSY96 D. F. Ferguson, C. Nikolaou, J. Sairamesh, and Y. Yemini, Economic models for allocating resources in computer systems, Market-based control: a paradigm for distributed resource allocation (1996), 156-183.

FRHB11 A. Ford, C. Raiciu, M. Handley, S. Barre, and J. Iyengar. Architectural Guidelines for Multipath TCP Development. Internet-Draft draft-ietf-mptcp-architecture-05, Internet Engineering Task Force, Jan. 2011. Work in Progress.

GEv07 P. Garbacki, D. H. J. Epema, and M. van Steen, An amortized Tit-For-Tat protocol for exchanging bandwidth instead of content in P2P networks, Proceedings of SASO 2007 (Boston, MA), July 2007.

GH05 F. D. Garcia and J. H. Hoepman, Off-line Karma: A decentralized currency for peer-to-peer and grid applications, 3th Applied Cryptography and Network Security (ACNS 2005) (New York, NY, U.S.A.) (J. Ioannidis, A. Keromytis, and M. Yung, eds.), Lecture Notes in Computer Science, vol. 3531, Springer Verlag, June 7-10 2005, pp. 364-377.

GKL01 Z.W. Geem, J.-H. Kim and G.V. Loganathan, A new heuristic optimization algorithm: harmony search, *Simulation* 76 (2001) (2), pp. 60–68.

GLBM01 P. Golle, K. Leyton-Brown, and I. Mironov, Incentives for sharing in peer-to-peer networks, EC '01: Proceedings of the 3rd ACM conference on Electronic Commerce (New York, NY, USA), ACM Press, 2001, pp. 264-267.

GLL08 Y. Guo, C. Liang, and Y. Liu, “AQCS: Adaptive Queue-based Chunk Scheduling for P2P Live Streaming,” in Proceedings of IFIP Networking, 2008.

GR04 H. Gravelle and R. Rees. Microeco*nomics. Prent*ice Hall, 3rd edition, 2004.

Gro03 C. Grothoff, An Excess-Based Economic Model for Resource Allocation in Peer-to-Peer Networks, Wirtschaftsinformatik 3-2003 (2003).

GSS06 R. Gupta, V. Sekhri, and A. K. Somani, CompuP2P: An architecture for internet computing using peer-to-peer networks, IEEE TPDS 17 (2006), no. 11.

HGL10 H. Hu, Y. Guo, Y. Liu, Mesh-based peer-to-peer layered video streaming with taxation, in: Proceedings of the 20th International workshop on Network and Operating Systems Support for Digital Audio & Video (NOSSDAV 2010), 2010.

HT95 N. Hardy and E. D. Tribble, The digital silk road, Agoric Systems: Market Based Computation (1995).

I5.1 ENVISION Internal report I2.1, Initial Specification of Metadata Management, specification of Enriched Content Metadata and Profile Management Functions and Information Model, Juin 2011, FP7 ICT ENVISION project, http://www.envision-project.org

MAEG11 S. Medjiah, T. Ahmed, E. Mykoniati and D. Griffin , Scalable Video Streaming over P2PNetworks: A Matter of Harmony?, In IEEE proceeding of CAMAD’11

Wu10 Q. Wu, Problem Statement for HTTP Streaming, draft-wu-http-streaming-optimization-ps-03 (work in progress), October 2010.

Zo10 N. Zong, Survey and Gap Analysis for HTTP Streaming Standards and Implemntations, draft-zong-httpstreaming-gap-analysis-01, October 2010.

PM11 R. Pantos and W. May, HTTP Live Streaming, June 2010, draft-pantos-http-live-streaming-07, September 2011.

Izu01 R. Izumi, The WAT system: An exchange system based on mutual appreciation, 2001.

Kla08 M. Klafft, Peer to peer lending: Auctioning microcredits over the internet, Proceedings of the 2nd International Conference on Information Systems, Technology and Management, ICISTM 2008 (Dubai), March 2008.

LB99 K. Lai and M. Baker, Measuring bandwidth, INFOCOM, 1999, pp. 235-245.

Lie04 Lietaer, Complementary currencies in japan today: History, originality, relevance, International Journal of Community Currency Research 7 (2004).

LLSB08 D. Levin, K. LaCurts, N. Spring, and B. Bhattacharjee, BitTorrent is an auction: analyzing and improving BitTorrent's incentives, SIGCOMM Comput. Commun. Rev. 38 (2008), no. 4, 243-254.

MKKB01 R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan, Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications, Proc. SIGCOMM, 2001.

MLL08 P. Marciniak, N. Liogkas, A. Legout, and E. Kohler, “Small is not always Beautiful”. In Proceedings of IPTPS, 2008.

MM05 F. M. Menezes and P. K. Monteiro, An introduction to auction theory, 1st ed., Oxford University Press, USA, January 2005.

MPTCP https://datatracker.ietf.org/wg/mptcp/charter/

Ms09 Microsoft Corporation, "IIS Smooth Streaming Technical Overview", March 2009.

NLE10 A. T. Nguyen, B. Li, and F. Eliassen. Chameleon: Adaptive Peer-to-Peer Streaming with Network Coding. In IEEE INFOCOM’10, 2010.

OIPF10 OIPF, "HTTP Adaptive Streaming (Release 2)", September 2010.

PA05 V. Pai et al., “Chainsaw: Eliminating Trees from Overlay Multicast,” Proc. IEEE INFOCOM ’05, Feb. 2005.

PR11 C. Pluntke and M. Rio, TCP-friendly Rate Control for non-TCP Multipath Flows. Poster at 8th International Conference on emerging Networking EXperiments and Technologies (ACM CoNEXT), December 2011.

RD01 A. Rowstron and P. Druschel, Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems, IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), November 2001, pp. 329-350.

RHW11 C. Raiciu, M. Handley, and D. Wischik. Coupled Congestion Control for Multipath Transport Protocols. Internet-Draft draft-ietf-mptcp-congestion-01, Internet Engineering Task Force, Jan. 2011. Work in Progress.

RO03 R. Rejaie and A. Ortega, PALS: peer to peer adaptive layered streaming. In Proceedings of NOSSDAV’03, Monterey, CA, USA, Jun. 2003.

RR06 R. K. Rajendran and D. Rubenstein, Optimizing the Quality of Scalable Video Streams on P2P Networks, Computer Networks: The International Journal of Computer and Telecommunications Networking, vol. 50, no. 15, pp. 2641–2658, Oct. 2006.

Sai06 K. Saito, i-WAT: The Internet WAT system - an architecture for maintaining trust and facilitating peer-to-peer barter relationships, Ph.D. thesis, Graduate School of Media and Governance, Keio University, February 2006.

Sol96 L. D. Solomon, Rethinking our centralized monetary system: The case for a system of local currencies, Praeger Publishers, 1996.

SKT http://www.mobilebusinessbriefing.com/articles/skt-debuts-mixed-mode-network-technology/21224/

SR00 D. S and K. W. Ross, Optimal Streaming of layered video. In Proceedings of IEEE Infocom 2000, pp:737-746, April 2000.

TCX09 Hossain. T, Yi Cui, Yuan. Xue, On the Optimality of Layered Video Streaming Rate in a P2P Mesh Network. In Proc of ICCCN, 2009.

TPM04 K. Tamilman, V. Pai, and A. Mohr, SWIFT: A system with incentives for trading, Proceedings of Second Workshop of Economics in Peer-to-Peer Systems, 2004.

TR04 D. A. Turner and K. W. Ross, A lightweight currency paradigm for the P2P resource market, 7th International Conference on Electronic Commerce Research, June 2004.

Var73 H. R. Varian, Equity, envy and efficiency, Working papers 115, Massachusetts Institute of Technology (MIT), Department of Economics, August 1973.

VCS03 V. Vishnumurthy, S. Chandrakumar, and E. Gun Sirer, KARMA: A secure economic framework for P2P resource sharing, Proc. of P2PEcon, 2003.

Vic61 William Vickrey, Counterspeculation, auctions, and competitive sealed tenders, The Journal of Finance 16 (1961), no. 1, 8-37.

WHH+92 C. A. Waldspurger, T. Hogg, B. A. Huberman, Jeffrey O. Kephart, and W. Scott Stornetta, Spawn: A distributed computational economy., IEEE Trans. Software Eng. 18 (1992), no. 2, 103-117.

WL05 W. Wang and B. Li, Market-driven bandwidth allocation in selfish overlay networks, INFOCOM 2005, 24th Annual Joint Conference of the IEEE Computer and Communications Societies, IEEE, 2005, pp. 2578-2589.

WRG11 D. Wischik, C. Raiciu, A. Greenhalgh, and M. Handley. Design, implementation and evaluation of congestion control for multipath TCP. In Proceedings of NSDI, 2011.

XSG08 X. Xiao, Y. Shi, and Y. Gao, On Optimal Scheduling for Layered Video Streaming in Heterogeneous Peer-to-Peer Networks, in In Proc. of ACM international conference on Multimedia, 2008, pp. 785–788.

YGM03 B. Yang and H. Garcia-Molina, PPay: micropayments for peer-to-peer systems, Proc. of CCS '03, 2003.

ZLY05 X. Zhang, J. Liu, B. Li, and T.-S.P. Yum, Coolstreaming/Donet: A Data-Driven Overlay Network for Efficient Media Streaming, Proc. IEEE INFOCOM ’05, Mar. 2005.

ZWLZG09 L. Zhou, X. Wang, Y. Li, B. Zheng, and B. Geller, Optimal Scheduling for Multiple Description Video Streams in Wireless Multihop Networks, IEEE Communications Letters, vol. 13, no. 7, pp. 534–536, Jul 2009.

ZXZY09 M. Zhang, Y. Xiong, Q. Zhang, and S. Yang, Optimizing the throughput of data-driven peer-to-peer streaming, IEEE Transactions on Parallel and Distributed Systems, vol.20, no.1, 2009.