Enriched Network-aware Video Services over Internet Overlay Networks

www.envision-project.org

Deliverable D3.3

Final Specification of the ENVISION Interface, Network Monitoring and Network Optimisation Functions

Public report, v2a, 6 March 2013.

Authors

UCL	Raul Landa, Richard Clegg, Eleni Mykoniati, David Griffin, Miguel Rio
ALUD	Nico Schwan, Klaus Satzke
LaBRI	
FT	Bertrand Mathieu, Irène Grosclaude, Selim Ellouze, Emile Stephan, Pierre Paris, Valery Bastide
TID	
LIVEU	

Abstract This deliverable presents the latest updates in the definition of the CINA interface and our efforts to push it in the standardisation bodies (IETF ALTO). The proposed drafts, issued from ENVISION, are mainly related to the discovery and the multi-cost and cost schedule approach. The progress on the implementation of the network services, such as the Multicast and the High Capacity node, is also described in this deliverable, in a complementary approach, with regards to the existing text in [D3.1] and [D3.2]. We also investigated the use of the CINA interface for Cloud services, which is presented with a presentation of additional network services a Cloud Provider is interested in and with the migration of VM in a Cloud system as an example. Finally, two studies are presented: The time and space traffic shifting study, initiated in [D3.2], completed with the latest evaluation results showing the interest for ISPs for such an approach, and another study, highlighting the benefit of a collaboration between service providers and network operators, as advocated by the ENVISION project, with exchange of maps in a bidirectional way (such as the cost map, the constraint map) is presented and showed great improvement in the network load for a lower cost for ISPs.

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

EXECUTIVE SUMMARY

This deliverable contains the main achievements of the work done in the last 6 months of WP6:

- Updates for the CINA interface and drafts pushed in the IETF ALTO working group: server discovery, multi-cost metric and cost schedule.
- Updates on implementation of the network services : multicast and high capacity node
- Investigation of the use of CINA for Cloud systems.
- Proposal for a solution to shift traffic in time and in space for reducing bills for ISP: presentation of the model and evaluation results.
- Model to highlight the benefit of a bidirectional communication between service providers and network operators and evaluation results.

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

In this deliverable, we present the results of work done within WP3 in the last 6 months. It mainly includes updates about the CINA interface and focusing on the drafts the project pushed in the IETF ALTO working group in section 2. Section 3 related to the monitoring in very short since no major updates on this activity has been done in the last months. In section 4, we present the main part of work done recently, related to network optimisation. We present the improvement done in the implementation of the network services we will test in a real testbed, namely the multicast in section 4.2 and the High Capacity Node in 4.3. We also present the outcomes of our investigation about the possibility to use CINA for Cloud systems: two possibilities have been explored: as an interface between cloud management systems and network operators to invoke network services in section 4.6, and as a standardised interface between end-users and Cloud management systems in section 4.7. An example depicts how it might be used. Finally two studies related to network optimisation are detailed : one related to the time and space traffic shifting, initiated in [D3.2] is completed in this deliverable with latest evaluation results in section 4.5 and another one highlighting the benefit of mutual exchange of information between the two actors with modelling and evaluation results in section 4.8.

2. CINA INTERFACE

The Collaboration Interface between Network an Applications (CINA) interface has been specified within the ENVISION project as a powerful instrument to foster the cooperation between the network and the application layer. This collaboration allows a mutual information exchange as well as the possibility for applications to invoke network services tailored to their requirements. D3.1 and D3.2 contain the detailed specifications of the CINA interface. In particular the specifications comprise the discovery of a CINA server by a CINA client, the information exchange methods from the network to the application and vice versa, the network service invocation methods as well as the security system employed to guarantee controlled access to CINA services. In section 2.1 we discuss the progress that has been made on the specification, in particular with respect to the active Internet Drafts that are currently under standardisation at the IETF. Therefore it is assumed that the reader is familiar with D3.2 and the underlying concepts of the CINA interface. In section 2.2 the update on the implementation of the CINA client library and the CINA server are presented.

2.1 Standardisation update

Although the ALTO standardisation effort started off with a focus on P2P overlay applications, recently new use cases around data centres and content distribution networks have emerged and received a lot of attention by the community. This has led to an informational BOF at IETF#83 to gather standardisation interest for an *Infrastructure to Application Extension* (i2aex) protocol. The BOF was organised around three use cases that motivate such a protocol. These use cases were presented and discussed and the ENVISION consortium has contributed to the slideshow to ensure the uptake of project results.

In the following we present currently ongoing updates of the active Internet Drafts that are contributions at the IETF by the ENVISION project. Further for each draft the respective section discusses the relevance of the suggested extensions in the emerged focus areas.

2.1.1 Server Discovery

The Server Discovery draft [KIE12] has undergone major revisions since a DNS expert review has proven several limitation of the previously designed solution. In particular the third party ALTO server discovery raises several issues, as typically only the IP address of the client is known that needs to suffice finding the corresponding ALTO server. This requires determining a FQDN as input for the U-NAPTR process. One option to do this is to use a reverse DNS lookup. However, reverse DNS has several limitations:

First, there is no established unique way of maintaining the DNS tree, and there are different practices in different networks. Furthermore, it is possible that a lookup fails or that the returned value is not valid. For instance, it can point to a different domain. As a result, any potential use of reverse DNS lookup for service discovery must be able to deal with failures of lookup and react accordingly. Second, determining a domain name from IP addresses by tree climbing is problematic, in particular for IPv6, which is discussed in [RFC4472]. Third, populating a DNS name space what looks like a reverse tree is a significant administrative DNS overhead. Finally, it must be emphasised that any tree walking procedure raises several issues. There is no single best way tree, and heuristics are needed.

Given these problems, the Server Discovery Draft does not specify a reverse DNS mechanism to determine a FQDN. Instead, it is limited to scenarios where the discovery procedure is done by the resource consumer. In case a third party needs to know the contact information of the server it recommended that the resource consumer discovers the server and then sends the contact information directly to the third party. As a potential alternative, the client could provide a valid FQDN, so that the third party can use this as input for the U-NAPTR process, but this variant has no

significant advantages. The options on how to transmit the contact information from the resource consumer to the third party are out of scope of the draft and are not specified.

A second major change is the specification of a Point-to-Point Protocol (PPP) extension for the provisioning of the Access Network Domain Name as an alternative to DHCP, which is critical for access networks that employ PPP to configure home user devices. One possible example that yielded by this extension could be:

```
example.com
```

The domain name is encoded according to Section 3.1 of [RFC1035] whereby each label is represented as a one-octet length field followed by that number of octets. Since every domain name ends with the null label of the root, a domain name is terminated by a length byte of zero. The high-order two bits of every length octet MUST be zero, and the remaining six bits of the length field limit the label to 63 octets or less. To simplify implementations, the total length of a domain name (i.e., label octets and label length octets) is restricted to 255 octets or less.

For example, the domain "example.com." is encoded in 13 octets as:

```
+---+---+---+---+---+---+---+---+---+---+---+---+---+
| 7 | e | x | a | m | p | l | e | 3 | c | o | m | 0 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

The Internet Protocol Control Protocol (IPCP) Option defines a method for negotiating with the remote peer the name of the Access Network Domain Name to be used on the local end of the link. A summary of the Access Network Domain Name Configuration Option format is shown below. The fields are transmitted from left to right.

```
 Type  Len Access Network Domain Name
+-----+-----+-----+-----+-----+-----+-----+----
| tba |  n  |  s1 |  s2 |  s3 |  s4 | s5  |  ...
+-----+-----+-----+-----+-----+-----+-----+----
```

The values s1, s2, s3, etc. represent the domain name labels in the domain name encoding. Note that the length field in the IPCP option represents the length of the entire domain name encoding, whereas the length fields in the domain name encoding the length of a single domain name label.

```
Type: to be assigned by IANA

Len: Length of the 'Access Network Domain Name' field in octets.

Access Network Domain Name: The domain name of the Access Network for
the client to use.
```

Use of Server Discovery outside of the P2P space

The server discovery enables client applications to dynamically find an appropriate server, no matter at what access network they are registered in. This dynamic discovery is typically needed to avoid that home users need to do this configuration manually. In emerging use cases the environment is often controlled and managed by a technically skilled administrator (i.e. in CDNs or Data Centres). Here a dynamic discovery of servers is expected to be unlikely; in contrast a manual configuration is probable. Thus it is expected that the Server Discovery mechanism will not play a prominent role outside of uncontrolled environments with home user engagement, such as P2P overlay applications.

2.1.2 Multi-Cost

The Multi-Cost extension draft [RAN12] is currently in its 6th iteration. A detailed description has been given already in the previous deliverable D3.2, thus this section is limited on describing the main updates of the draft version 06 only.

In this version one new use case has been added, centred on end systems that need to spare time by optimising their transactions. Additionally the main motivation and features for introducing multi-cost queries have been detailed. By using the multi-cost extension applications gain time and resources by transmitting information on N Cost Types in one transaction rather than in N transactions which saves N RTTs and thus increases performance and user QoE. In addition to the bandwidth saved during transmission, one Multi-Cost Map is further less bulky to store than N Single Cost Maps.

The most recent version of the draft further specifies several Multi-Cost services and illustrates example request and responses for each service. The services specified are

- Multi-Cost Map Service
- Filtered Multi-Cost Map Service
- Endpoint Multi-Cost Service

Use of Multi-Cost outside of the P2P space

The Multi-Cost extension in itself is an optimisation of the ALTO/CINA protocol. Retrieving Cost Maps for more than one cost could be done already by today with the base protocol; the benefits of Multi-Cost Maps in terms of performance and reduced network load are discussed above. These benefits become increasingly important the more different Cost-Types are useful for a client application. For P2P applications the predominant cost is expected to remain routing cost, however also latency might play a role in the future. For CDNs and Data Centres many more cost types are expected to be important, for example request routing decisions to one out of multiple deployed content or service instances are not only based on network metrics like routing cost, latency or bandwidth, but also on server load (memory, cpu). Thus it is expected that the Multi-Cost extension will remain important for use cases outside of the P2P space, in particular if the ALTO protocol receives a wide adoption as base protocol for network to application information.

2.1.3 Cost Schedule

Historically the extensions specified in the Cost Schedule draft [RAND12] have been described in the Multi-Cost draft described in the previous section. For consistency reasons these previously called ‘dynamic costs’ have been extracted and submitted in a dedicated draft.

The typical use cases for ALTO and CINA are applications that spatially shift traffic between network regions in order to lower routing cost for ISPs. Other applications however also have a degree of freedom of when to request or use a resource. These are typically non-real-time applications that for example need to receive a particular content at some point in time in the future or that need to have a computational job done at a certain point in time. In order to allow the applications to schedule their actions, the Cost Schedule extension enhances Cost Maps in a time horizon. Thus data centre or cache locations can express what the cheapest time for them is to execute a certain action, for example in diurnal hourly patterns. Examples that are described in the draft comprise pre-population of caches in CDNs, data-replication across time-zones between Data Centres of Online Social Networks and end systems with limited access to Data Centres that need to schedule their access to resources. The Cost Schedule extension lowers traffic peaks of bottleneck links and thus saves scarce resources and improves user QoE. By allowing different levels of abstractions, it further allows different levels of accuracy according to the requirements of the respective service operator, preserving confidentiality of the data where needed.

Use of Cost-Schedule outside of the P2P space

The Cost Schedule extension is focused on use cases where applications do have a degree of freedom on when to schedule a download. For P2P applications this is typically not the case (although theoretically possible) as there is no obvious user incentive that could be used to delay a download. The Cost Schedule extension in itself is motivated by other applications as described above, outside the P2P space, and thus it is expected that this extension remains important for these emerging use cases.

2.1.4 Incremental Updates

The Incremental Updates extension [SCH12] is motivated by the fact that Network and Cost Maps can become very large, for example a Network Map that comprises 5000 PIDs and 10 CIDRs per PID will hold up to 1.25 Mbytes of data. For a Cost Map the situation is even worse, for example one that provides costs for the above described Network Map results in a 5000x5000 matrix and thus approximately 417 Mbytes of data. Although the estimated update frequency of Network Maps is expected to lie in the order of once a day or two, for Cost Map the frequency is probably much higher. It is expected that for Cost Maps something will change every few minutes. Thus a mechanism that allows conditional and incremental updates of Network and Cost Maps is required.

The current draft analyses existing HTTP Mechanisms for conditional retrieval, such as the If-Modified-Since header or the If-None-Match header. Also partial retrieval mechanisms such as the Range header are considered. However the analysis has proven that although existing HTTP mechanisms might work for conditional retrieval of a full map, incremental retrieval of changes is violating the HTTP specification and is thus not a valid option.

For incremental updates two possible options are described. One is the use of JSON Patch which is designed to modify existing JSON encoded resources by *add, replace* and *delete* operations. However JSON Patch has been designed to operate on server resources, whereas in this case a mechanism that operates on client Maps is required. The second alternative which is considered thus is a separate CINA/ALTO service that is based on the existing Filtered Network Map and Cost Map response messages. In this incremental update service for Network Maps, each PID in the message replaces previous CIDRs with new CIDRs. To delete a PID, "*delete*" is used as the value (or alternatively an empty array). Finally PIDs which are not included not in the message stay the same. Similarly for Cost Maps costs in the message replace the previous costs for the respective source/destination PIDs. To delete a cost, again, the value "*delete*" (or alternatively "*-1*", or "*NaN*") can be used and costs not in the message remain the same. Additionally versioning for the Cost Map is introduced, a new MIME type for the incremental update services is specified and an example entry for the Information Resource Directory is given.

In order to decide whether to proceed with JSON patch or the new service option the next step will be to provide calculations of the effectiveness of the respective option. Currently, it is expected that the dedicated incremental update methods will be the preferred option in terms of practicality and performance gain.

Use of Incremental Updates outside of the P2P space

The importance and benefits of the Incremental Updates extension strongly depends on the size of the Network and Cost Maps, which in turn depends on the respective deployment scenario. While in some scenarios the server operator might be willing to share a lot of detailed information, it is also likely that in other scenarios the network view provided will remain on highly abstracted level. In particular in uncontrolled (P2P) scenarios, it is expected that network operators will be rather reluctant to provide detailed insights in their network topology. In contrast, in controlled environments, for example system internal request redirection within CDNs or Data Centres, the operator does not risk that the information gets spoiled. In conclusion it is expected that the Incremental Updates extension remains or even gains importance outside of the P2P space.

2.2 Implementation of the CINA interface

For the initial description of the CINA implementation, the reader can refer to the previous deliverable [D3.2]. The CINA library has been updated in the last months. The main updates are related to the development and integration of modules necessary to invoke the network services, such as the multicast and the caching. For the multicast service, the CINA server-side has been completed with the Tomcat software which enables to more conveniently and more efficiently processed the CINA request related to the invocation of the multicast network service (with the necessary parameters as input to the request). It better manages the fast-cgi scripts that are used for the CINA services.

The CINA client-side has also been improved with a more efficient software, taking less memory space, less redundant functions. For this, it better includes some Java libraries useful for our project, such as the JSON library. The encapsulation and decapsulation of messages into/from JSON messages is easier from a developer point of view, relying on the JSON library.

3. MONITORING

3.1 Monitoring Component

The monitoring activity was detailed in [D3.2] and few activities happened since then. The monitoring tool has just been updated to be able to monitor new network equipment, installed in the Orange tested. The main concepts of it did not change and just little adaptation has been done. The reader can find more information for the monitoring aspects in [D3.1] and [D3.2].

4. NETWORK OPTIMISATION

4.1 Introduction

In task 3.3 of the project, mechanisms aiming at optimising the network were investigated. It encompassed the network services a network operator can deploy for network load reduction, such as Multicast, Caching, High Capacity node. For those Network Services, an implementation has been done as a proof-of-concept. Their descriptions have been provided in [D3.1] and [D3.2] and their implementation in [D3.2], but some latest updates are also in this deliverable. We have also investigated the potential use of CINA for Cloud networking and the outcomes of our investigation are presented in this document. For the network optimisation, the project also performed some analytical evaluations and simulations to highlight the benefits. One related to the space/time shifting proposal has been initiated in [D3.2] and an updated version of this study can be found in this deliverable. In this deliverable, another one, related to the use of CINA for CDNs and Cloud systems is also described.

4.2 Multicast

The CINA interface for the lease of multicast resources has been described in [D3.2].

In this document we describe the final implementation of the multicast service and how it has been integrated in the CINA server.

As already described in [D3.2] the multicasters are implemented as NAT functions that translate unicast UDP flows to multicast flows. Our implementation relies on the *iptables* functions of the Linux kernel and on *smcroute* which is a static multicast router daemon for Linux. See [D3.2] for the exact configuration commands.

The set of multicasters, and the associates multicast resources are managed by a multicast controller integrated in the CINA server.

4.2.1 Multicast controller implementation

The multicast controller has been implemented in Java. The main classes are represented in Figure 1.

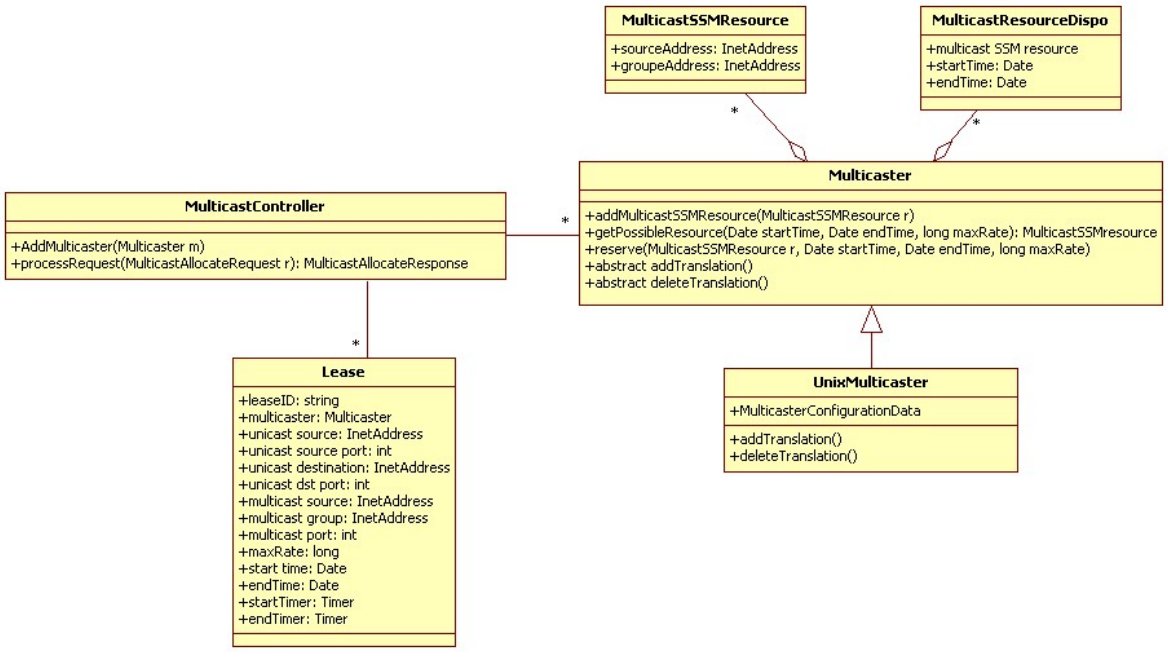

Figure 1: Multicast controller implementation

The multicast controller manages a set of objects "Multicaster" representing the physical multicasters, each one being initialised with a set of multicast resources (couple a multicast source and group addresses) and with a bandwidth capacity. Each multicaster manages the list of time intervals during which each multicast resource is available.

When the multicast controller receives a multicast allocation request, it verifies the resources available on each multicaster for the requested time interval and selects one resource. If several multicasters can be used, the choice can take into account the distance and/or the link state between the source and the multicaster. Then the chosen resource is reserved and an associated lease is created. This lease contains two Java Timers which triggers the configuration of the multicaster at the beginning and end of the lease time.

The class Multicaster is an abstract class, which allows the use of several types of multicasters. In our case we have implemented a specific UnixMulticaster class which establishes SSH connections with the Linux multicasters and sends the required configuration commands.

4.2.2 Integration in the CINA server

The multicast controller can easily be integrated in Java applets as a persistent and shared object.

The CINA web server is configured to forward all the requests corresponding to the multicast service URI (typically with the /multicast path) to the applet container.

For our implementation we have used the Apache Tomcat applet container, the open source implementation of JAX-RS (Java API for RESTful Web Services) Jersey[1], and the library Jackson[2] to convert Java Object to JSON.

4.3 High Capacity Node

The High Capacity Node (HCN) network service is designed to support overlay applications by seamlessly integrating into the overlay and helping to distribute content and thereby boosting the totally available bandwidth of the peer-to-peer overlay. The invocation of the HCN service is possible in ENVISION enabled network domains through the CINA interface in a dynamic and on-demand way. The instantiated nodes with their high capacities enable applications to support higher quality streams due to the extra bandwidth added by infrastructure nodes. The HCN service is designed as a network service that runs on a specialised routing platform deployed at strategic locations within the network. Thus HCNs can be instantiated at critical parts of the network, for example where bandwidth demand of peers exceed the currently available overlay upload capacity.

The High Capacity Node network service has been introduced and motivated in [D3.1] and [D3.2] already. In particular two implementation options have been discussed; either to implement the HCN as a loosely coupled application inside a virtualised environment on the Research Routing Platform (RRP), or to use the native IP forwarding for content distribution. A preliminary conclusion was made, that the IP based forwarding option was more promising, as it leverages the special capabilities of the platform very well and in addition is well suited for the tree-based IVCD (see D4.2) system. However the drawback of this option is the limited applicability as it only works for UDP based applications. As a broad applicability of the network service is desirable, the decision was revisited and instead the design decision was made to implement the network service as loosely coupled application.

The final architecture of the network service is detailed in the next section, followed by a call flow that shows how the network service is setup and how applications are able to invoke the service and instantiate a High Capacity Peer.

4.3.1 Final Architecture

The RRP provides several features that are exploited for the implementation of the HCN network service. It is able to host applications inside of virtualised machines, thus it provides the opportunity to host runtime environments suited for any requirements of third party applications. More details on the platform are given in [D3.2].

[1] See http://jersey.java.net/

[2] See http://wiki.fasterxml.com/JacksonHome.

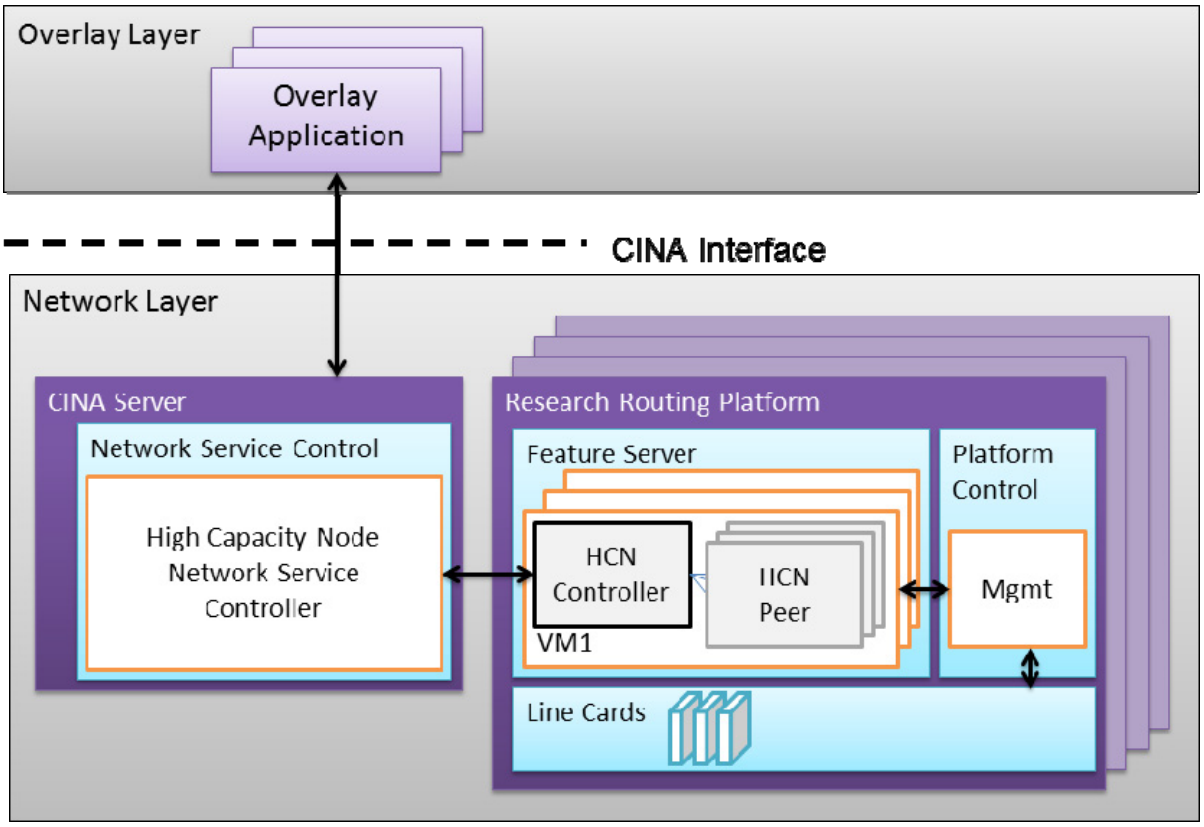

Figure 2: High Capacity Node Network Service Architecture

Figure 2 illustrates the final architecture of the HCN network service. On top it depicts various potential overlay applications that are able to invoke the network service through the CINA interface. On the network side they communicate with the CINA server, and specifically with the High Capacity Node Network Service Controller (HCNNSC). The HCNNSC aggregates a view of all available RRP and their hosted HCN Peers to a Service Cost Table, which can be requested by applications in order to know in which network regions the HCN Peers can be instantiated. It further controls the instantiation and teardown of HCN Peers and therefore triggers the HCN Controllers (HCNC) that resides in the virtual machines (VM) inside of the platforms. The HCN Controllers in turn receive these control messages and start or stop the respective process inside the VM. The executables that are started or stopped by the HCNC therefore need to be deployed, which typically means that they need to be compiled with the platforms SDK beforehand. It is envisaged that application developers that want to use the HCN network service register their application during an offline process. Each application (thus: P2P protocol) is assigned a unique identifier that ensures that the instantiated HCN is compatible with the running overlay protocol. As applications require different runtime environments, for example in terms of operating system, library versions, etc., multiple virtual machines can be created, meeting the requirements of each application. If two or more overlay applications are able to run in parallel in the same environment, they can be hosted in the same VM and thus are controlled by the same HCNC. Another task of the HCNC is to configure the forwarding rules of the platform in order route traffic to a newly instantiated HCN Peer. More details on this are depicted in the call flow in the next section. Although the architecture is designed to host multiple third party software versions, deployed in multiple VMs and on multiple RRPs; due to resource limitation the Proof-of-Concept implementation is limited to only a subset of the described functionalities.

4.3.2 HCN Call Flow

This section depicts in Figure 3 a call flow that shows (i) how the HCN network service is dynamically setup at the start-up phase and how a coherent view of the network service resources can be provided to overlay applications and (ii) how an overlay application interacts through the CINA interface with the HCNNSC to retrieve this information and to invoke the service.

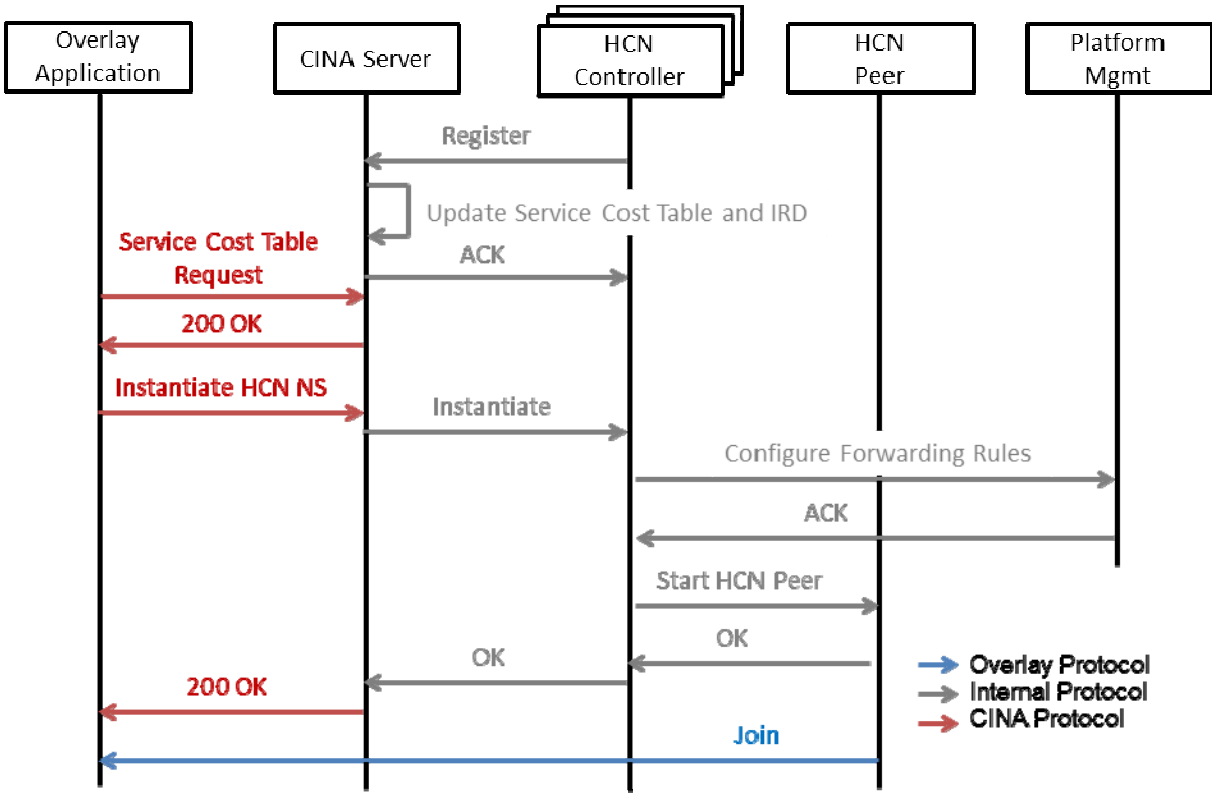

Figure 3: HCN Network Service Registration and Service Instantiation Call Flow

At start up each HCNC needs to register at the HCNNSC, which is embedded in the CINA server. Therefore it needs to gather and transmit different types of meta-information, which can either be retrieved from the RRP itself or needs to be provided by manual configuration. The meta-information includes

- Overlay application software details (executable path, unique identifier)
- HCNNSC contact information
- Load characteristics of the node (CPU load, memory usage, bandwidth capacity etc.)
- HCN cost information
- Interface addresses of the line cards

This information is sent to the HCNNSC, which in turn updates its internal database and calculates an update of the Service Cost Table (defined in [D3.2] Appendix B). Therefore it maps the IP addresses of the RRP line cards to the network regions (PIDs) of the configured Network Map and updates capacity and cost information respectively. The Service Cost Table thus depicts information about where a specific HCN service *can* be invoked, and where it *should* be invoked according to the preferences of the CINA service provider. An application can then choose between different locations and behave friendly towards the network operator, similarly to choosing between several destinations the one with the least routing cost. The information can be updated periodically in order to keep an up-to-date view on current resources of the HCN service.

After HCNCs have registered, the HCNNSC is able to advertise and instantiate HCN Peers. In the call flow an overlay application is illustrated which instantiates an HCN Peer. The call flow hereby does not show the first interactions between the overlay application and the CINA server, where the CINA server is discovered and the Information Resource Directory (IRD) is processed and thus the different types of available HCN services are discovered. The first request which is shown in the call flow retrieves the Service Cost Table for a specific HCN service (i.e. for a specific overlay protocol version). From the Service Cost Table it can now calculate where to request a HCN Peer (HCNP), taking into account its current overlay topology and potential bottlenecks in it. It then sends a request to the HCNNCS to instantiate a HCN, which in turn parses the request and may check credentials. The request is then forwarded to the HCNC, which before starting the HCNP, configures the forwarding plane through the PPRs Mgmt module. The added forwarding rule will allow the HCN Peer to be reachable at the configured address. As a second step the HCNC starts a process which runs the overlay application software. It also configures the necessary software parameters like parent peer or tracker contact point or maximal streaming capacity or amount of neighbour peers. Subsequently the overlay application is informed of the success of the instantiation operation. Finally the new HCNP joins the overlay using the potentially proprietary P2P protocol and is thus ready to support the content distribution.

4.4 Caching

The caching activity within WP3 is tightly coupled with the one in WP5. In our implementation for this network service, we rely on the LaBRI software, developed within WP5. Since the caching node is also part of WP5, for clarity, we decided to describe our implementation in WP5 only, instead of sharing this activity between WP3 and WP5. The reader can read [D5.3] for more information on this caching implementation.

The evaluation of the caching network service will be performed in WP6.

4.5 Network optimisation logic based on preferences: Shifting ISP traffic in time and space to reduce transit bills[3]

This section describes TARDIS (Traffic Assignment and Retiming Dynamics with Inherent Stability) a system which allows network operators to delay or redirect traffic in order to reduce their transit bills. Traffic shifts in space, between alternate locations, and time, to later periods, are treated in a unified and consistent manner. An algorithm is given for calculating the cost of traffic across both dimensions based on past demand. The resulting transit cost charged is designed to be consistent with the commonly used 95th percentile pricing method. TARDIS is then formulated as a dynamical system that determines how best to allocate traffic in space and time in order to minimise transit costs. A continuous approximation of this system is proved to be stable. TARDIS is evaluated through simulations using real user data from two different networks and is shown to reduce provider transit bills under a wide variety of conditions.

4.5.1 Introduction

Surges in demand and changes in traffic patterns can incur significant operating costs to network providers. On a short timescale, there is a variable cost which an operator must pay its own providers for transit traffic. On a longer timescale, there is a fixed cost in upgrading capacity at each provisioning cycle. Traditionally, networks have attempted to reduce both through a combination of

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

traffic shaping, artificially curbing demand, and *traffic engineering* through routing optimisations. Both methods have proved ineffective in containing costs without degrading end user performance or inducing potentially oscillatory behaviour. Given these shortcomings, there has been great interest in alternative forms of traffic management which leverage the nature of Internet content.

The proliferation of highly replicated content across peer-to-peer (P2P) systems, content distribution networks (CDNs) and one-click hosting services (OCH) allow users to download the same content from multiple sources. In [AMSU11] the authors investigate CDN and hosting infrastructures to establish the proportion of content unique to a single provider and show that some hosting infrastructures have as much as 93% of their content available elsewhere. The prevalence of content replication is already exploited for the purpose of reducing ISP costs in proposals such as ALTO/P4P [XYK+08] and ONO [CB08]. In [FPS+12] the authors propose *content aware traffic engineering* (CaTE), which allows ISPs to take advantage of content available in multiple locations to reduce link utilisation. By causing users to select alternative download sources, traffic can be effectively rerouted within the network. This can be performed transparently (CaTE) or with user collaboration (ALTO). Such systems can be deemed to *shift traffic in space*.

In parallel, multiple proposals have explored the potential for shifting delay-tolerant traffic to off-peak hours. In [LR08] the authors describe a mechanism that offers users higher bandwidth off-peak if they deliberately delay some of their traffic. Further contributions [JHC11b, JHC11a, CLRS10] represent similar attempts to *shift traffic in time* by providing incentives to users.

Both temporal and spatial traffic shifting share the same underlying premise: that reallocating traffic according to content properties can reduce network costs. Despite this, no work has thus far addressed the challenges of managing traffic across both dimensions simultaneously. The present work fills this void.

While TARDIS can potentially be applied to both outbound and inbound traffic, its most compelling use case is for the latter. For one, existing traffic management tools are far more effective at exerting control over outbound traffic. There is a distinct lack of viable solutions for managing inbound traffic. Additionally, the characteristics of content consumption are more amenable to traffic shifting in space and time than those of content production and distribution. This is emphatic for "eyeball" ISPs in particular, whose costs are dominated by users retrieving highly replicated and often delay-insensitive content. Even beyond such networks however, TARDIS can prove a valuable asset in managing traffic and reducing costs by redirecting and retiming of traffic.

The contributions of this work are threefold. Firstly, a unified mathematical framework is presented for reallocating traffic across both time and space. Secondly, a means to compare the costs associated with traffic flows from different sources at different times and subject to different pricing schemes. Thirdly, a mechanism is described to allocate choices for time-delayed or location-altered traffic so that a provider can inform the host where and when to perform a given download. The dynamical system representation of this mechanism is shown to converge to a beneficial state for the system under weak conditions. The properties of TARDIS are verified through simulation, using traffic traces collected from a large European ISP and an Asian academic network.

4.5.2 Related work

Time dependent pricing has been studied as a method for enabling time shifting. In [JHC11a] and [JHC11b], the authors describe a model which uses a control loop to adapt the prices that ISPs charge users in response to changes in their bandwidth consumption. This provides an incentive for users to shift part of their traffic to off-peak times. Another deferred download scheme is the *Internet Post Office*, in which users request files and the ISP downloads them off-peak and temporarily stores them so that users can quickly retrieve the local copies when they next log on. The idea is further developed in [CLRS10] which uses real user data to estimate the cost reductions provided by such time delays.

Spatial shifting is commonly studied within the context of P2P systems, often requiring sharing information between the ISP and the application, either explicitly (such as provided by ALTO/P4P [XYK+08]) or implicitly (as achieved by ONO [CB08]). The ALTO/P4P approach is to introduce an interface between ISPs and overlay applications with the purpose of facilitating the selection of overlay nodes based on locality. In this context, TARDIS provides a mechanism for calculating the costs when the objective is not solely traffic localisation but the reduction of the ISP's transit bills in general. In [CLY+11] the authors assess the extent to which BitTorrent swarms can be *localised*, i.e. downloads can be kept within an ISPs own network. The authors consider various strategies to bias overlay topology construction towards local peers, and develop the concept of *inherent localisability*, which assesses the download performance of swarms using largely local connections within an ISP. Unfortunately, the degree of localisability depends heavily both on the nature of the torrent and the ISP.

Opportunities for space shifting are also widely recognised within CDNs, which present both high content replication and transparency in traffic redirection. In [PFA+10], for example, alterations are made to DNS servers in order to serve traffic from different CDN hosts transparently to the end user. As previously mentioned, CaTE [FPS+12] allows ISPs to change the user's download location in order to reduce link utilisation. The authors show that the gains can be substantial: more than 32% of the traffic in their dataset can be delivered from at least 8 different subnets, and almost 40% of traffic can be obtained from 3 or more locations. This estimate appears to focus on traffic from major content providers. With the inclusion of other kinds of traffic (e.g. P2P), the numbers might rise even higher.

An analysis of the benefits of space shifting achieved by a cooperative approach to content server selection and traffic engineering is undertaken in [JZRC09]. A Nash bargaining solution is used to combine the ISP objective of reducing congestion across the network and the content distribution objective of increasing the performance perceived by the users. A more lightweight form of cooperation is also considered where the two processes remain independent but they share information about the network status. While this approach ensures that cooperation leads to win-win scenarios without the requirement for additional incentives for space shifting, it does not explicitly take into account transit costs and it requires the ISP to expose sensitive information to third parties.

One-click hosting services provide yet another opportunity to take advantage from destination diversity, as they already contribute a share of traffic that can exceed that generated by on-line video services [AMD10]. For instance, Rapidshare (one of the largest OCH providers) distributes each piece of content between several multi-homed servers on different subnets in the same geographical location [AMD10].

4.5.3 Conclusions

TARDIS is a system which determines the best way for ISPs to redirect or retime their traffic in order to reduce bills. The system calculates pricing for a unit of traffic on a link which is priced at the 95th percentile in order that it can be compared with a link with a fixed price per unit traffic. The problem of reducing the ISP cost by moving traffic was reframed as a problem of equilibration over choice sets. A dynamical system was demonstrated which, in the continuous context, stably moves the system to an equilibrium position where no user can reduce the cost by moving their traffic in space or time.

The dynamical system was implemented as a discrete time system and tested on real data. The tests showed that under a very wide variety of assumptions ISP transit bills can be substantially reduced by rerouting or retiming only a small proportion of its traffic.

4.6 Network services invocation for Clouds via CINA

4.6.1 Limitations of current Cloud architectures

Cloud services built using cloud infrastructures are located on one or several Data Centres (DCs). Customers remotely access cloud services hosted in the cloud provider's DCs. Consequently, the network connection between a customer and their access DC and the network connection among DCs involved in delivering a cloud service play an important role in its perceived performance [ARM10]. Indeed even if a cloud service is well provisioned within the DC domains, if the network connectivity between customers and DCs is not up to the task, the perceived quality of service will not be satisfying for the customers. Thus, the network aspect of the cloud service delivery should be carefully addressed.

In current deployments, the selection of a cloud service instance to serve an end user request is performed without involving the network operator which will carry the traffic. As a result, the network segment that delivers the cloud service traffic is selected with little consideration regarding its actual performance.

Indeed, within DCs, cloud customers are able to provision on-demand various cloud services such as application instances, platforms and infrastructure, via a Cloud Management System (CMS for short, the entity that manages the cloud infrastructure and services) through a web or a programming interface. This enables them to create, modify, delete, scale up or down and monitor their cloud services in an on-demand fashion. However, this level of flexibility is lacking of network resources. There is no interface for dimensioning the availability, security and performance of the networks used to deliver the cloud service data. Instead, CMS need to negotiate with network operators separately, following non-standard manual processes to request the desired network resources, undergo a technical validation process and once this validation is obtained, a network administrator will undertake the configuration of the associated network devices. A simple example is a customer that scales up an Infrastructure as a Service (IaaS) service to face a peak in demand by doubling the number of its Virtual Machines (VMs). While the creation of VMs and the related LAN configuration in DCs takes few minutes, readjusting the bandwidth of the L2/L3 VPNs used by the customer to access this service will require few days delay following the manual process described above. This delay is a barrier for customers to benefit from the most appealing characteristic of cloud services that is the elastic resource provisioning.

The complete separation between cloud and network resource management (see figure 10), in addition to the absence of automation for the network service management introduce major drawbacks like lack of responsiveness to demand changes. We advocate the cooperation between Cloud Management System and network operators via an open interface, where the CMS can access network performance information and dynamically request the instantiation and configuration of network services.

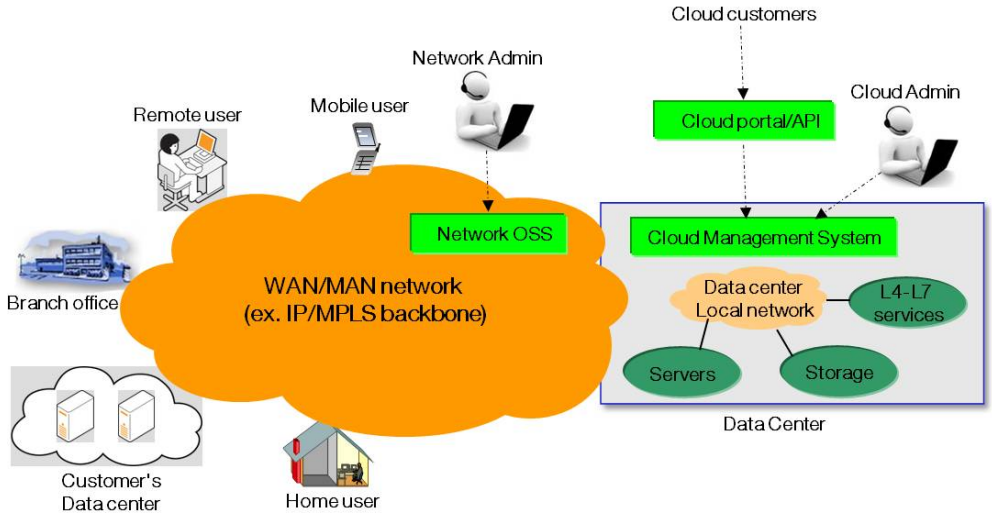

Figure 4: Current Cloud and network architectures

4.6.2 Cooperation between network operators and CMS via CINA

The objective of this cooperation is to allow CMS and cloud customers to request network information, instantiate, configure and customise in an on-demand fashion, ISP-provided network services. The CINA interface could perfectly do this job, allowing a cloud OS or a cloud user to directly query network operator for network performance information and network services (cf. figure 11).

While the CINA specifications do not currently cover all possible network services, where extensions are required to satisfy requirements specific to CMS , these can be easily added. The invocation of network services does not require knowledge of the underlying network technology or specific low level network parameters. It is the job of the network operator to later map a CINA service request into network level configuration request. For example, if the cloud customer requests the Cloud Management System to add a new Virtual Machine (VM) for supporting voice services, the cloud may request the network operator to assign a voice class of service quality for the voice traffic that will be generated by this VM. The network operator then has first to deduce the set of network devices that should be configured, for example these could be the CPEs located on the different customer remote sites that need to access the voice service on the VM created, their corresponding PEs and the PE of the cloud data centre. Then, a configuration request is sent to the NMS to configure these devices. This request must include all the necessary network level parameters that need to be configured for the voice traffic (e.g. DiffServ code as per RFC2475, the new subnet and the bandwidth reservation). In the same way, for a data transfer service (bulk transfer of cloud data), network services can be requested so as to transfer the data as fast as possible. For instance, the services could be related to bandwidth reservation, to traffic prioritisation that the network operators will activate for this specific CMS, for the necessary duration.

However, this cooperation could not happen without security. Indeed, since the network services could be invoked by the Cloud Management System or directly by the cloud service customers, security must be addressed at these two levels. The security considerations underlying the interface between Cloud operator and network operator are not new and available security mechanisms for communications between two entities with established agreements and authentication profiles could be applied. The end-user to network operator interface is much more challenging. Over this interface, strong security mechanisms (such as automated strong authentication) should be applied to maintain control over the network services an end user is able to request. This is particularly true for network services which activation or modification triggers network control plane protocol

messages. For example, in case of a network service allowing on-demand bandwidth reservation for a VPN connection where a request to upgrade or downgrade allocated bandwidth triggers signalling messages in the backbone network, strict authentication and authorisation checks and DDoS protection techniques must be applied on these requests to prevent malicious entities from issuing repetitive requests that may impact the core network control plane.

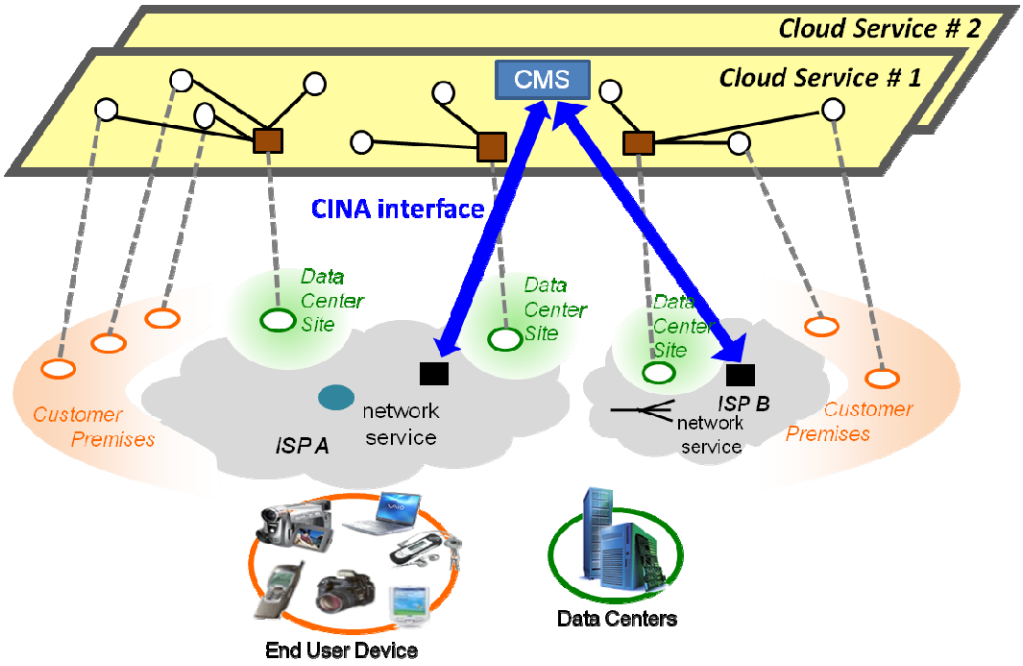

Figure 5: CINA interface for Cloud services

In addition to well known network services such as VPN connectivity, bandwidth on-demand and monitoring, the network operator could provide more advanced services such as virtual network slices and network-based performance and security services (such as firewalling and application acceleration). The next section briefly introduces some of the network services a network operator can offer.

4.6.3 Network services of Interest for CMS

In deliverables D3.1 and D3.2, we presented several network services that the overlay applications might benefit. In this section, we introduce some of the network services a network operator can offer to CMS . Even if those already defined in previous deliverables might be usable by applications deployed in Data centres, we prefer to focus here in those useful for the management of the Cloud.

Information on network performance and ISP preferences

A Cloud Management System could benefit from an improved knowledge of the underlying network performance such as the load, number of hops, delay and jitter of the paths connecting Cloud Data Centres or the paths connecting users to the Cloud Data Centres. Further, assuming appropriate incentive schemes are established between the ISPs and the cloud operators, the ISP preferences for shifting traffic to particular destinations and/or at particular time slots may be taken into account by the cloud operators to reduce theirs and the ISPs' costs. This information will be used to better select the Data Centre where cloud resources should be provisioned for a given customer at any given time. It could also be used to select an alternate DC where a VM should be moved to under certain conditions (ex. the initial DC is overloaded) e.g. by employing VM live migration. This guidance might be based on various metrics such as the needed bandwidth, the latency, the topology (e.g. closest to

the users or triggers for SLA conformance). This can also be useful when users connect to a Cloud service which is distributed across many Cloud Data Centres at different locations: having such an interface could allow the Cloud Management System to redirect the end-users to the appropriate Cloud Data Centre that will provide the best service quality to the client.

Dynamic VPN connectivity

With dynamic VPN connectivity, the Cloud operator and cloud users are enabled to manage dynamically CMS reachability of VPNs that are provisioned by the network operator. A customer having a managed VPN can add, delete and modify the reachability of various cloud resources inside his VPN domain almost in real time. A typical example of this is the hybrid cloud service: a customer having a L3 (MPLS based) business VPN subscribes to an IaaS offer and wants this IaaS service to belong to its intranet, i.e., to be reachable inside the VPN. Once the Cloud operator has provisioned all of the Data Centre infrastructure, it may request the network operator to include this IaaS service reachability into the customer's VPN. Once this is done (which could be done within minutes) the VMs provisioned in the IaaS service will appear in the customer's intranet. The requests sent by the Cloud operator or the cloud customer to query dynamic VPN connectivity will include user-level parameters. The network operator will use these parameters to deduce the network-level parameters required to establish reachability to the cloud resources. For instance, the request may include identification parameters and remote customer sites names that should be enabled to reach the cloud provisioned resources.

Bandwidth on-demand

Offering bandwidth on-demand consists in allowing the cloud operator or cloud customers located in the same network domain to adjust dynamically the network bandwidth from the customer's location up to the cloud service location (ex. a VM on a host). There are various cloud services use cases where this service could be very attractive. An example is business customers who need non-regular massive data transfer to and from the cloud (ex. to answer peak demands during special events). It would be very interesting for these customers to increase their bandwidth (used to access cloud resources) only when this is required instead of paying for a permanent over-provisioned bandwidth.

Cloud internal data transfer scheduling

Typically CMS at some point in time need to replicate data, such as video or user generated, to other data centres especially for global services. These bulk-data transfers between data centres profit from intelligent scheduling, as avoiding diurnal peak traffic demand can help increasing user experience and decreasing billing costs for data centre operators. Provisioning information about time varying traffic patterns can be provided by CINA. In particular in scenarios where multiple administrative domains are connected, a standardised and abstracted way of this load information is required, and can be provided by the 'Cost Schedule'.

L4-L7 network services

The network operator could provide L4-L7 services, including caching and content adaptation that are already explored in ENVISION, but also firewalling, application acceleration and load balancing (dynamically balance the traffic to the DCs according to network conditions). These services are mandatory to guarantee the security and performance of cloud services but are usually provided by specialised devices located on both the cloud provider's DC and on the customer premises. Providing these in-network services on-demand, i.e. within the ISP's backbone network infrastructure, would accelerate their provisioning times allowing the customers to benefit instantaneously from secure and reliable reachability to CMS . An example of this *is an enterprise that purchases a SaaS service* such as a Customer Relationship Management application (CRM such as SAP). These applications have been initially designed to function on a LAN environment, but recently they are being deployed over the WAN (through a VPN); it is necessary to deploy an application acceleration service to guarantee this application performance on the WAN. Typically, the customer will need to order a

specialised device, install and operate it on its premises. With the network operator providing this service on-demand, the application acceleration services may be hosted at the ISP PoPs that are near the customers' sites. Using that, it would be very easy for the customer to provision such a service for applications that are partially hosted in the cloud within minutes or hours.

Measurement and monitoring

Network performance and utilisation reports are an important aspect that a network operator should be able to provide to a Cloud operator as well as to cloud customers. It includes all the relevant indicators on network service utilisation statistics, measurements and alerts handling especially those regarding network SLA violation. Indeed, one of the key characteristics of cloud services is the pay-as-you-go billing model. With the introduction of the network services, an important question is whether the same billing model should be also applied to network services provided by the network operator.

The answer is far from being evident. While the various aspects of the "pay-as-you-go" concept for CMS have been deeply addressed by the industry and are currently in use with access to a large set of related tools, almost no work has been done on the pay-as-you-go aspects of the consumption of network resources that the CMS allocate and utilise. Firstly, we need to define, for each of the network services provided by the network operator, what a pay-as-you-go billing model does exactly means for that service. Then, we will need to see how each such billing model could be implemented within the network infrastructure. It is obvious that, the more fine-grained the usage billing is, the more the technical challenges will come to the fore because that would require more precise measurement indicators. . A technical-economical study should be undertaken to see what could be the various costs and gains of the various billing models, taking into account parameters such as the costs of implementing the required technical solutions and the integration costs with existing Network and Cloud billing systems.

4.6.4 Example of one use-case: Virtual Machine Mobility

AS an example, we can define a scenario where a CMS needs to migrate some Virtual Machines (VMs) between DC locations. This scenario illustrates a seamless VM mobility service providing a "follow-the-sun" approach. We explained it using the topology of the Orange testbed we have set up in the project, with the network regions defined by PIDs in and two CINA servers, one CMS, three DC emulation machines and several end users located at different regions (see figure 12).

The scenario is the following:

- EU (End-User) 1 is working with the application running in DC (Data Centre)
- At 18:00, EU1 finishes her working hours and EU2 takes over her. EU2 is located far from DC1. The CMS (Cloud Management System) detects that DC2 and DC3 are possible candidates for hosting the VM (i.e. they have the necessary resources and adequate functions).
- The CMS then requests the CINA server to get the network map and the cost map with the latency metric as parameter (the application EU1 and EU2 are using is latency-sensitive). Based on network monitoring, the CINA server responds with lower costs for DC2. We assume that the link between PID6 and PID5 is congested and very slow. Providing a dynamic metric of the network performance such as latency in the cost map allows a more efficient selection than a typical ALTO-based system where only routing cost are taken into consideration, which in this case, would have indicated DC3 as the best candidate.
- Before moving the VM, to make the migration as fast as possible, the CMS requests the CINA server to allocate bandwidth dedicated to this transfer between DC1 and DC2 and to prioritise its packets (e.g. using DiffServ marking and forwarding of the packets to a codepoint associated with the VM migration traffic). The CINA server then makes the

necessary network configuration using the appropriate protocols and actions specific to the network operator. Once it is done, the CMS can launch the VM migration.

- EU2 can now work on the application running in DC2.

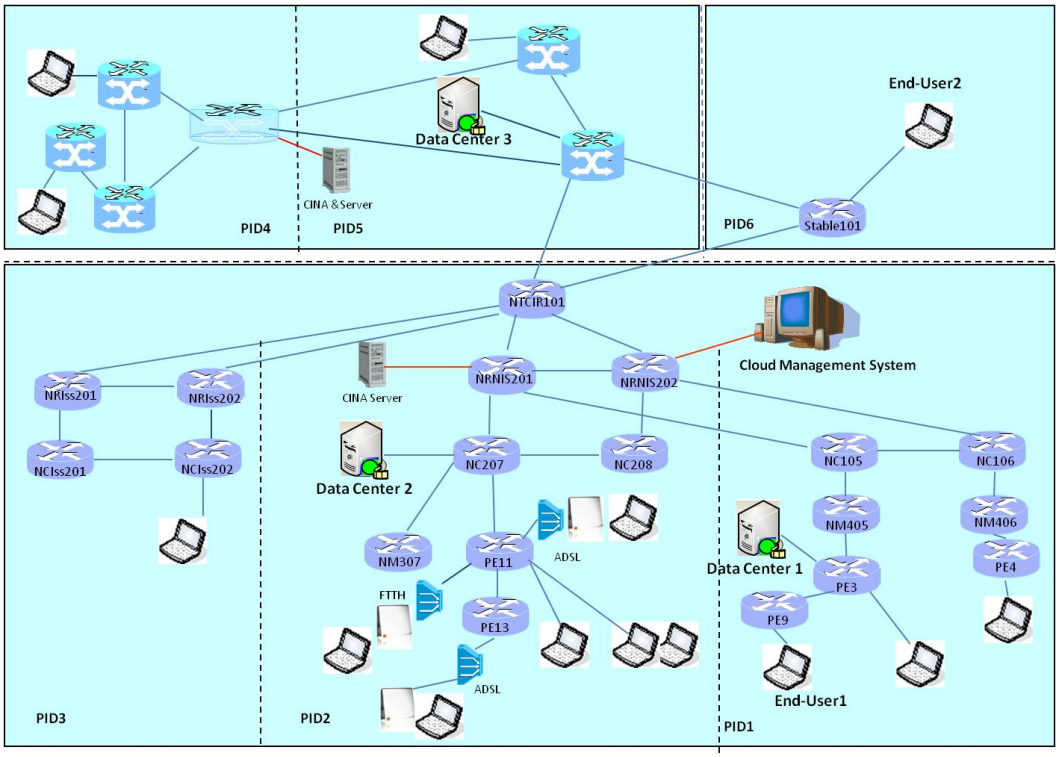

Figure 6: Testbed for CINA & Cloud example

4.7 CINA as interface between Cloud and End-Users

In 4.6, we presented a potential use of CINA between Cloud Data centres and the network operator. A second potential option for CINA is to enable information exchange and service invocation between user applications and cloud infrastructure. Current cloud APIs are proprietary and mostly limited to single service management functions, such as starting a particular service, monitoring a service instance or scaling a service up and down. However they lack of a comprehensive and standardised option for client applications to determine current state of different data centres or particular services, for example in terms of network or processing load or cost. Further there is no automated way of discovering service offerings of a data centre and instantiating the respective services without a major administrative and off-line procedure. In these cases CINA could serve not as a horizontal interface between the application and the network layer, but as a vertical interface between applications and cloud infrastructure, as depicted in the following figure. Here the CINA server is an integral part of the Data Centre and operated by the Cloud provider. In the following section we discuss this deployment option for CINA."

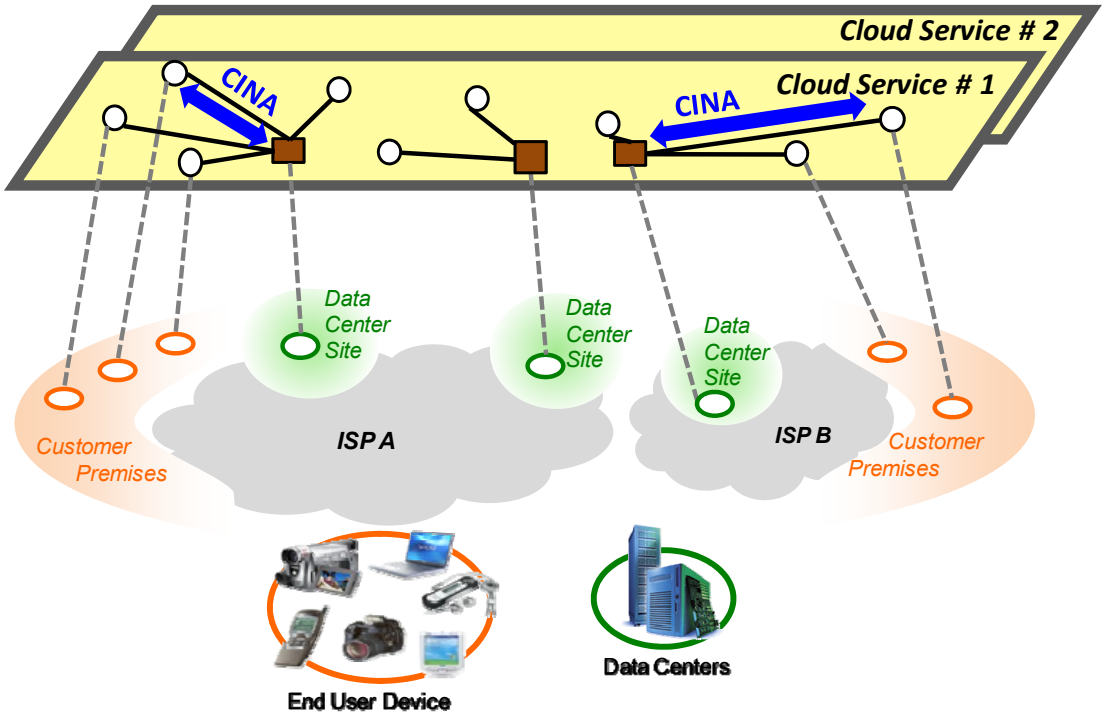

Figure 7: CINA between end-users and CMS

Information provisioning

Although currently cloud environments already monitor various metrics, a typical application, running on a user device or the cloud infrastructure itself, has no standardised way of discovering what kind of information is available and to query this information. Even worse, partially applications do their own monitoring, although the cloud infrastructure is typically in a better position to monitor and could provide this information in greater accuracy. Furthermore an application on its own cannot observe the behaviour of other applications or services, and thus no joint optimisation across application boundaries is feasible as of today.

CINA is one promising option to fill this gap, due to its flexibility in terms of scope an abstracted map provides, and due to its extensibility in terms of what information is provided. CINA is able to meet requirements such as security and privacy of information, by restricting access to information only to trusted applications. Further through the possible abstraction of network regions or server clusters a cloud operator can adjust the level of detail depending on the trust relationship it has. However there are some aspects of CINA that require further research. For example highly dynamic information, such as CPU load, may require a publish/subscribe mechanism that allows application to explicitly react on events quickly. This mechanism is currently not supported in CINA.

One typical use case of information usage by an application is request routing in case several potential content or service replicas exist as destination. Request routing can be done based on various metrics, according to the requirements of the application, e.g. the load of the CPU, memory usage, available bandwidth etc. The selection process for a particular request can then be based on the information provisioned by a CINA server which is operated by a cloud service provider. The CINA service can be queried by an end user application directly or by a external request redirector, based on standard mechanisms such as DNS or HTTP redirect.

Service Instantiation

Today typical cloud offerings require a user to manually administrate and configure her application. The CINA service invocation methods could provide several enhancements and atomisation of this

process. One drawback that could be compensated by CINA is dynamic service discovery on different data centres. For example a service that currently runs on one data centre, but can potentially be instantiated closer to a user as well, can be advertised and finally instantiated by the means of CINA in a fully automated process and thus without the need of a user involvement. Furthermore CINA offers primitives to query service capacity and cost, which can help in the decision process.

Typical use cases for CINA enabled cloud service offerings comprise Network Services as defined in D3.2 e.g. the High Capacity Node or the Caching service. In particular in future cloud environments that are likely to have processing resources highly distributed in the network these Network Services will help applications boosting their capacity in terms of bandwidth or storage thus increasing the user experience.

4.8 A Traffic Optimisation mechanism for CDNs and Clouds based on the CINA interface

New challenges arise for network operators concerned by the growing traffic demands while appropriate Quality-of-Experience (QoE) should be provided for end-users. Recent traffic studies [SAND11] has shed some light on the impact of video streaming services on networks. Netflix, a content provider company, and YouTube, both relying on Content Delivery Networks (CDNs), account alone for more than 35% of the traffic in peak hours in North America. We reasonably expect these services to continue their evolution with more subscribers, higher definition streams, richer content libraries and greater device support. In the same vein, Clouds offer promising development opportunities towards enterprises and extra services dimensions towards consumers. We can expect from these developments serious impacts on the networks requiring the attention of CDN and Cloud Services Providers (CCSPs) and Network Operators (NOs). In such a context, the CINA interface provides the opportunity for these actors to collaborate in order to achieve aligned objectives rather than throwing bandwidth at the problem or digging into solitary responses on each side. We present in the following sub-sections a mechanism for optimising the utilisation of the network resources by upper services. We introduce first a new metric for determining the cost of using the network resources, exposed by the CINA interface or ALTO. We present then the utility of a bidirectional exchange of information between the network and service layers for the traffic optimisation. We expose finally the optimisation problem.

4.8.1 A new metric for determining the utilisation cost of network resources

4.8.1.1 Traditional metrics

ALTO/CINA can provide different types of information to CDNs and other overlay applications. Policies, explicit agreements with network operators and technical challenges statue upon the availability of each type of information. However, the relevance of the exposed costs to requesting services profiles should also be taken into consideration. In the context of a particular service profile such as CDN media streaming or Cloud bandwidth-consuming services, constant and high transmission rates are required for a non negligible period of time. By contrast, web content consists generally in small files which could be retrieved in fraction of time from surrogates. While bandwidth or traffic load metrics may seem appropriate for profiles of heavy-load services, delay and proximity could be sufficient for small web contents. Actually, metrics such as distance in terms of number of hops or delay are the main metrics used in the literature and expected to be provided by ALTO.

Distance is generally computed in terms in number of hops between the source and the destination. However, a smaller number of hops do not always ensure that the path is well provisioned for handling the traffic between the source and destination. In particular situations where two or more paths with the same number of hops are available, a random choice may lead to sub-optimal decision.

CINA costs may also expose end-to-end path delays between the different PIDs defined within the network map. These values can be retrieved by measuring RTTs between edge RN directly or by monitoring the latency between adjacent routers and aggregating the values along the different paths between PIDS. Similarly to distance information, delay is not an appropriate candidate for heavy-load services such as content distribution. Indeed this profile of services bears with some additional delay but cannot afford shortages in transmission rates. Not only delay provides indications and rough estimations about the state and capacity of paths, but also spread measurements in time affects the efficiency of delay-based RR.

In addition, network operators may expose available bandwidth between PIDs as one class of information provided by CINA servers to CINA clients. However two major challenges stand in the way of its deployment. On one hand, aggregating the bandwidth information of the different links composing the path is not possible. On the hand, such information is considered critical by network operators and is not meant to be shared.

Another class of information eventually destined to be exposed through CINA is the routing costs information. However, an insightful understanding of this information is essential to value its relevance for upper services. OSPF costs for instance are defined for building the shortest paths between network nodes. By default, an OSPF cost is inversely proportional to the link capacity. These values can be customised by network administrators. However, they are defined to allow the convergence of the Dijsktra algorithm while satisfying the preferences of the network administrators. They do not provide information about the network conditions compared to other metrics and do not expose the availability of network resources.

4.8.1.2 The delivery cost metric

Although the classes of information we discussed in the previous subsection is not an exhaustive list, it includes the main metrics expected from an ALTO server. Besides, to the best of our knowledge, no other type of information is suggested in the literature or by the IETF ALTOWG. The analysis of the existing ALTO information we carried showed that while some are unusable, others can contribute to the improvement of the overall system performance. However they might incidentally lead to sub-optimal or even counter-productive results. To address the different loopholes of other metrics regarding the profile of heavy-load services offered by CDN or Clouds, we propose a different approach for computing the cost values exposed by the CINA server.

4.8.1.2.1 The delivery cost computation

We define the delivery cost between a source PID and a destination PID as the cost of delivering data on the end-to-end path between them. When comparing multiple paths, lower values indicate higher preferences. To compute the delivery costs, we propose an approach based on the utilisation metric of network links. CDNs and other overlay applications are not allowed to get this information. They have no access to the know-how of NOs including physical links between PIDs, their capacities, the routing schemes and policies, etc. However the cost map provides them with the delivery costs between PIDs. These costs are presented as abstract values that offer no opportunity for inferring other information about the network except delivery costs.

The delivery cost of a path is the accumulation of the delivery costs of the links composing the path. For instance the delivery cost of the path $\pi_{4,6}$ between PID-4 and PID-6 is the sum $\sum(\mathrm{C1}, \mathrm{C5}, \mathrm{C6}, \mathrm{C7}, \mathrm{C8})$ (see Figure 14). However we should first define a safety policy for avoiding using links which utilisation ratios exceed a certain limit, for instance 80% of the capacity of each link. We consider the remaining 20% as a safety margin for uncontrolled traffic. Thus, the first step for computing the delivery costs is to check whether the traffic on any link reaches 80% of its capacity. In this case, the link cost is put to -1, i.e. the link is temporarily unusable (e.g. C3 = −1 make the paths $\pi_{2,6}$ and $\pi_{2,7}$ unusable). This safety policy is decided by NOs and could be customised for each link separately. The next step is the calculation of PID-to-PID delivery costs. If the traffic is below the

safety limit, we calculate the link cost for link I using a link cost function presented in the next subsection. Then we can calculate the end-to-end Path Cost PC_{ij} for $\pi_{i,j}$. If there is one link on the path with a -1 cost, then $PC_{ij} = -1$, i.e. the path $\pi_{i,j}$ must not be used. Otherwise the path cost is defined as the sum of links costs:

$$\boldsymbol{PC_{ij}} = \sum_{\boldsymbol{L_u \in \pi_{ij}}} \boldsymbol{LC_u} \qquad (5)$$

Where L_u is the network link number u.

4.8.1.2.2 The cost function

The cost function definition should fulfil different requirements. On one side, it has to provide appropriate information for the routing functions of overlay services. On another side, it has to respect operators' policies, avoid disclosing confidential information and efficiently reflect the cost of using the network links. As previously discussed, heavy-load services are more concerned about bandwidth than delay. In order to cope with this requirement, we propose a function taking into account an equivalent metric to the utilisation ratio of network links. The utilisation ratio of link u can be defined as the traffic load on the link (LT_u) over its capacity ($LCAP_u$). Meanwhile the cost function should increase as this ratio increases. It has to provide abstract values and satisfy the safety policy. As this function is proper to each network operator, she decides what should be the cost of a link at different utilisation ratios. Given a data set of utilisation ratios and corresponding cost values, mathematical approaches based on interpolation such as Newton-Gregory, Lagrange or polynomial approximations [RAM06], can define a cost function for extrapolating cost values from any utilisation ratio. In our case, we propose an exponential root function which will be used for our simulations:

$$LC_u = \begin{cases} -1, & LT_u \geq 0.8\, LCAP_u \;\forall\, u \\ LC_u = a \times \exp\left(b \times \dfrac{LT_u}{LCAP_u - LT_u}\right), & LT_u < 0.8\, LCAP_u \;\forall\, u \end{cases} \qquad (6)$$

The parameter a is a cost scale factor while b is the growth amplification factor. These parameters may be randomly generated to prevent overlay applications from inferring information about the utilisation ratios of network paths. We considered the exponential function for calculating links' costs because its growth is faster than polynomial ones, i.e. ensuring more cost increase particularly for higher values of traffic. The term $LCAP_u - LT_u$ ensures that the cost tends to infinity when traffic gets closer to the links capacity.

4.8.1.2.3 Analysing the cost function

A path delivery cost is an accumulation of the delivery costs of its links. Thus it genuinely reflects the state of the links. The link cost value is function of the utilisation of the link. It depends on the link's capacity and the actual traffic load or inversely the available bandwidth. By contrast to other classes of information such as distance or delay, delivery costs expose information about the available bandwidth which is more appropriate for high and constant transmission rate services. However, we discussed previously the issues behind exposing bandwidth information to upper services. In this sense, delivery cost information is the best candidate. Indeed, the cost function converts the utilisation of links into abstract values used to compute end-to-end path cost. To some degree, the utilisation of delivery costs is very similar to those of some routing protocols such as OSPF. By default, the OSPF cost of an interface is $^{10^8}/_{capacity}$ in bps. Our cost function takes an additional parameter to capacity, the available bandwidth. At the end, the cost of the path is the accumulation of the costs of links (see Figure 14). OSPF is used at the network layer for determining the best paths

between network nodes. Equivalently, ALTO providing delivery costs can be used at the application layer for determining the best paths between PIDs (see Table 1).

Another aspect which should be carefully addressed as well is the cost function. Results provided by this function are an interpretation of the utilisation of network links. A linear function for instance values much less higher utilisation ratios of network links than an exponential or polynomial function. Let us consider the following linear function $L(u) = 10 \times u, u \in [0,1]$ where u is the utilisation ratio.

We compare the delivery costs for two different paths between the linear function and the exponential function we proposed earlier with a = 1 and b = 2 (see Table 2). The second path should be preferred because one of the links in the first path has reached 70% of its capacity whereas links in the second path are at 40%. We notice that the results of the linear function favour the first path whereas the exponential function recommends the second one. Indeed, the higher utilisation ratio of link L_2 results in a high cost with the exponential function which filters out the first path. Consequently, NOs should pay the utmost attention to the definition of such cost function.

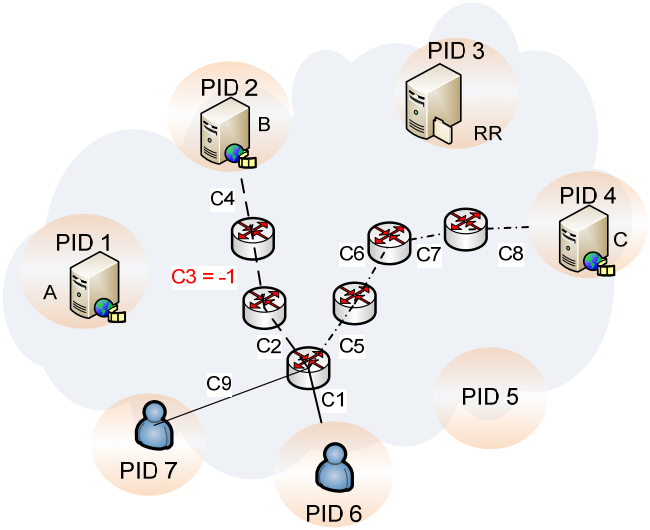

Figure 8: Overview of costs of links between PIDs

4.8.2 Traffic optimisation using the bidirectional CINA interface[4]

4.8.2.1 ALTO limitations

The ALTO framework allows CCSPs to get network-related information from NOs. Traffic optimisations are carried at the application level. The Network Map is used by CCSPs for locating service resources and end-users within the network, i.e. determining the PID where is located each overlay entity from its IP address. Then the Cost Map is used for improving data delivery between CCSPs' end-users and resources. For instance, the CCSP entity responsible for the redirection of end-users to surrogates can use the delivery costs between PIDs to determine the most appropriate surrogate for each user considering the state of network paths. In fact, for each PID containing clients, there is one corresponding PID among those containing surrogates to which users should be redirected. The later is the PID which end-to-end path offers the lowest delivery cost to the PID

[4] The optimisation problem formulation and algorithm design have been suppressed from the public version of this deliverable as the content is currently under review for publication.

hosting the users. For instance, let us consider a Network Map composed of 5 PIDs and a CCSP disposing of surrogates in PIDs 3 and 5. When receiving the Cost Map exposed in Figure 15, the CCSP can define a new table mapping each PID containing clients to a PID containing surrogates as shown in Table 3. For instance, this table states that clients in PIDs 1, 4 and 5 will be redirected to surrogates in PID 5.

Table 1 Comparison between OSPF and ALTO costs

Parameter	OSPF	ALTO
Layer	Network layer	Application Layer
Cost computation parameters	Capacity	Available Bw and capacity
Algorithm	Dijkstra	Iterative
Path cost computation parameters	link-states (OSPF metrics, neighbors, subnets, etc)	routes and delivery costs
Purpose	best path between routers	best path between PIDs

Table 2 Comparison between different cost functions

Path	Link Utilisation (%)			Linear function	Exponential function
	L1	L2	L3		
1	20	70	20	11	11.33
2	40	40	40	12	8

Table 3 Cost Map of proportions of traffic between PIDs {1-5}

PIDs	1	2	3	4	5
1	0	0	70	0	30
2	0	0	65	0	35
3	0	0	0	0	0
4	0	0	0	0	100
5	0	0	0	0	0

As the Cost Map provides static values of delivery costs, the mapping between PIDs stays valid until an update of the Cost Map. Thus end-users in the same PID will be using the same path as they are redirected to the surrogate within the PID corresponding to their PID in the mapping table. Consequently, the Cost Map approach could lead to sub-optimal decisions if it is not updated frequently. Indeed, since path utilisation increases with an increasing number of incoming end-users, a different path to an alternative surrogate in another PID may offer lower delivery costs after a certain period of time. In the same vein, in periods of dense traffic where the network resources

utilisation ratios are already high, an important number of clients in a PID redirected using the same Cost Map may bring heavy traffic on the utilised network links and could lead to congestion problems, loss of packets and poor QoE perceived by end-users. Nevertheless, a frequent update of the Cost Map to mitigate the problem is not an easy task. It requires heavy monitoring on critical network equipments. In order to avoid such situations, we propose in the following section a cross-layer framework implicating CCSPs and NOs by contrast to ALTO where optimisations are carried at the application level.

4.8.2.2 CINA information and services

By taking into consideration the needs and constraints of the different CCSPs competing for their resources, NOs can provide enhanced guidance information to CCSPs compared to static delivery costs.

4.8.2.2.1 Cost Map of proportional type

We introduce a new *'cost-type'* for the Cost Map, *'proportional'*. Instead of providing routing costs to CCSPs, NOs provide the proportions of traffic or sessions between PIDs containing end-users and PIDs containing surrogates providing services to these end-users. Figure 15 shows an illustration of a Cost Map providing the information about proportions of traffic between the different PIDs containing end-users (PIDs 1, 2 and 4) and those containing surrogates (PIDs 3 and 5). In this example, the *'cost-mode'* is *'percentage'*. For instance, 70% of the ingress traffic to PID 1 should be delivered by surrogates in PID 3, the remaining by surrogates in PID 5.

Such a cost type mitigates the limitations of static routing costs defined by ALTO. Indeed, it ensures a good balance of traffic on network links between surrogates and end-users with respect to the links utilisation ratios. NOs avoid situations where certain network links are heavily exploited while others remain poorly used. Similarly, CCSPs benefit from better QoE by using network paths presenting better availability and lower congestion risks.

```
{
"cost-mode" : "percentage",
"cost-type" : "proportional",
"map-vtag"  : "225006996",
"map" : {
"PID1":{"PID3":70,"PID5":30},
     "PID2":{"PID3":65,"PID5":35},
     "PID4":{"PID5":100}
   }
}
```

Figure 9: Illustration of a Cost Map of proportions

In order to be able to determine the Cost Map of proportions specific to each CCSP, NOs need to be aware of the location and resource availability of surrogates and of the needs in terms of traffic. For instance, a NO cannot recommend a number of sessions to be redirected to a surrogate exceeding the surrogate's capacity. Similarly, it cannot recommend the balancing of a PID traffic for a Cloud service without knowledge about the total inbound and/or outbound required traffic. To allow such interactions between the layers, we introduce a new service to the CINA interface allowing CCSPs to expose information to NOs without disclosing critical information such as IP addresses.

4.8.2.2.2 Constraint Map service

This service allows CINA clients implemented by CCSPs to expose their needs and constraints to NOs. The information uploaded to the CINA server may provide for instance the locations and capacities of

surrogates or the locations and traffic requirements for end-users. Depending on the CCSP type and preferences, information may disclose the number of entities or amounts of traffic. Additional constraint types and modes can be defined for future usages. An example of a Constraint Map is shown in Figure 16. It provides multi-constraint information following the multi-cost scheme proposed to the ALTO WG [RAN12]. The first constraint exposes the location and capacities of a CDN provider surrogates in terms of traffic in Gbps. For instance, the surrogate in PID 3 is capable of handling 20 Gbps of the service traffic. The second constraint exposes the forecast of the CDN provider in terms of inbound traffic to each PID where a certain number of end-users is expected. CCSPs can rely on their statistical analysis of logs for estimating the expected number of users or network traffic to be generated.

```
{
   "constraint-mode": ["Gbps" , "Gbps"]
   "constraint-type":["surrogate
   capacity", "inbound traffic "]
   "map-vtag"  : "225006996",
   "map" : {
     "PID1":{[0,6]},
     "PID2":{[0,4]},
     "PID3":{[20,0]},
     "PID4":{[0,10]},
     "PID5":{[5,0]},
   }
}
```

Figure 10: Illustration of a Constraint Map

NOs and CCSPs use the CINA interface for exchanging the information required for the optimisation. Figure 17 illustrates the basic call flow for exchanging information in the form of maps between a CINA client and a CINA server.

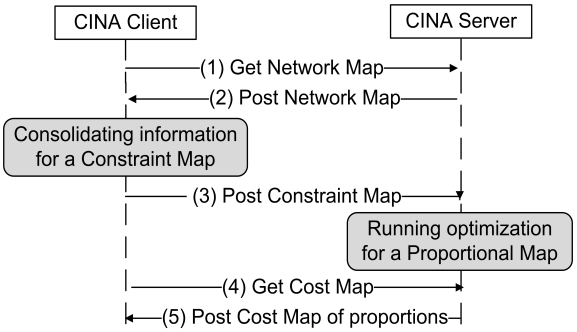

Figure 11: Overview of the exchange of maps through the CINA interface

5. CONCLUSIONS

5.1 Conclusions from this report

In this deliverable, we have presented the latest updates in the definition of the CINA interface and our efforts to push it in the standardisation bodies, more precisely the IETF ALTO working group for this work. The proposed drafts, issued from ENVISION, are mainly related to the discovery and the multi-cost approach.

The progress on the implementation of the network services, such as the Multicast and the High Capacity node, is also described in this deliverable, in a complementary approach, with regards to the existing text in [D3.1] and [D3.2]. The network services will be evaluated in a real testbed, to prove the feasibility of such approach. For instance, the multicast will be deployed in the Orange testbed with several end-users, in a defined scenario, to prove the efficiency of being able to dynamically invoke the instantiation of multicast service by the service provider. This will be done within WP6 and be documented in the final WP6 deliverable.

As discussed in the last review meeting with the reviewers, we also investigated the use of the CINA interface for Cloud services. It is depicted in this deliverable with a presentation of additional network services a Cloud Provider is interested in. An example is introduced to highlight it : the migration of VM in a Cloud system.

The time and space traffic shifting study, initiated in [D3.2], has been completed and is detailed in this document, with the latest evaluation results showing the interest for ISPs for such an approach.

Finally, another study is described in this deliverable to highlight the benefit of a collaboration between service providers and network operators, as advocated by the ENVISION project. With analytical model and simulation, we proved that exchanging maps in a bidirectional way (such as the cost map, the constraint map) can lead to great improvement in the network load (utilisation of link) for a lower cost for ISPs.

5.2 Overall conclusions on the ENVISION Interface, Network Monitoring and Network Optimisation Functions

The work done in WP3 related to the specifications of the CINA interface as well as discussions we had when presenting it at IETF ALTO meetings or in conferences highlighted the need for such a collaboration interface between service providers and network operators. Service providers are in favour of being able to request some network services an operator can expose, if it allows to optimise the delivery and improve the QoE for end-users. We can now see some discussions between OTTs and network operators to find agreements about the OTTs data to be delivered in the networks in an efficient way and without negative effects for the network operators. The CINA interface might be a way to formalize this kind of collaboration.

Technically speaking, the choice of a Web service, with a Restful approach and JSON encoding for implementing CINA, was a good choice, because it allows rapid debugging and has proven to be efficient enough for such primitives. Furthermore, it is now widely used and its adoption by others actors might be facilitated.

The network services (multicast, caching, high capacity node) we have thought as added-value services for both the service providers and the network operators were good choices since they really enable an optimisation of the network while improving the QoE for the services. We can now see recent work related to multicast and bittorrent for instance, proving the interest for such a P2P network to use a native multicast facility. The discussions we had with some providers also highlighted their interest for being able to deliver their content with multicast capabilities. Concerning the caching and the high capacity node, their choice was also effective since we can now

see work related to CDN infrastructures and ALTO in the IETF ALTO and IETF CDNi working groups. It was something we proposed in Envision, even if it was with CINA instead of ALTO, but having similar objectives. Finally, in the second part of the project, we also investigated and presented the use of CINA for Cloud services and this idea is now becoming adopted by the Cloud networking community, in relation with the SDN approach, and it can be seen as the North interface of the SDN architecture. CINA might be an instance of this North interface, in an approach named “Network-as-a-Service” in the Cloud environment.

The evaluation performed in WP6 has proved the technical feasibility of the CINA approach and the dynamic instantiation of network services and the simulation performed within WP3 and WP6 highlighted the benefits the actors can get using such a collaboration interface, both in terms of network optimisation and Quality of Experience. The various simulations, related for example to the time-space shifting of traffic or the selection of paths depending on the load for CDN or Cloud systems, allows to argue that the CINA approach proposed within the Envision project is a promising one and that actors can go in this direction.

REFERENCES

[AMD10] Antoniades, Demetris and Markatos, Evangelos P. and Dovrolis, Constantine. One-click hosting services: a file-sharing hideout. Proc. of ACM/SIGCOMM IMC, pages 223–234, 2009.

[AMSU11] Ager, Bernhard and Mühlbauer, Wolfgang and Smaragdakis, Georgios and Uhlig, Steve. Web content cartography. Proc. of ACM/SIGCOMM IMC, pages 585–600, 2011.

[ARM10] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," Commun. ACM, vol. 53, no. 4, pp. 50–58, Apr. 2010. [Online]. Available: http://doi.acm.org/10.1145/1721654.1721672

[CAIDA] The Cooperative Association for Internet Data Analysis . (2012, Mar.) Trace Statistics for CAIDA Passive OC48 and OC192 Traces. [Online]. http://www.caida.org/data/passive/trace_stats/

[CB08] David R. Choffnes and Fabián E. Bustamante. Taming the torrent: a practical approach to reducing cross-ISP traffic in peer-to-peer systems. Proc. of ACM SIGCOMM, 2008.

[CLRS10] Parminder Chhabra and Nikolaos Laoutaris and Pablo Rodriguez and Ravi Sundaram. Home is where the (fast) Internet is: Flat-rate compatible incentives for reducing peak load. Proceedings of ACM HomeNETS, 2010.

[CLY+11] Ruben Cuevas Rumin and Nikolaos Laoutaris and Xiao Yang and Georgos Siganos and Pablo Rodriguez. Deep Diving into BitTorrent Locality. Proc. of IEEE INFOCOM, 2011.

[D3.1] The ENVISION project, "D3.2: Initial Specification of the ENVISION Interface, Network Monitoring and Network Optimisation Functions", February 2011 (updated in May 2011), http://www.envision-project.org/deliverables/envision_d3.1-v2.pdf

[D3.2] The ENVISION project, "D3.2: Refined Specification of the ENVISION Interface, Network Monitoring and Network Optimisation Functions", January 2012, http://www.envision-project.org/deliverables/envision_d3.2_final_public.pdf

[DHKS09] Xenofontas Dimitropoulos and Paul Hurley and Andreas Kind and Marc Ph. Stoecklin. On the 95-percentile billing method. Proc. of PAM, 2009.

[FPS+12] Benjamin Frank and Ingmar Poese and Georgios Smaragdakis and Steve Uhlig and Anja Feldmann. Content-aware Traffic Engineering. Proceedings of ACM SIGMETRICS/Performance 2012, London, UK, 2012.

[GQX+04] Goldenberg, David K. and Qiuy, Lili and Xie, Haiyong and Yang, Yang Richard and Zhang, Yin. Optimizing cost and performance for multihoming. Proc. of ACM SIGCOMM, pages 79–92, 2004.

[JHC11a] Carlee Joe-Wong and Sangtai Ha and Mung Chiang. Time-Dependent Broadband Pricing: Feasibility and Benefits. Proc. of IEEE ICDCS, 2011.

[JHC11b] Carlee Joe-Wong and Sangtai Ha and Mung Chiang. Time-Dependent Internet Pricing. Proc. of Internet Technologies and Applications Conference (ITA), 2011.

[JZRC09] W. Jiang and R. Zhang-Shen and J. Rexford and M. Chiang. Cooperative Content Distribution and Traffic Engineering in an ISP Network. ACM SIGMETRICS, 2009.

[KH95] Anatole Katok and Boris HasselBlatt. Introduction to the Modern Theory of Dynamical Systems. Cambridge University Press, 1995.

[KUHN51] H. W. Kuhn and A. W. Tucker, "Nonlinear programming," Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, Berkeley and Los Angeles, 1951.

[KIE12] Kiesel, S., Stiemerling, M., Schwan, N., Scharf, M., Song, H., "ALTO Server Discovery", draft-ietf-alto-server-discovery, 2012.

[LINX] LINX. https://www.linx.net/service/servicefees.html, 2012.

[LIU02] E. Liu, "A Hybrid Queueing Model for Fast Broadband Networking Simulation," PhD dissertation, Queen Mary, University of London, London, 2002.

[LR08] Nikolaos Laoutaris and Pablo Rodriguez. Good things come to those who (can) wait: or how to handle Delay Tolerant traffic and make peace on the Internet. Proc. of ACM HotNets, 2008.

[LSRS09] Nikolaos Laoutaris and Georgios Smaragdakis and Pablo Rodriguez and Ravi Sundaram. Delay Tolerant Bulk Data Transfers on the Internet. Proc. of ACM SIGMETRICS, 2009.

[NOC06] J. Nocedal and S. J. Wright, "Sequential Quadratic Programming," in Numerical Optimization. Springer, 2006, ch. 18.

[O01] Andrew Odlyzko. Internet pricing and the history of communications. Comp. Netw., 36:493–517, 2001.

[PFA+10] Poese, Ingmar and Frank, Benjamin and Ager, Bernhard and Smaragdakis, Georgios and Feldmann, Anja. Improving content delivery using provider-aided distance information. Proc. of ACM/SIGCOMM IMC, pages 22–34, 2010.

[RAN12] Randriamasy, S., Schwan, N., "Multi-Cost ALTO", draft-randriamasy-alto-multi-cost, 2012

[RAND12] Randriamasy, S., Schwan, N., "ALTO Cost Schedule", draft-randriamasy-alto-cost-schedule, 2012

[RFC4472] Durand, A., Ihren, J., and P. Savola, "Operational Considerations and Issues with IPv6 DNS", RFC 4472, April 2006.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

[SAND11] Sandvine, "Global Internet Phenomena Spotlight - Netflix Rising," Sandvine, 2011.

[S84] Michael J. Smith. The stability of a dynamic model of traffic assignment – an application of a method of Lyapunov. Transp. Science, 18(3):245–252, 1984.

[SCH86] K. Schittkowski, "NLPQL: A FORTRAN subroutine solving constrained nonlinear programming problems," Annals of Operations Research, vol. 5, pp. 485-500, 1986.

[SCH12] Schwan, N., Roome, W., "ALTO Incremental Update", draft-schwan-alto-incr-updates, 2012

[SLR10] Rade Stanojevic and Nikolaos Laoutaris and Pablo Rodriguez. On economic heavy hitters: Shapley value analysis of 95th percentile pricing. Proc. of ACM/SIGCOMM IMC, 2010.

[VLF+11] Vytautas Valancius and Christian Lumezanu and Nick Feamster and Ramesh Johari and Vijay Vazirani. How many tiers? Pricing in the Internet transit market. Proc. of ACM SIGCOMM, 2011.

[W52] J. G. Wardrop. Some theoretical aspects of road traffic research. Proc. of the Inst. of Civil Engineers II, 1:325–378, 1952.

[WONG81] J. W. Wong and S. S. Lam, "Queueing Network Models of Packet Switching Networks," North Holland Publishing Company, 1981.

[XYK+08] Haiyong Xie and Yang Richard Yang and Arvind Krishnamurthy and Yanbin Liu and Avi Silverschatz. P4P: Provider portal for applications. Proc. of ACM SIGCOMM, 2008.