Enriched Network-aware Video Services over Internet Overlay Networks

www.envision-project.org

Deliverable D4.1

Initial Specification of Consolidated Overlay View, Data Management Infrastructure, Resource Optimisation and Content Distribution Functions

Public report, Version 2, 20 May 2011

Authors

UCL Eleni Mykoniati, Raul Landa, David Griffin, Miguel Rio

ALUD Nico Schwan, Klaus Satzke

LaBRI Ubaid Abbasi, Toufik Ahmed

TID Oriol Ribera Prats

LIVEU Noam Amram

Reviewers Nico Schwan, Toufik Ahmed, Bertrand Mathieu, Noam Amram

Abstract This document elaborates on the functionality that is required to perform network-aware content distribution with cross-layer optimisation between the network and the application layers using the capabilities of the CINA interface. A description of the related problems and the associated requirements, a thorough review of the state of the art and high-level specifications are specified for the supporting functions required to gather and consolidate information about the network and the preferences of the ISPs, to update and discover the application overlay and network resources in an efficient and scalable way, and finally the functions for the distribution of live and interactive video using the information and services offered by the ISPs.

Keywords Network-aware Content Distribution, Cross-layer Optimisation, Consolidated Overlay View, Distributed Resource Data Management, Live and Interactive Video

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

EXECUTIVE SUMMARY

This document is the first WP4 deliverable of the ENVISION project.

The project advocates the cross-layer optimisation between network and application overlay functions through the *Collaboration Interface between Network and Applications* (CINA). Drawing from the requirements of the future networked media applications and the particular use cases and business models considered in [D2.1], a high-level functional architecture has been defined, capturing the cross-layer optimisation functions and the interactions between them. Details for the specification of the CINA interface, the metadata exchanged between the network and the applications, and the services offered by the network are specified in [D3.1], while [D5.1] documents the techniques and the enhancements required to perform content adaptation in a scalable and efficient way, taking into account the information discovered through the CINA interface.

Departing from the use cases, the requirements and the architecture defined in [D2.1], the capabilities of the CINA interface captured in [D3.1] and the requirements of the content adaptation functions defined in [D5.1], this document focuses on the functionality and the research challenges associated with the content distribution optimisation functions and other supporting functions required to retrieve and consolidate information from the network and the application overlay, and to manage this information in an efficient way.

In particular, this document provides a description of the related problems and the requirements that accompany them in order to refine the scope of the work, a thorough review of the state of the art and high-level specifications for the following functionality required at the overlay layer:

- *consolidated overlay view* functions that retrieve information from the underlying ISPs through the CINA interface, complement it with information from the overlay passive and active monitoring and consolidate it to produce predictions about the end-to-end network performance between overlay nodes and a consistent view of the preferences of the underlying ISPs;
- *resource data management* functions that handle the information about the application layer resources, the overlay nodes, the users, the available content etc. in an efficient way, supporting frequent updates and high query rates, and being aware of the network properties of the registered resources;
- *content distribution* functions for live and interactive video applications that create and optimise the data distribution overlay paths, by mobilising and interconnecting the overlay resources in a way that produces the best performance for the users, and it is cost-efficient for the application and the underlying ISPs.

This document concludes with the high-level specifications of the above functionality and the definition of the interactions between the identified functional blocks in WP4 and WP5. The work in WP4 will continue with the detailed specification of the protocols and optimisation algorithms identified here, the compilation of an implementation plan for the functionality that needs to be developed and further evaluated in WP6, and a preliminary software release at M18 of the project (June 2011), while the specifications of the developed protocols and algorithms and the refined functional specifications will be documented in D4.2 at M24 (December 2011).

TABLE OF CONTENTS

1. INTRODUCTION

In WP4, the focus is on developing techniques at the overlay layer for optimising the content distribution with the collaboration of the ISPs. To this end, we have identified three high-level subsystems that need to be in place and that each presents its own research challenges:

- *consolidated overlay view*: includes the functionality that retrieves information from the underlying ISPs through the CINA interface, complements it with information from the overlay passive and active monitoring and consolidates it to produce predictions about the end-to-end network performance between overlay nodes and a consolidated view of the preferences of the underlying ISPs;
- *resource data management*: includes the functionality for handling the information about the application layer resources, the overlay nodes, the users, the available content etc. in an efficient way, supporting frequent updates and high query rates, and being aware of the network properties of the registered resources;
- *content distribution*: includes the functionality that is responsible for the creation and optimisation of the data distribution overlay paths, by mobilising and interconnecting the overlay resources in a way that produces the best performance for the users, and it is cost-efficient for the application and the underlying ISPs.

The optimisation of the content distribution, and in particular of the algorithms that build the overlay topology, depends greatly on the type of the content, the capabilities and the requirements of the particular application and the way the users interact between them and with the application to produce and consume the content. Therefore, a significant part of the work in this first period was dedicated in identifying the most prominent research challenges that would produce the results with the greatest impact and that can be dealt with within the resources and during the lifetime of the ENVISION project.

Departing from the use cases captured in [D2.1] and investigating additional applications like virtual world and augmented reality games for their particular content distribution requirements, we have concluded on two applications for which we will investigate specialised content distribution optimisation techniques: a large-scale live video streaming application and an interactive video application, inspired by the bicycle race and the web conference use cases respectively.

As a result of this decision, in WP4 we are not considering the implications from content caching techniques, as these mainly apply to less interactive applications like Video on Demand or file sharing. Although one could think of live and interactive video applications with additional features for recording and accessing non live content, these would in effect require Video on Demand techniques, which are considered out of scope in WP4.

Figure 1 shows the main subsystems that are identified in the context of the work in WP4, including WP4 and WP5 functions at the overlay layer, the interactions between them and with WP3 functions. Please note that only the functionality that is in the scope for WP4 is included, while other functionality studied in WP3 and WP5 independently of the WP4 functions is not considered.

The consolidated overlay view functions interact with the *network data management* function in WP3 through the CINA interface to retrieve information about the network. They also interact with the content distribution functions to extract passive monitoring information. This information is then processed and fed to the resource data management, which uses it to filter and rank the resources and return for every query only the ones that yield the best network performance and are most favoured by the ISPs.

Queries for resources are issued mostly by the content distribution functions which need to discover which overlay nodes have a particular content object and what are their network performance properties, but also from the content adaptation and consolidated overlay view functions which also

rely on mobilising resources dynamically, as the application overlay and the user demand grows to act as content adaptation and resource data management nodes respectively. Finally, the resource data management function provides information about the application back to the ISP and the *network management* function through the CINA interface.

Figure 1 – High-level Overlay Functions

As users become active, they register their terminals as overlay nodes with the resource data management function and notify the consolidated overlay view so that it begins retrieving information about them. When they decide to consume a particular content object, they would trigger the corresponding procedures with the content distribution function, which would establish and maintain the necessary connections between the user nodes and other user or provider nodes that already participate in the distribution of this content object.

When the participant resources are not sufficient to provide a satisfactory performance as defined by the content distribution functions monitoring the quality of the content, the reliability of receiving it with low delay and other applicable application performance metrics, the content distribution functions would invoke network services like multicast through the CINA interface, by first interacting with the *network management* function to activate and setup the network service, and then by interacting with the *network services control* function for as long as the service is active to add or remove overlay nodes to the service, to exchange any required control messages and other information to ensure the optimum operation of the service.

Finally, when the user decides to produce and make available a new content object, then the appropriate WP5 *content generation* and *content adaptation* functionality is triggered, including appropriate encoding, error protection and transcoding functions. The adapted content is then fed to the content distribution subsystem, which is responsible to distribute it to all the content consumers in an optimum way. The content adaptation functions also determine which level of quality is best

for each content consumer, based on the available content formats, the capabilities of the user terminals and the network conditions. This information is passed to the content distribution functions that modify the overlay topology accordingly. When there are not enough overlay resources to sustain the bitrate required for a particular format, then the content distribution functions would notify the content adaptation to consider alternatives.

The identified functions map to the more generic functional blocks defined in the ENVISION architecture in [D2.1]. The content adaptation and the content distribution are shown to span across the *overlay management* and the *services control* functions of the ENVISION architecture. However, this is only because of the level of detail of the functional decomposition and the fact there is no distinction at this point between the management and the control processes for these two functions.

The rest of the document defines the problem statement and elaborates on the requirements, the state-of-the-art and the high-level specifications for each of the WP4 subsystems introduced above. In particular, section 2 presents the consolidated overlay view subsystem, section 3 the distributed data management infrastructure, section 4 the content distribution subsystem for live video, and section 5 the content distribution subsystem for interactive video. Finally, the document concludes with section 6.

2. CONSOLIDATED OVERLAY VIEW FUNCTIONS

2.1 Problem Statement

2.1.1 Description

One of the aims of the ENVISION project is to explore the ways in which collaboration between ISPs and application overlays can enable the deployment of advanced multimedia applications over the current Internet. By taking the current approaches to overlay-ISP collaboration and measurement consolidation as a starting point, ENVISION will develop techniques for incorporating high-quality, ISP-provided measurement information to end-to-end overlay measurements, in order to increase their quality and usefulness. By simultaneously optimising measurements for increased precision and lower network impact, the Consolidated Overlay View functions (COV) will provide a unified view of the current state of the network that can help overlay applications make better use of the network infrastructure at their disposal.

The first variable to explore in the context of the COV is the communication mechanisms between application overlays and their underlying ISPs. Recently, an explicit interface between the overlay and the network has been proposed and is currently being standardised by the IETF ALTO working group [PMG09]. Instead of performing measurements at the overlay layer in order to deduce the topology and performance of the underlying network, this interface provides access to information available directly at the network in the form of ratings on overlay links between peers. The ISP rates these links with its own traffic engineering criteria, that may include load balancing metrics such as available bandwidth, cross-domain traffic, and cost to the user. The ALTO interface is defined in a content-agnostic manner. Thus, it is impossible for the application and the rating algorithm to exchange information regarding particular content requirements.

The scope of the overlay-ISP interaction in ALTO is limited to the viewpoint of a single ISP and the peers located on its domain. Further, this interaction is only explored in the context of improving the selection of overlay nodes for a new node compared to random and only the first time the node joins the overlay, as opposed to ENVISION which is concerned with continuously optimising the overlay connections. Given that ENVISION applications are global in coverage and require end-to-end traffic optimisation involving different networks, it is necessary to collect information from many underlying networks. There are several problems associated with collecting and using this information: data from one network may conflict with that provided by another; the quantity and quality of the information may differ from ISP to ISP and some may not offer any information at all; the preferences expressed by different ISPs may be using different criteria and may produce rankings of nodes that are not directly comparable. The harmonisation of the information gleaned from the ISPs, the aggregation of the information collected from different ISPs, its auditing and augmentation with additional data collected by the overlay and its subsequent use for the global optimisation of the application is one of the major research challenges of this project.

Regarding the use of network-layer information for the overlay optimisation, most of the research literature has focused on locality-awareness using network latency measurements (see section 4.2.4 for details on the related state of the art). The reason for this is that many large-scale Internet applications can benefit from round-trip time predictions without resorting to explicit measurements, which can be unattractive if the cost of measurement outweighs the benefits of exploiting proximity information. One system that achieved this in a scalable fashion is Vivaldi [DCKM04], a lightweight algorithm that performs an approximate isometric embedding of the Internet delay space in a pseudo-Euclidean space. By assigning synthetic coordinates to hosts such that the distance between the coordinates of two hosts predicts the communication latency between the hosts, Vivaldi allows the use of locality information in a simple and scalable way. Many other synthetic coordinate systems have been proposed after Vivaldi [BP08, LGS07, ST04], but most focus on network latency rather than on general network measurements.

Since interaction with the underlying network can provide essential information to a number of application functions that need to be optimised in order to support the future networked media environments, and since Internet-wide applications might involve a number of ISPs, ENVISION will benefit from algorithms that take overlay and ISP-provided measurements to provide a consolidated view of the network for the entire overlay.

2.1.2 Requirements

The Consolidated Overlay View (COV) functions will provide a set of algorithms that take both overlay and ISP-provided measurements and integrate them, so that they can be used by overlay applications. This work assumes that the COV builds on top of the CINA interface capabilities, captured in the ENVISION deliverable D3.1 [D3.1].

The following sections identify the requirements that apply to each of the individual steps of producing and maintaining the COV, namely the information retrieval from the ISPs, the overlay monitoring, and the auditing and consolidation of these two sources of information.

2.1.2.1 ISP Information Retrieval

- The COV functions should use the CINA interface to discover the information provided by the ISPs, in particular:
  - Which ISPs support ENVISION, how to contact them and which of these ISPs can provide information for each of the overlay nodes
  - Which network services are supported by which ISP and under which conditions
  - Which metrics / information about the network are provided by which ISP
  - Which entities and granularity is a particular metric provided at, e.g. loss per link or per path, or for arbitrary groupings of links/paths
  - Which time period and statistical value a particular metric is provided at, e.g. aggregate over a number of minutes/days, min/max value or a particular percentile or EWMA with a given weight, etc.
  - What is the accuracy associated with each measurement
- The COV functions should maintain an up-to-date view of the information provided by the ISPs for the active set of overlay nodes, e.g. by subscribing to notifications, polling the interface periodically or only when the information is to be used
- The COV functions will minimise the volume and frequency of the interactions with the ISP, complying with any restrictions imposed by the ISP, and minimising any penalty on the accuracy of the information

2.1.2.2 Overlay Monitoring

- End hosts will contact each other directly to perform measurements or exchange measurement information
- End hosts may collect passive monitoring information from their overlay data connections and/or establish dedicated active monitoring connections
- End hosts will provide raw measurements and/or statistical measurement descriptions
- The COV functions will re-use existing information, minimising the overhead required to gather it
- The COV functions will produce estimates for the accuracy of gathered measurements using information provided by both end hosts and ISPs

- The COV functions will optimise the trade-off between the overhead of monitoring jobs and the accuracy of the gathered information

2.1.2.3 Information Consolidation

- For all metrics of interest that represent objective information and not the preference of the ISP, the COV functions will perform dedicated overlay measurements to allow comparison of the exact values, or of the relative values between different pairs of nodes, or more complex hypothesis tests
- The COV functions will assign weights to the information received by the ISPs, based on an associated level of confidence derived from the analysis of long-term data summaries
- The COV functions will produce statistically consolidated data objects that allow the estimation of relevant network layer measures based on the raw or statistical measurements and information gathered both from application overlay monitoring and ENVISION-enabled ISPs
- The COV functions will provide estimates for the accuracy of the statistically consolidated data objects
- When consolidating information from multiple ISPs, the COV functions will be able to cope with:
  - Information provided through relative metrics rather than absolute metrics
  - Missing input from ISPs that do not support the CINA interface
  - Conflicting input received by different ISPs, i.e. in terms of preferences, or network performance metrics that cannot be verified by auditing
  - Input provided in different granularity, statistical aggregation, accuracy, etc.
- The statistical consolidation models used by the COV functions will converge to a stable, usable state. Furthermore, the system will minimize the impact that transient events have on the underlying statistical model
- The COV functions will map the requirements of the overlay application to requests for information from the ISPs and the overlay nodes in terms of geographical scope, metrics, frequency, time aggregation, etc.
- The COV functions will communicate the updates or refinements of the consolidated information to the data management infrastructure

2.2 State-of-the-art and Innovation

This section presents the state-of-the-art for overlay monitoring and information consolidation techniques, and discusses briefly the possible avenues of research to produce innovative solutions, in particular for the information consolidation techniques in the context of network performance information and preferences retrieved by the ISPs. Techniques for retrieving information from the ISP are not discussed in this section, as there is virtually no prior related work and they are more relevant to the specification of the CINA interface, documented in [D3.1].

2.2.1 Overlay Monitoring

The state of the art provided in this section elaborates on some monitoring approaches and related techniques. In the ENVISION project we are not aiming to generate novel monitoring techniques rather to use existing techniques and exchange information over the CINA interface between the network providers and the applications for mutual benefits.

2.2.1.1 Passive Monitoring

Monitoring approaches are distinguished into passive and active monitoring methods. Passive monitoring is defined as a monitoring technique that does not generate new packets for the purpose of monitoring, thus the application data packets are used to extract monitoring information from source to destination and vice versa. This type of monitoring is most suitable to active data sessions. The portion of monitoring information in proportion to other application traffic is usually minor (i.e. less than 1%), which is also the main advantages of passive monitoring. The disadvantage is that it is not suitable between overlay nodes where connections are not established yet, thus active monitoring is required to complement it.

2.2.1.2 Active Monitoring

Active monitoring techniques generate dedicated traffic for the purpose of monitoring. Such monitoring could be used in order to check connectivity, count number of hops between peers, collect statistics and so on. There are two main approaches for active monitoring. One relies on collaboration of the routers along the path such as traceroute. This assumption is more and more not valid due to routers that do not support it, named anonymous routers or due to Layer 2 switching and new technologies like MPLS that does not provide Layer 3 routing information. The other one is network tomography which is based on end-to-end measurements between end hosts. The second one is more relevant to WP4 and peer-to-peer overlay monitoring in general.

2.2.1.2.1 Tools to Measure Network Performance

There are many network tools to measure the connectivity and performance of the network. The following are some examples which may be used in ENVISION to aid in the construction of high-performance distribution overlays:

- Ping: the classic network tool for verifying IP connectivity as well as estimating the round trip time interval.
- Traceroute: sends a series of UDP packets with an increasing TTL, and by watching the ICMP time expired replies it can discover the hops to a host (assuming the hops actually decrement the TTL).
- TTCP: Test TCP is a freeware tool to test the throughput between two network endpoints. It needs to be run on both the source and the destination, and there is a Java version of TTCP[1] that runs on many different operating systems. Since the measurement requires the creation of congestion losses on the bottleneck link, this tool floods the network with traffic.
- Pathchar: this tool identifies network bottlenecks. It operates like traceroute, but rather than printing response time to each hop it prints bandwidth between each pair of hops. This tool works by sending "shaped" traffic over a long interval and carefully measuring the response times. It doesn't flood the network like TTCP does. Since it relies on the detection of queuing correlations between consecutive packets, it does not require to load the network as TTCP does.

2.2.1.2.2 Network Tomography Techniques

This paper [HLW07] proposes a probing noise resilient available bandwidth estimation scheme, called JitterPath, which is based on one-way delay jitter and queuing delay propagation for multimedia applications. The presented JitterPath solution deals with queuing region classification, queuing delay propagation, and initial queuing delay determination. The captured traffic ratio and a binary search-based probing rate adjustment mechanism have been proposed to iteratively approximate the available bandwidth with an error that is within the preset estimation resolution.

[1] http://www.ccci.com/tools/ttcp

This paper [NXTY10] proposes a general framework for designing topology inference algorithms based on additive metrics. The framework can flexibly fuse information from multiple measurements to achieve better estimation accuracy. A proposed novel sequential topology inference algorithm, significantly reduces the probing overhead under unicast probing. In addition, it can efficiently handle dynamic node joining and leaving and thus is particularly desirable for applications and networks where node dynamics are prevalent.

This paper [GSG02] present a tool (referred as King) that accurately and quickly estimates the latency between arbitrary end hosts by using recursive DNS queries. Compared to previous approaches, King has several advantages. King does not require the deployment of additional infrastructure, and does not require end hosts to agree upon a set of reference points. The main idea behind it is that the delay between the DNS servers the end hosts are associated with, is of the same order to the delay between the hosts themselves.

2.2.2 Information Consolidation

The existing techniques and the research challenges associated with the consolidation of information vary significantly, depending on the type of information that needs to be consolidated. The following sections address three distinct technical topics that are relevant to this work: consolidation of preferences, and consolidation and auditing techniques for measurements.

2.2.2.1 Consolidation of Preferences

In addition to measurements, ISPs may expose to application overlays preferences regarding protocol operations. An example of this is ALTO, where the ISPs provide a preference relation over peers that capture their preferences with respect to overlay topology formation. ALTO provides increased performance when there is alignment of incentives between the overlay and its underlying ISP network. In a multi-ISP setting, however, the incentives of the multiple ISPs might not align in the same way with those of the overlay. In this case, it may be beneficial to the overlay to incorporate the rankings provided by of all the relevant ISPs into a single ranking that integrates the information provided by all relevant ISPs and that harmonises the interaction between them. In this case, the objective of the COV is to integrate various resource rankings from different ENVISION-enabled ISPs into a single, composite ranking.

The harmonisation of preferences exposed by various ISPs is a novel area of study that has not been explored in the literature. However, the general problem of aggregating preferences has been analysed in Economics. The field has been shaped by Arrow's impossibility theorem [Arr70], a fundamental result that states that if three or more outcomes are being considered, there is no function that aggregates rankings between outcomes that simultaneously satisfies some basic consistency properties and which is not a dictatorship (a system in which one of the preference rankings asserts itself over all the rest). This result can be adapted to social choice scenarios, where a given outcome or set of outcomes are selected according to the aggregated social choice functions [Tay05]. These impossibility results seem to suggest that the design of incentive-compatible social choice functions is too limited to yield useful results. However, by considering tightened assumptions or introducing money (or other ways in which to change preferences) it is possible to achieve useful results for many practical applications. Thus, by considering Mechanism Design [NRTV07, HR06] and its application to welfare and Social Choice [Sen77] it is possible to find preference aggregations that form tradeoffs between rankings.

The usual way in which this problem has been addressed by the networking community is through policy conflict detection and resolution [CFP+09, ASH04, LS99]. In this case, the policies themselves are represented using a logical framework that allows a deductive engine to find policy conflicts. When these are identified, they are resolved using application-specific mechanisms that represent the allowable tradeoffs between the policies. Usually, these resolution mechanisms select one possible ranking on the basis of priority. It is also possible, however, to produce policies on the fly

that represent the tradeoffs between the conflicting policies. Unfortunately, the focus of these works has been on the computational logic challenges of the problem, rather that its strategic implications for rational agents. This assumption is not a problem when considering a single administrative domain, but becomes troublesome in the case of multiple ISPs which is the focus for ENVISION.

The problem of aggregating policies can be understood directly from the point of view of game theory. To achieve this, the preferences of each ISP must be presented as strategies. This means that each ISP can inform the overlay what actions it will take towards its traffic depending on how closely the actions of the overlay match the preferences of the ISP. In this case, a game-theoretical formulation is natural: Each ISP is a player, their preferences are interpreted as strategies in a game, and the equilibria of this game are the possible tradeoffs between the different strategies available to each ISP. If these can be enumerated, the overlay can then choose that provides the greatest utility.

This framework, however, assumes that different ISPs have visibility of each other's strategies. This may be impossible to achieve, thus making the game theoretical formulation unnecessarily complex. In this case, the problem of harmonising preferences can be formulated as an optimisation problem with the function to optimise representing the utility of the overlay, and the rankings of each ISP affecting the result by changing this utility or providing restriction for the optimisation process.

The ENVISION project will consider the tradeoffs required by these techniques and select the one that better implements the requirements that have been presented in section 2.1.2.

2.2.2.2 Consolidation of Measurements

The CINA interface will run over Internet-scale infrastructures where not all underlay ISPs will be ENVISION compliant. In addition, even if all underlying ISPs were ENVISION compliant, they would still be under different administration domains. Thus, the ISP level measurements available to each one of the underlying ISPs may not be available to all the ISPs concerned. In this case, it will be the responsibility of the overlay to consolidate all these measurements into a single, cohesive set of end-to end measurements that make use of the measurements provided by all involved ISPs. Of course, in order for the system to be useful it will need to support various measurement types, be applicable to both envision-enabled ISPs and those which do not support the interface, an it must be scalable to Internet-scale distributed applications.

The most direct way of integrating measurements is to have the overlays perform end-to-end measurements, and then enrich them using the local information provided by each ISP.

Since a simple database approach is not only unfeasible but also undesirable, explicit summarisation techniques are needed. One family of techniques, which has proven effective in the representation of network delays and other peer-related measures, relies on an error-minimisation framework based on metric space embedding [BP08, DCKM04, ST04]. In essence, the idea is to map measurements defined over a discrete metric space (the Internet, when understood as a graph) to distances defined over a continuous metric space, usually based on a Euclidean geometry. In this way, a subset of measurements can be used to map each node to a point in the space so that their geodesic distance is a good estimation of a given network measurement between them.

The COV will make use of these techniques, enriching them by considering measurement information provided by the ISPs through the CINA interface. A simple way in which ISP-provided preferences might be assimilated into these embedding-based frameworks is to insert weights in the end to end calculation that represent these preferences. This would give ISPs a way to guide the distributed computation by artificially modifying the distance between nodes. This, however, defeats the purpose of the COV as a data source. Another way in which this could be done would be to use SNMP link-level measurements to estimate end-to-end measurements. This has been conventionally called Inverse Network Tomography [Var96], and has been used for traffic matrix estimation [GJT04]. Using this technique, an ISP could provide estimations that may be used between intra-domain

overlay participants. Of course, this technique cannot be used end-to-end for overlays that span many ISPs, not all of them ENVISION compliant. Another possible avenue of research is the use of probabilistic embedding [BV09]. The ISP could generate highly accurate embeddings for its own nodes, and these local coordinate systems could then be composed with a egress-to-ingress embedding, hopefully reducing embedding error.

The ENVISION project will consider strengths and weaknesses of each one of these methods for information consolidation and select that one that better implements the requirements that have been presented in section 2.1.2.

2.2.2.3 Auditing of Measurement Information

As is usually the case for systems in which there might be misalignment of incentives, there is the possibility for the interactions through the CINA interface to be used strategically. This means that, rather than exposing truthful measurements, the ISP or the overlay might expose that information that will yield the greatest benefit for them. Thus, audit mechanisms might be needed to verify the degree to which overlay-collected data is compatible with ENVISION-provided data, and vice-versa.

One way this could be accomplished is by using inverse and direct network tomography [CCL+04] along with topology estimation [CCN+02, SH04]. Using these techniques, the ISP can use estimate intra-domain measurements and compare them with those provided by the overlay, and the overlay can estimate link-level measurements and compare them with those provided by the ISP. These methods, however, have important drawbacks. In addition to being unsuitable for inter-domain end-to-end use, they may require knowledge of either the network topology or link level measurements. Making these measurements available may be undesirable for many ISPs.

Auditing can be performed in two ways. In one of them, historical statistics are used to assess the probability that the data provided fits the probability distribution of the audit measurements. To achieve this quickly, we could draw upon contributions in the areas of traffic anomaly detection [LCD04] and classification [CDGS07]. Another way in which this can be accomplished is by direct comparison of simultaneous interactions, to verify if the rankings provided by the interface match those that the peer actually experiences [MZPP08, DMG+10].

Although the ENVISION project is concerned primarily with the cooperation between application layer overlays and their underlying ISPs, it may be beneficial to consider adversarial scenarios in order to make the protocols and interfaces robust against manipulation. In the event that such an analysis is required, the scope and benefits of each one of the techniques presented above will be assessed for inclusion within the analytical framework of the ENVISION project.

2.3 High-level Specifications

Figure 2 depicts the consolidated overlay view functions and the control and metadata interactions between them and with the other application and network layer functions. These interactions are summarised below and the following sections provide a short description for each function.

Control Interactions:

- The network information consolidation function controls the frequency and granularity of the active monitoring and ISP information retrieval functions to achieve the required level of accuracy with a minimum overhead.

Metadata Interactions:

- The ISP discovery and peer mapping function triggered by the user profile management, it establishes the information about the user overlay node and, when available and not already active, communication with the CINA interface of the user's ISP.

- The ISP discovery and peer mapping function registers the discovered ISP capabilities and the mapping of overlay nodes to the ISP with the resource data management functions.
- The ISP discovery and peer mapping function informs the network information consolidation about the existence of a new node and possibly a new ISP for which information needs to be gathered.
- The ISP information retrieval receives information about the network, the ISP capabilities and the ISP preferences through the CINA interface.
- The passive monitoring interacts with the content distribution functions to gather passive measurements.
- The network information consolidation receives information from the ISP information retrieval, passive and active monitoring functions.
- The ISP information retrieval and active monitoring functions receive information from the resource data management functions about the nodes that can be used to poll the ISP and establish active monitoring jobs respectively.

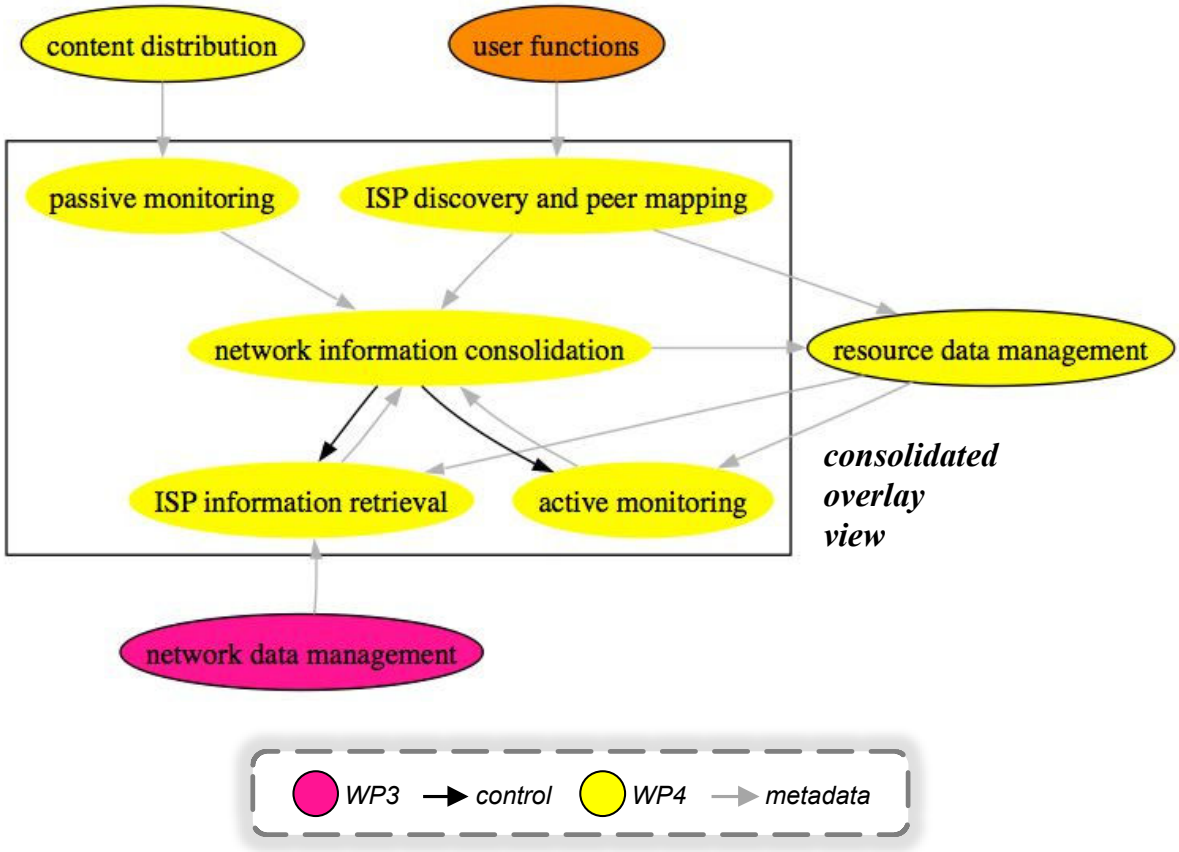


Figure 2 – Consolidated Overlay View Functions

2.3.1 ISP Information Retrieval Function

The ISP information retrieval function uses the CINA interface to retrieve information regarding the ISP capabilities and the network services it supports, as well as more dynamic information about the performance of the network and the preferences of the ISP. The ISP information retrieval function will implement the CINA protocol stack, discussed in [D3.1].

The ISP information retrieval function is responsible for polling or maintaining subscriptions with the ISPs for receiving up-to-date information, covering all the locations where there are active overlay

nodes, at a frequency and granularity that is sufficient to accurately determine the network performance between any two nodes. While the ISP information retrieval function is responsible for retrieving the information, the network information consolidation function is responsible for determining what is the appropriate frequency and granularity of information that is required to derive accurate network performance statistics and predictions. Detailed requirements for the ISP information retrieval can be found in section 2.1.2.1.

The information that needs to be retrieved from the underlying ISPs grows as the application overlay grows in size and spreads across many different ISPs and locations with diverse network characteristics. A centralised approach to implement this function is therefore considered inappropriate, as this would impact the scalability of the application. At the other extreme, all the nodes participating at the overlay could retrieve information from their ISPs. This approach, however, would impose an unnecessarily high load to the CINA servers at the ISPs. Alternatively, a number of stable provider- or user-provisioned overlay nodes could be selected that can share the load and the results between them in a coordinated way to retrieve enough information while eliminating any possible overlaps. The specification of the associated protocols is work in progress.

2.3.2 Active and Passive Monitoring Functions

The passive monitoring function gathers measurements of the end-to-end network performance experienced by the data flows between any two nodes of the overlay, summarises these measurements to statistics at each node and possibly per groups of nodes at the same location and provides these statistics to the network information consolidation function. Although some metrics can be extracted with very little overhead, e.g. throughput, there are others that may require dedicated fields to be added to the data packets increasing the data protocol overhead, e.g. timestamps to measure delay. The design of the passive monitoring function depends on the application, and will be further investigated in the context of the live and interactive video distribution use cases considered in ENVISION.

The active monitoring function is responsible to initiate and maintain a sufficient number of active monitoring jobs between any two nodes of the overlay that are not necessarily exchanging data between them. The acquired end-to-end network performance measurements are processed and fed to the network information consolidation function, to complement the information received by passive monitoring and the underlying ISPs. Similarly to the ISP information retrieval, the active monitoring is responsible to produce the required information, but it is network information consolidation function that determines what information is needed, how frequently and from how many measurement points per location. Finally, the active monitoring function selects the overlay nodes where to activate active monitoring from a pool of nodes that support active monitoring discovered through the resource data management functions. The nodes are selected based on their capabilities to perform active monitoring without overloading them, and such that they collectively provide enough information about the network performance at a particular location to other remote network locations.

Detailed requirements regarding the passive and active monitoring functions can be found in section 2.1.2.2.

2.3.3 Network Information Consolidation Function

The network information consolidation function is responsible for consolidating network performance information and information regarding the preferences and policies of the ISPs where the overlay nodes reside.

The network information consolidation functions receives the following information:

- end-to-end network performance measurements between pairs of overlay nodes from the active and passive overlay monitoring functions; these may include round-trip network delay, loss, throughput etc.
- network performance statistics and routing information retrieved from the ISPs that support the CINA interface; these may include load at the access network or at the path from the egress router of a local end host to the egress router for a particular remote destination end host, intra-domain hop count or AS hop count to a particular destination, etc.
- preferences and/or policies from the ISPs that support the CINA interface; these may be expressed by ranking destination nodes, or assigning weights, etc.[2]

The network information consolidation function uses this information in order to:

- assess the accuracy of the collected information by comparing information received from different sources, and in particular overlay monitoring compared with information retrieved by the ISPs
- produce predictions about the end-to-end network performance between any two pairs of overlay nodes even when no statistics are gathered between them
- produce a normalised view of the preferences of different ISPs, by resolving the incompatibilities or conflicts and consolidating the different rankings and weights set by individual ISPs into a common set of weights based on the optimisation criteria of the application.

The consolidated network performance and ISP preferences information is used by the resource data management function in order to perform the filtering and ranking of the resources that give the best network performance from a particular location. The network information consolidation function maintains an up-to-date view of the active overlay nodes by receiving updates from the ISP discovery and peer mapping function. When there is not enough information to produce accurate predictions, or when nodes are added to new locations for which there is no previous information, then the network information consolidation function instructs the ISP information retrieval and the active monitoring functions to increase accordingly the information they are gathering.

Detailed requirements about the network information consolidation function and a discussion on related techniques that will be further investigated in the project can be found in sections 2.1.2.3 and 0 respectively.

2.3.4 ISP Discovery and Node Mapping Function

The ISP discovery and node mapping function is responsible for ensuring that, as users arrive and leave the application and the corresponding nodes are added and removed from the overlay, the application is aware of all the ISPs where active overlay nodes reside and can communicate with them if they support the CINA interface to retrieve information about these nodes. When there is no information already available in the system about a particular ISP, then this function is responsible for discovering the CINA contact point. A detailed description of the protocol considerations for the ISP discovery can be found in [D3.1].

[2] Details for the network performance metadata and the preferences of the ISP can also be found in [D3.1].

3. DISTRIBUTED DATA MANAGEMENT INFRASTRUCTURE FUNCTIONS

3.1 Problem Statement

3.1.1 Description

An application with logic and content distributed over a potentially massive number of participants needs an appropriate infrastructure to communicate highly dynamic control data efficiently, reliably and in a scalable way. BitTorrent is known for its scalability, but when the peer tracker is centralised, a dependency is created with this centralised component that may impact of the availability of the BitTorrent application [NRZ+07, PGES05]. Alternative infrastructures have been proposed, where storage and access of data is done in a distributed manner over a set of nodes organised in an overlay network.

Due to the dynamic nature of the content and the participating nodes, users seldom have the exact or complete knowledge of the advertised information in the system. Instead, queries are often based on partial knowledge about a target advertisement. This mandates an efficient search mechanism that is capable of resolving queries based on partial or incomplete information about the target advertisements, and can handle the dynamism in nodes and content's availability. Typical properties that such systems need to achieve include support for high volume of data registration / update / retrieval requests, high performance in terms of request response time and result accuracy, features like publish-subscribe, locating the closest member of a group, etc. The challenge is even greater for highly distributed overlay systems that employ not reliable participant resources, as they need to be able to cope with the often very dynamic arrival and departure rates of the participant nodes over which the data management infrastructure is built.

It is important to clarify that the problem of persistent storage and efficient retrieval of content data over a highly distributed and flexible infrastructure is not considered here. Rather, ENVISION is concerned with the management of application metadata and not the content itself, as this is a much more general problem underlying all highly distributed applications, whether they are about retrieving static content from a scalable pool of storage resources or about the efficient distribution of dynamic content. Application metadata include but are not limited to state of the application functions, information about the application resources and their properties, information about the underlying network, as well as content metadata used to discover the content items that best match the interest of the application users.

The problem of matching user interest to content items based on the user preferences and the content metadata, varies significantly with the type of content. In some applications the association of content sources to content consumers is deterministic and subject to the application logic and the user interactions. Such applications are for example virtual world applications, where the position of the user in the virtual space, the visibility settings and the layout of the virtual area determine the static and dynamic content that the user needs to receive, e.g. landscape objects or the position and state of mutable objects and of avatars of other players.

Another type of application is one where the content consumers explicitly express their interest to content by setting their preferences and the application is responsible for identifying the best available content at any point in time. An example of such an application is the bicycle race use case captured in [D2.1], where the users may express their interest to follow a particular athlete, or to watch the race from a particular viewpoint, and the application is responsible for the dynamic matching of metadata extracted from the face recognition processing modules or from the camera GPS tracking with the preferences of the user, for as long as the content consumer remains active.

Finally, content search is an major component of some applications like file sharing, where each file is associated with a set of keywords describing the content of the file and metadata capturing the properties of the content, e.g. the owner, the format, the publication date, etc. Such applications typically operate on a per-request basis without constantly evaluating previously submitted search requests, and the content for which they maintain metadata is not necessarily distributed by the applications themselves, as in the case of web searches for example.

The data management techniques and the research challenges that are relevant to the discovery of application resources and to the content search for the application types identified above, vary greatly. To maximise the impact of the project results, ENVISION addresses the most general problem which is the resource data management. In the distributed applications considered in ENIVISION, the discovery of resources is subject to their application and network properties, and in particular to network proximity or other quantitative metrics that can be used to filter and rank the resources matching a particular resource discovery query. As such, although the focus of the work in ENVISION is set to resource discovery, the developed techniques could equally apply to content discovery that is based on numeric values, as for example in the case of coordinate-based content discovery for the virtual world applications or applications broadcasting video annotated with GPS coordinates.

The following section identifies the requirements related to resource data management.

3.1.2 Requirements

3.1.2.1 Functional Requirements

- The resource data management system should allow for the dynamic addition, update and removal of application resources, including:
  - participant nodes, network and third-party provider nodes
  - network and third-party provider services and capabilities
  - network performance meta-data associated with the above resources
  - resources that are participating to a particular content item distribution performing a particular function, etc.
- The resource data management system should resolve queries for the discovery of application resources, satisfying a combination of criteria and/or restrictions, including for example participation to a particular content item distribution at a particular ISP, or ranked by proximity to a certain location, summing up to a certain value etc.
- The resource data management system may support queries for aggregate information regarding the application resources, e.g. return the number or the sum of resources at a particular ISP.
- The resource data management system will optimise on all or a subset of the following criteria:
  - Search completeness: Search completeness is measured as the percentage of registered resources matching the query that were discovered by the data management system. Required level of search completeness varies from application to application and type of resources. A search mechanism should guarantee the discovery of rare objects. In the case of queries that result to a large number of matches, exhaustive search would be unnecessary and a predefined number of matches would suffice for most cases, allowing for saves in the consumption of bandwidth and processing resources.
  - Accuracy: Accuracy can be defined as the percentage of the registered resources matching the query criteria better than any other resources that were discovered by the data management system. Because the search mechanism may be distributed and not all local

indexes up to date, or because the system is aggregating results during the indexing process, accuracy in discovering the current best results in a particular query might suffer. Similarly to search completeness, accuracy requirements may be different depending on the application. Discovering the closest in terms of network delay resources may tolerate some inaccuracies as the performance penalty is small for differences of milliseconds in RTT, while discovering the least expensive available relaying nodes might have greater accuracy requirements as this has a direct impact on the application cost.

  - Query Response Time: The resource data management system should minimise the time required to respond to a query issued by the users. It is expected that the query response time will be higher for queries that cannot be resolved locally but require exhaustive search across all the distributed indexes.
  - Efficiency: The resource data management system should be able to store, update and retrieve index information without consuming unnecessarily large storage and bandwidth resources.

- The resource data management system should take into account information provided by the ISP and refined by the overlay monitoring functions to improve the efficiency of the query resolution protocols and the accuracy of matching resources to queries with network performance criteria. If this information changes dynamically the resource data management system should adapt as timely as possible without undermining the stability of the system.
- The resource data management system should distribute the load fairly across the overlay nodes implementing the indexing and query resolution protocols according to their storage, processing power and bandwidth capabilities.

3.1.2.2 Performance Requirements

- The resource data management system should be designed to support up to millions of resources to be registered with the system spread across any number of arbitrary locations, participating in up to thousands of different roles and content distribution overlays.
- The resources data management system should be designed to return complete and accurate results for highly and sparsely populated resources, ranging from a single match registered with the system to hundreds of thousands of matches.
- The resource data management system should be designed to support up to hundreds of thousands of users and/or instances of functions querying the system at an average frequency of once every few seconds.
- The resource data management system should be designed to accommodate without significant disruptions high levels of churn in the participation of the nodes to the overlay and select the most stable nodes to participate in the resource data management system.

3.2 State-of-the-art and Innovation

In the following section we are presenting a review of the state-of-the-art on distributed data management systems for the discovery of resources with application and network performance criteria. Additionally, section 3.2.2 presents the state-of-the-art on distributed keyword-based search that has been undertaken in ENVISION in order to assess the complexity of the related issues and decide the feasibility of addressing them within the limits of the project resources.

3.2.1 Resource Search

The discovery of resources in a distributed overlay application involves two basic query operations:

- the exact match of an identifier to define the desired type of resources (relaying nodes, transcoding nodes, ISP CINA nodes etc.) and the particular content distribution overlay they are participating in (e.g. relaying nodes for a particular live video feed or telepresence conference session) and
- the filtering and/or ranking of the matching resources based on application and network performance criteria, e.g. the relaying nodes of a particular video feed that have the smallest playout delay from the source, or the smallest network delay to the querying node, etc.

While the first part has been extensively studied in distributed environments with the introduction of DHTs and many off-the-self software components are available, the second part and the combined problem are not widely addressed. The following sections summarise existing approaches in distributed environments related to the second part of the problem, i.e. the numeric filtering and ranking of the results.

3.2.1.1 Numeric Query Operators

One of the most common numeric query operators is the range filter, where the numeric values of one or more attributes of the returned results must fall within the range(s) specified in the query. Distributed hash table (DHT) designs [RFH+01, SMK+01, ZKJ01] are widely proposed for exact match search operations. While hashing is crucial for DHTs in order to get good load balancing properties, it is also the main barrier in implementing range queries as it destroys the structural associations of the keys. As opposed to DHTs, SkipNet [HJS+03] provides controlled data placement and routing locality by organising data and nodes over an hierarchical string naming scheme. By arranging content in name order rather than dispersing it, efficient operations on ranges of names are possible. Furthermore, by using names rather than hashed identifiers to order nodes in the overlay, the locality based on the names of objects is preserved. In Mercury [BAS04] the authors address the problem of range queries explicitly, allowing for multi-attribute range queries and introducing dedicated mechanisms for preserving the load balancing properties of DHTs. Mercury creates a routing hub for each attribute in the application schema. Each routing hub is organised into a circular overlay of nodes and data are placed contiguously on this ring. Queries are passed to exactly one of the hubs corresponding to the attributes that are queried, while a new data item is sent to all hubs for which it has an associated attribute.

Another approach to support range queries is based on the Distributed Segment Tree (DST), a DHT-based structure [ZSLS06]. A DST is a balanced binary tree where each node represents a segment of the entire possible value range. The two child nodes of each inner DST node equally divide their parent's segment into two halves with the root node being responsible for the entire value space. Each DST node is mapped onto the DHT node associated with the hash value of the DST node's value range segment. The query range is split into the union of the corresponding smallest segments in the DST tree, and queries are forwarded to the corresponding DHT nodes using the underlying DHT routing protocols. DSTs are designed to support both range and cover queries. While in range queries the system registers keys to specific values and returns all the keys that are within a particular range, in cover queries the system registers keys to a range of values and returns all the keys whose registered range covers the query range. An example of a cover query is searching for all the relaying nodes that include in their cache a particular range of data packets.

Range or rather radius queries and *k*-nearest-neighbour (kNN) queries are addressed in the context of proximity search problems. Proximity search involves finding closest points in metric spaces and is based on gradual rather than exact relevance using a distance metric. Proximity search is used, among others, in content search involving complex data types such as images, videos, time series, text and DNA sequences. Radius queries return results with values in the metric space that are within a given distance from the point specified in the query. In some cases where there is no strict predefined threshold that determines the relevance of the results, the discovery of the *k* best matches with kNN queries is more suitable than the discovery of all the matches within a given

range, which, depending on the registered data, it may return an unknown number of results or even no results at all. Several techniques have been developed to support radius and kNN queries for proximity search over peer-to-peer systems, organising the peers into a flat or hierarchically structured overlay.

In the flat-based approach, each node is mapped to a non-overlapping region of the multi-dimensional metric space and adjacency links are added when the regions of the nodes satisfy a certain condition, e.g. being contiguous forming a Voronoi diagram [BkS04]. In [LLSL04] additional links are added between distant nodes to reduce the routing distance among them following the small-world model. The query resolution is reduced to routing over these links to the node(s) whose regions contain the query range. The flat-based approach is less efficient in high dimensionality because of the many adjacency links a node has to maintain with neighbour nodes. In the hierarchy-based approach, a multi-dimensional indexing tree structure is used to provide a mapping of nodes to indexes in a hierarchy that can be navigated to resolve queries, building on the success of indexing tree structures in traditional databases [LLS06, MYK04]. Alternatively, EZSearch [TN08] builds on top of Zigzag [THD04], a dual-head cluster hierarchy designed for media streaming.

The techniques reviewed in this section organise the indexes and the connections between the nodes to perform range or kNN queries assuming no additional condition to match the query to results. In ENVISION, however, there is the requirement of finding a certain group of resources identified by an identifier that needs to be matched exactly. Although these techniques are not directly applicable to ENVISION, they provide significant insight in how to efficiently build an overlay to match on numeric operators, which will be used as the basis of developing techniques to address the dual ENVISION problem.

3.2.1.2 Virtual World Object Discovery

In social or game virtual worlds, as the user avatars are moving they encounter objects and other avatars that are present in the new virtual areas they are entering. The distributed scalable discovery of the objects and avatars as they are moving and the exchange of state information between them for as long as they remain visible to each other is a large field of research the recent years. The related techniques could be used in ENVISION to register and discover resources by substituting the proximity based on the virtual world coordinates with the proximity in application and network performance metrics.

One of the typical approaches is [KLXH04], which builds on top of the Pastry DHT [RD01] to organise the nodes and uses Scribe [CJK+03] for sending updates regarding leaves, arrivals and state updates within the areas of the virtual world. The virtual world is divided into regions of fixed size. The peers and the regions are assigned with hashed identifiers and for each region the node that has the identifier that is numerically closer to the region identifier becomes the region coordinator. The coordinator serves as the root of the Scribe multicast tree for the region and the users inside the virtual region subscribe to the root node to send and receive updates from other users. Coordinator nodes maintain links between them, facilitating the user transition between regions. The assignment of coordinator nodes and inner multicast tree nodes is done based on identifier proximity rather than the nodes virtual world or network proximity. As a result, the users that operate the coordinator node and the inner multicast tree nodes are unlikely to be present in the corresponding virtual area, reducing the efficiency of the overlay.

To improve this inefficiency, [BPS06] proposed to assign each node with the responsibility to maintain the status of all the objects that are nearer to this node in the virtual world, and to use a DHT overlay for the discovery of these nodes. However, instead of appointing coordinators the nodes establish direct connections between them as long as their visibility areas overlap, allowing thus for more efficient exchange of state messages directly over the interested nodes.

A different approach is presented in [HL04], where the virtual world is partitioned to a Voronoi graph with points the virtual world coordinates of the user avatars and the virtual area around them the non overlapping cells of the Voronoi graph. A node maintains direct connections with all the nodes in cells adjacent to its own. Through these direct neighbours it also discovers boundary neighbours, which are the nodes within a given radius from its virtual world coordinates that determine the avatar's visibility area. As the avatars move in the virtual space, the nodes that are direct neighbours detect the updates and report any relevant changes to their boundary neighbours.

The techniques for discovering the objects in the virtual world are of interest to ENVISION mostly from the point of view of organising the overlay nodes and assigning responsibilities for controlling a particular area of the metric space. These techniques become even more relevant if the resources registered in the system change their position in the metric space, when for example there is a significant change in the network performance to reach these resources, and this dynamicity needs to be taken into account when designing the ENVISION solution.

3.2.1.3 Application-Layer Anycast and Manycast

3.2.1.3.1 Anycast

A distributed search query can be thought of as the parallel execution of a query on many subindexes. However, for scalability reasons, it is necessary to limit the number of subindexes where the query is run to those which yield the greatest benefit to the query originator. Since this benefit may be a function of network properties such as latency, throughput or loss, these become important variables to consider.

The prevailing approach to this problem is from the point of view of anycast [FBZA98, KW00, ZAFB00, RK02, SAZ04, CDKR03, BF05, FLM06], of which two kinds can be traditionally distinguished: network layer anycast (NLA) where the routers themselves perform group management and querying functions, and application layer anycast (ALA) where these are delegated to overlay nodes. Typically, the former have access to network layer functions such as multicast or data link services, whereas the latter usually have access to metrics such as server load and available bandwidth. Bhattacharjee et. al [BAZS97] propose the use of anycast domain names (ADNs) with ADN resolvers maintaining a database of metrics, such as server load. These metrics are useful only if they accurately reflect the current state of the network, and thus, techniques are needed to accurately propagate them. Work presented by Zegura [ZAFB00] provides a hybrid server push technique for the maintenance of the metrics database, and shows that lower response times can be obtained when compared with randomly chosen servers.

Much NLA research focuses on solving its scalability issues. Work by Katabi [KW00] overcomes some of the shortfalls in NLA by using route caching techniques; Ballani proposes the use of proxies [BF05] as a means to reduce the size of routing tables as the number of groups increases. As another example, the Internet Indirection Infra-structure (i3) [SAZ04] provides an anycast service. However, this solution may be of limited scalability in practice, as it either relies on a single node that maintains all members of each group, or requires a tree to be explicitly constructed and maintained.

Early work done on ALA [BAZS97, FBZA98] was geared towards small, stable groups, and thus did not focus on the control and management of churn. This last assumption is rarely justified in peer-to-peer overlays [SR06], and is addressed in [CDKR03] by utilising a proximity-aware overlay spanning tree based on Pastry [SR06]. In addition, [CDKR03] proposes a solution suitable for applications with a small number of large groups, and with a loose requirement for accurately routing to the closest group member. In [CWWK06], the authors propose an anycast service optimised for high-volume content distribution that is based on a distributed, locality-aware object-level directory. Other proposals for the solution of the ALA problem include Oasis [FLM06], that elaborates on reducing the measurement overhead involved in locating the closest group member, and [RK02], an algorithm based on attenuated Bloom filters that uses Tapestry [ZKJ01] as a locality-aware DHT. These systems,

however, are only partially adequate for advanced media applications as required by ENVISION: [FLM06] requires the search to begin at a known rendezvous point for each group, and [RK02] is insensitive to performance penalties on congested or long-distance links.

3.2.1.3.2 Manycast

Anycast sends messages to a single member of an anycast group, however there may be instances when a user wants to notify a larger subset of the anycast group. Routing messages to a subset of nodes in an anycast group that satisfy a given criterion is typically called n-casting or manycasting, and it is an operation that fills the spectrum of network communication space between anycast and multicast [CYRK03].

Many works have addressed the development of scalable techniques for n-casting. One of them is [CDKR03], an N-casting system based on Scribe [RKCD01] and built using Pastry, a tree-based distributed hash table [CDHR02]. For standard anycasting, where each group has been allocated a groupID, messages are routed to the root node of the group tree. At each hop, the node handling the message checks whether it is a member of the group referenced in the message. If there is a match, the message is not forwarded, with a depth-first search of the group tree being initiated instead. The system keeps a count of how many nodes within the group have handled the message, stopping when the desired number of nodes had been visited.

Other methods to enable N-casting include beaconing, proposed by Kommareddy et al. [KSB01]. The authors propose a system where groups are assigned to beacons, reference points throughout the network. Each group has a beacon, with all nodes within that group periodically polling and reporting the distance between them and the beacon. The beacon acts as a repository of distances, which is queried by external nodes when the closest node in a group is required. When answering queries, the beacon returns a list of nodes that are within given bounds of a distance parameter (for example, IP hops). This procedure of polling beacons for nodes within a pre-determined distance is repeated, with the querying node performing an intersection of the node lists it has accumulated. As more beacons are queried the intersection is reduced; the node stops querying beacons when the number of nodes after the intersection is down to a desired amount.

Anycast, and in particular manycast techniques are of extremely relevant to ENVISION, as they address the dual problem of finding results that match exactly an identifier (the anycast group identifier) and that are selected based on some proximity metric to the query originator node. In ENVISION, however, the requirements for achieving high scalability, resilience to churn and independency from a particular infrastructure, impose some requirements that are not met by these existing techniques and where the ENVISION innovation will mainly lie.

3.2.1.4 Distributed Databases

Much of the early work in distributed database systems focused on the caching of results not on server infrastructure, but on equipment closer to the end user. For instance, the IBM DBCache system [ABK03] provides a local cache on which queries can be resolved. Those queries not resolved locally are resolved in a central server, and the results cached on the client. A similar system, Microsoft MTCache [kLGZ04], relies on manual cache assignments and applies improved cost considerations when determining whether to execute the query locally or remotely.

There has been a substantial amount of research on the formal aspects of distributing search operations over a collection of separate, local databases managed by different organisations, and with no global schema. The Local Relational Model (LRM) [BBG02] and the work by Franconi et al in [FKLS04] both introduce formal models (using relational algebra) in which peers consult neighbour nodes to obtain the answer to queries which they cannot answer. Additional works focus on either the management of update messages, the decentralised maintenance of database consistency [VS05], or the formal semantics of the system [Maj04].

Some works in peer-to-peer database rely on the building of interest groups. In [GZ02], each node is responsible for holding metadata about groups pertaining to different generic query topics. Routing is decided on the basis of these groups, and queries are forwarded and resolved recursively; peers keep a record of their acquaintances with respect to individual queries and construct graphs of how a query is evaluated. Other works have focused on the determination, for unsuccessful queries, of the part of the query that cannot be answered. In particular, [MRP99] presents a query difference operator formulated on the basis of relational algebra expressions operating over advertisements generated by peers and conforming to a known global schema aggregated from those of all the participating peers.

There is a set of works focusing on the provision of database services based on structured peer-to-peer overlays, as opposed to gossip-based systems that usually operate on the basis of TTL-limited message broadcasts and with no guarantee of complete answers [Hos02]. One of these is [GHI01], which concentrates particularly on the problem of where to initially position the data in the network. Peers work collectively in “spheres of co-operation”, pooling their data resources and sharing routing responsibilities. Peers broadcast unanswerable queries amongst their sphere of co-operation and optimally request different parts of these unanswerable queries using a particular cost model. Another example, the PIER [HHL03] Internet query engine, provides database query processing techniques based on a DHT. PIER query plans are written in its own dataflow language, UFL, and can be sent by any user to any node, which then becomes a proxy for that user. The proxy parses the UFL into Java bytecode, and sends the query plan to nodes which can answer the query. A DHT-based indexing system is used to keep track of which nodes can resolve a query.

Some works approach peer-to-peer databases from the point of view of query planning and optimisation. The SQPEER Middleware [KC05] is a query processor designed to operate in semantic overlay networks. If the query submitter does not know how to obtain some parts of the query, these parts of the plan are left blank, for the peers that receive the plan to complete. The Mutant Query Plans [PMT03] system splits data into interest areas, each one served by a base server. Other types of servers – index servers, meta-index servers and category servers – keep records of base servers and overlapping interest areas. As a final example, the query trading algorithm [PI06] is designed for distributed networks of autonomous database systems and combines query optimisation techniques with market equilibrium as a distributed optimisation technique. Queries are viewed as commodities, with users submitting queries modelled as buyers and those providing the answers to the queries as sellers.

Finally, much recent work focuses on the continuous querying of real-time data streams from decentralised data sources. These include Medusa [CBB03], PIER [HHL03], IrisNet [GKK03] and Borealis [AAB05]. These systems provide a general-purpose distributed query processing engine that usually offers a relational data model and provides an elaborate set of operators to applications. Conceptually similar to these, S4 [NR10] is a general purpose, scalable, partially fault-tolerant platform for the processing of unbounded data streams. By creating topologies of processing elements that operate on key-value event streams, S4 can implement many real-time applications such as clickthrough measurement.

Although all these database-inspired systems are very flexible and expressive in their query construction and processing capabilities, they impose important requirements on the overlay peers. Given that groups could be large in ENVISION applications, systems in which control of group metadata is delegated to a single node are unsuitable for these applications. Moreover, since peers in ENVISION are expected to be subject to high levels of churn, the performance of full-fledged database systems may be insufficient in many cases. Thus, rather than general-purpose data management systems, ENVISION will research optimised systems which provide good locality performance while mitigating peer churn.

3.2.2 Keyword-Based Content Search

The efficient discovery of the content, based on the fractional knowledge about the content becomes a challenging problem in large-scale P2P systems. P2P networks are highly dynamic in nature. Due to this dynamicity of peers and content, it is difficult to obtain the complete knowledge of the available content in the system. As a consequence a P2P system requires an efficient search mechanism that utilizes the partial knowledge. While keyword search is a popular query type over the Web, how to implement keyword search mechanism efficiently on P2P systems remains a challenging task. Different from traditional web search engines, it is often difficult, if not impossible, to maintain a centralized content index in a large scale P2P network [CJW08].

A centralised file system, as present in any traditional operating system, permits sophisticated search operations involving wildcards and partial keywords. Enabling existing P2P file sharing systems with efficient wildcard search capability will allow users to perform flexible search. Besides inexact keyword matching, many problems, such as partial service description matching for service discovery systems, and data records for P2P database systems can be mapped to the keyword based search problem.

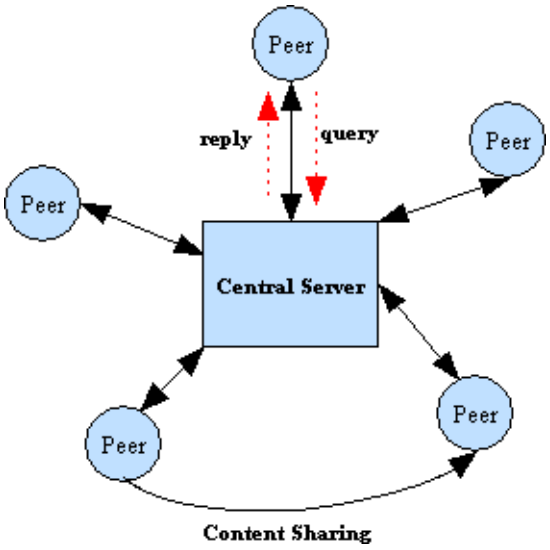
Peer-to-Peer Content Sharing

P2P content sharing (file, video streaming) is the most interesting and widely accepted P2P application. The P2P systems use several mechanisms for content lookup and distribution. Although content distribution takes place between two peers, the search mechanism usually involves intermediate entities. To facilitate effective search, an object is associated with an index file that contains the name, location, and a description of the content. The search for content typically involves matching a query against the index files. P2P systems differ in how this index file is distributed over the peers (architecture) and what index scheme is used (i.e., index structure). Existing P2P retrieval mechanisms provide a scalable distributed hash table (DHT) that allows every individual keyword to be mapped to a set of content/nodes across the network that contain the keyword. Using this single-keyword based index, a list of entries for each keyword in a query can be retrieved by using existing DHT lookups.

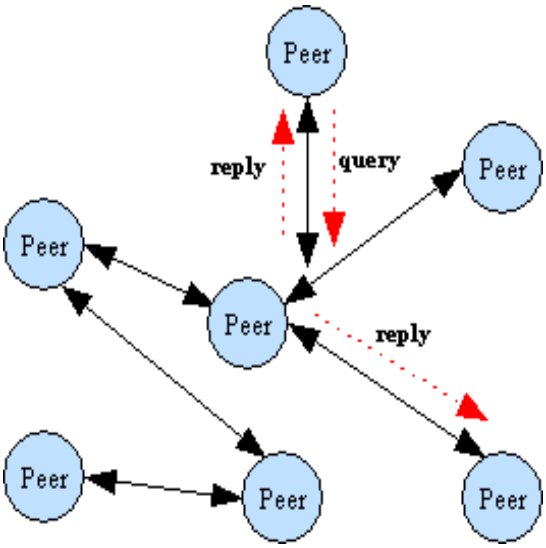
P2P systems are divided into three categories on the basis of their architecture (Figure 3). It can be centralized, decentralized, or partially-decentralized. Centralized P2P systems are characterized by the existence of a central index server, whose sole task is to maintain the index files and facilitate content search. Although centralized P2P systems are simple and easy to manage all data information, they have some obvious problems which are not suitable for large scale systems. The indexing server is a bottleneck and a single point of failure.

Decentralized architectures resolves this problem by having all peers index their own local content, or additionally cache the index of their direct neighbours. Content search in this case consists in flooding the P2P network with query messages (e.g., through TTL-limited broadcast in Gnutella). When a node receives a request for a key which represents the data item, it attempts to retrieve the content locally if possible, otherwise it forwards the request to another node. When a request is successful or failed, the desired content or failure report is returned to the requester along the same path of the incoming request. A decentralized P2P system such as Gnutella is highly robust, but the query routing overhead is overwhelming in large-scale networks. Also there is no guarantee that the content location information is reliable or not.

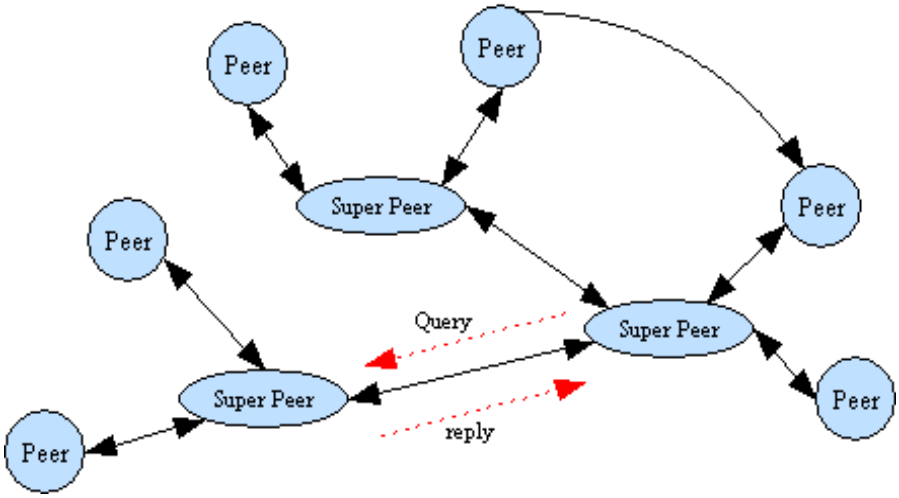
Recognizing the benefit of index servers, many popular P2P systems today use partially-decentralized architectures, where a number of peers (called super-peers) assume the role of index servers. Each super-peer is in charge of maintaining the index file for its peers. Content search is then conducted at the super peer level, where super-peers may forward query messages to each other using flooding. The selection of super-peers is difficult in such a scheme, as it assumes that some peers in the network have high capacity and are relatively static (i.e., available most of the time).



Centralized Architecture



Decentralized Architecture



Partially Centralized

Figure 3 – P2P Architecture

3.2.2.1 Structured techniques

3.2.2.1.1 DHT Based Solutions

In general DHT-techniques (Chord, CAN, and Tapestry) are not suitable for solving the partial keyword matching problem (and the DPM problem) mainly for two reasons. Firstly, DHT techniques use numeric distance based clustering of patterns which is not suitable for pattern matching and results into multiple DHT-lookups per search. Secondly, DHT-techniques cannot handle common keywords problem well. Popular keywords can incur heavy load on the peers responsible for these keywords as a result, the distribution of load will become unbalanced among the participating peers. Inability to support partial keyword matching is considered a handicap for DHT techniques. In the last few years a number of research efforts have focused on extending DHT-techniques for supporting keyword search. Most of these approaches adopted either of the following two strategies:

- Build an additional layer on top of an existing routing mechanism, like Chord, CAN or Tapestry. The aim is to reduce the number of DHT lookups per search by mapping related keywords to nearby peers on the overlay.

- Combine structured and unstructured approaches in some hierarchical manner to gain the benefits of both paradigms.

Keyword Fusion is also an inverted indexing mechanism on top of Chord. This work extend Chord to support keyword-based queries in a straightforward manner by maintaining <keyword, list of values> information at each DHT node, in place of <file ID,value>. Note that because the same keyword can appear in multiple files, unlike in the case of file ID, the right hand side of the mapping is extended to store a list of values to include the locations of all files containing that keyword. When a user searches files with a keyword, the extended Chord forwards the query to the node that contains the location of the files annotated by that keyword. If a user specifies multiple keywords to locate a file, the extended Chord should take the intersection of the results for each keyword before returning the results to the user.

3.2.2.1.2 Non-DHT Solutions

There are also some Non-DHT solutions to the search problems in P2P networks. SkipNet [HJS+03] and SkipGraph [AS03] are the example of these approaches. Both techniques use the Skip list [PUG90]. In both SkipGraph and SkipNet, nodes responsible for the upper level elements of the Skip List become potential hot spots and single points of failure. To avoid this phenomenon, additional lists are maintained at each level. This in turn increases the degree of each node. A multi-level indexing mechanism for keyword search based on SkipNet has been proposed in [SYW04] However, none of these approaches can efficiently support partial keyword search because the underlying data structure used by these techniques, i.e., Skip List, supports prefix matching only.

3.2.2.2 Non-structured Techniques

A structured system identifies the object by keys, normally generated by applying hash function. Key-based query routing is highly efficient as compared to keyword-based unstructured query routing, however key-based query routing requires accurate search parameters. The non-structured P2P systems identify the object as keywords. In these systems the queries are expressed in terms of keywords. These keywords are associated with the shared objects. There are several un-structured systems than can be used for partial matching queries. These include flooding [GNU10] and Random walk [LPE02]. But, due to the lack of proper routing information, the generated query routing traffic would be very high. Moreover the search completeness is not guaranteed.

There are several techniques for search in un-structured system. The routing hints are used by several systems. In [CRB03], queries are routed to peers having higher capacity with higher probability. In [TR03] and [YM02] peers learn from the results of previous routing decisions and bias future query routing based on this knowledge. In [CFK03] peers are organized based on common interest, and restricted flooding is performed in different interest groups.

Bloom Filters is another technique used by un-structured system for improving the query routing efficiency. In [LLS03] each peer stores Bloom filters from the peers at a distance of one or two hops.

Three ways of aggregating Bloom filters are also presented. Experimental results presented in [LLS03] show that logical OR-based aggregation of Bloom filters is not suitable for indexing information from peers more than one hop away. In [RK02] each peer stores a list of Bloom filters per neighbour. The *ith* Bloom filter in the list of Bloom filters for neighbour *M* summarizes the resources that are *i*-1 hops away from neighbour *M*. A query is forwarded to the neighbour with a matching Bloom filter at the smallest hop-distance. This approach aims at finding the closest replica of a document with a high probability. An approach similar to [RK02] has been presented in [KXZ05] which uses an exponentially decaying Bloom filter for indexing neighbour content.

3.3 High-level Specifications

Figure 4 depicts the resource data management functions and the control and metadata interactions between them and with the other application and network layer functions. These interactions are summarised below and the following sections provide a short description for each function.

Metadata Interactions:

- The user profile management function registers the type of resources that the user is willing to provide to the application, in terms of both the capacity and the functionality to be provided, e.g. relaying node, active monitoring node etc.
- The consolidated overlay view functions register updates on the network properties of the overlay nodes, the capabilities of the ISPs and the mappings between overlay nodes and ISPs.
- The consolidated overlay view functions provide additional information regarding network conditions; these are then used to filter the results of the resource discovery function (e.g. the delay between nodes that determines which nodes are considered closer to a particular node with a query for resources).
- The content distribution functions register the nodes that participate in the distribution of a particular stream.
- The resource registration function populates the data structures required by the resource discovery to retrieve the registered data.
- The resource discovery function provides information to the content adaptation, content distribution and consolidated overlay view functions regarding the overlay node and ISP resources, and provides the ISP with information about the overlay.

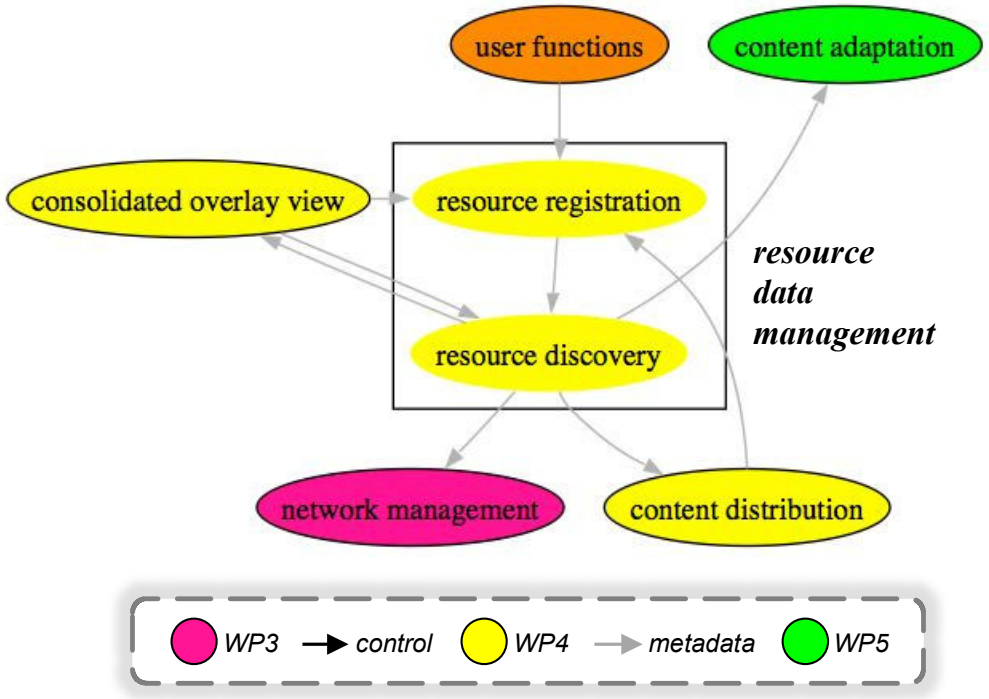


Figure 4 – Resource Data Management Functions

3.3.1 Resource Registration

The resource registration function is responsible for storing information about application resources and their network performance properties.

Resources provided by participants are registered by the user profile management functions when the users join the application or when they modify their profiles to increase or decrease the amount and type of resources made available to the application. Resources offered by the ISPs are registered by the consolidated overlay view functions that discover their existence as part of the capabilities of the underlying ISPs. Finally, the content distribution functions maintain the associations between overlay resources and the distribution of content objects: when a node starts receiving a particular content object with a particular format or a given set of properties (e.g. buffer size or playout lag) this is stored at the resource data management system.

The resource registration function should scale with the number of the information items that it needs to store, which grows with the number of overlay nodes, resources and content objects. Therefore, it should be highly distributed and with the capability to offload processing and storage functions from infrastructure-provisioned resources and onto user-contributed capacity. Further, the resource registration function should partition and distribute the storage of information to overlay nodes to allow for the optimisation of resource discovery functions in terms of accuracy, query response time and other related performance metrics.

More detailed requirements for the resource registration and discovery functions can be found in section 3.1.2. The logic for partitioning the information to be stored to distributed indexes, the data structures that will be used to store the distributed indexes and the protocols required for updating these indexes are not finalised at this point and require further investigation.

3.3.2 Resource Discovery

The resource discovery function is responsible for processing queries regarding application resources as accurately and quickly as possible using the distributed indexes produced by the resource registration function.

As explained in section 3.2.1, the resource discovery problem involves two basic operations: (a) the matching of an exact identifier that defines the type of resources that are of relevance to the query, and (b) the filtering and/or ranking of the matching resources based on application and network performance criteria. The resource identifier should be unique for each type of resources, e.g. content relaying nodes, content adaptation nodes, idle high capacity nodes, etc., as well as, where applicable, for each content object that these resources are used for, e.g. content relaying nodes relaying a particular video stream. The filtering and ranking of the resources could then take advantage from network performance information such as delay and throughput from the querying node.

Any function that is dynamically distributed over a number of overlay nodes as the size of the overlay grows will interact with the resource discovery function to locate available resources. Examples of such functions include the consolidated overlay view, several content adaptation and distribution functions, and the resource data management function itself. In addition, resource discovery is used by content distribution functions to discover candidate senders that have the content and are best positioned to serve it to other nodes that need to receive it. Finally, the resource discovery is responsible for aggregating and filtering information required by the ISPs in exchange for the network performance or other information they provide. Examples of such information include the number of nodes within the ISP's domain that consume a particular content object at a particular bitrate, or the number of nodes that could benefit from the establishment of a multicast tree.

As the indexes are distributed over a number of overlay nodes, it is expected that queries will have to be forwarded over a number of nodes hosting the indexes to retrieve a sufficient number of results. The logic for constructing and updating the query forwarding tables to optimise the accuracy and query response time depends greatly on the way the distributed indexes are built and updated and is currently under investigation.

4. OVERLAY RESOURCE OPTIMISATION AND CONTENT DISTRIBUTION FOR LIVE VIDEO

4.1 Problem Statement

4.1.1 Description

The demand for advanced Internet multimedia, such as video streaming, is increasing [HLL+07]. Simultaneously, the deployment of increasingly higher bandwidth technologies in the network edge makes peer-to-peer overlays increasingly attractive to achieve systems with high scalability and low publishing cost. By removing the need to deploy specialised resources, this architecture could allow anyone to broadcast and reach consumers anywhere in the world. Although some principles and some components of such an architecture could be used by many applications, including video distribution, real-time multi-user 3D environments and games, the techniques for optimising the content distribution differ with the type of content and the associated QoE metrics. In this particular section, we are considering the content distribution techniques and research topics pertaining to live video applications with large coverage and unpredictable demand.

Although the use of multicast and quality of service (QoS) mechanisms at the network layer seem ideal for delivering live content, access to these resources to overlay applications is still beyond the capabilities of most distribution networks. Although the lack of access to multicast and QoS capabilities can be overcome through the use of unicast connections or network over-provisioning, these solutions unattractive both due to their reduced effectiveness and their wasteful use of network resources. Borrowing from traditional content delivery networks (CDNs) for Web content, some advanced media service providers create "walled garden" intranets with native multicast, traffic differentiation, abundant capacity, and dedicated infrastructure. Although this approach gives acceptable results, even for 8 Mb/s high definition (HD) H.264 TV channels, it is neither scalable nor accessible to a wide selection of content producers.

Recently there have been numerous studies and even deployed systems that rely on peer-to-peer protocols to transmit TV streams, including live channels, from a single fixed source to a set of recipients. Such systems inherently scale with the number of participants and allow for the cost-efficient distribution of streams with very few participants otherwise impossible over the expensive IPTV infrastructure. The best performance of currently deployed peer-to-peer systems can be summarized as having 20 seconds startup delay and 1 min playout lag with acceptable playout continuity for streams of 350 kb/s [SMG+07]. This is far from even the average performance of current IPTV “walled gardens”, that exhibit 2 seconds startup delay and 2 seconds playout lag with good playout continuity for streams of 3.5 Mb/s.

The content distribution techniques developed in ENVISION aim to close the gap between these two scenarios by enabling cooperation between advanced distribution overlays and their underlying ISPs. By complementing application layer swarming with network layer techniques such as QoS and multicast, performance-critical applications can be deployed scalably. Furthermore, by making overlay networks aware of the content that they transport, resource allocation decisions can be made more intelligently.

Finally, ENVISION is mainly about enabling the future networked media applications, and the content distribution techniques to be developed need to meet requirements regarding user interactivity and scalability to support unpredictably large number of participants, located anywhere in the world. The following section provides a set of requirements that are indicative of the scale and performance requirements associated with these applications.

4.1.2 Requirements

4.1.2.1 Functional Requirements

- The content distribution system should be able to support various encoding/decoding formats, bandwidth and processing capacity levels, both at the content source and at consumer terminals
- The content distribution system will optimise on all or a subset of the following criteria:
  - Playout continuity: The percentage of video data units successfully rendered by the consumer within the required time constraints. High playout continuity enables the local reconstruction of the media session with minimal distortion.
  - Quality Level: The quality of the video that can be sustainably received at any point in time, in terms of spatial, temporal and video signal resolution.
  - Playout lag: The delay between a video data unit being sent by the video source and its rendering by the consumer. Low playout lag means that media object rendering is more "live."
  - Startup delay: The time between a consumer's request to gain access to a particular video live stream and the stream being available for continuous local rendering. Low startup delay allows quick switching between video streams, aka zapping.
- The content distribution system should interact, support and take into account information provided by the content adaptation overlay functions (see also [D5.1]), including for example:
  - Data encoding priority and FEC protection levels input to the topology construction and data scheduling functions
  - Multiple access link capabilities input to the topology construction and data scheduling functions
  - End-to-end path performance and QoE evaluation across a number of end-points to provide feedback to overlay resource management, FEC and content adaptation functions
  - Triggering of alarms when low performance is attributed to content adaptation and FEC functions
- The content distribution system should include functionality to support the invocation of network services and resources, including for example:
  - Evaluation of the cost/benefit of using multicast or ISP-provisioned relaying nodes
  - Triggering the establishment or teardown of multicast trees for a stream or a subset of a stream (e.g. only the most popular SVC layers)
  - Election of overlay nodes to act as multicast sources and protocols to coordinate the handover from overlay to multicast reception of content
  - Invocation of relaying resources (application, user or ISP-provisioned nodes that are not consuming a particular stream, the equivalent of seeders for file sharing)

4.1.2.2 Performance Requirements

- The content distribution functions should be designed to support of up to thousands of simultaneous live video sources, some of which might be reliable mobile video feeds (e.g. professional vehicle camera systems), and some unreliable fixed video feeds (e.g. amateur spectator cameras)

- The content distribution functions should be designed to support of up to millions of simultaneous content consumers across all video streams located across many ISPs
- The content distribution functions may allow for a small number of consumers to receive up a larger number of streams simultaneously (e.g. studio editors)
- The content distribution functions should be designed to accommodate without significant disruptions high levels of churn in the interest of content consumers to particular streams
- The content distribution functions should be designed to accommodate a long tail for content popularity: a few popular streams will retain most of the content consumers, a large number of much less popular streams attract interest only temporarily

4.2 State-of-the-art and Innovation

To address the problem of live video distribution and meet the requirements set in the previous section, a number of technical challenges need to be addressed: optimisation of overlay topology and scheduling algorithms for live video streaming and resilience to churn caused by user departures switching off overlay nodes, resilience to churn caused by short-lived interest of users to particular content, content-aware streaming optimised for delivering different content formats, mobilisation of resources, and optimum resource management and finally cross-layer optimisation with the network cooperation.

While significant advances have been made in the literature to address these research challenges, they typically provide partial solutions and assume none or very limited collaboration with the underlying network. ENVISION aims at providing a unified solution for live video distribution with emphasis on the collaboration with the network through the CINA interface based on the results and enhancing the techniques available in the state of the art. The following sections present a survey of the most relevant work on the research challenges presented above.

4.2.1 Live Streaming Topologies

The challenge of distributing live content is somewhat different to that of distributing recorded content on demand. A major difference is that delay is important to optimise so that consumers can view streams as live as possible, whereas throughput only needs to be large enough to view the stream – a user cannot download faster than the rate the stream is generated.

Early peer-to-peer streaming systems, based on application layer multicast, organised peers in a tree with the content source at the root. After an initial consumer request, the entire stream would be pushed to the consumer from a peer with available upload bandwidth and probably in close network proximity [VYF06]. This scheme gives good startup delay and playout continuity if the tree is stable under churn (peer arrivals and departures), and good playout lag if the tree is also short (low number of peers between content source and consumer).

More recently, Bos [LZ07] proposed a method which constructs a data distribution tree containing the *Euclidean Minimum Spanning Tree*, where the distance in the Euclidean space represents the network delay. A subset of stable and high capacity nodes are elected to become *super peers*. Super peers are interconnected to form a *Yao graph*, a structure which contains the Euclidean Minimum Spanning Tree. Normal peers attach directly to the closest super peer. The source routed multicast tree is built over the super peers topology based on the compass routing protocol.

The departure of a node in a single distribution tree results in complete loss of connectivity for all the nodes in the underlying subtree (i.e., single point of failure). To overcome this problem, several studies investigate streaming the data over a forest of multicast trees, each of which carries only part of the stream. CoopNet [PWC02] is a forest-based streaming approach, where the authors identify a trade-off between efficiency in terms of locality and path diversity required for resilience to node departures. Upon addition of a new node, the source returns a significantly large set of candidate

parent nodes to ensure diversity. As an optimisation the candidate parent nodes are selected so that are they are *nearby* the newly added node.

Techniques for constructing trees typically assume global knowledge and at least one interaction with the source. Alternatively, overlay topologies can be constructed with local knowledge, where the connections are determined by each node and the data flow may take many alternative and potentially overlapping paths. In [RKHS02] a technique for clustering nodes to bins based on their locality is proposed. As a case study of this technique, the *BinShort-Long* overlay construction method is presented, where each node connects to k/2 randomly selected nodes from within its cluster (bin) and k/2 random nodes from anywhere in the system. A similar technique is proposed in [BCC+06] as an improvement for the BitTorrent protocol. The clustering here is done primarily to distinguish between nodes located in the same ISP, and nodes in different ISPs. Out of the total BitTorrent peers discovered by a new peer through the local tracker, all but k are selected to be local peers, with typical values 35 for total peers and 1 for the k external peers. This is done to reduce the traffic over the inter-domain links while still maintaining enough connections with external peers to receive the data. Finally, in [RLC08] the authors formulate the *Minimum Delay Mesh problem* and prove that it is NP-hard. They propose a heuristic for constructing a shallow (low number of hops) and locality-aware (low delay at each hop) overlay topology. In order to minimise the number of hops, nodes with higher capacity need to be connected closer to the source. The selection of the nodes to establish connections with is done after calculating the power of each node as a function of the node's locality and bandwidth availability.

The construction of efficient topologies for live video streaming has been addressed by the NAPA-WINE [NAPA10] project by proposing delay-aware modifications to a class of hybrid push/pull, contention-free protocols presented in [SHMA07]. The *push* function is geared towards the propagation of new information, while the *pull* function is devoted to retrieving locally missing chunks [RULC10].

While topology-aware tree-based topologies with global knowledge create optimum topologies minimising the playout lag, they can be very expensive to maintain for large-scale dynamic overlays. In addition, they might have low resilience to churn (see section 4.2.2). Therefore, the ENVISION live video distribution algorithms could adopt a more flexible distribution topology (mesh, or hybrid mesh and multiple-tree) to establish connections, also taking into account network performance and ISP preferences.

4.2.2 Resilience to Churn

In the highly dynamic Internet environment, variations in connectivity and unpredictable departures of the user-provisioned overlay nodes, have so far kept tree/push systems within research settings or applicable only to small scale deployments. On the other hand, file distribution systems like BitTorrent [PGES05] achieve high resilience to churn through swarming: splitting the file into small data units and ensuring that these pieces are distributed among the set of peers participating in the download. In this way peers obtain parts of the file from many peers in parallel, increasing resilience. Unfortunately, BitTorrent was designed for file transfers, which means that content can only be consumed once the full download has been completed.

Many variations on BitTorrent have been proposed for on-the-fly consumption of swarmed content. One such proposal is Tribler [PGW+07], a modified BitTorrent client that supports both live and on-demand P2P video streaming. To achieve this, it introduces special mechanisms to bypass problems such as having unknown video length or future content availability, unsuitable piece selection policies, and the lack of seeders [MBP+09].

Other recent peer-to-peer live streaming systems, based on similar mesh/pull techniques [HLL+07, SGGS09] have been recently deployed. However, although these systems are much more resilient to churn due to their to local optimisation of delivery and free selection of peers and chunks, they can

be susceptible to relatively long startup delays and playout lags. Furthermore, the degree to which these systems are aware of the limitations and capabilities of their underlying networks is very limited [CGH+10].

The performance of peer-to-peer media streaming systems can be particularly affected when the interest of the users to content is short-lived, also known as zapping in TV multi-channel streaming systems. To address the challenge of creating a stable topology, in [WLR09a] the authors propose to decouple what a peer uploads from what it consumes, bringing stability and enabling cross-channel resource sharing. Each peer is assigned to one or more channels, with the assignments made independently of what the peer is viewing. This has the effect of creating a semi-permanent distribution swarm for each channel at the expense of additional overhead since each peer now needs to upload into its assigned swarm as well as to peers outside the swarm that want to view the channel.

For a live video distribution application with high churn on the interest of the users to streams and built-in support for scalable video encoding, it is important to achieve a good tradeoff between resilience to churn and optimum playout lag, taking into account the importance to the video quality and protection level of the data that are carried over the different connections. ENVISION will study this tradeoff taking into account all the involved parameters.

4.2.3 Resource Optimisation

Peer-to-peer overlays are self-configuring systems where autonomous peers contribute some portion of their resources to collectively distribute data in an efficient and scalable way. In the particular case of QoS-sensitive applications, which peers provide resources to which other peers, and the end-to-end data distribution topology thus created, is critical for the performance of the system. On the other hand, peers in these systems can behave selfishly, leading to freeloading and ultimately to widespread service degradation [HCW05, AH00, PIA+07]. This is one of the main challenges underlying the design of incentive mechanisms, which seek to align the utility of the system with that of individual peers.

In most peer-to-peer overlays, the only way that a peer can provide value to the system is through the contribution of some of its resources, and the only way that a peer can extract value from the system is by consuming the resources of other peers. Thus, the problem of aligning selfish peers with a desired protocol behaviour becomes one of resource allocation, and an incentive mechanism becomes a set of rules that stipulate which peers can gain access to which resources, and under what circumstances. This leads to one of the problems of interest for ENVISION: the design of resource allocation rules that optimise the performance of the system while providing contribution incentives. One approach that has drawn a lot of attention involves the use of Vickrey auctions for strategy-proof resource allocation.

Vickrey auctions have been repeatedly proposed in the literature as a means for adaptable resource allocation. An example of this is CompuP2P [GSS06], a system that implements an open market for peer resources quantised into different markets. Pricing is achieved using iterated single-item Vickrey auctions. 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. The system also uses iterated single-item Vickrey auctions.

The use of iterated single-item auctions 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. Auctions have also been used in the specific context of peer-to-peer streaming. In [TJ06], the authors present a substream-based system in which each substream is distributed using application-level multicast trees. These trees are built by using iterated, sealed-bid, first-price, multiple-item auctions. Another example is [WLL08], where auctions are used to allocate upload capacity to multiple coexisting streaming overlays.

Assuming that an appropriate incentives scheme is in place, there is the issue of how to best utilise the additional resources offered by the peers. Most approaches in the literature develop techniques for optimally allocating the additional upload capacity to peers within a particular stream, with the objective of optimising the overlay topology. More recent studies however recognize that, in deployed peer-to-peer systems, more than one stream are distributed at any time, creating the opportunity to use the additional upload capacity in streams other than these to which the peer already contributes. Measurement studies [HLL+07, HLR07] indicate that there is indeed scope of improvement as the current deployments show that some streams suffer bad quality due to bandwidth deficit, while others have unused surplus bandwidth.

In [MLS+07] the authors introduce the concept of non-consuming peers (NCPs), that request only a fraction of the chunks in a given stream and try to send it as many times as possible, providing a multiplication effect on the availability of data. An extra advantage of NCPs is the reduction of playout lag, as they can be used to create extra capacity near the source, making chunks with less playout lag more widely available. While the previous study assumes the presence of idle non-consuming peers, [WLR09a] assumes that actively consuming peers need not to be restricted to uploading only the stream they consume, to improve the adverse effects of churn in user interest. The same authors go further in [WLR09b] to propose queuing models to study rules for assigning peers to streams. These models include other desirable properties, such as peer churn, peer bandwidth heterogeneity, and Zipf-like channel popularity.

The explicit control of overlay resources has been addressed previously in the context of EU projects. Such is the case for the *highly active peer* mechanism of SmootIT [SMOO10], which relies on a centralised controller to promote peers to superpeer status and to allocate their resources. Another example is the *Give-to-Get* resource allocation policy used by *NextShare*, a research prototype produced by the P2PNext project [P2PN10]. In this case, peers have to upload (give) chunks that they have received from other peers in order to receive additional chunks from those peers. This behaviour provides an incentive for good forwarders and against freeloaders.

In ENVISION, resource optimisation will be studied in the context of invoking network services through the CINA interface compared with relying on user-provided resources based on existing techniques, and the associated tradeoffs will be investigated from the perspective of the application and the underlying networks.

4.2.4 Cross-layer Optimisation

If one considers the edge-to-edge traffic pattern most useful to a given peer-to-peer system or managed overlay, it is clear that it will be a function of the preferences of the overlay peers regarding QoS, resource availability and data caching and replication. On the other hand, if one considers the edge-to-edge traffic pattern most desirable to the underlying ISP, it will be a function of its link and node costs, the background traffic that it carries and its traffic engineering policies. Thus, a tension between the preferences of the overlay and the ISP arises. To resolve this tussle and enable the optimisation across both the application and the network layers, the explicit interaction of overlay applications and ISPs has received increasing attention by the research community in the last 5 years. In most of this work, communication is only from the ISP to the overlay, with no information being fed back into the ISP.

One of the first works to focus in BitTorrent locality was [BCC06], where peers submit peer lists to the ISP, which returns them after tagging each peer as *local* or *external* to the ISP. This allows the

overlay to set a soft limit on *external* peers, thus biasing peer selection to give preference to intra-domain overlay links. The authors compare such localisation with traffic throttling by the ISP, and conclude that although average download time is very similar to the non-biased peer selection case, variability is greatly reduced.

In [AFS07], the authors present a system in which an *oracle* receives peer lists from the overlay and returns it ranked according to the preferences of the ISP. Although many possible metrics are presented as candidates (AS hop path length, IGP metric distance, geographical information, expected delay or bandwidth and link congestion) the authors focus in AS hop length. The system operates by allowing a Gnutella overlay to perform topology construction using oracle-provided peers, and achieves reduced AS distance while keeping other metrics unchanged (such as graph connectivity, mean node degree, graph diameter and mean overlay path length). In [AAF08] the system is extended to take into account peer upload bandwidth, and achieves inter-domain traffic reductions of 20 to 40 per cent. By taking into account CDNs, [PFA10] improves upon these works by allowing the oracle (in this case referred to as *PaDIS*) to enrich DNS responses with network information.

Another contribution in this area that relies on explicit information exchange between the overlay and the ISP is P4P [XYK08], which became the basis for one of the main standardisation efforts in this area [PMG09]. In its original presentation, [XYK08] relied on the ISP grouping peers into groups called PIDs and then providing a set of end-to-end prices between them. Conversely, the overlay used these ISP-provided prices to calculate its own optimal traffic matrix. This scheme assumes that the overlay incentives are aligned with those of the ISP, as these prices are simply a means by which the optimisation problem of the ISP can be decomposed into decoupled subproblems that can be partly solved by the overlays. Tie ISP solved its own optimisation subproblem using a supergradient projection approach, thus calculating the optimal prices to provide to the overlays given the traffic matrices that they have provided. Although P4P provides increased performance over ISP- agnostic operation, its performance, when compared delay localization, is not superior from the point of view of the clients. However, from the point of view of the ISP, P4P significantly reduces bottleneck link traffic utilisation.

There has been particular attention to BitTorrent locality in the context of overlay-ISP interaction. In [SCPR09], the authors present a modified BT tracker that answers client queries for swarm members with peers the are either selected from the 25 per cent closest peers as determined from delay synthetic coordinates or randomly selected over the entire swarm. By simply modifying the trackers in the conventional BitTorrent architecture, [SCPR09] achieves 11 to 16 per cent increases in inter-domain traffic while at the same time reducing file download completion time and average overlay link latency. These benefits increase with swarm size. In [BLD10], modified BT trackers keep logs of the number of peers external to a given ISP that have been given to peers in that ISP, and keep this number under a given threshold. By using these localised peer sets for topology construction, the system achieves a 40 per cent reduction in inter-domain traffic.

The user incentives for BitTorrent localisation, however, have been called in to question. In [PMJ09], the authors consider many of the popular localisation techniques described, and show that they can be frequently exploited to increase ISP revenue are frequent, which can lead to average path length increases of up to 72.6 per cent. Furthermore, if peer throughput is not limited by latency, BitTorrent localisation will fail to deliver good performance to the users. Rather than assuming that the ISP preferences always correspond to locality, which at the same time also reflects the preferences of the overlay application, in ENVISION we distinguish between information on objective network performance metrics the ISP may provide, and information regarding their preferences, which can be subjective and arbitrary. Regarding network performance metrics, the application could validate the information received by the ISPs by periodically comparing it with measurements at the overlay layer. Regarding ISP subjective preferences, and in particular for these cases when they are not aligned with the optimisation objectives of the applications, the ISPs would need to provide

incentives for the applications to comply. The study of schemes for expressing subjective preferences and providing incentives at the ISP layer, and of techniques for performing the tradeoffs between network performance and compliance with the ISP preferences at the overlay layer is ongoing work in ENVISION.

Another Overlay-ISP interaction technique developed in the context of CDNs is [CB08], a client-only solution that clusters peers using their DNS resolutions for CDN content as a similarity metric. Thus, the CDN provider indirectly gives the overlay information on latency-based peer clustering, and allows significant reduction in both the number of router level and as-level hops between nodes in the same cluster, with over one third of overlay paths not leaving the origin AS. Further, the system reduces latency by two orders of magnitude and loss by 30 per cent when compared with random choice BitTorrent topology construction, and improves upload and download times by 42 and 31 per cent respectively in average. However, in median, there is no improvement unless ISPS give higher bandwidth to intra-AS overlay paths.

Finally, the authors in [AFJ08] present a market-based system where inter-domain providers present a set of *network prices* that are incorporated by the overlay peers into a pricing mechanism that performs resource allocation through multilateral trade. This system achieves efficient resource allocation while allowing transit ISPs to minimise inter-domain traffic.

The use of ISP-provided information has been addressed previously by both [SMOO10] and [NAPA10]. In the former case, the issue is addressed through a locality promotion mechanism implemented by a centralised controller [EMTS10]. In the latter case, provisions are made for the peer selection and scheduling mechanisms to be made aware of network-layer measurements such as delay [RULC10] or upload capacity [CLMM08].

In ENVISION, the live video distribution optimisation algorithms will take into account the preferences of the ISPs gathered through the CINA interface in order to fine tune the establishment of connections between the overlay nodes in a way that is beneficial for both the application and the underlying networks. At this point, there are no assumptions regarding the compatibility of the preferences of different ISPs or any one particular model for setting these preferences, to favour locality, to encourage traffic at particular inter-domain links or other. The impact of these different policies to the performance of the application will be studied further in ENVISION.

4.2.5 Content-Aware Streaming

Most of the currently deployed P2P streaming systems do not use scalable video streams. This means that they serve a single version of the video stream to all peers, and they have limited support for heterogeneous peers. To address these issues, a number of works have proposed P2P streaming systems with scalable video streams, e.g. [CN03, HH08, LZX+07, MH09, RO03, MR06]. Cui and Nahrstedt [CN03] present an algorithm that allows each peer to decide how to request video layers from a given set of senders. They assume layers have equal bitrate and provide equal video quality. Hefeeda and Hsu [HH08] study this problem for Fine-Grained Scalable (FGS) videos, taking into account the rate-distortion model of the video for maximizing the perceived quality. Rejaie and Ortega [MR06, RO03] present a framework for layered P2P streaming, where a receiver coordinates the transmission of video packets from multiple senders using a TCP-friendly congestion control mechanism. Lan et al. [LZX+07] propose a scheduling algorithm for peers to request data from senders. The allocation of seed server resources in P2P streaming systems with scalable videos has also been considered in [MH09].

Scalable content delivery techniques can be used also to mitigate the problem of unreliable short-lived peers in current peer-to-peer streaming systems. CoopNet [PWC03] is a peer-to-peer video streaming system where stream reliability is increased by introducing spatial network diffusion redundancy, as well as multiple description coding. Multiple Description Coding [Goy01] is a coding technique where a single media stream is separated into a number independent substreams

(descriptions). Each description is then divided into packets and routed over uncorrelated, mutually disjoint end-to-end paths. Although any combination of descriptions can be received and decoded, the quality of the reconstructed signal is proportional to the number of recovered descriptions: the more descriptions recovered, the lower the distortion of the original signal. Thus, network QoS problems such as congestion or loss will not interrupt the signal playout, as they would have to affect all descriptions simultaneously. Instead, the reconstructed signal will exhibit transitory episodes of reduced quality. CoopNet uses tree-based topologies and it tries to overcome the fragility of the distribution tree to failures, peer disconnections or QoS problems at peers close to the root by using, instead of a single multicast tree, a set of n uncorrelated multicast trees, each carrying an independent substream (MDC description).

Scalable content delivery can be used to provide incentives for the participating peers to contribute more upload resources. In [HC06], the authors propose the use of rank-order tournaments as a basis for an incentive mechanism scheme to provide differentiation of the throughput and the playout lag experienced by the peers according to their resource contributions. The relative contribution of a peer is used as a signal that affects the strategic QoS decisions of the others. This study was based on PROMISE [HHB+03], and used PlanetLab [pla07] to conduct experiments over the wide-area Internet. The rationale of the incentive mechanism is that a given peer will refuse to give service to any peer that has a smaller cooperation score than itself - peers only cooperate with those peers that have cooperated at least as much as themselves. Thus, predictably good peers (those that have largest scores) have much greater flexibility in choosing their peers, and thus are capable of obtaining better throughput and playout lag, thus increasing the Quality of Experience of the users. Finally, the authors compare the effectiveness of their proposed mechanism to the effectiveness of FEC (Forward Error Correction) for codec-level packet loss, and find that up to 35% of FEC overhead is required to have the same loss performance as the incentive mechanism. The system allows deferred benefit from contributions and is resistant to whitewashing attacks, but is vulnerable to sybil attacks.

The distribution of scalable content using overlay networks has received attention by EU researchers in the context of [P2PN10]. In particular, [CEMP10] presents a mapping from SVC layers to pieces for distribution in the peer-to-peer system along with descriptions of the producer and consumer site architectures of the system.

In ENVISION existing techniques for scalable video streaming will be integrated to the live video distribution algorithms, producing a unified solution that optimises the content quality as well as minimising the playout lag, while taking into account the ISP preferences expressed through the CINA interface.

4.3 High-level Specifications

Although one could imagine a single function orchestrating the establishment of overlay connections and controlling the bandwidth and data scheduling at each connection, such a centralised architecture is unattractive for dynamic large-scale content distribution systems. Rather, content distribution functions will need to be distributed either at every node, or at selected nodes coordinating a group of overlay nodes. In the general case, however, a closed group of nodes will be unable to satisfy requests for content without establishing connections with external nodes – content source nodes may be outside this group, or there might be not enough resources within the group to distribute the content. For this reason, separate content retrieval and sending functions are needed, with the first handling the requests for content at any particular node or group of nodes and the second handling the announcement of content availability and the allocation of upload capacity to the requests received at any particular node or group of nodes.

Figure 5 depicts the content distribution functions for live video content and the control and metadata interactions between them and with the other application and network layer functions. These interactions are summarised below and the following sections provide a short description for each function.

Control Interactions:

- The content retrieval function receives instructions from the content consuming user function when a new stream is selected, and from the quality level selection function if content adaptation is required (modification of the SVC layers / bitrate at which this stream is to be received).
- The resource management function controls the resources made available to the relaying resource allocation and multicast management functions, e.g. decides to establish or teardown a particular tree.
- The multicast management and relaying resource allocation function communicate with the ISP to control the corresponding network services.
- The multicast management function controls the content retrieval function of the nodes that are to receive through multicast, and the content sending function of the nodes elected to send the content to the multicast tree.
- The relaying resource allocation function controls which relaying nodes will relay which streams or layers or parts of a stream.

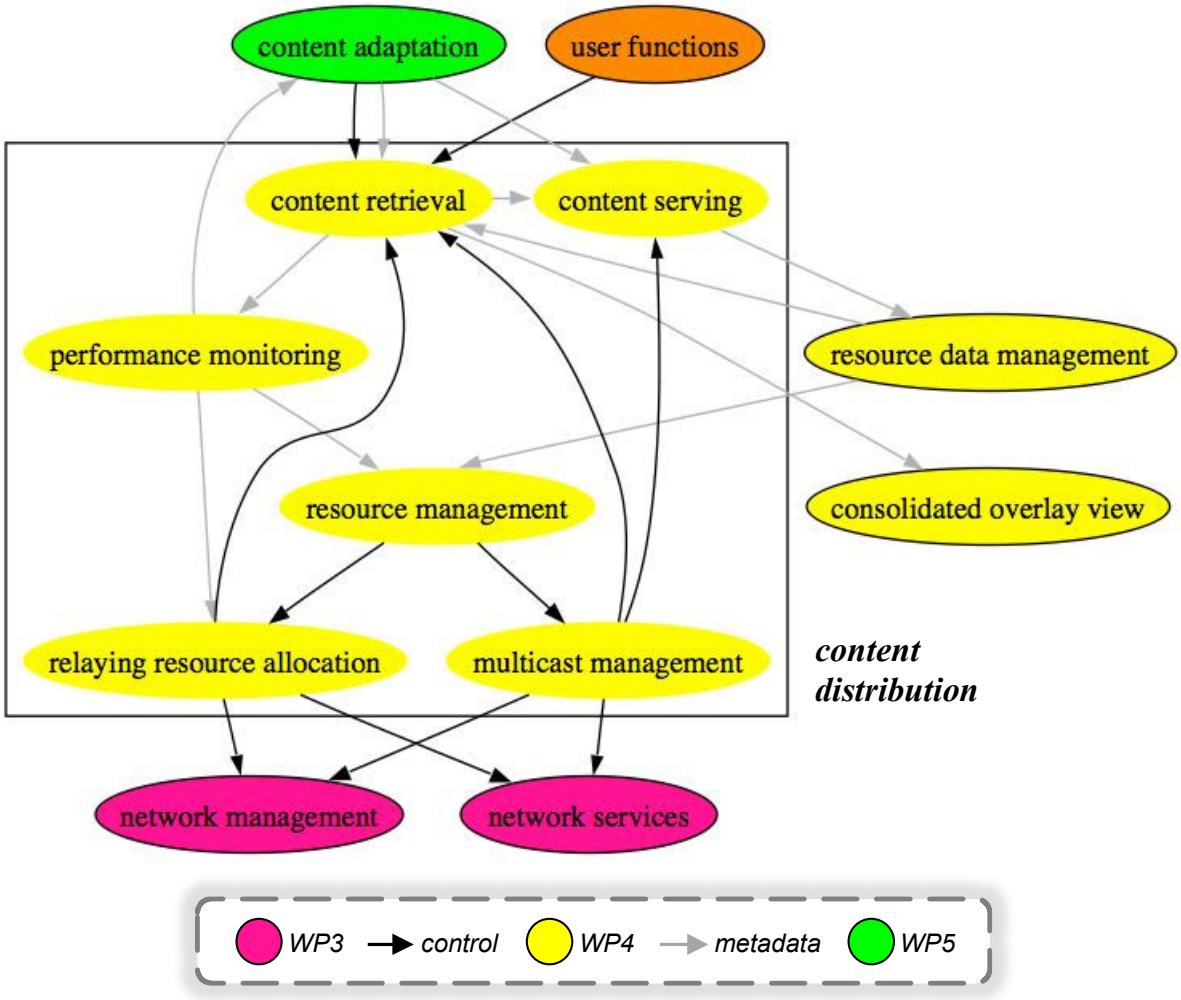


Figure 5 – Content Distribution Functions

Metadata Interactions:

- The content retrieval function informs the content serving function with the required metadata for sending the stream (e.g. chunk availability).

- The content serving function informs the resource data management function of the availability of data for retrieval from other nodes.
- The content retrieval function receives data from the resource data management function regarding the nodes that participate in the distribution of a particular stream.
- The content retrieval function provides feedback on network conditions to the consolidated overlay view, and feedback on the performance of the content distribution system to the performance monitoring function.
- The performance monitoring function provides feedback regarding the performance of the content distribution system to the resource management, relaying resource allocation and content adaptation functions.
- The resource management functions retrieves information regarding the overlay nodes and ISP network services that are available to be used, e.g. overlay and ISP relaying nodes.

4.3.1 Content Retrieval Function

The content retrieval function ensures that any overlay node receives the data it requires, either to be consumed by the user that operates the node, or to be processed and re-transmitted to other nodes when the node implements a content adaptation or a content relaying function.

A user who is interested in consuming a particular content object will invoke the content retrieval function to retrieve it. Content adaptation functions will be responsible for determining which resolution and quality level is adequate for the particular user and the current network conditions, and instruct the content retrieval function to retrieve the appropriate content format or set of layers in the case of scalable encoding formats.

Overlay functions for processing and relaying data may be implemented at provider-provisioned overlay nodes, as well as overlay nodes operated by the users themselves. These functions have specialised requirements for content retrieval. High capacity nodes used to relay content would only require, for example, to receive a fraction of the content object and upload it as many times as possible to other nodes.

The content retrieval function is responsible for discovering which nodes have the content objects of interest through the resource data management functions, which would return the nodes that have the data and that satisfy the network performance criteria set by the content retrieval function (e.g. those with the smallest network delay). Based on this information, the content retrieval function is responsible for selecting the best candidate sender nodes, and issue requests for establishing connections and receiving content data. In the case of dynamic systems, it is expected that the content retrieval function does not have perfect information when selecting the senders, and it may therefore select senders that have already exhausted their capacity serving other receivers. The maintenance of a set of connections through which the content can be received reliably, with resilience to churn and at the best possible quality is the main role of the content retrieval function. Requirements for the content retrieval function can be found in section 4.1.2.

4.3.2 Content Serving Function

The content serving function is responsible for managing the upload capacity resources at a particular overlay node or group of nodes. The content serving function receives updates about the content available at a node by the content retrieval function, and updates the resource data management function accordingly when, for example, a new content object is being received, or when the content time range available at the node shifts and the stream is available at an earlier or later playout point (in the case of live content). Given the set of requests received by content retrieval functions, the content serving function decides how many and which connections to establish between which nodes, and how to allocate the bandwidth among these connections. While

the content retrieval logic optimises the performance for receiving the content for the user, the content serving logic would favour the connections that maximise the performance of the overlay as a whole. As an example, the content serving function at the source node would accept to establish connections with the nodes that have the maximum capacity and the lowest delay, so that they can then distribute the low playout lag stream to more nodes at the overlay and increase the overall performance.

4.3.3 Performance Monitoring Function

The performance monitoring function is responsible for maintaining an estimation of the performance achieved by the content retrieval functions for the nodes that correspond to content consuming users across the entire overlay, or to groups of nodes at particular locations or connected through particular ISPs. This information is required for the management of the overlay, and in particular by content adaptation and resource management functions.

The performance monitoring function receives updates from the content retrieval function regarding the achieved video stream performance in terms of stream-specific metrics such as playout continuity, resolution, playout lag (see also requirements in section 4.1.2). These raw metrics are then summarised into appropriate statistical representations for the specified groups of nodes, and when significant changes occur on the aggregate values, notifications could be sent to the appropriate functions. When quality drops at a particular area due to low availability in upload bandwidth, for example, resource management functions could be triggered to invoke high-capacity nodes in the area. Alternatively, the content adaptation function may decide to lower the bitrate and the quality received by the nodes in this area.

Note that the performance monitoring function is different from the passive and active monitoring functions documented in sections 2.2.1.1 and 2.2.1.2 respectively, as it is not concerned with the monitoring of network performance metrics over an overlay link between two nodes, but rather, with the monitoring of the application performance at groups of content consuming nodes.

4.3.4 Resource Management Function

The resource management function is responsible for ensuring that there are sufficient storage, processing and bandwidth resources available at the overlay layer to perform the required content distribution and other supporting functions. As the size of the overlay changes with the dynamic arrival and departure of users and their corresponding overlay nodes, this function is responsible for invoking resources when and where appropriate to ensure a minimum level of performance for the users and does not incur unjustified costs for the application.

The resources that are available to the overlay include application provider provisioned resources, ISP and third-party provisioned resources, ISP network services, and the resources of the users, which may actively participate in the application and the distribution of the content they produce or consume. In this particular version of the functional model, we have modelled the invocation of two types of resources: upload bandwidth or relaying resources, and the multicast network service. These are considered at this point the most prominent to further investigate; more details for the corresponding functions can be found in the following sections.

4.3.5 Relaying Resource Allocation Function

The invocation of a new high-capacity node at an ISP, or of a set of idle user resources to act as replication points at a particular region, can be potentially valid for longer than the lifetime of the particular content objects that triggered the allocation of additional resources. This stems from the assumption that application users can join a number of different video streams during a session, while their media display capabilities or their average bitrate may not change significantly over time.

Therefore, there is a need for a function that decides how to best allocate a given pool of relaying resources to the distribution of the video streams (content objects) that require additional upload capacity. This is performed by the relaying resource allocation function, that takes the resource pool allocated by the resource management function and information from the performance monitoring function as inputs, and selects the content object to be relayed on this basis. When a decision is made instructing an overlay node to start or stop relaying a particular content object, the relaying resource allocation function interacts with the content retrieval function responsible for this node to implement this decision. Finally, although the decision to allocate additional overlay resources is made by the resource management function, the actual allocation of these additional resources is the responsibility of the relaying resource allocation function, which is also responsible for undertaking the corresponding interactions with the network management and network services functions at the ISP through the CINA interface.

4.3.6 Multicast Management Function

The multicast management function is responsible for invoking the multicast service offered by an ISP through the CINA interface, selecting and controlling the nodes that will send and receive data through multicast, and ensuring a smooth transition between the unicast and multicast overlay connections. Details regarding the research challenges of hybrid multicast overlays and their configuration can be found in [D3.1].

The multicast management function controls the content retrieval function of the overlay nodes that require data which are available from an established multicast tree. The content retrieval function should in that case discontinue the overlay unicast mode of operation in order to start receiving data through the multicast tree. Finally, the multicast management function controls the content serving function of the node(s) selected to become the sources for the data sent through a multicast tree, and ensures that backup nodes are immediately re-allocated in case of failures. This ensures that the impact to all the nodes receiving through multicast is minimised.

4.3.7 Multi-Link Considerations

Unlike the client-server model where the user nodes maintain a single connection to the application server, overlays and peer-to-peer systems rely on establishing multiple connections between nodes for data distribution. The topology construction logic, that decides which connections should be established, typically takes into account the resources of the nodes and the network performance between them. However, in case of nodes with multiple Internet access links through a number of different access networks with different network performance characteristics, the decision to create these connections and to allocate the traffic between them becomes more complicated as it needs to take into account the network performance to any other nodes through any of the available links. The following sections elaborate on some of the issues related with multi-link enabled peers (MLEPs) and further details can be found in [D5.1].

4.3.7.1 MLEP

The multilink enabled peer (MLEP) refers to a peer that has multiple network interfaces that could be used simultaneously for delivering P2P traffic. When such a peer is the source of live video/audio delivery, it can perform scheduling algorithms over the interfaces to make sure the delay is minimised, or, that more important data are delivered with greater priority, thus increasing the quality of the swarm for other peers. Content delivery from MLEP to MLEP through different access networks and ISP domains can be used when high quality data is required to be delivered from mobile locations through wireless networks as described in the bicycle use case in [D2.1] and the multi-link enabled peers section in [D5.1].

However, techniques developed for communication between two MLEP may also be useful for other cases where multiple communication paths exist. In many of these cases, however, devices will have a single network interface. This is discussed in the following sections where the MLAP is defined.

4.3.7.2 MLAP

The multilink aware peer (MLAP) refers to a peer that, through the overlay logic, is made aware of multiple communication paths and is able to distribute its traffic between them and set specific priorities to content units sent through these different connections. The MLAP itself does not need to have capabilities to distribute the content through different interfaces, it shall however consolidate the requirements of the application and service with the availability and costs of the delivery chains through multiple domains, to improve the quality the end user receives.

4.3.7.3 Cross layers Multi-Link scheduling in P2P swarms

Based on the consolidated information gathered as defined in section 2, the MLEP will perform advanced streaming techniques. The discovery of the different interfaces and links through the CINA interface will enable peers to make use of sophisticated scheduling algorithms which will take into consideration both the requirements of the upper layers for a given media quality and the requirement from the ISPs to localize the swarms and obey to the ISP recommendations for peer selection. Moreover, the upper layers may wish to distribute content units that have several importance and protection levels. The consideration of this cross layers optimization problem remains to be addressed in work to come.

5. OVERLAY RESOURCE OPTIMISATION AND CONTENT DISTRIBUTION FOR INTERACTIVE VIDEO

5.1 Problem Statement

The Interactive Video Content Distribution (IVCD) algorithm introduced in this section allows the system to create an overlay topology specifically for the distribution of interactive HD AV content across dynamic many-to-many groups of users. Therefore routing and scheduling algorithms create a distribution topology which is optimised for the transmission of live media streams that are typical for an interactive conferencing application where a user group exchanges media data in real time. To achieve an immersive interactive user experience the focus of the content distribution algorithm is to minimize the end-to-end latency of the media streams between the users. Therefore the topology will be built according to the network information gained through the CINA interface and additionally network services will be placed and invoked where suitable.

5.1.1 Description

The main function of the Interactive Video Content Distribution algorithm will be to distribute live media streams amongst users that are part of an interactive AV session. The IVCD will offer applications that implement such a service the opportunity to efficiently distribute content by creating an overlay tree topology between nodes that are part of the media session. Hereby the system abstracts the content distribution functionalities on the network level for the application, which only needs to provide or receive the media data of the media session from internal buffers. In this manner, the application developer does not have to deal about the application requirements on the network level.

The IVCD communicates with several functional blocks in the overall system. The search functionality will be used to register available media streams and the respective source node that manages the distribution topology of those streams. Nodes that want to receive a particular media stream will thus use the search function to find these source nodes in order to retrieve those peers that can be contacted to join the distribution tree.

In order to create an efficient overlay topology the node that manages the distribution tree will further invoke the functionality which consolidates network information received through the CINA interface or through end-to-end measurements between network nodes. For example the information is used to decide where in the distribution tree a node should join the network. Another example is the invocation of network services by the IVCD: Nodes that are in the same native multicast domain will be grouped together in the overlay topology. This will allow switching to native multicast data distribution if needed.

5.1.2 Requirements

5.1.2.1 Functional Requirements

- The system has to allow applications the instantiation of media stream distribution trees.
- The system has to support the distribution of simultaneous media streams, typically at least one per user.
- The system has to support differently encoded video and audio formats which are not known a priori to the overlay.
- The system will optimise content distribution scheduling according to content format specific prioritisation where applicable.

- The system should allow applications to specify minimum distribution requirements in terms of latency and bandwidth that should be met by the system.
- The system should provide an indication whether it is able to distribute a media stream according to the application requirements and notify the application if in case this condition changes.
- The solution must be able to integrate dedicated network resources and services like high capacity nodes or native multicast where these services are needed. These services are regarded as optional improvement; the solution must not rely on them. The services should be integrated in case the solution exceeds requirement threshold parameters as defined by the application.

5.1.2.2 Performance Requirements

- The overlay has to scale up to telepresence and web conferencing typical amounts of users (2 to 20).
- The overlay has to provide short latency to support a good QoE of the interactive application. Typically the latency should not exceed a limit of 300 milliseconds.
- The overlay has to be able to deal with churn as peers may leave and join during the conference. However, it is assumed that users in general participate to the overlay until the end of a session.

5.2 State-of-the-art and Innovation

During the last years video streaming applications evolved to create the biggest fraction of today's Internet traffic [CISC10]. Due to the lack of efficient point-to-multipoint content distribution services, namely the deployment of native multicast services, application developers basically have different possibilities to distribute video content. One is to welcome users on portal websites as YouTube and to redirect the clients to caches of a server based content distribution infrastructure, as for example the Akamai CDN offering [NSS10], which distributes the stored video content. Another option is application-level multicast solutions which stream multimedia video and audio content from a source to a large set of end users. Here end hosts create an overlay application, for example a peer-to-peer (P2P) network being a self-organized network of unicast connections across participating overlay nodes.

These content distribution systems are categorized according to the created overlay topology. In general it is distinguished between:

- single tree-based approaches, such as ZIGZAG [THD03] or NICE [BBK02]. Here the nodes are organized in a single tree structure where content is forwarded from the root of the tree towards the leaf nodes;
- mesh-based approaches, such as CoolStreaming [ZLL05], Gridmedia [ZZT05] or PRIME [MaRe07], where each node establishes overlay connections to random neighbour nodes. Content distribution trees are then generated on top of this mesh in a way each content chunks is sent over its own tree;
- forest-based approaches, such as CoopNet [PWC02] or SplitStream [CDK03], where the distributed content is split among different trees formed by the overlay. Here nodes participate in each tree in a way, that leave nodes in one tree are interior nodes in another tree. By this the distribution load is split in a relatively equal way between all participants.

All described approaches create overlay topologies that are built up without knowledge of the underlying network. Thus nodes that are neighbours within the overlay are usually not neighbours in terms of the underlying access network. Thus data might travel unnecessarily long distances, for example content is loaded from a peer located on a different continent, whereas actually the content would be available at a peer close by. One main goal of the approach presented here is to overcome

this limitation by leveraging information gained through the CINA interface. In order to minimize the end-to-end latency of the content distribution algorithm overlay nodes are grouped together which are neighbours in the underlying network.

While mesh-based approaches are usually considered to be more resilient compared to tree-based approaches, the distribution of content here follows random connections between nodes. Chunks that belong to the same content are distributed over separate tree structures on top of the mesh. Optimizing the multitude of tree structures according to the underlying network would result in equal tree topologies and thus contradict the resilience provided by the random connections. This is similar to forest-based approaches, where the nodes of the different trees are in different roles (intermediate node – leave node) in order to distribute load equally. By this definition it is impossible to organize two trees of the same stream at the same time according to the underlying network, as the optimizing one tree would automatically affect the other tree. For these reasons the design choice for the IVCD is to use a single tree-based approach. The major drawback of single tree-based approaches is the weakness in terms of resilience. However due to the fact that we expect a low churn of the participating nodes this weakness seems to be negligible.

Another categorization of the overlay applications is made in terms of the scheduling algorithms that are used to distribute data:

- Push-based algorithms are typically used in tree-based approaches, such as ZIGZAG [THD03] or NICE [BBK02]. Here the parent node decides which content chunks it pushes next to its child node.
- Pull-based algorithms are used for example in CoolStreaming [ZLL05] or PRIME [MaRe07]. Here the receiving node requests the next content item from one of its neighbouring nodes.
- Push-pull algorithms are a hybrid version, used for example in [ZZT05], where chunks are requested in one phase of the content distribution (pull) and relayed directly in another phase of the content distribution (push).

The IVCD system will be optimized for interactive video distribution. One critical requirement for this type of application is a minimized latency in order to achieve an immersive QoE for the user. For this reason the scheduling algorithm will be push-based, in order to avoid the extra RTT of a client requesting specific content items.

5.3 High-level Specifications

This section provides an overview on the IVCD system discusses the design choices and explains the basic operation. Further it introduces the high level software architecture of the overlay nodes and maps it to the overall ENVISION architecture.

5.3.1 Single Streaming Tree

The IVCD is a subsystem creates a content distribution topology for streaming data with a minimal end-to-end delay. Figure 6 gives an overview on the single tree distribution topology, showing the streaming order from the source over intermediary towards leave nodes. Source and intermediary nodes relay the streaming data to their child nodes, whereas the more capabilities a node has, the more child nodes it supplies. Content is ingested only by the source peer. Typically the source peer of the IVCD subsystem is instantiated by a node of an overlay application that wants to use the system, or the source peer created by a network node which assists a client with low capabilities (e.g. in terms of bandwidth) in distributing the content of a client. Besides the function that establishes and maintains the overlay topology, which is common for all peers, the source peer additionally provides a function called ‘resource Manager’ (RM). The RM is a central instance which manages the overlay topology. An additional external search function in the resource data management of the functional model is used to distribute a description of the content that is streamed over a particular tree and

the according contact point of the RM. This external search function is part of the global overlay, shared between all peers of the application and used as a global repository of available content. A user or an application may browse this search function in order to look for interesting streams. Once it has found a particular content it retrieves the contact data of the responsible RM from the search function and instantiates an IVCD peer by itself.

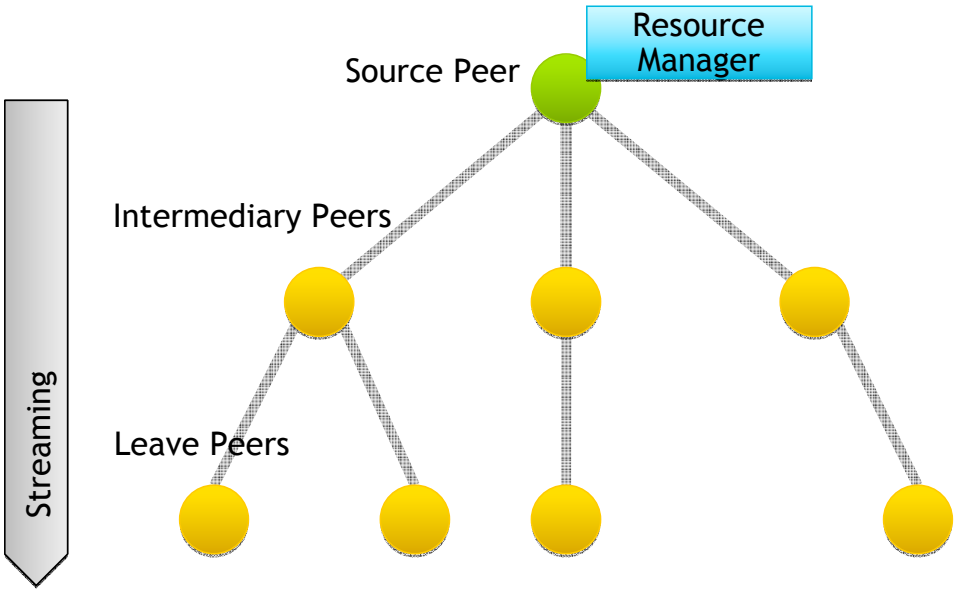


Figure 6 – IVCD Streaming Tree

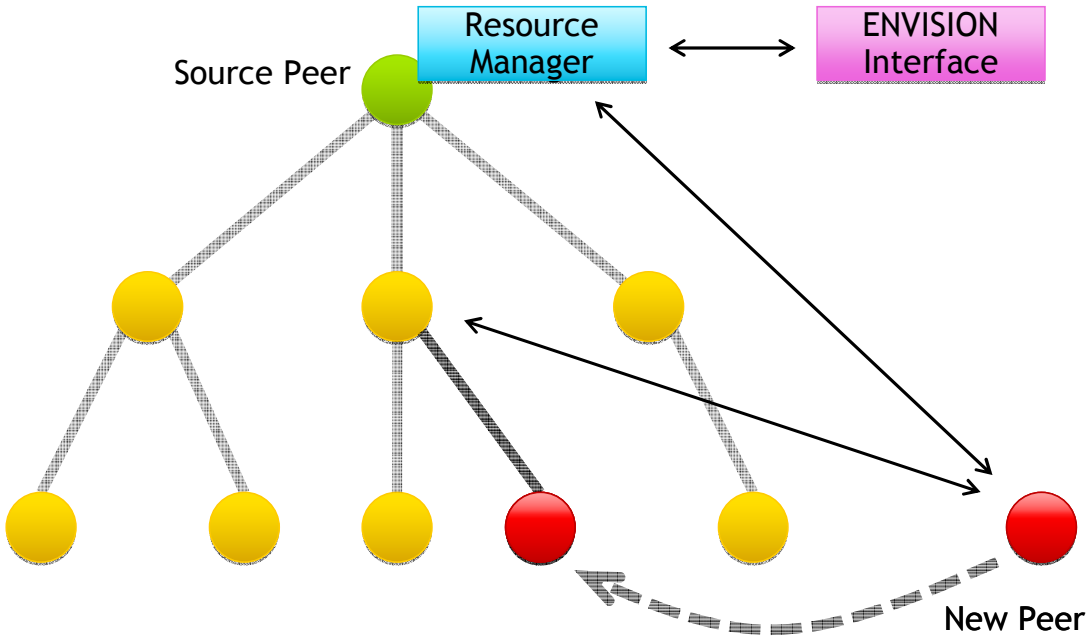


Figure 7 – Join procedure

Figure 7 illustrates this new peer that wants to join the network. The peer first contacts and registers itself at the RM. The RM now advices a free slot within the tree to the peer and provides it with the contact details of its future parent peer. The RM hereby runs an algorithm that takes different metrics into account, such as the available capacity of the respective peers of the system. For example some peers may have a high uplink streaming capacity; some only a low or none at all (e.g.

mobile peers). Some of the used metrics are hereby provided by the peers themselves, but additionally the RM may contact the CINA interface to gain additional knowledge. This additional knowledge can for example be used to group peers that are located in the same access network or Autonomous System together to keep traffic local and latencies small. After the joining peer receives the contact point of its parent peer it establishes a direct connection to it and the parent peer starts to stream the content. After the connection is set up the peers within a neighbourhood maintain a heartbeat that is used to detect node failures. Thus, in case a peer leaves the streaming tree, gracefully or ungracefully, the RM can be informed about the topology change and the RM's view is kept up-to-date.

5.3.2 Architecture overview

In this section the high-level architecture of an IVCD node is introduced. Figure 8 shows the different functional modules of the software. The central controlling instance is the PeerControl function, which manages the scheduling algorithms and the maintenance of the overlay topology. It hereby comprises a view on the neighbourhood of the peer, which contains information about the parent node as well as potential child nodes and also recovery nodes that help to compensate the failure of the parent node. Also the RM function is located in the PeerControl; however only if the node is the source of a content stream the RM function is instantiated. PeerHeartbeat and PeerSignalingSystem are minor support modules for the PeerControl, responsible to create the periodic heartbeat and message handling.

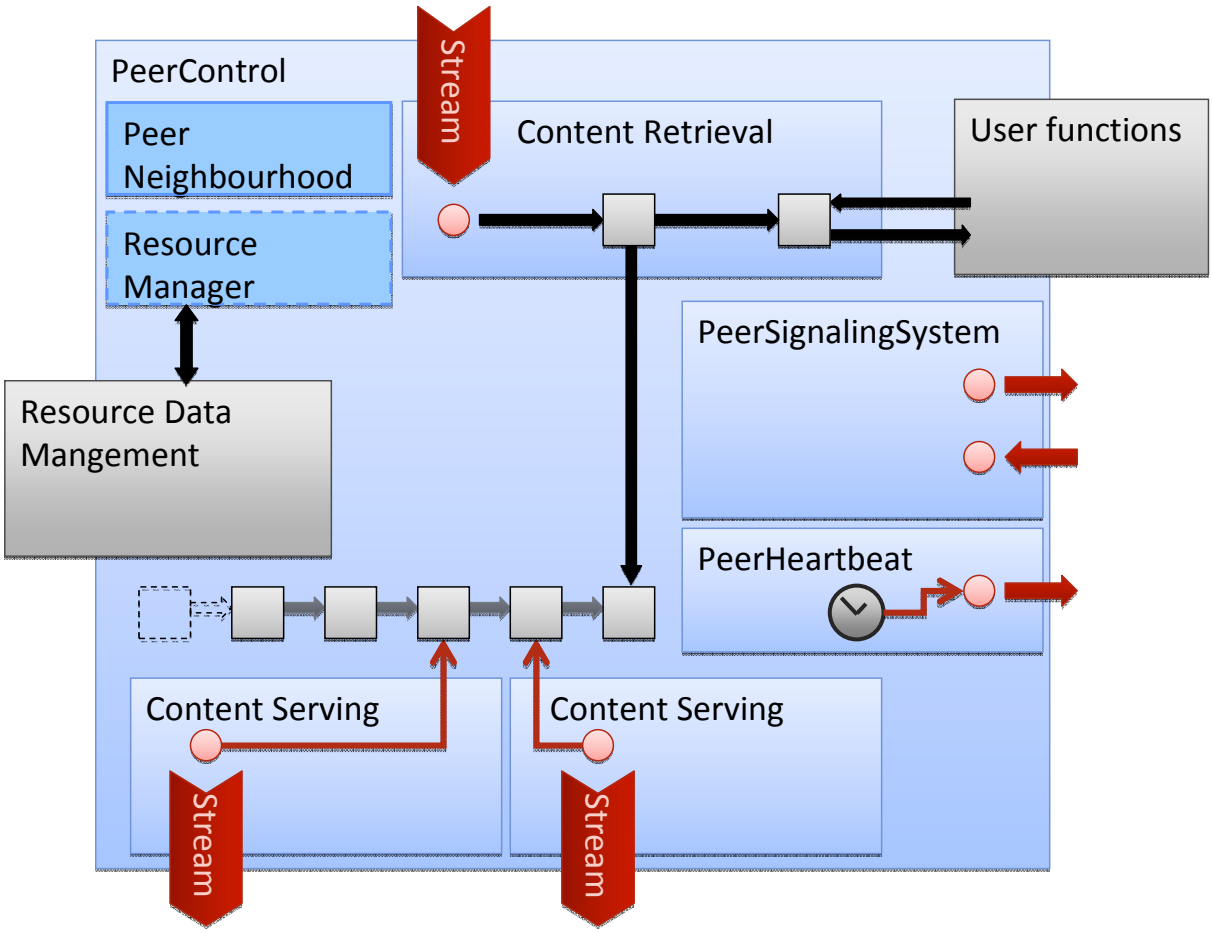


Figure 8 – Architecture Overview

A streaming tree node further comprises several modules that receive and send the streaming data. The Content Retrieval function receives the data from the parent node, or, if it is the source node, from the streaming device, for example a camera. Incoming data blocks are forwarded to the

PeerControl unit, where they are put into a buffer queue and supplied to the Content Serving functions. For each child node one Content Serving function is instantiated, which retrieve data from the data buffer in order to redistribute it. Additionally, incoming data is copied into a buffer provided by the application, if existent, to allow the application usage of the data, for example media playback to the user.

All of the blue modules of the node architecture are comprised in block 5 'ENVISION Overlay Management' of the overall architecture. The RM module uses the interfaces C5 and M5 as well as C0 and M0 to communicate with the Resource Data Management to gain knowledge on the peers that want to join the tree topology as described in section 5.3.1. Finally the user functions (grey module of Figure 8) are connected to the IVCD software through the interfaces C2 and M2.

6. CONCLUSION

This document elaborates on the functionality that is required to perform network-aware content distribution with cross-layer optimisation between the network and the application layers using the capabilities of the CINA interface. In the context of future networked media applications, where the users interact, generate and consume content through the application and the demand for content cannot be accurately predicted both in terms of volume and location, flexible and scalable solutions are required that are able to dynamically mobilise available resources, including participant and provider-provisioned resources and network services, combining them to form the application overlay.

While the CINA interface exposes the capabilities of a single ISP, the application nodes are potentially located over several ISPs, and overlay links may traverse several ISP domains. Based on the local network performance information provided by each ISP, the ISP preferences and end-to-end passive and active measurements gathered by the application at the overlay layer, the *consolidated overlay view* functions create and maintain a view of the overlay nodes and the end-to-end overlay links that is used to make the content distribution and content adaptation functions network-aware, and to allow the cross-layer optimisation with a consistent end-to-end view of the network. The consolidation of measurements of different granularity and accuracy, and the consolidation of the ISP preferences potentially using proprietary algorithms for ranking and weighting the overlay links is the main research challenge, and section 2 elaborates on the related issues.

Large scale applications need to scale sufficiently not only in terms of the resources required for content distribution, but also in terms of their supporting functions and the information that needs to be exchanged between them in a scalable and timely manner. The *resource data management* functions create and maintain a separate control overlay for storing and retrieving metadata about the application resources, including the active participant nodes, content objects, interests of users to content objects, and resources offered by the ISPs. Resource registration and discovery in the application need to support a high volume of dynamic updates and queries issued by the distributed functions across the application overlay, and should return optimum results tailored to the query taking into account application and network performance criteria. Section 3 elaborates on the existing approaches and the research issues that are of concern in ENVISION.

Based on the consolidated overlay view and relying on efficient resource discovery, the *resource optimisation* and *content distribution* functions provide the logic for creating the overlay topology and ensuring that the content can be produced and injected into the system dynamically, distributed with the highest bitrate and the best level of quality that the network and application resources can support and consumed by all the users that express interest for it, without incurring excessive unnecessary costs for the application or the network providers. Overlay topologies cannot be universally optimised as the performance requirements at the application layer have a major impact on the choices for interconnecting nodes. An example is the case of interactive multi-participant video where the connections between the video source and the consumer can only tolerate up to 300 milliseconds of playout delay, imposing a very hard restriction on the number of overlay links between any two nodes that need to receive this content. In this document we have focused in the research challenges involved in content distribution for multi-participant live and interactive video, and sections 4 and 5 respectively elaborate on the related issues.

This document concludes with the high-level specifications of these functions. The work in WP4 will continue with the detailed specification of the protocols and optimisation algorithms identified here, the compilation of an implementation plan for the functionality that needs to be developed and further evaluated in WP6, and a preliminary software release at M18 of the project (June 2011), while the specifications of the developed protocols and algorithms and the refined functional specifications will be documented in D4.2 at M24 (December 2011).

7. REFERENCES

[AAB05] Daniel J. Abadi, Yanif Ahmad, Magdalena Balazinska, Ugur Cetintemel, Mitch Cherniack, Jeong H. Hwang, Wolfgang Lindner, Anurag S. Maskey, Alexander Rasin, Esther Ryvkina, Nesime Tatbul, Ying Xing, and Stan Zdonik. The Design of the Borealis Stream Processing Engine. In 2nd Biennial Conference on Innovative Data Systems Research (CIDR'05), pages 277–289, 2005.

[AAF08] Vinay Aggarwal, Obi Akonjang, and Anja Feldmann. Improving User and ISP Experience through ISP-aided P2P Locality. In Proc. of the Global Internet Symposium, 2008.

[ABK03] Mehmet Altinel, Christof Bornhövd, Sailesh Krishnamurthy, C. Mohan, Hamid Pirahesh, and Berthold Reinwald. Cache tables: paving the way for an adaptive database cache. In Proceedings of the 29th international conference on Very large data bases - Volume 29, VLDB '2003, pages 718–729. VLDB Endowment, 2003.

[AFJ08] Christina Aperjis, Michael J. Freedman, and Ramesh Johari. Peer-assisted content distribution with prices. In Proceedings of the 2008 ACM CoNEXT Conference, pages 17:1–17:12, USA, 2008. ACM.

[AFS07] Vinay Aggarwal, Anja Feldmann, and Christian Scheideler. Can ISPs and P2P users cooperate for improved performance? SIGCOMM Comput. Commun. Rev., 37:29–40, July 2007.

[AH00] E. Adar and B. Huberman. Free riding on Gnutella. Technical report, Xerox PARC, August 2000.

[Arr70] Kenneth J. Arrow. Social Choice and Individual Values. Cowles Foundation Monographs Series. Yale University Press, September 1970.

[AS03] J. Aspnes and G. Shah. Skip graphs. In Proceedings of Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pages 384-393, 2003

[ASH04] E.S. Al-Shaer and H.H. Hamed. Discovery of policy anomalies in distributed firewalls. In Proceedings of INFOCOM 2004, volume 4, pages 2605 - 2616 vol.4, March 2004.

[BAS04] A. R. Bharambe, M. Agrawal, and S. Seshan, Mercury: Supporting scalable multi-attribute range queries, in ACM SIGCOMM, Portland, OR, August-September 2004.

[BAZS97] S. Bhattacharjee, MH Ammar, EW Zegura, and V. Shah. Application-layer anycasting. Proceedings of INFOCOM '97, 3, 1997.

[BBG02] Philip A. Bernstein, Fausto Giunchiglia, Anastasios Kementsietsidis, John Mylopoulos, Luciano Serafini, and Ilya Zaihrayeu. Data management for peer-to-peer computing: A vision. In Proc. Workshop on the Web and Databases (WebDB'02), pages 89–94, 2002.

[BBK02] S. Banerjee, B. Bhattacharjee, C. Kommareddy, Scalable application layer multicast, in Proceedings of ACM SIGCOMM, Pittsburgh, PA, USA, 2002

[BCC+06] R. Bindal, P. Cao, W. Chan, J. Medved, G. Suwala, T. Bates, and A. Zhang. Improving traffic locality in bittorrent via biased neighbor selection. Distributed Computing Systems, 2006. ICDCS 2006. 26th IEEE International Conference on, pages 66–66, 2006.

[BCC06] Ruchir Bindal, Pei Cao, William Chan, Jan Medved, George Suwala, Tony Bates, and Amy Zhang. Improving traffic locality in BitTorrent via biased neighbor selection. In ICDCS '06: Proceedings of the 26th IEEE International Conference on Distributed Computing Systems, page 66, Washington, DC, USA, 2006. IEEE Computer Society.

[BF05] H. Ballani and P. Francis. Towards a global IP anycast service. In Procceedings of SIGCOMM '05, August 2005.

[BkS04] F. Banaei-Kashani and C. Shahabi, SWAM: A family of access methods for similarity-search in peer-to-peer data networks, in ACM International Conference on Information and Knowledge management, Washington, DC, November 2004.

[BLD10] Stevens Le Blond, Arnaud Legout, and Walid Dabbous. Pushing bittorrent locality to the limit. Computer Networks (to appear), 2010.

[BP08] Nicholas Ball and Peter Pietzuch. Distributed content delivery using load-aware network coordinates. In Proceedings of the 2008 ACM CoNEXT Conference, CoNEXT '08, pages 77:1-77:6, New York, NY, USA, 2008. ACM.

[BPS06] A. Bharambe, J. Pang, S. Seshan, Colyseus: A distributed architecture for interactive multiplayer games. NSDI 2006.

[BV09] Ph. Blanchard and D. Volchenkov. Probabilistic embedding of discrete sets as continuous metric spaces. Stochastics, 81(3):259 - 268, November 2009.

[CB08] David R. Choffnes and Fabián E. Bustamante. Taming the torrent: a practical approach to reducing cross-ISP traffic in peer-to-peer systems. In Proceedings of the ACM SIGCOMM 2008 conference on Data communication, SIGCOMM '08, pages 363–374, USA, 2008. ACM.

[CBB03] Mitch Cherniack, Hari Balakrishnan, Magdalena Balazinska, Donald Carney, Ugur Çetintemel, Ying Xing, and Stanley B. Zdonik. Scalable distributed stream processing. In In Proc. 1st Conference on Innovative Data Systems Research (CIDR), 2003.

[CCL+04] Rui Castro, Mark Coates, Gang Liang, Robert Nowak, and Bin Yu. Network tomography: Recent developments. Statistical Science, 19(3):pp. 499-517, 2004.

[CCN+02] Mark Coates, Rui Castro, Robert Nowak, Manik Gadhiok, Ryan King, and Yolanda Tsang. Maximum likelihood network topology identification from edge-based unicast measurements. SIGMETRICS Perform. Eval. Rev., 30:11-20, June 2002.

[CDGS07] Manuel Crotti, Maurizio Dusi, Francesco Gringoli, and Luca Salgarelli. Traffic classification through simple statistical fingerprinting. SIGCOMM Comput. Commun. Rev., 37:5-16, January 2007.

[CDHR02] Miguel Castro, Peter Druschel, Y. Charlie Hu, and Antony Rowstron. Exploiting network proximity in distributed hash tables. In Ozalp Babaoglu, Ken Birman, and Keith Marzullo, editors, International Workshop on Future Directions in Distributed Computing (FuDiCo), pages 52–55, June 2002.

[CDK03] M. Castro, P. Druschel, A. Kermarrec, A. Nandi, A. Rowstron, A. Singh, Split-Stream: high-bandwidth multicast in cooperative environments, in Proceedings of the 19th ACM Symposium on Operating Systems Principles (SOSP'03), The Sagamore, Bolton Landing, NY, USA, 2003

[CDKR03] Miguel Castro, Peter Druschel, Anne-Marie Kermarrec, and Antony I. T. Rowstron. Scalable application-level anycast for highly dynamic groups. In Networked Group Communication, pages 47–57, 2003.

[CEMP10] N. Capovilla, M. Eberhard, S. Mignanti, R. Petrocco, and J. Vehkaperä, "An Architecture for Distributing Scalable Content over Peer-to-Peer Networks", Proceedings of the 2010 Second International Conferences on Advances in Multimedia (MMEDIA'10), Athens, Greece, June 2010.

[CFK03] E. Cohen, A. Fiat, and H. Kaplan. Associative search in Peer-to-Peer networks: Harnessing latent semantics. In Proceedings of IEEE INFOCOM, 2003.

[CFP+09] M. Charalambides, P. Flegkas, G. Pavlou, J. Rubio-Loyola, A.K. Bandara, E.C. Lupu, A. Russo, N. Dulay, and M. Sloman. Policy conflict analysis for diffserv quality of service management. Network and Service Management, IEEE Transactions on, 6(1):15 -30, March 2009.

[CGH+10] D. Ciullo, M.A. Garcia, A. Horvath, E. Leonardi, M. Mellia, D. Rossi, M. Telek, and P. Veglia. Network awareness of p2p live streaming applications: A measurement study. Multimedia, IEEE Transactions on, 12(1):54-63, January 2010.

[CISC10] Cisco Visual Networking Index 2010, http://www.cisco.com/go/vni

[CJK+03] M. Castro, M. B. Jones, A.-M. Kermarrec, A. Rowstron, M. Theimer, H. Wang, and A. Wolman. An evaluation of scalable application-level multicast built using peer-to-peer overlays. In Infocom'03, April 2003.

[CJW08] H. Chen, H. Jin, J. Wang, L. Chen, Y. Liu, and L. M. Ni. Efficient multi-keyword search over P2P Web. In WWW'08: Proceeding of the 17th International World Wide Web conference, Beijing, China, 2008.

[CLMM08] Ana Paula Couto da Silva, Emilio Leonardi, Marco Mellia, Michela Meo, "A Bandwidth-Aware Scheduling Strategy for P2P-TV Systems," p2p, pp.279-288, 2008 Eighth International Conference on Peer-to-Peer Computing, 2008.

[CN03] Yi Cui and Klara Nahrstedt. Layered peer-to-peer streaming. In Proceedings of NOSSDAV 2003, pages 162 171, New York, NY, USA, 2003. ACM.

[CRB03] Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker. Making Gnutella-like P2P systems scalable. In Proceedings of ACM SIGCOMM, pages 407-418, 2003.

[CSS06] Peter Cramton, Yoav Shoham, and Richard Steinberg. Combinatorial Auctions. MIT Press, 2006.

[CWWK06] Byung-Gon Chun, Peter Wu, Hakim Weatherspoon, and John Kubiatowicz. ChunkCast: An anycast service for large content distribution. In Proceedings of IPTPS '06, Santa Barbara, CA, February 2006.

[CYRK03] Casey Carter, Seung Yi, Prashant Ratanchandani, and Robin Kravets. Manycast: exploring the space between anycast and multicast in ad hoc networks. In MobiCom '03: Proceedings of the 9th annual international conference on Mobile computing and networking, pages 273–285, New York, NY, USA, 2003. ACM.

[D2.1] ENVISION deliverable D2.1, Final Specification of Use Cases, Business Models and the System Architecture, December 2010, FP7 ICT ENVISION project, www.envision-project.org

[D3.1] ENVISION deliverable D3.1, Initial Specification of the ENVISION Interface, Network Monitoring and Network Optimisation Functions, December 2010, FP7 ICT ENVISION project, www.envision-project.org

[D5.1] ENVISION deliverable D5.1, Initial Specification of the Metadata Management, Dynamic Content Generation and Adaptation, Adaptation and Caching Node Functions, December 2010, FP7 ICT ENVISION project, www.envision-project.org

[DCKM04] Frank Dabek, Russ Cox, Frans Kaashoek, and Robert Morris. Vivaldi: a decentralized network coordinate system. SIGCOMM Comput. Commun. Rev., 34(4):15-26, 2004.

[DMG+10] Marcel Dischinger, Massimiliano Marcon, Saikat Guha, Krishna P. Gummadi, Ratul Mahajan, and Stefan Saroiu. Glasnost: Enabling End Users to Detect Traffic Differentiation. In Proceedings of NSDI, 2010.

[EMTS10] Towards the Future Internet - Emerging Trends from European Research. Ed. by Georgios Tselentis, Alex Galis, Anastasius Gavras, Srdjan Krco, Volkmar Lotz, Elena Simperl, Burkhard Stiller, Theodore Zahariadis. IOS Press, 2010.

[FBZA98] Z. Fei, S. Bhattacharjee, E. W. Zegura, and M. H. Ammar. A novel server selection technique for improving the response time of a replicated service. In Proceedings of INFOCOM '98, pages 783–791, 1998.

[FKLS04] Enrico Franconi, Gabriel Kuper, Andrei Lopatenko, and Luciano Serafini. A robust logical and computational characterisation of peer-to-peer database systems. In Karl Aberer, Manolis Koubarakis, and Vana Kalogeraki, editors, Databases, Information Systems, and Peer-to-Peer Computing, volume 2944 of Lecture Notes in Computer Science, pages 64–76. Springer Berlin / Heidelberg, 2004.

[FLM06] Michael J. Freedman, Karthik Lakshminarayanan, and David Mazières. Oasis: anycast for any service. In Proceedings of NSDI '06, volume 3, Berkeley, CA, USA, 2006. USENIX Association.

[FLZ05] Michal Feldman, Kevin Lai, and Li Zhang. A Price-Anticipating Resource Allocation Mechanism for Distributed Shared Clusters. In Proc. ACM EC, June 2005.

[GHI01] Steven Gribble, Alon Halevy, Zachary Ives, Maya Rodrig, and Dan Suciu. What Can Databases do for Peer-to-Peer? In Proc. WEBDB'01, the 4th International Workshop on the Web and Databases, 2001.

[GJT04] Anders Gunnar, Mikael Johansson, and Thomas Telkamp. Traffic matrix estimation on a large IP backbone: a comparison on real data. In Proceedings of the 4th ACM SIGCOMM conference on Internet measurement, IMC '04, pages 149-160, New York, NY, USA, 2004. ACM.

[GKK03] P.B. Gibbons, B. Karp, Y. Ke, S. Nath, and Srinivasan Seshan. Irisnet: an architecture for a worldwide sensor web. Pervasive Computing, IEEE, 2(4):22 – 33, 2003.

[GNU10] The Gnutella website, http://www.gnutella.com.

[Goy01] Vivek K. Goyal. Multiple description coding: Compression meets the network. IEEE Signal Processing Magazine, 18(5):74-93, September 2001.

[GSG02] Krishna P. Gummadi, Stefan Saroiu, and Steven D. Gfibble, King: Estimating Latency between Arbitrary Internet End Hosts, IMW'02, Nov. 6-8, 2002, Marseille, France.

[GSS06] Rohit Gupta, Varun Sekhri, and Arun K. Somani. CompuP2P: An architecture for internet computing using peer-to-peer networks. IEEE TPDS, 17(11), 2006.

[GZ02] Fausto Giunchiglia and Ilya Zaihrayeu. Making peer databases interact - a vision for an architecture supporting data coordination. In Proceedings of the 6th International Workshop on Cooperative Information Agents VI, CIA '02, pages 18–35, London, UK, 2002. Springer-Verlag.

[HC06] Ahsan Habib and John C.-I. Chuang. Service differentiated peer selection: an incentive mechanism for peer-to-peer media streaming. IEEE Transactions on Multimedia, 8(3):610-621, 2006.

[HCW05] Daniel Hughes, Geoff Coulson, and James Walkerdine. Free riding on Gnutella revisited: The bell tolls? IEEE Distributed Systems Online, 6(6):1, 2005.

[HH08] Mohamed Hefeeda and Cheng-Hsin Hsu. Rate-distortion optimized streaming of fine-grained scalable video sequences. ACM Trans. Multimedia Comput. Commun. Appl., 4:2:1 2:28, February 2008.

[HHB+03] Mohamed Hefeeda, Ahsan Habib, Boyan Botev, Dongyan Xu, and Bharat Bhar- gava. Promise: peer-to-peer media streaming using collectcast. In MULTIME- DIA '03: Proceedings of the eleventh ACM international conference on Multi- media, pages 45-54, New York, NY, USA, 2003. ACM Press.

[HHL03] Ryan Huebsch, Joseph M. Hellerstein, Nick Lanham, Boon Thau Loo, Scott Shenker, and Ion Stoica. Querying the internet with pier. In Proceedings of the 29th international conference on Very large data bases - Volume 29, pages 321–332, 2003.

[HJS+03] N. Harvey, M. B. Jones, S. Saroiu, M. Theimer, and A. Wolman. SkipNet: A scalable overlay network with practical locality properties. In Proceedings of the USENIX Symposium on Internet Technologies and Systems (USITS), Seattle,March 2003.

[HL04] S.-Y. Hu, G.-M. Liao, Scalable Peer-to-Peer Networked Virtual Environment, SIGCOMM Workshop on Network and system support for games 2004

[HLL+07] Xiaojun Hei, Chao Liang, Jian Liang, Yong Liu, and K.W. Ross. A measurement study of a large-scale P2P IPTV system. Multimedia, IEEE Transactions on, 9(8):1672 1687, December 2007.

[HLR07] Xiaojun Hei, Yong Liu, and K.W. Ross. Inferring network-wide quality in P2P Live streaming systems. IEEE Journal on Selected Areas in Communications, vol. 25, no. , pp. 1640-1654, December 2007.

[HLW07] Yu-Chen Huang, Chun-Shien Lu and Hsiao-Kuang Wu, JitterPath: Probing Noise Resilient One-Way Delay Jitter-Based Available Bandwidth Estimation, IEEE Transactions on Multimedia, Vol. 9, No. 4. (2007), pp. 798-812.

[Hos02] Wolfgang Hoschek. A unified peer-to-peer database framework for scalable service and resource discovery. In Proceedings of the Third International Workshop on Grid Computing, GRID '02, pages 126–144, London, UK, 2002. Springer-Verlag.

[HR06] Leonid Hurwicz and Stanley Reiter. Designing Economic Mechanisms. Cambridge University Press, May 2006.

[KC05] George Kokkinidis and Vassilis Christophides. Semantic query routing and processing in p2p database systems: The ics-forth sqpeer middleware. In Wolfgang Lindner, Marco Mesiti, Can Türker, Yannis Tzitzikas, and Athena Vakali, editors, Current Trends in Database Technology - EDBT 2004 Workshops, volume 3268 of Lecture Notes in Computer Science, pages 433–436. Springer Berlin / Heidelberg, 2005.

[kLGZ04] Per Åke Larson, Jonathan Goldstein, and Jingren Zhou. Mtcache: Transparent mid-tier database caching in sql server. Data Engineering, International Conference on, 0:177, 2004.

[KLXH04] B. Knutsson, H. Lu, W. Xu, B. Hopkins, Peer-to-Peer Support for Massively Multiplayer Games, INFOCOM 2004.

[KSB01] C. Kommareddy, N. Shankar, and B. Bhattacharjee. Finding close friends on the internet. In IEEE ICNP, pages 301–309. Citeseer, 2001.

[KW00] Dina Katabi and John Wroclawski. A framework for scalable global IP-anycast (GIA). In Proccedings of SIGCOMM '00, pages 3–15, 2000.

[KXZ05] A. Kumar, J. Xu, and E. W. Zegura. Efficient and scalable query routing for unstructured peer-to-peer networks. In Proceedings of IEEE INFOCOM, 2005.

[LCD04] Anukool Lakhina, Mark Crovella, and Christophe Diot. Diagnosing network-wide traffic anomalies. In Proceedings of the SIGCOMM, pages 219-230. ACM, 2004.

[LGS07] Jonathan Ledlie, Paul Gardner, and Margo I. Seltzer. Network coordinates in the wild. In NSDI. USENIX, 2007.

[LLS03] M. Li, W. Lee, and A. Sivasubramaniam. Neighborhood signatures for searching P2P networks. In Proceedings of Seventh International Database Engineering and Applications Symposium (IDEAS), pages 149-159, 2003.

[LLS06] M. Li, W.-C. Lee, and A. Sivasubramaniam, DPTree: A balanced tree based indexing framework for peer-to-peer networks, in IEEE International Conference on Networking Protocols, Boston, MA, November 2006.

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

[LLSL04] M. Li, W.-C. Lee, A Sivasubramaniam, and D. L. Lee, A small world overlay network for semantic based search in p2p systems, in IEEE International Conference on Network Protocols, Berlin, Germany, October 2004.

[LPE02] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker. Search and replication in un- structured Peer-to-Peer networks. In Proceedings of the International Conference on Supercomputing (ICS), 2002.

[LS99] Emil C. Lupu and Morris Sloman. Conflicts in policy-based distributed systems management. IEEE Trans. Softw. Eng., 25:852-869, November 1999.

[LZ07] Eng Keong Lua and Xiaoming Zhou. Bos: Massive scale network-aware geometric overlay multicast streaming network. Global Telecommunications Conference, 2007. GLOBECOM ’07. IEEE, pages 253–258, Nov. 2007.

[LZX+07] Xuguang Lan, Nanning Zheng, Jianru Xue, Xiaoguang Wu, and Bin Gao. A peer-to-peer architecture for efficient live scalable media streaming on the Internet. In Proceedings of the 15th international conference on Multimedia, MULTIMEDIA '07, pages 783--786, New York, NY, USA, 2007. ACM.

[Maj04] Zoran Majkic. Weakly-coupled ontology integration of p2p database systems. In Proceedings of the MobiQuitous’04 Workshop on Peer-to-Peer Knowledge Management, Boston, MA, 2004.

[MaRe07] N.Magharei, R.Rejaie, PRIME: Peer-to-Peer Receiver-driven Mesh-based Streaming, in Proc. of IEEE INFOCOM, 2007

[MBP+09] J. J. D. Mol, A. Bakker, J. A. Pouwelse, D. H. J. Epema, and H. J. Sips. The design and deployment of a BitTorrent live video streaming solution. In Proceedings of the 2009 11th IEEE International Symposium on Multimedia, pages 342--349, 2009.

[MH09] K. Mokhtarian and M. Hefeeda. Efficient allocation of seed servers in peer-to-peer streaming systems with scalable videos. In Quality of Service, 2009. IWQoS. 17th International Workshop on, pages 1 9, July 2009.

[MLS+07] E. Mykoniati, R. Landa, S. Spirou, R. G. Clegg, L. Latif, D. Griffin, M. Rio, Scalable Peer-to-Peer Streaming for Live Entertainment Content, IEEE Communications, Feature Topic on Consumer Communications and Networking - Gaming and Entertainment, vol. 46, no. 12, pp. 40-46, IEEE, December 2008.

[MR06] Nazanin Magharei and Reza Rejaie. Adaptive receiver-driven streaming from multiple senders. Multimedia Systems, 11(18):550--567, June 2006.

[MRP99] Michael Minock, Marek Rusinkiewicz, and Brad Perry. The identification of missing information resources through the query difference operator. In Proceedings of the

Fourth IECIS International Conference on Cooperative Information Systems, COOPIS '99, pages 304–, Washington, DC, USA, 1999. IEEE Computer Society.

[MYK04] A. Mondal, Yilifu, and M. Kitsuregawa, P2PR-tree: An r-tree-based spatial index for peer-to-peer environments, in ICDE/EDBT PhD Workshop, Crete, Greece, 2004.

[MZPP08] Ratul Mahajan, Ming Zhang, Lindsey Poole, and Vivek Pai. Uncovering Performance Differences Among Backbone ISPs with Netdiff. In Conference on Network Systems Design & Implementation (NSDI), 2008.

[NAPA10] http://napa-wine.eu - NAPA-WiNe: Network-Aware P2P-TV Application over Wise Networks. Small or Medium-Scale Focused Research Project (Ref. FP7-2008-ICT-214412).

[NR10] Leonardo Neumeyer, Bruce Robbins, Anish Nair, and Anand Kesari. S4: Distributed stream computing platform. In Proc. International Workshop on Knowledge Discovery Using Cloud and Distributed Computing Platforms (KDCloud 2010), 2010.

[NRTV07] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V. Vazirani. Algorithmic Game Theory. Cambridge University Press, New York, NY, USA, 2007.

[NRZ+07] Neglia, G., Reina, G., Honggang Zhang, Towsley, D., Venkataramani, A., Danaher, J., Availability in BitTorrent Systems, INFOCOM 2007. The 26th Conference on Computer Communications. IEEE, pages 2216-2224, May 2007.

[NSS10] Erik Nygren, Ramesh K. Sitaraman, Jennifer Sun, "The Akamai Network: A Platform for High-Performance Internet Applications", ACM SIGOPS Operating Systems Review, Vol. 44, No.3, July 2010.

[NXTY10] Jian Ni, Haiyong Xie, Sekhar Tatikonda and Yang Richard Yang, Efficient and Dynamic Routing Topology Inference From End-to-End Measurements, IEEE/ACM Transactions on Networking (TON), Volume 18, Issue 1, February 2010.

[P2PN10] http://www.p2p-next.org - P2P-NEXT: Next generation peer-to-peer content delivery platform. Large-scale Integrated Project (Ref. IST-216217).

[PFA10] Ingmar Poese, Benjamin Frank, Bernhard Ager, Georgios Smaragdakis, and Anja Feldmann. Improving content delivery using provider-aided distance information. In Proceedings of the 10th annual conference on Internet measurement, IMC '10, pages 22–34, USA, 2010. ACM.

[PGES05] J. A. Pouwelse, P. Garbacki, D. H. J. Epema, and H. J. Sips. The Bittorrent P2P file-sharing system: Measurements and analysis. In 4th Int'l Workshop on Peer-to-Peer Systems (IPTPS), volume 3640. LNCS, Feb 2005.

[PGW+07] J. Pouwelse, P. Garbacki, J. Wang, A. Bakker, J. Yang, A. Iosup, D. H. J. Epema, M. Reinders, M. van Steen, and H. Sips. Tribler: A social-based peer-to-peer system. Concurrency and Computation: Practice and Experience, 2007.

[PI06] Fragkiskos Pentaris and Yannis Ioannidis. Query optimization in distributed networks of autonomous database systems. ACM Trans. Database Syst., 31:537–583, June 2006.

[PIA+07] Michael Piatek, Tomas Isdal, Thomas Anderson, Arvind Krishnamurthy, and Arun Venkataramani. Do incentives build robustness in BitTorrent? In Proceedings of NSDI'07, Cambridge, MA, April 2007.

[pla07] PlanetLab, 2007.

[PMG09] Jon Peterson, Enrico Marocco, and Vijay Gurbani. Application-Layer Traffic Optimization (ALTO) working group, 2009.

[PMJ09] Michael Piatek, Harsha V. Madhyastha, John P. John, Arvind Krishnamurthy, and Thomas Anderson. Pitfalls for ISP-friendly P2P design. In Proceedings of the Eighth ACM SIGCOMM Workshop on Hot Topics in Networks, USA, 2009. ACM.

[PMT03] Vassilis Papadimos, David Maier, and Kristin Tufte. Distributed query processing and catalogs for peer-to-peer systems. In In Proc. 1st Conference on Innovative Data Systems Research (CIDR), pages 5–8, 2003.

[PUG90] W. Pugh. Skip lists: a probabilistic alternative to balanced trees. Communications of the ACM, 33(6):668-676, 1990.

[PWC02] V. N. Padmanabhan, H. J. Wang, P. A. Chou, and K. Sripanidkulchai. Distributing streaming media content using cooperative networking, in Proceedings of NOSSDAV, Miami Beach, FL, USA, 2002

[PWC03] Venkata N. Padmanabhan, Helen J. Wang, and Philip A. Chou. Resilient peer-to-peer streaming. In ICNP '03: Proceedings of the 11th IEEE International Conference on Network Protocols, page 16, Washington, DC, USA, 2003. IEEE Computer Society.

[RD01] A. Rowstron and P. Druschel. Pastry: scalable, decentralised object location and routing for large-scale peer-to-peer systems. In Proceedings of the 18th IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), November 2001.

[RFH+01] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, A Scalable Content-Addressable Network. In Proceedings of the SIGCOMM'01 Symposium on Communications Architectures and Protocols, San Diego, California, August 2001.

[RK02] S. C. Rhea and J. Kubiatowicz. Probabilistic location and routing. Proceedings of INFOCOM '02, 3:1248–1257, 2002.

[RKCD01] Antony Rowstron, Anne-Marie Kermarrec, Miguel Castro, and Peter Druschel. Scribe: The design of a large-scale event notification infrastructure. In Jon Crowcroft and Markus Hofmann, editors, Networked Group Communication, Third International COST264 Workshop (NGC'2001), volume 2233 of Lecture Notes in Computer Science, pages 30–43, November 2001.

[RKHS02] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker. Topologically-aware overlay construction and server selection, INFOCOM, 2002.

[RLC08] D. Ren, Y.-T. Li, and S.-H. Chan. On reducing mesh delay for peer-to-peer live streaming. INFOCOM 2008. The 27th Conference on Computer Communications. IEEE, pages 1058-1066, April 2008.

[RO03] Reza Rejaie and Antonio Ortega. Pals: peer-to-peer adaptive layered streaming. In Proceedings of NOSSDAV 2003, pages 153 161, New York, NY, USA, 2003. ACM.

[RULC10] A. Russo and R. Lo Cigno, "Delay-Aware Push/Pull Protocols for Live Video Streaming in P2P Systems," in IEEE ICC 2010, (Cape Town, South Africa), May 2010.

[SAZ04] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana. Internet indirection infrastructure. IEEE/ACM Transactions on Networking (TON), 12(2):205–218, April 2004.

[SCPR09] Marco Slot, Paolo Costa, Guillaume Pierre, and Vivek Rai. Zero-day reconciliation of bittorrent users with their isps. In Proceedings of the 15th International Euro-Par Conference on Parallel Processing, pages 561–573, Berlin, Heidelberg, 2009. Springer-Verlag.

[Sen77] Amartya Sen. Social choice theory: A re-examination. Econometrica, 45(1):pp. 53-88, 1977.

[SGGS09] Salvatore Spoto, Rossano Gaeta, Marco Grangetto, and Matteo Sereno. Analysis of PPLive through active and passive measurements. In Proceedings of the 2009 IEEE International Symposium on Parallel and Distributed Processing, pages 1--7, 2009.

[SH04] Meng-Fu Shih and A.O. Hero. Network topology discovery using finite mixture models. In Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on, volume 2, pages ii-433 - ii-436 vol.2, May 2004.

[SHMA07] S. Sanghavi, B. Hajek, and L. Massoulié, "Gossiping with Multiple Messages", in Proc. INFOCOM, 2007, pp.2135-2143.

[SMG+07] A. Sentinelli, G. Marfia, M. Gerla, L. Kleinrock, and S. Tewari. Will IPTV ride the peer-to-peer stream? Communications Magazine, IEEE, 45(6):86--92, June 2007.

[SMK+01] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan, Chord: A scalable peer-to-peer lookup service for internet applications. In Proceedings of the SIGCOMM'01 Symposium on Communications Architectures and Protocols, San Diego, California, August 2001.

[SMOO10] http://www.smoothit.org - SmoothIT: Simple economic management approaches of overlay traffic in heterogeneous Internet topologies. Small or Medium-Scale Focused Research Project (Ref. FP7-2008-ICT- 216259).

[SR06] D. Stutzbach and R. Rejaie. Understanding churn in peer-to-peer networks. In Proceedings of IMC '06, pages 189–202. ACM Press New York, NY, USA, 2006.

[ST04] Y. Shavitt and T. Tankel. Big-bang simulation for embedding network distances in euclidean space. Networking, IEEE/ACM Transactions on, 12(6):993 - 1006, 2004.

[SYW04] S. Shi, G. Yang, D.Wang, J. Yu, S. Qu, and M. Chen. Making Peer-to-Peer keyword searching feasible using multi-level partitioning. In Proceedings of International Workshop on Peer-to-Peer Systems (IPTPS), pages 151{161. Springer, 2004.

[Tay05] Alan D. Taylor. Social Choice and the Mathematics of Manipulation. Outlooks. Cambridge University Press, May 2005.

[THD03] D. A. Tran, K. A. Hua, T. Do, ZIGZAG: an efficient peer-to-peer scheme for media streaming, in Proceedings of IEEE INFOCOM, San Francisco, CA, USA, 2003.

[THD04] D. A. Tran, K. Hua, and T. Do, A peer-to-peer architecture for media streaming, IEEE Journal on Selected Areas in Communications – Special Issue on Advances in Service Overlay Networks, vol. 22, no. 1, January 2004.

[TJ06] Guang Tan and Stephen A. Jarvis. A payment-based incentive and service differentiation mechanism for peer-to-peer streaming broadcast. Proc. Of IWQoS, 2006.

[TN08] D.A. Tran, T. Nguyen, Hierarchical Multidimensional Search in Peer-to-Peer Networks, Computer Communications, Feb 2008

[TR03] D. Tsoumakos and N. Roussopoulos. Adaptive probabilistic search for Peer-to-Peer networks. In Proceedings of International Conference on Peer-to-Peer Computing (P2P), 2003

[Var96] Y. Vardi. Network tomography: Estimating source-destination traffic intensities from link data. Journal of the American Statistical Association, 91(433):365-377, 1996.

[VS05] David Del Vecchio and Sang H. Son. Flexible update management in peer-to-peer database systems. Database Engineering and Applications Symposium, International, 0:435–444, 2005.

[VYF06] V. Venkataraman, K. Yoshida, and P. Francis. Chunkyspread: Heterogeneous unstructured tree-based peer-to-peer multicast. Proceedings of ICNP '06, pages 2--11, November 2006.

[WHH+92] Carl A. Waldspurger, Tad Hogg, Bernardo A. Huberman, Jeffrey O. Kephart, and W. Scott Stornetta. Spawn: A distributed computational economy. IEEE Trans. Software Eng., 18(2):103--117, 1992.

[WLL08] Chuan Wu, Baochun Li, and Zongpeng Li. Dynamic bandwidth auctions in multi-overlay P2P streaming with network coding. IEEE TPDS, 19(6), 2008.

[WLR09a] Di Wu, Chao Liang Yong Liu, and Keith Ross. View-upload decoupling: A re-design of multi-channel P2P video systems. In Proceedings of IEEE Conference on Computer and Communications (INFOCOM Mini-Conference '09), 2009.

[WLR09b] Di Wu, Yong Liu, and Keith Ross. Queuing network models for multi-channel P2P live streaming systems. In Proceedings of IEEE Conference on Computer and Communications (INFOCOM '09), 2009.

[XYK08] Haiyong Xie, Y. Richard Yang, Arvind Krishnamurthy, Yanbin Grace Liu, and Abraham Silberschatz. P4P: Provider portal for applications. SIGCOMM Comput. Commun. Rev., 38(4):351–362, 2008.

[YM02] B. Yang and H. Garcia-Molina. Improving search in Peer-to-Peer networks. In Proceedings of International Conference on Distributed Computing Systems (ICDCS), 2002.

[ZAFB00] E. W. Zegura, M. H. Ammar, Z. Fei, and S. Bhattacharjee. Application-layer anycasting: a server selection architecture and use in a replicated Web service. IEEE/ACM Transactions on Networking (TON), 8(4):455–466, 2000.

[ZKJ01] Ben Y. Zhao, John D. Kubiatowicz, and Anthony D. Joseph. Tapestry: An infrastructure for fault-tolerant wide-area location and routing. Technical Report UCB/CSD-01-1141, EECS Department, University of California, Berkeley, Apr 2001.

[ZLL05] X. Zhang, J.C. Liu, B. Li, P. Yum, CoolStreaming/DONet: A data-driven overlay network for efficient live media streaming, in Proceedings of IEEE INFOCOM, Miami, FL, USA, 2005

[ZSLS06] C. Zheng, G. Shen, S. Li, S. Shenker, Distributed Segment Tree: Support of Range Query and Cover Query over DHT, Proceedings of the Fifth International Workshop on Peer-to-Peer Systems (IPTPS), Santa Barbara, California, February 2006.

[ZZT05] M. Zhang, L. Zhao, Y. Tang, J. Luo, S. Yang, Large-Scale Live Media Streaming over Peer-to-Peer Networks through Global Internet, in Proceedings of ACM Multimedia 2005, Singapore, Singapore, 2005