COntent Mediator architecture for content-aware nETworks

European Seventh Framework Project FP7-2010-ICT-248784-STREP

Deliverable D3.1
Interim Specification of Mechanisms, Protocols and Algorithms for the Content Mediation System

The COMET Consortium

Telefónica Investigación y Desarrollo, TID, Spain
University College London, UCL, United Kingdom
University of Surrey, UniS, United Kingdom
PrimeTel PLC, PRIMETEL, Cyprus
Warsaw University of Technology, WUT, Poland
Intracom SA Telecom Solutions, INTRACOM TELECOM, Greece

For more information on this document or the COMET project, please contact:

Prof. George Pavlou
g.pavlou@ee.ucl.ac.uk
Department of Electronic & Electrical Engineering
University College London
Torrington Place, London,
WC1E 7JE
UK

Document Control

Title: Interim Specification of Mechanisms, Protocols and Algorithms for the Content Mediation System

Type: Public

Editor(s): Wei Koong Chai, Ioannis Psaras

E-mail: w.chai@ee.ucl.ac.uk, i.psaras@ee.ucl.ac.uk

Author(s): Wei Koong Chai, Ioannis Psaras, Marinos Charalambides, George Pavlou (UCL)

Gerardo García de Blas, Luis Enrique Izaguirre Gamir, Francisco Javier Ramón Salguero (TID)

Ning Wang, Lei Liang (UniS)

Andrzej Beben, Jaroslaw Sliwinski, Jordi Mongay Batalla, Wojciech Burakowski, Piotr Wisniewski (WUT)

Sergios Soursos, Vasiliki Kamariari and George Petropoulos (ICOM)

Sergey Kuleshov and Chrysovalanto Kousetti (PrimeTel)

Doc ID: D3.1-v1.0.doc

AMENDMENT HISTORY

Version	Date	Author	Description/Comments
V0.1	September 17, 2010	Wei Koong Chai	First version, ToC
V0.2	September 28, 2010	Wei Koong Chai	Revised ToC
V0.3	October 16, 2010	Wei Koong Chai	First integrated version of the deliverable with initial contributions from all partners.
V0.4	November 16, 2010	Wei Koong Chai	Second integrated version of the deliverable with incomplete consolidated contributions.
V0.5	November 30, 2010	Wei Koong Chai	Third integrated version using proper deliverable template with all consolidated contributions.
V0.6	December 7, 2010	Wei Koong Chai	Fourth integrated version.
V0.7	January 2, 2011	Wei Koong Chai	Fifth integrated version – complete full version.
V0,8	January 11, 2011	Wei Koong Chai	Last version before final submission.
V1.0	January 14, 2011	Wei Koong Chai	Final version ready for submission.

Legal Notices
The information in this document is subject to change without notice.
The Members of the COMET Consortium make no warranty of any kind with regard to this document, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The Members of the COMET Consortium shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Table of Contents

1 Executive Summary 6

2 Introduction 7

2.1 The Role of WP3 and this Deliverable 7

2.2 The Structure of this Deliverable 7

3 Overview 9

3.1 Background 9

3.2 High-level View of Content Mediation 10

4 State-of-the-Art 11

4.1 Current Systems 11

4.1.1 The Domain Name System (DNS) 11

4.1.2 The Handle System 14

4.2 Recent Content-based Proposals 15

4.2.1 Introduction 15

4.2.2 Approaches 16

4.2.3 Conclusion 20

4.3 Resource and Content Identification 20

4.3.1 The Current Resource Identification Scheme 20

4.3.2 The Recently Proposed Content Naming Scheme 22

4.3.3 Synthesis of the Schemes 24

5 Content-based Resolution and Delivery 26

5.1 Content Record-based Resolution Approach 26

5.1.1 Overview 26

5.1.2 Content Naming Scheme 27

5.1.3 Unification of Content Access Scheme 28

5.1.4 Content Publication 30

5.1.5 Content Resolution 32

5.1.6 Preparation for Content Delivery 44

5.1.7 Interim Study 46

5.2 Coupled Resolution and Delivery Approach 47

5.2.1 Overview 47

5.2.2 Content Naming Scheme 48

5.2.3 Unification of Content Access Method 49

5.2.4 Overview of Basic Operations 49

5.2.5 Content Publication 52

5.2.6 Content Resolution 53

5.2.7 Preparation for Content Delivery *55*
5.2.8 Interim Study *55*
6 Awareness in the Content Mediation Plane **58**
6.1 Content Server awareness 58
6.1.1 Achieving Content Server-awareness in the decoupled approach *59*
6.1.2 Achieving Content Server-awareness in the coupled approach *61*
6.2 Network-awareness 63
6.2.1 Scope of network state information *63*
6.2.2 Achieving Network-awareness in the Decoupled Approach *64*
6.2.3 Achieving Network-awareness in the Coupled Approach *65*
7 COMET-ENVISION Interface **67**
8 Summary and Conclusions **68**
9 References **70**
10 Abbreviations **73**
11 Acknowledgements **76**

1 Executive Summary

This deliverable presents the interim specifications on the protocols and algorithms associated with content mediation plane (CMP) functionalities for content publication and resolution in multi-domain environments.

The document begins with a short overview of the 2-plane approach in the COMET project stemming from the discussion on the transition of the Internet model from a host-centric to a content-centric one. We then proceed to provide a comprehensive state-of-the-art review on pertinent publication and resolution frameworks. We study and compare the systems that are currently in use in the real world and those that are proposed in the recent research literature, learning from the past lessons while keeping our design ahead with the latest and future foreseeable requirements and challenges. Specifically, the Domain Name System (DNS) and the Handle system are analyzed while the new approaches (e.g., Van Jacobson's Networking Named Content proposal, LIPSIN from the EU PSIRP project, etc.) are studied. In addition to those, since content is being considered as the central primitive in the network, we have also attempted to synthesize the current resource identification schemes (e.g., URL, DOI) with the recently proposed content naming schemes (e.g., self-certifying flat names).

Within the COMET project, we develop two approaches for content naming and resolution based on the high-level COMET architecture. The content record-based approach, a more natural evolution of the current Internet system towards a content-centric one, is presented first. It follows a look-up based paradigm with its foundation built upon the current DNS system which would allow gradual deployment with minimal disruption. The second approach, on the other hand, follows a hop-by-hop hierarchical resolution paradigm. It involves more fundamental changes to the current Internet operations whereby the argument is that the original design principles of today's Internet are essentially unsuitable for the current content-based usage of the Internet. We specify how content is named and the content access methods for both approaches. The content publication and resolution operations are then described. We discuss how the different functional blocks under the COMET architecture collaborate to achieve the two content-based frameworks. We introduce several new entities in both approaches that house these content manipulation functions. Finally, we also explain the preparatory stage of the content delivery operation for both approaches.

While section 5 describes the core publication and resolution operations that enable the discovery and access of content across the Internet, section 6 details how the COMET mediation system achieves awareness that enhances the two aforementioned operations. In essence, there are two types of awareness that we discuss – (1) content server-awareness and (2) network-awareness. With the added awareness, the content discovery and retrieval can be optimized based on various conditions. For instance, the best server can be chosen for delivery of the content during the resolution operation rather than simply returning the first server found. The section describes separately how the two approaches achieve the two types of awareness as they follow distinctly different methodologies in the content manipulation functions.

In section 7, we provide the initial work on the collaboration between COMET and ENVISION projects. We provide the preliminary high-level specification of the COMET-ENVISION interface and the possible performance metrics that could be communicated from the COMET system to the ENVISION one.

All the interim design and specifications specified in this deliverable will be further enhanced in WP3 during the rest of the period of the corresponding tasks, and their final version will be presented in D3.2 in Month 21.

2 Introduction

2.1 The Role of WP3 and this Deliverable

The overall objective of the Content Mediator architecture for content-aware networks (COMET) project is to define a novel content-oriented Internet architecture that radically simplifies content access and supports content distribution in a network-aware manner. Work Package 3 of the COMET project undertakes the specification and implementation of a location-independent content naming and resolution architecture that forms a unified content-centric platform. It elevates content such as videos, pictures, blogs etc in the Internet to the fore where the network operations revolve around these media content rather than the network elements or communication infrastructure (e.g. network routers). For instance, a request to access for a specific content in the Internet is based on the location-independent name of this content rather than the location of the server hosting it.

To achieve this shift from the current host-centric networking paradigm to a content-centric one, there are several main issues that require re-design / re-thinking in a holistic manner. The core questions are:

- How content in the Internet is labelled or identified in the COMET system in a unified way without being dependent on the communication infrastructure (such as the location of hosts)?
- Based on the content naming or labelling scheme, how can COMET support efficient, robust and fast content request resolution (i.e. from the content request issued by an end user to the commencement of the transmission of the requested content itself to the requesting user)?
- Upon the successful discovery of a content, how can COMET mediate the delivery path of the said content to ensure the best quality of experience for the end users?

This document reports the results from WP3 in the first year of the COMET project. Specifically, two main approaches on content naming and resolution are being developed. The first approach follows more closely the current operation of Internet where the aim is to build a content-based resolution framework that is readily deployable in the current Internet with minimal disruption. It draws upon the current Domain Name System (DNS) [15] (Section 4.1.1) and Handle System [19] (Section 4.1.2) which are both widely used. A benefit of this approach is the support for gradual incremental deployment of the developed content resolution framework.

The second approach, on the other hand, follows a more radical path and tackles the problem without being constrained by the current Internet infrastructure. With fewer constraints, the design aims to change the way Internet works today from the root, breaking some of the design principles of the original Internet. This approach is envisioned to be disruptive to the current Internet as it requires substantial changes to many components (e.g. networking protocols, addressing etc).

2.2 The Structure of this Deliverable

This deliverable mainly focuses in providing the interim specifications of the two approaches mentioned above. The rest of the document is structured as follows. Chapter 3 provides a high-level view of the COMET mediation framework so as to put readers in context for better understanding the rest of the document. Chapter 4 provides the state of the art review where past and recent effort on the various design issues and aspects are examined critically so as to learn from the previous lessons while improving on the present. Chapter 5 constitutes the interim specifications and descriptions on the two main content-based resolution and delivery approaches for COMET (i.e. the decoupled and coupled approaches). It details the concepts and the main operations of these approaches while providing some insights into how the approach should work when finalized.

Chapter 6 describes how COMET achieves or extends its awareness outside its own system so as to provide enhanced or optimal decision making on several issues such as optimal content delivery path and server selection. In Chapter 7, the initial specification of the COMET-ENVISION interface is provided. Chapter 8 gives an intermediate summary on the achievements so far of WP3 in COMET and its progress in relationship with the other work packages.

3 Overview

3.1 Background

Since its introduction, the Internet has continuously grown and evolved to encompass various uses both in work and social life. The original Internet model focused mainly on connecting machines whereby addresses point to physical end-hosts and routing protocols compute routes to specific destination endpoints. However, nowadays the Internet is primarily used for transporting content/media, where a high volume of both user-generated and professional digital content, e.g. webpages, downloadable movies/songs, live video streams, etc., is delivered to users who are usually only interested in the content itself rather than the location of the content sources. Human needs along with the nature of communication technologies have transformed the Internet into a new content marketplace generating revenue for various stakeholders. In fact, the Internet is rapidly becoming a super-highway for massive digital content dissemination.

However, the initial design of the Internet architecture has not taken such scenario into account. In fact, despite the fact that content access and distribution becoming the main use of today’s Internet, the “content” itself does not have its own namespace within the current architecture.

In today’s Internet, there are two global namespaces, namely the domain name system (DNS) [15] names which (somehow) reflects domain structures and IP addresses which reflects network topology. For content access, the workaround currently being employed basically uses DNS system to identify content. This, however, overloads the DNS names and rigidly binds them with specific domain / network location. Such an approach makes accessing some contents more difficult especially in the case where a content is being moved or replicated. Another implication of this approach is that DNS names inherently reflect administrative structure while the content itself not necessarily so. Since the domain structure rarely changes, content is often changed either in ownership or location.

As it has been observed, the current communication model adopted by the Internet is host-centric (i.e. host to host communication) whereby addresses point to physical hosts and routing algorithms also compute routes directed to specific hosts. The need of a paradigm shift is apparent and many research studies are being directed towards realizing a content-centric network whereby contents no longer take a backseat but are considered foreground in the Internet. Table 1 shows the paradigm shift from a host centric to a content centric communication model.

Table 1: Paradigm shift from host centric to content centric communication model

Host Centric	Content Centric
Host to host communication model	Access to content but oblivious of host location(s)
Address to host	Name for content
Routing to host	Routing to content
Security considerations focus on host authentication	Inherent security features on content
Reliance on securing the transmission infrastructure (the host, the channel etc)	Content integrity must be verifiable; not relying on secure transmission channel.

The shift from host centric to content centric communication model requires foremost a novel content naming and resolution architecture. This is the root to realizing the paradigm shift as almost all other functionalities adapting to content centric model are dependent on how content is named.

3.2 High-level View of Content Mediation

The COMET project proposes the Content Mediation as the way to move from the host centric to the content centric communication model in the Internet. COMET sets out to unify the diffusion of digital media content in the Internet by mediating content access on the underlying data transmission and network technologies. It aims to harmonize the two worlds (contents and network) so that a robust, efficient and flexible content delivery can be achieved. The COMET system allows Internet Service Providers (ISPs) to act as mediators for content publication and distribution. This mediation simplifies access to content by offering unified interfaces to both content providers and consumers whereby content is treated as the first-class citizen in the Internet. As of now, the Internet does not recognize content as a class of object but only treats it as a bitstreams to be shuffled around different nodes. This leads to serious limitations to how content can be manipulated more efficiently.

Moreover, by taking into account the network and server conditions through the COMET system, the mediation offered by ISPs can also improve content delivery in terms of quality and effective bandwidth utilization. Since today's networks are unaware of the content they are transporting, networks cannot apply the most appropriate end-to-end transport strategy to provide the adequate quality of experience for the end users. Therefore, content delivery is done purely based on the network conditions, neglecting the specific requirements of different content being delivered.

The COMET project follows a 2-plane approach for mediating content dissemination. We define an overlay named the Content Mediation Plane (CMP) that offers the necessary services for content access and distribution. CMP provides the unified interfaces to both content providers and consumers. The second plane in the 2-plane approach is called the Content Forwarding Plane (CFP) which is in charge of the delivery of content. Both planes collaborate to achieve network and server awareness to ensure optimal content delivery across the Internet. Detail elucidation of the COMET 2-plane approach and the architecture for supporting this approach can be found in D2.1 [47] and D2.2 [48] respectively.

In COMET, a content to be made available for access in the Internet goes through both content publication and consumption stages where the content consumption stage further consists of content resolution and content delivery stages. This deliverable mainly focuses on the content publication and content resolution stages with D4.1 [49] providing the details on the content delivery.

4 State-of-the-Art

4.1 Current Systems

In this sub-section, we present two current resolution systems that are widely used. We study their naming and resolution systems in relation to the COMET's objectives, drawing lessons and experiences from these past successful systems.

4.1.1 The Domain Name System (DNS)

The first system that has been assessed in order to develop a suitable naming architecture and resolution system for the COMET architecture is the Domain Name System (DNS).

This system is supported by the fact that end users can access data, resources or content stored in the Internet by following a content naming scheme based on URLs (Uniform Resource Locators) [17]. These URLs, which are described in 4.3.1.1, are strings of characters used to identify a name or a resource on the Internet. URLs are defined as:

```
scheme://host:port/path,
```

where the *scheme* indicates how to access the resource, the *host* and *port* indicate where to access the resource (the *host* can be identified either by the IP address or, much commonly, by a domain name), and the *path* indicates the specific resource.

Typically domain names are used to identify hosts, so that, for the sake of simplicity, URLs can be seen as *scheme://domain:port/path*. *Domains* are strings of characters used in order to facilitate end users to remember and identify hosts, and therefore facilitate the access to resources. These domains [15][16] are organized in a tree structure which starts with the root level that points to a set of Top Level Domains (TLD), and further sub-domains are built in subsequent branches of the TLDs as can be shown in Figure 1.

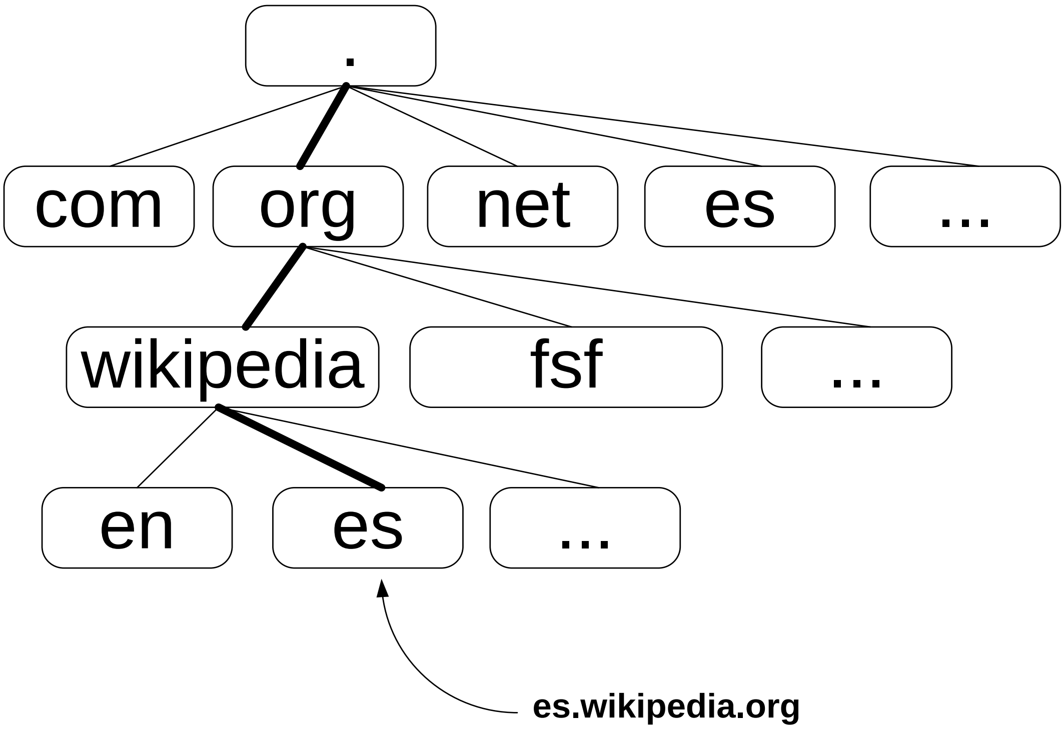


Figure 1: Domain names tree structure

The DNS allows the association of these domains to certain properties — IP addresses, mail agent, etc. — needed to solve where a specific resource resides; these properties are stored in so-called DNS Resource Records associated to a domain. Typically, the DNS is used by end user devices to translate from domain names in a URL to IP addresses, as a first step before accessing a host.

Name servers are the repositories of both the tree structure and the DNS Resource Records. The tree structure is maintained by requiring each name server to know the data that allows access to name servers of the sub-domains. For example, root name servers contain a list of names and numeric IP addresses of the name servers for all TLDs, so they can answer requests returning a list of the designated name servers for the appropriate TLDs.

Name servers can answer queries in a simple manner; the response can always be generated using only local data, and either contains the answer to the question or a referral to other name servers "closer" to the desired information. This process is illustrated in Figure 2.

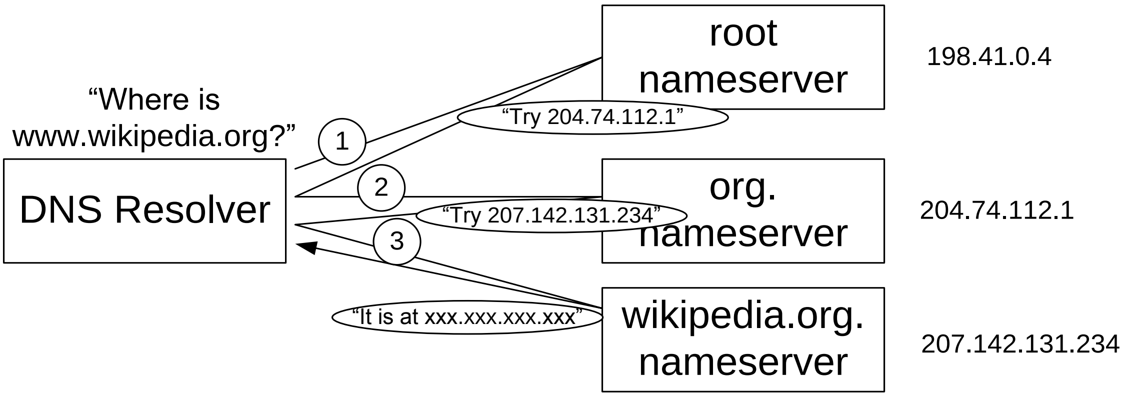


Figure 2: DNS resolution of domain name www.wikipedia.org

DNS Resource Records are the set of resource information associated to a specific name of the DNS tree. These DNS Resource Records are categorized in types [21], being TYPE A and TYPE NS the most utilized:

- A, that indicates a 32 bit Internet address
- NS, that indicates a host which should be the name server of the specific domain

For each TYPE, a specific resource data structure — RDATA — is defined. It is important to note that not all types can be applied to all resources in the DNS tree — e.g. it might not make sense to associate an MX type, which identifies a mail exchange for the domain, to a TLD such as .org —.

The DNS also defines two types of queries:

- Non-recursive, where the name server can answer queries using only local information: the response contains an error, the answer, or a referral to some other server "closer" to the answer, or
- Recursive, where the name server returns either an error or the answer, but never referrals: the name server behaves as a client when it receives a referral.

To improve efficiency, reducing DNS traffic across the Internet and increasing performance in user-applications, the DNS supports DNS cache servers, which implement the recursive algorithm necessary to resolve a given name starting with the DNS root through to the name server of the queried domain, and store DNS query results for a period of time determined by the TTL of the requested Resource Record. ISPs typically provide this kind of caching to their customers. Figure 3 shows this recursive DNS resolution process.

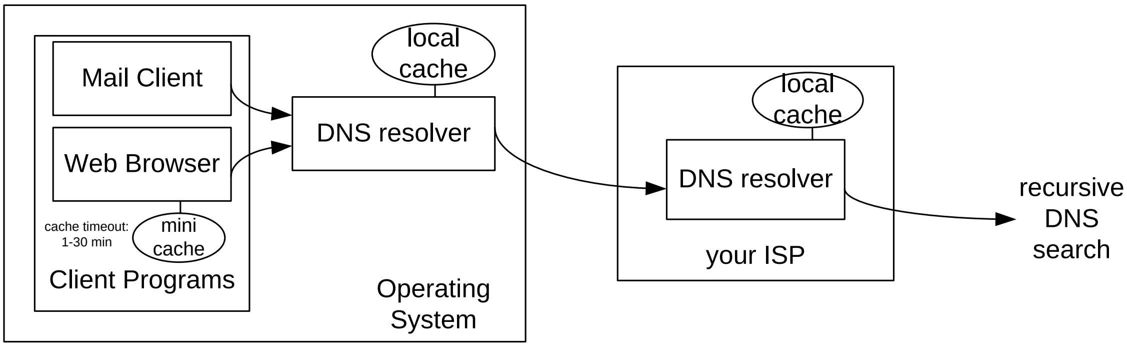


Figure 3: Recursive DNS resolution in a DNS cache server

But with the existing implementation, the DNS system has some problems. One of them is that the DNS is the meeting point of all communications initiation; however, with the current technology it is not possible to know a priori what end users will do after this initial interaction with the DNS: whether they will consume a web page, a real-time content, download a file, etc. The specific details of the application and transport protocol, resource within the domain, and MIME type [22] needed for such communications are completely unknown to the DNS — they can be known a priori by the end user client or exchanged between client and server via an application protocol such as HTTP/HTML [23], SIP/SDP [24][25], etc. —.

Since the DNS is unaware of the previous details, this situation results in inefficiency:

- Application layer negotiation — such as the exchange of applications and protocols supported by the client's device and servers — is done after the initial DNS transaction, delaying the communication set-up further.
- Actions in the network — such as QoS provision, server load balancing, DoS, etc — cannot be applied based on the initial interaction with the DNS. In order to apply these control policies and load balancing, operators have to invest in other additional specific devices (e.g. DPI equipment, layer 4 load-balancer switches, etc.).
- No valuable information can be extracted from the DNS queries-responses log, wasting the DNS rendezvous property.

In order to use the DNS system (or an enhanced version) as a possible Content Resolution infrastructure for the COMET system, the following properties must be highlighted:

- Performance: the system achieves content resolution in a range between 50 and 150 ms depending on whether cache techniques are applied or not.
- Capillarity: almost all devices with access to the Internet have a DNS client embedded
- Scalability: due to the distributed nature of the DNS tree there is no restriction on the number of content providers and content names
- Reliability: both DNS cache-resolvers and DNS authoritative servers have developed along the years a bunch of tools that guarantees maximum reliability. These are, for instance, anycast techniques, primary-secondary redundancy techniques, load balancing techniques and protection against attacks techniques
- Security: although still not widely deployed, appropriate authentication and integrity mechanisms for the conveyance of data have been standardized.

4.1.2 The Handle System

The second system that has been assessed for the development of the COMET naming architecture and resolution system is the Handle System.

The Handle System [19] was developed at the Corporation for National Research Initiatives (CNRI) with support from the Defense Advanced Research Projects Agency (DARPA), in order to provide, manage and resolve persistent identifiers to digital objects and other resources over the Internet. These persistent identifiers are called handles and are associated with sets of values containing all the appropriate information to locate and access the identified resources. Handles in the Handle System are defined as:

```
<Handle> :: <Naming Authority> "/" <Local Name>,
```

where the <Naming Authority> is a prefix that identifies an administrative entity (person or organization) and the <Local Name>, which is locally unique, characterizes an object under a naming authority. Both of them may consist of any characters from the Unicode 3.0 standard, except from the "/" character which is reserved for distinguishing the <Naming Authority> from the <Local Name>.

Each naming authority constitutes a local handle namespace, is globally unique within the Handle System and may have a set of child naming authorities, resembling a hierarchical structure. The <Naming Authority> is constructed from left to right, and its child naming authorities are separated by the ASCII character ".". The <Local Name> is an alphanumerical string, identifying the digital object, which is locally unique under its respective naming authority. The uniqueness of naming authorities and local names under their respective naming authorities ensure that each handle is globally unique in the Handle System.

Each handle resolves to a set of values that contain information about the digital object that is identified, such as its location, type, administrative permissions and many more. Each value record has a common data structure, containing the following components: index (identifier of a value record), type (data type for each value), data for each value record (based on type), permission (administrative permissions for this value record), TTL, timestamp (last time this value record was updated) and reference (number of other handles that this handle references).

The Handle System supports a two-level hierarchical service model (see Figure 4). The top level consists of a global service, the Global Handle Registry (GHR), which manages the namespace of naming authorities. The lower level consists of several Local Handle Services (LHS) that manage the handles under naming authorities. Every naming authority has to be registered under the GHR with a naming authority handle "0.NA/<Naming Authority>". A naming authority handle provides information about the handle service that manages all handles under a specific naming authority (i.e., which LHS serves a specific naming authority). An LHS might be responsible for several local namespaces, each identified by a unique naming authority. An LHS and its associated local namespaces must be registered to the GHR. Each of these services may consist of one or more service sites. Each service site is a complete replication of every other site in the service. Each service site may consist of one or more servers. All handles, and hence all handle requests directed at a given service site will be distributed across these servers.

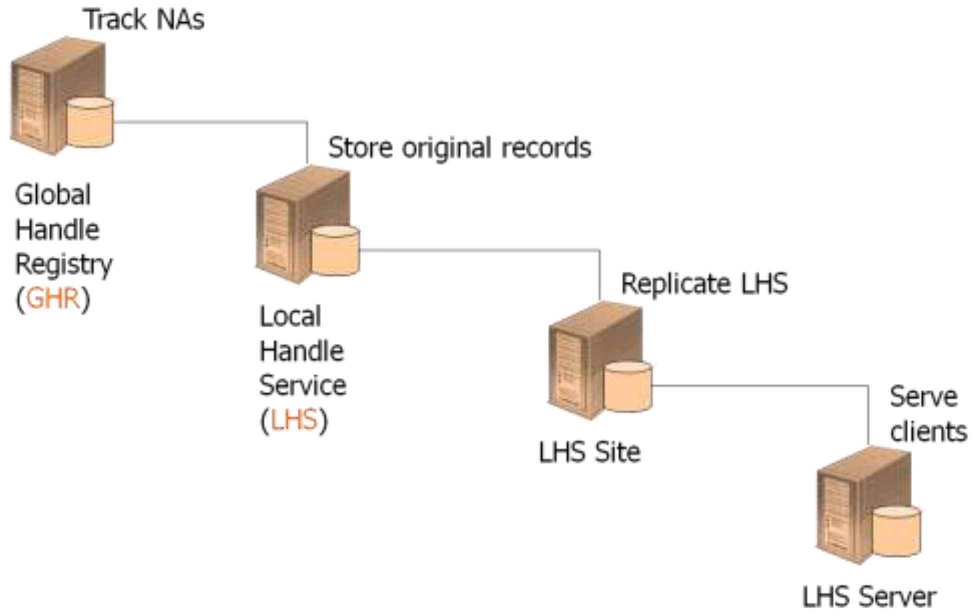


Figure 4: The Handle system hierarchy

Thus, during handle resolution, a client has to contact first the GHR to resolve the naming authority handle, receive the relevant service information, query the handle in the responsible LHS server and finally receive the requested set of values. The Handle System also supports an authentication protocol for client authentication to handle administration and resolution.

The Digital Object Identifier (DOI) System[1] is an implementation of the Handle System, uniquely identifying objects with persistent identifiers and associating metadata, so that users could access them. As in Handle System, DOIs consist of:

```
<DOI> :: <prefix> "/" <suffix>,
```

where <prefix> defines the registrant of the DOI and <suffix> defines the local identifier of the object. Every DOI prefix is a number string, starting with "10.", distinguishing every DOI name from other implementation of the Handle System. On the other hand, every DOI suffix is an alphanumeric string, which is chosen by the registrant and it can be either a sequential number or an existing legal identifier, e.g.: "10.1000/123456" and "10.1000/ISBN1-900512-44-0" are 2 examples of DOI names.

The DOI System besides adopting most of the benefits of the Handle System such as persistence and location-independence offers many advantages to content providers, intermediaries and end-users, such as persistence even if ownership or legal rights change, incorporation of existing legal identifiers and dynamic modification of content metadata.

4.2 Recent Content-based Proposals

4.2.1 Introduction

The Internet as we know it today is designed and engineered to solve the problem of sharing resources with remote machines. The principles on which this exchange of resources is done have been around since the 1970's. This exchange essentially involves the conversation between two machines, one providing the resource and the other requesting it. The increasing use of the Internet to share content (and content that may be popular among multiple users) calls out for a

[1] http://www.doi.org/

new architecture that will shift the focus from the location of the content to the content itself. Thus, several content-oriented networking approaches have been appearing over time, either focusing on the evolution of the current network architecture or even redesign it [12][39][45].

Van Jacobson et al. in [39] and Teemu Koponen in [12] identify, in combination, the following major issues that affect users from the incompatibility between today's models:

- **Availability**: How fast content arrives to the client? Currently this problem is solved via application specific mechanisms and/or pre-planned mechanisms like CDNs and P2P networks.
- **Security**: Deciding which hosts to trust, relaying on untrustworthy locations and connection information. In particular, security considerations were added after the creation of the Internet and thus, they are not native to the operation of it. What we see nowadays is that security technologies often attempt to secure the communication infrastructure rather than the content itself.
- **Location-dependence/Persistence**: Associating a piece of content with a specific location requires the overhead of configuration and implementation of network services. Also because the content is closely bound to location, when content changes location there is a need for reconfiguration / update.

4.2.2 Approaches

This section describes some of the new proposals in the realm of content oriented networks. This is not the complete list of potential approaches but some of the most recent and popular choices.

4.2.2.1 Content-centric Network (CCN)

This proposal is probably one of the most popular one in content oriented networking approaches. It bases its architecture on named content instead of named hosts. Van Jacobson et al. present their CCN architecture in [39], where they explain that the main way in which CCNs and IP differ is in the strategy and security layers. Figure 5 shows how the IP stack on the left compares to the CCN stack on the right [39].

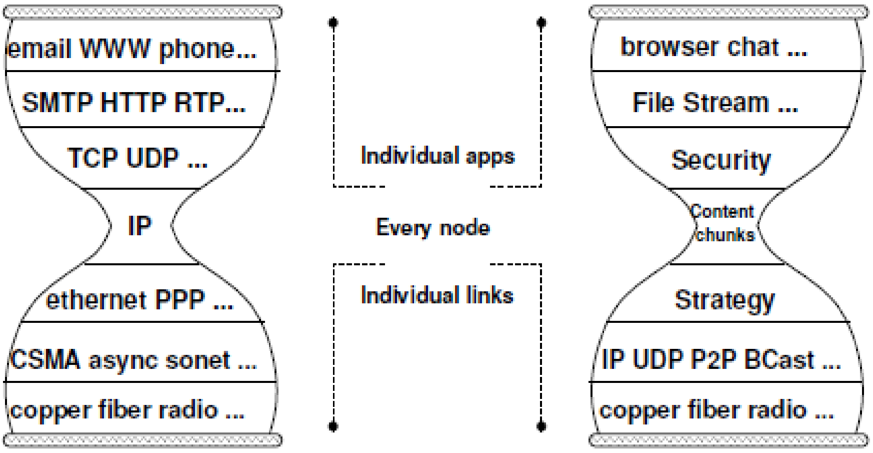


Figure 5: CCN moves the universal component of the network stack from IP to chunks of named content [39]

In this architecture there are two types of packets: *Interest* and *Data* packets. An *Interest* packet requests a specific *Data* packet. A content consumer makes a request for content or a content chunk by broadcasting *Interest* packets. Any node in the network that receives this *Interest* and has this content responds with the corresponding *Data* packet [39][44].

A CCN node has three main data structures: Forwarding Information Base (FIB), Content Store and Pending Interest Table (PIT). A packet can arrive to the CCN node via one of the available node faces. The CCN forwarding engine model is shown in Figure 6 below. The FIB is used to forward any *Interest* packets to potential providers of the content. It differs from the IP FIB only in the way that the CCN FIB allows for multiple outgoing faces to be specified for each entry. The Content Store in the CCN node is what the memory buffer is in an IP node but differs in its replacement policy. CCN packets are independent, self-authenticating and they can be useful to many consumers. For this the CCN tries to “remember” Data packets it receives for as long as possible using the Least Recently Used (LRU) replacing policy. The PIT structure is used to keep a record of *Interests* forwarded to other nodes so that the data that has been requested can find its way back to the content consumer. PIT entries can be thought of as a sort of a breadcrumb back to the original requestor [39][44].

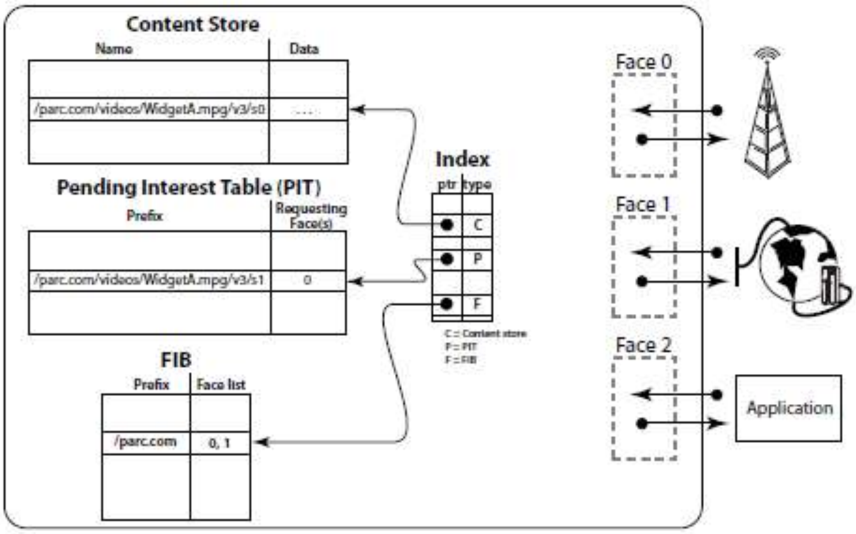


Figure 6: CCN forwarding engine model

The name of each piece of content is unique and is composed by a number of components that can be encrypted for privacy. The CCN naming architecture is flexible and hierarchical in the way that it allows for different granularity levels to be specified. Names are composed in such a way so that they are meaningful to the upper layers of the stack (Figure 5) whereas to the transport layer all that matter is the structure of the components [44].

When a consumer needs to request a piece of content they can either specify exactly the piece of content they need but in most cases the exact name of the chunk of *Data* is not known. Since the CCN tree is lexicographically ordered, the next and previous chunks can be interpreted easily by the CCN transport with any other knowledge [39].

By design, CCNs have the notion of content-based security built in, meaning that a packet is secure on its own and can travel through any route and still be secure and trusted. All the content that travels through the CCN framework is authenticated with digital signatures and is also capable of further encryption for private data. This content-based security enables for great flexibility when it comes to caching content and retrieving it from the closest provider. It means you do not need to have trust in the provider itself but to the publisher of the content [39].

Van Jacobson et al. in [39][44] implemented a prototype of the CCN network stack and demonstrated how the CCN can be an improvement both for content distribution and point-to-point protocols like VoIP (Voice over IP). VoCCN is the prototype implementation of CCN for VoIP and is described by Van Jacobson et al. in [44] where they attempt to prove that CCN can be used to support a full range of Internet application and offer improvements and advantages of traditional IP. The evaluation proved that VoCCN is functionally and performance at least equivalent to VoIP but offers simpler architecture and fewer configurations. Furthermore the built in security features of CCN make VoCCN a more secure implementation over VoIP at its simplest configuration. Finally due to the way the naming architecture of CCN is designed, VoCCN is fully interoperable with VoIP via basic and stateless gateways [44].

4.2.2.2 Data Oriented Network Architecture (DONA)

DONA, which is proposed by Teemu Koponen et al. in [12], attempts to address several issues including persistence and authenticity with the use of flat, self-certifying names. Through its name resolution technique it also hopes to solve the availability issue. It argues that the shift from host-centric to content-centric networking paradigm mostly requires changes at how Internet names are structured and resolved.

For name resolution, DONA uses the route-by-name paradigm. It replaces DNS servers with a new network entity called resolution handlers (RHs). Resolution handlers have two primitives which allow them to route content requests and responses from the consumer and provider: FIND and REGISTER. Each RH in DONA must hold and maintain a large forwarding table which provides information for the next hop for every piece of content in the network. DONA exploits the business relationship between ASes to achieve the dissemination of this information. Using the forwarding table in the RHs the content is located and once this is done standard IP routing is used to route packages from the provider to the consumer [12][39] [44].

4.2.2.3 Publish / Subscribe for Internet Perspective (PSIRP)

The PSIRP project aims to solve similar problems as CCNs and tries to redesign the entire Internet architecture via the publish/subscribe approach [40][46]. Some of the aspects that PSIRP aims to address, in order to create an efficient and effective architecture include: security, routing, wireless access, architecture design, and network economics [46]. PSIRP takes up the challenge of addressing some of today's problems where security and mobility are of the greatest importance and where multicast and caching are to replace unicast.

The conceptual architecture of PSIRP is based on the PSIRP component wheel, which acts as a modular and extensible core. The components of the PSIRP component wheel are shown in Figure 7 and they may be decoupled in space, time and context. The operation of the model is based through efficient structuring of the information identifiers and the way they interact with other network elements in such a way that allows for future expansion. The main inter-domain components, depicted also in Figure 7 are: forwarding, routing, rendezvous and caching [46].

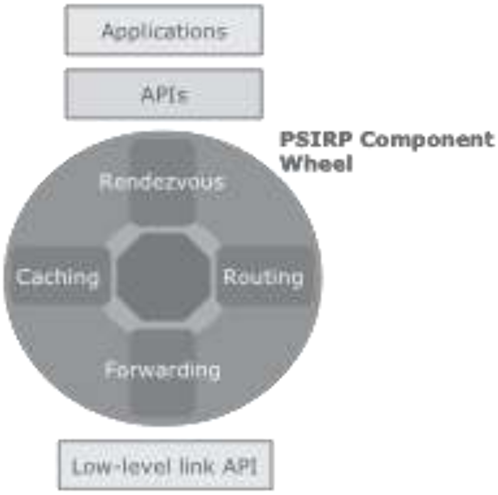


Figure 7: PSIRP component wheel

As mentioned above, information identifiers are of great importance in the PSIRP architecture. There are four types of classes of such identifiers [46]:

- AId – Application identifiers: These identifiers are used directly by the publishers and subscribers.
- RId – Rendezvous identifiers: They are used to bridge higher-level identifiers with lower layer identifiers.
- SId – Scope identifiers: Scope identifiers are used to delimit the reachability of given information.
- FId – Forwarding identifiers: FIds are used to define network transit paths and transport publications across the network.

In the PSIRP architecture, each domain can have at least three kinds of nodes [46]:

- TN – Topology Nodes: They are in charge of the intra-domain topology, load balancing between BNs and interchanging the routing vector among different domains.
- BN – Branching Nodes: They are responsible for routing subscription messages and caching popular content.
- FN – Forwarding Nodes: FNs implement a simple, cheap and fast forwarding algorithm which uses a Bloom filter.
- Rendezvous Points: They are used to locate publications within the network and they form rendezvous points which are globally connected by hierarchical DHTs which allows for greater system scalability.

From a security perspective, PSIRP uses elliptic curve cryptography and packet level authentication [46].

Within this project, a novel multicast forwarding fabric named Line Speed Publish / Subscribe Inter-Networking (LIPSIN) [40] is proposed, aiming for large-scale topic-based publish/subscribe. LIPSIN focuses on identifying links and using Bloom filters[2] to encode source-route-style forwarding information as part of the packet header. This enables forwarding without dependency on addressing the sending and receiving hosts. The independence from end-to-end addressing provides several advantages over the current model:

- Taking forwarding decisions becomes easier and forwarding tables are smaller. Smaller tables potentially means that switches may be faster, smaller and require less energy to operate as compared to the ones used today.
- It provides native support for data-oriented naming and in network caching.

The LIPSIN model has the overall aim of allowing both stateful and stateless operations by having a balance between the headers of the packet and the nodes in the network. This is achieved by using the Bloom filter thinking and inverting it. Instead of using the Bloom filter on each network node and matching the incoming packets to the defined filters, LIPSIN instead uses Bloom filters themselves in the packets to allow the nodes in the path of the package to determine to which links the packet will be forwarded [40].

Besides making forwarding simpler and more energy efficient, it also offers some very useful security advantages. Paper [40] shows the results from a small prototype made that could eventually be used to operate at a full bandwidth.

[2] A Bloom filter is a space efficient probabilistic data structure used to test whether an element is a member of a set, conceived by Burton Howard Bloom.

The LIPSIN model mainly focuses on the publish/subscribe mechanism. It is an incomplete approach in the sense that it does not particularly focus on the other two components of large-scale multicast; data-oriented naming and in-network caching [40].

4.2.2.4 Internet Indirection Infrastructure (i3)

The i3 architecture offers a rendezvous-based abstraction that is both flexible and powerful. Applications can easily build a variety of communication services on top of it, such as: multicast, anycast and mobility. i3 looks at combining the generality of IP-layer solutions with the deployability of overlay solutions to avoid the challenges and shortcomings of the two and take advantage of their benefits [4].

The i3 architecture looks similar to that of today's IP multicast but has some basic differences that make it more flexible. It uses indirection to separate the sender from the receiver. i3 uses DHTs Distributed Hash Tables (DHTs) in combination with packet identifiers. The receivers insert a trigger with the data identifier in the DHT, along with their address. The trigger is routed to the appropriate sender who responds to the request with the appropriate data. The returned data contains the same id as the request. This allows for greater flexibility as it lets receivers control the routing of the packet. The flexibility enables the creating of services, such as mobility, anycast and service composition at the application level. Furthermore, the infrastructure can be simplified, made more robust and scalable by passing on the responsibility of efficient tree construction to the end-hosts [4].

The results of evaluating i3 in [4] show that the system is highly flexible and can support demanding applications that require mobility, multicast and/or anycast. One of the applications on which it was tried out involved multicasting an MPEG video which was transcoded on the fly to H.263 format.

4.2.3 Conclusion

The IP interoperation layer has served the Internet well for a few decades now, due to its simplicity and flexibility of supporting multiple and different classes of applications. Despite this, over the years, the use of the Internet has shifted to become more about retrieving content. The focus of users and applications is no longer on 'where' the content is but on 'what' the content is. In the same manner, the approaches mentioned above try to decouple the way consumers find and retrieve resources from the notion of hosts. There are several considerations to take into account when evaluating these approaches, since not only should they allow for a wide spectrum of applications to operate through them but they must also take into account scalability, cost, complexity and performance issues.

4.3 Resource and Content Identification

4.3.1 The Current Resource Identification Scheme

4.3.1.1 URNs and URLs

Internet resource identification has been debated in the IETF and the W3C since the early 90s. The intention has always been to distinguish resource reference from resource access. This requirement led early to the concepts of Uniform Resource Names (URNs) and Uniform Resource Locators (URLs). A URN is a unique and persistent identifier for an Internet resource and it is independent from the location and availability of the resource. A URL [17] explicitly expresses the location of an Internet resource and can be used to access that resource. URLs appear on the address bar of web browsers, so most Internet users are today familiar with them. The IETF has also defined Uniform Resource Identifiers (URIs) [33] as a generic naming framework to encompass both URNs and URLs. In agreement with the W3C, the IETF has even suggested the deprecation of URNs and URLs in favour of the more general URIs [34].

URNs are designed primarily for machine consumption. Semantics to allow easy interpretation by humans are not required for URNs [35]. The syntax of URNs [36] is simple and generic in order to accommodate existing identifier namespaces (e.g., ISBN):

```
<URN> ::= "urn:" <NID> ":" <NSS>
```

where <NID> is the Namespace Identifier, which might be composed of several subordinated sub-name spaces, and <NSS> is the Namespace Specific String. An example of a URN could be "urn:issn:1535-3613", where "issn" is the namespace identifier and "1535-3613" is the namespace specific string.

URN resolution, while still under debate, is the process of translating a URN to one or more URLs, to resource metadata, or to the actual resource. While requirements and operations for URN resolvers are specified [37], each NID scheme is free in architecting and implementing such systems. This means that clients must potentially be extended for each scheme (e.g., with plug-ins). To remedy this, a service for discovering URN resolvers has been specified [38], but no implementation is widely available yet.

4.3.1.2 PURLs

Persistent Uniform Resource Locators (PURLs)[3] were developed in the mid-90s by the Online Computer Library Center (OCLC), a worldwide library cooperative. PURLs were created as a simple, practical, and short-term means of persistently naming web resources.

A PURL is a valid URL, but it does not point to the location of the desired resource. Instead, a PURL points to an intermediate location that stores a record of the current location of the desired resource. The record is used to redirect requests to the actual resource. When a resource relocates, its PURL remains unchanged (i.e., it persists), and only its record is updated to point to the new location. For example:

```
http://purl.oclc.org/OCLC/OLUC/32127398/1
```

this PURL consists of the protocol (http://), the resolver address (purl.oclc.org/) and the name (OCLC/OLUC/32127398/1).

HTTP is always used as the protocol for resolving PURLs. The resolver address is either the hostname or the IP address of the intermediate location (known as a PURL server or resolver). As such, PURLs depend on HTTP, and to a lesser extent, on DNS.

PURL names point to records in the resolver. Names are organized hierarchically in domains. Domains are separated by "/" in the name syntax, with the left-most domain at the top of the hierarchy. While resolver addresses inherit the global uniqueness of hostnames and IP addresses, names are unique only under the associated resolver address. However, taken together, resolver addresses and their local namespaces result in globally unique PURLs. Resolvers are independent of one another and each one is responsible only for its namespace. Persistence of PURLs is therefore dependent on the commitment of the resolver owner for running and maintaining the resolver and its records.

A PURL resolver receives HTTP requests, retrieves the record associated with the requested PURL name, and replies with a standard HTTP redirection to the actual resource. PURL records are inserted into the database of a resolver by its registered users. The creator of a PURL is known as its maintainer and is responsible for keeping its PURLs pointing to the correct actual resources at all times. In general, administration of PURLs consists of creation, modification, searching, validation, and deletion. To maintain persistence, deleted PURLs are only disabled. For administration purposes, PURL resolvers expose a web interface and a RESTful API.

[3] http://purl.org/

PURLs are still in use today, with the main resolver maintained by the OCLC. There's no way however to discover all available PURL resolvers. The OCLC has recently begun optimizing the original code base of the PURL resolver.

4.3.2 The Recently Proposed Content Naming Scheme

The work on content-centric networks has evolved over the past decade with various efforts pertaining to content naming. In this section, those relevant proposals are reviewed.

4.3.2.1 The TRIAD

The TRIAD [1] proposed the explicit inclusion of a content layer at content routers to handle contents. They observed that content should be treated explicitly in today's Internet. As such, contents should be addressed. In this proposal, contents are named using Uniform Resource Locators (URLs) i.e. the end-to-end identification are based on URLs while IP addresses are used as transient routing tags. Thus, this proposal requires globally unique name for each content. Specifically, the DNS part of the URL is used for routing purpose while the file name portion of the URL specifies the content within the host. The implication to this is that the resolution relies on the semantics and hierarchy of domain names to aggregate routes to content names. TRIAD also creates location-independent end-host identifiers. The resolution process in TRIAD is coupled with the routing process, thereby aiming at improving latency.

4.3.2.2 The IP Next Layer (IPNL)

Starting from a different motivation, the IP Next Layer (IPNL) [2] proposed to extend network address translation (NAT) to solve the issue on depletion of IP address by building the work on top of the current IPv4 infrastructure for minimal disruption. In this work, fully qualified domain names (FQDNs) are used as host identifier in packets. The idea is to isolate local site addressing space from the global IP addressing space. The proposal divides the addressing space into high and low order parts:

- Globally-addressed part of the Internet → middle realm
- Privately-addressed realms → private realm

NAT-boxes are used like gateways to and from private networks. By dividing the Internet into zones, IPNL allows re-use of addresses within each private realm.

4.3.2.3 The Split Naming / Forwarding (SNF) Scheme

To achieve the identity / location split, [3] proposed a split naming / forwarding (SNF) scheme. SNF divided addressing into naming and location by creating two sub-layers in the network layer.

- Forwarding layer – this sub-layer provides locators so that the network is capable of delivering packets to the destination.
- Naming layer – this sub-layer is responsible for providing name to locator mappings.

Similar to [2], SNF uses FQDNs to identify nodes. In addition to that, IP addresses are used to denote the location of nodes. A 64-bit number is used to identify packet flow at the transport layer.

4.3.2.4 The Internet Indirection Infrastructure (i3)

In [4], an overlay rendezvous-based communication abstraction named Internet Indirection Infrastructure (i3) is proposed whereby a level of indirection is introduced. The key idea in i3 is to decouple the act of sending from the act of receiving. The proposal is built on top of the Chord [5], a distributed lookup protocol for peer-to-peer networks that falls under the distributed hash table (DHT) category. Data location is implemented by associating a key to each data item and storing the key/data item pair at the node to which the key maps. The key space for Chord is flat. In i3, the identifiers are attached to packets and matched with triggers working in a publish / subscribe communication model [6][7].

4.3.2.5 The Forwarding directive, Association and Rendezvous Architecture (FARA)

Taking a more general and clean-slate approach, the Forwarding directive, Association and Rendezvous Architecture (FARA) [8] proposed an abstract model based upon the decoupling of end-system names from network addresses. The basic components of FARA are:

- Entity (e.g. host)
- Association (e.g. connection)
- Communication substrate (e.g. forwarding behaviour)

By using these basic components, the generic framework enables different instantiations for different design and requirements. It should be noted that FARA avoids the introduction of new namespace and uses rendezvous approach. Nevertheless, although FARA avoids the notion of host identity, it does not explicitly consider content. It remains to be seen how a content-centric network can be derived from the FARA meta-architecture.

4.3.2.6 The Host Identity Protocol (HIP)

The Host Identity Protocol (HIP) [9], on the other hand, proposed to separate the location and host identity information via the introduction of a new namespace with cryptographic capabilities for host identities with the IP addresses remain to be used for packet routing only.

- Host identity = a public cryptographic key of a public-private key pair

The public key identifies the party that holds the only copy of the private key. Packet transmission is preceded by an identity verification stage using IPSec. Using this approach, location change of hosts does not break the connection as packets are routed based on the identity. Additionally, thanks to the cryptographic features included in the protocol, a host receiving an unintended packet will not be able to open it since it does not possess the correct private key. Note that this proposal focused on the host identity separation but do not explicitly consider the content-centric networking issues.

4.3.2.7 The Layered Naming Architecture

Drawing upon past work such as [4][9], Balakrishnan et. al. proposed a layered naming architecture [10] which attempts to synthesize the different available proposals focused on narrower goal into a larger whole. It described a 3-level name resolution approach:

- User-level descriptor to service identifier
- Service identifier to endpoint identifier
- Endpoint identifier to IP address

The proposal considered both services and data as 1st class objects in the Internet and attempted to decouple the process of locating the content and the process of retrieving the content. It advocated globally unique flat names for both service and endpoint identifiers and used DHT solutions for resolution. Following this paradigm, the application layer will deal with service identifiers, transport layer deals with endpoint identifiers while the network layer will be responsible in routing using IP addresses.

4.3.2.8 The Delegation-Oriented Architecture (DOA)

Another proposal advocating globally unique identifiers in a flat namespace is the delegation-oriented architecture (DOA) [11]. The central to this proposal is the use of intermediaries (or middleboxes). It represents a small step towards achieving the location / identity split which requires no changes to IP or IP routers. In this architecture, packets carry references serving as persistent host identifiers. In other words, each packet contains an identifier that unambiguously specifies its ultimate destination. Each referenced host will have a chosen delegate responsible in resolving these references.

4.3.2.9 The Data Oriented Network Architecture (DONA)

The Data-Oriented Network Architecture (DONA) [12] is another attempt in proposing a clean-slate naming and name resolution architecture for a content-centric network. It argues that the shift from host-centric to content-centric networking paradigm mostly requires changes at how Internet names are structured and resolved. Instead of following earlier proposals such as [1][2][3] in using URLs as content names, DONA proposed a naming scheme organized around principals. The names are application-independent and globally unique. It also proposed to make the names self-certifying. Each named entity is associated with a principal and each principal is associated with a public-private key pair. The name is structured in the form of P:L where P is the cryptographic hash of the principal's public key and L is a label chosen by the principal who ensures that they are unique. Also, in place of DNS name resolution, it proposed a name-based anycast primitive above IP layer, advocating route by name paradigm as oppose to the current DNS look-up system.

4.3.2.10 The NetInf

The NetInf architecture [13] is yet another more recent proposal in unifying some of the related work in naming and resolution in content-centric networks. The NetInf naming architecture is based on zonal separation similar to [2]. The information model consists of information object and bit-level object. Similar to later proposals, NetInf put much effort in providing strong security to content. The metadata in this architecture contains various security related components such as public keys, content hashes, certificates and digital signature, reflecting the importance of security in the content-centric networks.

4.3.3 Synthesis of the Schemes

From the review of the available literature related to content naming and resolution, several trends can be observed on the evolution of the proposals over the past 10 years.

- Naming scheme – the earlier works (e.g. [1][2][3]) mainly borrowed from DNS and re-used the URLs to name content. However, later, we found a new batch of proposals advocating flat naming schemes (e.g. [4][9]). The debate on the scalability of flat naming schemes is inconclusive. Finally, we observed the possibility of combining the two extremes (e.g. [13]) to create a hybrid naming scheme consisting of both hierarchical and flat naming schemes.
- Name resolution – The name resolution approaches basically can be categorized into two; (1) lookup-based resolution and (2) route by name. The earlier work which followed closely the current DNS system mostly proposed lookup-based resolution where the name resolution depended on lookup operation on a name database. The later proposals, notably DONA [12], proposed a hop-by-hop resolution approach which moves away from the conventional DNS-like resolution approach.
- Security in the naming architecture – Earlier works have largely neglected intrinsic security design into their proposals. However, the community has acknowledged the importance of incorporating security features from the beginning and thus we see proposals in the last few years put much effort into building security features into the naming architecture.

Synthesizing these past proposals, we identified four main requirements for a content naming architecture:

- Scalability – the naming architecture must scale to high number of possibly dynamic contents.
- Security – Although many security features have been developed for the current Internet, they are inherently an add-on to the original architecture. For a content-centric network, the security features must be designed and implemented as part of the new system rather than an afterthought. Furthermore, the resolution process has to identify the exact content that has been requested (i.e., avoid spoofing, DDoS etc.).

- Routability – the naming architecture should facilitate simple and efficient routing of request and possibly the content retrieval as well. In other words, the efficiency and scalability of routing in a content-centric Internet depends highly on the way names are structured.
- Persistence – names should be persistently valid for the contents (especially due to mobility). As a general rule, the more “information” encoded into the name, the less persistent the name.

5 Content-based Resolution and Delivery

In this chapter, we detail two content resolution and delivery approaches under the high-level COMET architecture described in D2.2 [48]. The first approach (content-record based, or decoupled approach) advances the current Internet system, inheriting the required functionalities along with content-based enhancements. It is developed with less disruption to deployment in view. The second approach (coupled approach), however, follows a more disruptive design principle. It is not constrained by the current use of the systems like DNS. The approach exploits to some extent the hierarchical nature of the current Internet topology and employs a domain-level "hop-by-hop" paradigm to content publication and resolution.

5.1 Content Record-based Resolution Approach

5.1.1 Overview

This section proposes an approach for the Content Naming Scheme and the Content Resolution architecture as part of the Content Mediation Plane (CMP) of the COMET system. In this approach, the content resolution operation is based on a global directory system which stores content information, in a similar way as the current host-centric Internet uses global DNS directory system to support domain name resolution.

At first, section 5.1.2 presents the common proposal for the Content Naming Scheme to be followed in this architecture approach, while section 5.1.3 details the advantages of the unification of the Content Access Method.

Later on, section 5.1.4 describes the operation of Content Publication in the COMET system. For this operation, two different approaches have been followed. Although both of them follow a common naming scheme, one approach is based on the DNS System and the other one is based on the Handle System, so they have different mechanisms which are detailed in this section.

Section 5.1.5 deals with the operation of Content Resolution. In the context of this operation, there are some processes that are also described in this section: Name resolution, Path Discovery and Decision process. Path configuration, although part of the Content Resolution, is described in D4.1 [49]. As it happened with the Content Publication, different approaches have also been followed to describe these processes. For instance, the Name Resolution process could be based on the DNS System[4] or on the Handle System. Figure 8 shows a general scheme of the structure of the Content Record-based Resolution section:

[4] Telefónica I+D could seek for patent on the Content Resolution based on DNS which is explained in this section.

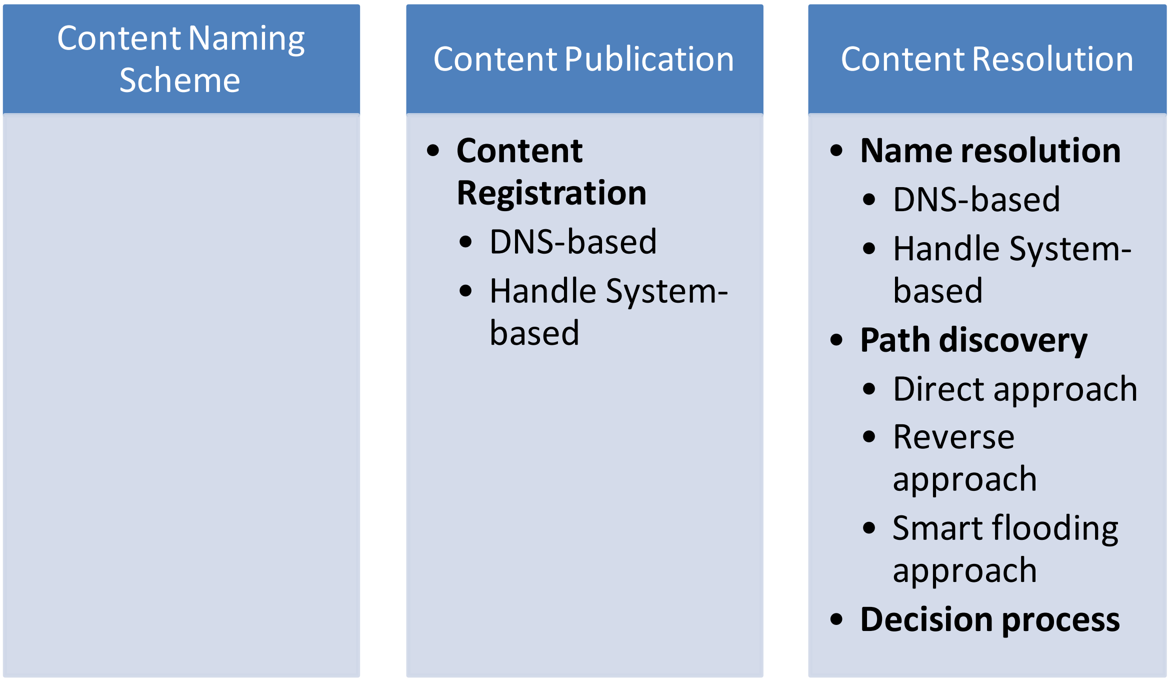


Figure 8: Content Record-based resolution approaches scheme

5.1.2 Content Naming Scheme

In COMET, content is associated with a Content Identifier, so that it can be accessed later on by content consumers via this Content Identifier. The Content Identifier is globally unique (different content objects have different Content Identifiers). In addition, the Content Identifier is location-independent and application-independent, so that if application protocols or server addresses change, the Content Identifier persists unmodified.

We distinguish between two types of Content Identifiers for identifying content: Content-IDs (CID, a non human-readable byte string of fixed length) and Content Names (CN, human-readable character string). In the Content Record-based Resolution Approach, the Content Name is represented with a Content Naming Scheme based on the Handle System.

We argue that content should be identified by a short, human-readable and variable-length Content Name, which is mnemonic to humans and has the following structure:

```
<CN> :: <who> "/" <what>,
```

where the <who> part of the <CN> defines the naming authority for that content (i.e. its owner or provider) and the <what> part defines the local name of content under its naming authority, separated by the reserved ASCII character "/".

Naming authorities identify the entities that create or own the rights of the content and want to publish them in the Internet (Content creators) or the entities that store and make content available to content consumers (Content providers). Any content creator or consumer may obtain a globally unique naming authority.

As in the DNS and Handle System, the <who> parts of content name reflect a hierarchical structure, in which every naming authority (parent) may have one or more child naming authorities. Thus content naming scheme for COMET can be represented as a tree, where each naming authority (parent) may have one or more naming authorities (children) underneath, which can be registered to COMET system only through their parent naming authority. The segments of the <who> part of content name are separated by the ASCII character ".", which is reserved in the case of <who> part, but not in the case of <what>, and, unlike DNS, are constructed from left to right. For example, "intracom.cdn" defines the naming authority of the CDN department of INTRACOM TELECOM.

We also propose an alternative structure for end-users (content prosumers) who are publishing their content, as their e-mail address can be used as the human-readable format of their naming authority in a way that would still maintain the left-most hierarchical structure of naming authorities in Handle System. For example, a content prosumer with an email address user@gmail.com may register to Handle System to obtain a naming authority "com.gmail@user". We can foresee a scheme where e-mail providers register the e-mail addresses of their users as naming authorities.

On the other hand, the <what> part of content name represents the local name of the content under a specific naming authority. Local names are case-sensitive, can contain any ASCII character except from the reserved "/" and must be locally unique under their specific naming authority, thus globally unique in COMET system and Handle System in general. Below, 2 examples of content names are presented:

```
com.abc/aShow
com.gmail@user/aShow
```

5.1.3 Unification of Content Access Scheme

As it is known, in recent years there has been a growing proliferation of user-generated Internet content, including blogs, photos, video, etc. The increasing trend of users generating their own content has led to an abundance of intermediaries. This large number of intermediaries makes many contents accessible only for particular user communities, resulting in global content search and direct access being fragmented.

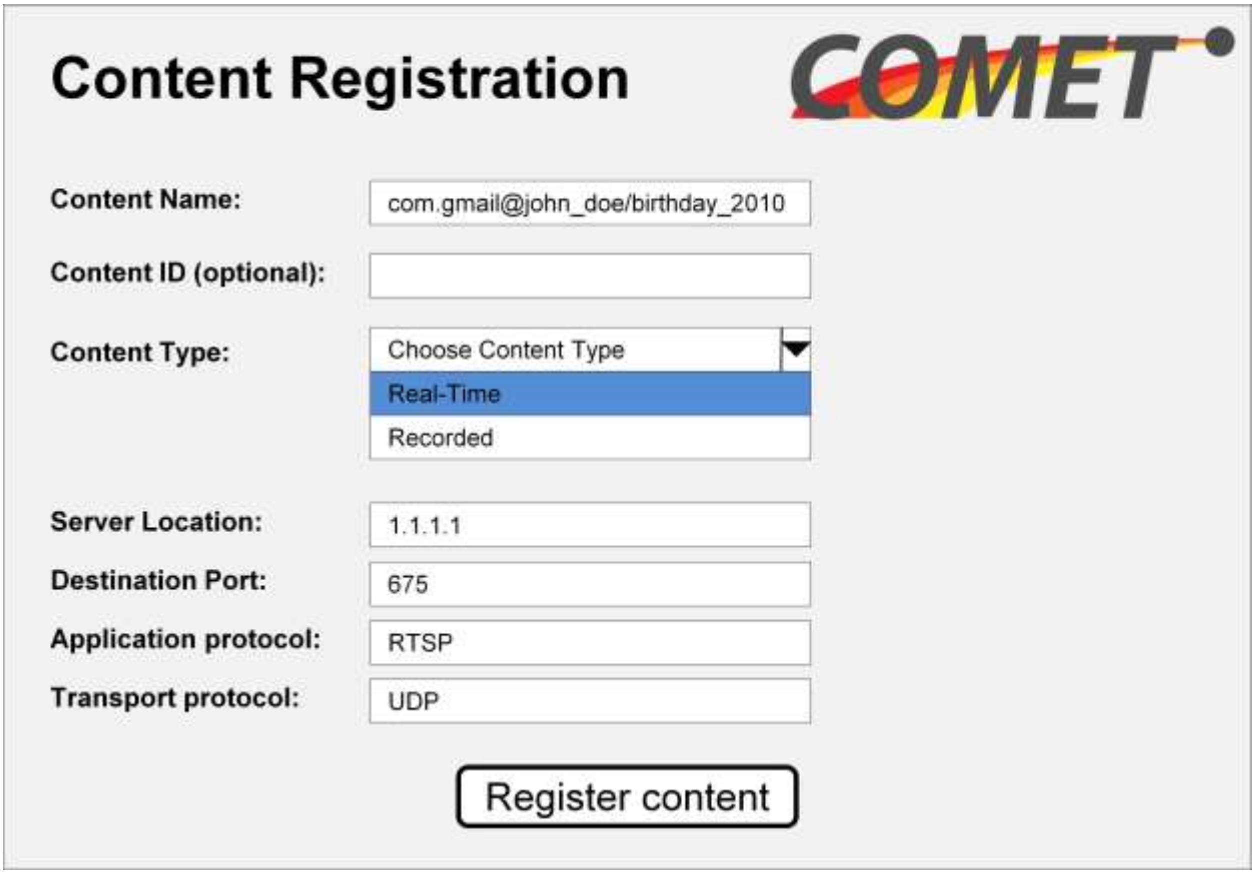
So, in the current Internet, global content search and direct access is fragmented into different content providers. The key problem with the current approach is that users need to know the URL in order to access the content, or, in the worst case, search for the content through the relevant intermediary.

By the use of the global Content Naming Scheme described in the previous section, COMET intends to offer unified interfaces for both content publication and consumption. In this way, the end users will access content independently of the specific content location and the distribution technique used by the content provider. Besides, it is expected that the new content publication would be linked to search engines, so that content could still be searched through Google.

Regarding Content Publication, COMET will offer a unified interface for registering content, with the usage of the Content Publisher module. Every Content Provider, either large organization or individual end-user, which wants to register its content to the COMET system, will be provided with such a module. At this point we assume that the Content Provider has registered with the COMET system, obtaining a <who> name (e.g. com.gmail@john_doe) and has also all the relevant credentials to access the Content Publisher interface securely. Through this interface, a `Publish(CN or CID`[5]`, content record parameters)` message will be sent to CRE during registration of a specific content, providing COMET with all relevant information, such as its location, nature, application and transport properties, which are going to be included in content record. In Figure 9 we present an example of how this interface might look.

In addition to this, Content Providers will be able to update content records through the same interface. `Update(CN or CID, modified content record parameters)` messages will be exchanged for that purpose. We assume that only registered users (e.g. Content Provider's administrator) will have access to update or delete their respective content records.

[5] In the decoupled approach, Content Identifiers, Content-IDs or simply CIDs are human-readable byte strings of fixed length that have a direct mapping to Content Names (human-readable character strings of variable length). Although ideally content publication and consumption should deal with Content Names, we foresee that for some situations a fixed-length content ID could be better for high-speed processing (e.g. to enable content-aware forwarding based on Content IDs).



Content Registration

Content Name: com.gmail@john_doe/birthday_2010

Content ID (optional):

Content Type: Choose Content Type

Server Location: 1.1.1.1

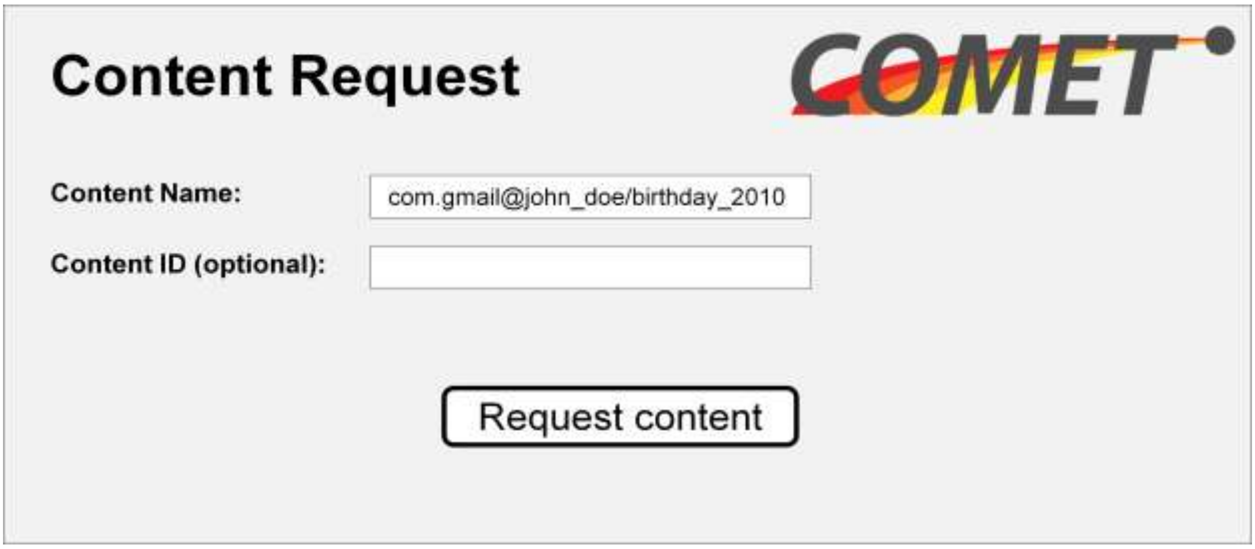
Destination Port: 675

Application protocol: RTSP

Transport protocol: UDP

Figure 9: Content Publisher Interface

During content consumption, a content consumer requesting a specific content will have access to COMET system through its Content Client module. Therefore, upon every content request, a `Request(CN or CID)` message will be sent to content consumer's adjacent CME, probably including certain preferences of the consumer, such as the desired QoS. Upon content resolution the Content Client will receive all relevant application and transport properties to send an application-level request towards the selected content server and finally receive content. In Figure 10 we present an example of how the Content Client interface might look.



Content Request

Content Name: com.gmail@john_doe/birthday_2010

Content ID (optional):

Figure 10: Content Client Interface

Both interfaces (content publisher and content client) might be integrated into existing applications, such as web browsers. More details about the unified interfaces, messages exchanged and information needed, will be discussed throughout the project and included in the upcoming deliverables.

5.1.4 Content Publication

We consider Content Publication as the process of making content available to Content Clients. This process includes two steps: Content storage and Content registration. Both suboperations are detailed below:

1. The first step of the Content Publication is Content storage. In this suboperation, which is out of the scope of COMET, the Content Provider stores the content in its servers for a later registration. Two different options are considered for this storage:
    - In the first option, the Content Provider or the end-user who wants to publish a particular content stores the content in their premises (Content Servers, Hard disk…).
    - In the second option, the end-user who wants to publish a particular content, decides to delegate the publication to another entity (Facebook, YouTube, etc.). In this case, this latter entity would act as the Content Publisher. The end-user must store the content in the Content Publisher premises.
2. After the replicas of the specific contents are stored in the Content Servers, the second step is the Content Registration.

    The Content Providers, which are the owners of the Content Servers, must notify the COMET system that they have these contents in their premises in order to make them available through the Internet. For this purpose, the Content Providers require the functionalities of an entity called Content Publisher. This entity interacts with the Content Resolution Entity (CRE, in charge of the Content Resolution Function) to perform the Content Registration.

    This Content Publisher, which basically consists of an application to interact with the CRE, can be executed by the Content Provider itself or, in some cases, a third party agent could offer publication services to the Content Providers and perform the Content Registration on their behalf. Accordingly, this suboperation involves two COMET entities: the Content Resolution Entity and the Content Publisher.

    The Content Registration will be made through a global and unique COMET publication interface accessible for all Content Publishers. As part of this process, some parameters associated to the content itself are passed to the CRE. With all this information retrieved, this entity creates a file called Content Record which includes a set of parameters associated to the content that is being published.

[[Please note that some text has been removed from the public version of this document due to Intellectual Property Protection procedures, but will be included in a later edition once concluded]]

Regarding the architecture that will perform the CRF, two different systems have been analyzed. The first one is based on the DNS and the second one is based on the Handle System. Depending on the approach followed to implement the CRF in the COMET System, the Content Publication operation has different mechanisms. The insights of both approaches for the Content Publication are detailed in the following sections.

5.1.4.1 DNS-based Content Publication

[[Please note that some text has been removed from the public version of this document due to Intellectual Property Protection procedures, but will be included in a later edition once concluded]]

5.1.4.2 Handle System-based Content Publication

The second approach analyzed for the Content Publication operation in COMET is based on the Handle System. This system provides a handle-to-values set mapping, meaning that each time there is a request for a specific handle, a set of values is returned. Similarly in COMET, each Content Name is associated with a Content Record, containing relevant parameters and properties about the content, thus there exists a Content Name-to-Content Record mapping.

We assume that a content provider or creator, who can be either a large organization or a single end-user, has stored its content to a content server, with a known host identifier and also has contacted and made an agreement with a Handle System Registrar to obtain a globally unique naming authority (<who>). The operation of obtaining a naming authority is performed only once, during the first time a content provider wants to publish a content, thus this step can be skipped in future content publications. In addition to this, content provider is assigned to a specific LHS server, which will handle content name requests for its published contents and can access and publish content to COMET system through its content publisher component. For this purpose, the Handle System Registrar creates a <who> handle ("0.NA/<who>"), which includes all this relevant information (service information) and stores it in the GHR.

During Handle System-based content publication, 2 entities are involved, the Content Publisher, located in the side of content provider and the Content Resolution Entity (CRE), which in this case is the Handle System. We consider Handle System as an entity external to COMET system, with its internal hierarchical structure and protocols. Thus, a content creator or provider may publish its content to COMET system:

1. Through its content publisher function, it sends a `Publish(CN or CID, content record parameters)` message to the CRF. Content record parameters include relevant information such as the Content Name, the host identifier of the content server where content is stored and the host identifier of the edge CME adjacent to the content server, as well as other content metadata that might be useful during content resolution or consumption. Components of content records were described in previous section of this chapter.

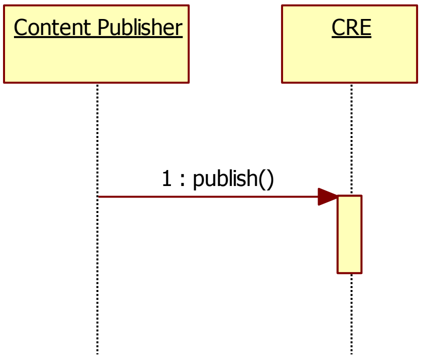


Figure 15: Handle System-based content publication

Upon receiving such a message, the Handle System will create the content record, using the specified content record parameters and will store it to the relevant LHS server, as specified in the service information included in the <who> handle. Since the Handle System is considered as an entity external to COMET, we assume that the Content Publisher is a Handle System client and the `Publish` message is based on the existing implementation of the Handle System protocol for handle publication.

The COMET system could also extend the current structure of value records, through the creation of new value types and fields. Such extensions could include certain COMET-oriented types, such as the host identifier of CME, the host identifier of Content Server, content metadata, application, session and transport information and many more.

5.1.5 Content Resolution

We consider Content resolution as the process responsible for the discovery of the requested content based on the given Content-ID or Content Name. Particularly, in the Content Record-based resolution approach, all the information related to the content, servers and its characteristics is stored in Content Records, which are located in the CREs, hierarchically organized. These entities are accessible by protocols based on IP. Besides, these Content Records are reachable either through the Content-IDs or through the Content Names, which will be known by the content client in order to access that content.

In COMET, the Content Resolution process will be able to locate all the copies of the same content if the content has been replicated and hosted at various content servers and the paths to reach those particular servers. This is crucial for optimization/enhancement of the content delivery and also, enabling capabilities such as anycast.

There are different ways to decide the "best" duple server/path as a result of the Content Resolution. For instance, a possible option is to use the number of hops as the metric in deciding the nearer Content Server. Alternative options are considered in the section that deals with the Decision Process, considering also QoS requirements and metrics. A more sophisticated scheme may involve the computation of the network resources in the domains involved in the transportation of the content and the current or even projected future load at the content server(s).

The operation of Content Resolution consists of different suboperations. These are: Name resolution, Path discovery, Decision process and Preparation for content delivery. The following sections detail all these processes and the different mechanisms that can be followed for their performance.

5.1.5.1 Name Resolution / Content Record Retrieval

We consider Name resolution as the process responsible for getting the Content Record associated to a particular content requested by a Content Client given the Content Name (or Content-ID) of the content. After the process of the name resolution, the Content Mediation Entity (CME) will obtain the set of content properties necessary to take later decisions regarding the best server and path for the delivery of the content.

The whole process is triggered right after the Content Client requests a content to the CME. In this step, the Content Client sends a Content Name or a Content-ID in order to express his purpose of retrieving a particular content. At this point, we assume that an end-user has found the CN or the CID of the content he wants to consume, either by searching it through keywords in search engines or by obtaining it with other ways (e.g. via email).

After this message, the CME starts the Name Resolution operation which consists of sending this Content Name or Content-ID to the appropriate Content Resolution Entity in order to obtain the Content Record associated to the content.

As it has been detailed in the Content Publication section, there is a Content Record for each content which has all the information associated to this content. From this Content Record, the CME selects the appropriate content source, application protocol, etc., and provides the Content Client only the information necessary in order to make him possible to retrieve the content from the Content Server.

Following the same reasoning as in the Content Publication section, depending on the system used to perform the Content Resolution Function (the DNS or the Handle System), two different approaches can be followed for the operation of Name resolution. The details of both of them are explained below.

DNS-based Name resolution

[[Please note that some text has been removed from the public version of this document due to Intellectual Property Protection procedures, but will be included in a later edition once concluded]]

Handle System-based Name resolution

As described in the section of content publication, a specific CN is associated with a content record, containing content metadata and properties. During name resolution, the COMET system will resolve the requested CN to its Content Record, and will extract these properties and use them during other processes of content consumption, such as path discovery and decision process. The basis for the proposed name resolution is Handle resolution.

In this procedure, 3 entities are involved, the Content Client, based on content consumer's end-machine, the Content Mediation Entity (CME), adjacent to content client and the Content Resolution Entity (CRE), which in this case is the Handle System which is considered as external to COMET. Handle System-based Name Resolution is described further in the following steps (as depicted in the respective steps of Figure 18):

1. A content consumer wants to search for a specific content with CN "who/what" and sends through its content client module a request(who/what) message to the CMF of its adjacent CME. At this point we assume that the client has previous knowledge of its respective CME or can find it in some way. The request message contains the requested CN and may also include the desired QoS class (CoS_req) for the delivery of the content.
2. The CME will then send to the CRE a resolve(0.NA/who) message in order to receive the service information for the specific naming authority <who>.
3. CRE returns the service information for the requested <who> handle.
4. Upon receiving the service information, the CME sends a resolve(who/what) message to the CRE, in order to receive content record.
5. Finally, the CRE resolves content name to its respective content record and returns parameters of the content record to CME.

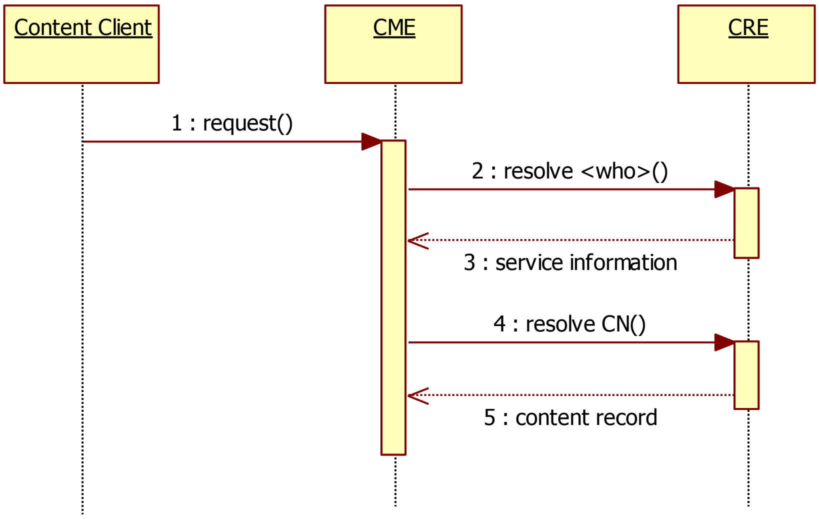


Figure 18: Handle System-based Name Resolution

Since CRE is the Handle System and is external to COMET system, we assume that a Handle System client will exist within the CME, responsible for sending requests to and receiving responses from the Handle System. Thus, `resolve` messages sent towards the CRE are based on the existing implementation of Handle System protocol for handle resolution. Handle System client based in the CME receives content record and forwards specific content record parameters towards other modules of the CME, which are going to be used in other COMET procedures.

In summary, during Handle System-based name resolution we define one COMET message:

- `request(CN)`

5.1.5.2 *Path Discovery*

We consider Path Discovery as the reactive process during the Content Resolution to obtain the properties of the paths from Content Servers to the Content Client, necessary to take decisions in the later Decision process. This is always performed by the CMF block. Depending on the mechanism that is used to perform this operation and where we calculate the path from, we have studied three different approaches. These are:

1. Direct approach. Path discovery based on path awareness provided by client's RAE
2. Reverse approach. Path discovery based on path awareness provided by servers' RAEs.
3. Smart flooding approach. Experimental solution based on flooding from the servers' domains.

In all the three approaches, for the Path Discovery process the CME requires the Routing Aware information that must be supplied by the Routing Awareness Entity (RAE).

On the one hand, the CME needs to be aware of the paths that can be followed between the different domains and the CoS that can be achieved to deliver the contents through these paths. This information is necessary in order to take decisions about the candidate server and path to deliver the content, so the CME needs to interact with the RAE in order to get this information.

On the other hand, the RAE knows about the routes and the topology of the network by its interfaces with other RAEs and provides routing awareness to the CME. This routing awareness will be enriched with information about the CoS of the paths between domains, that is, the NLRI exchanged between RAEs includes information about the CoSs that can be achieved while transferring contents along these paths. The Deliverable 4.1 [49] provides a detailed description of these processes and how the COMET System deals with Classes of Service.

Direct approach

In the Direct approach, the CME holds in a database all the information gathered from the RAE as part of the routing awareness process. This process is performed in an offline manner. When this routing information is needed as an input for the Decision Process, the CME just have to look up for the required information in the database related to the paths that can be followed between the domains that are going to take part in the delivery of the content.

It is assumed that the RAE does not provide more than one path for each CoS that can be achieved to reach each domain.

The table that holds the routing information received from the RAE includes subnets, the Classes of Service that could be achieved to reach that subnet and the distance to the subnets.

The scheme that is followed for the exchange of information between the CME and the RAE could be potentially different to that followed in the table that the CMEs maintain. But what it is necessary is that there must be a clear correspondence between the Network CoSs (RAEs level) and the CoSs related to content specification (CMEs level), in order to be possible for the CMEs to map between them.

Reverse approach

The second approach assessed for the Path Discovery process is the Reverse approach, which uses reverse path retrieval and caching mechanisms. The objective of this approach is to gather the information about paths going from the server domain towards the client domain (called reverse paths). The resulting information allows the CME located in client domain to perform the following tasks:

- selection of the best pair <path, server> (Decision process),
- enforcement of the selected path (Path Configuration process).

In this approach we distinguish two processes. The first one is the routing awareness process that collects the information about the inter-domain network paths. Those paths define the interdomain connectivity for the particular COMET Classes of Service (CoS)[6]. This process is performed by the Routing Awareness Entity (RAE). Note that the RAE located in client domain provides information about paths going towards any server domain, so called "forward paths". However, during the decision and consumption process we need information about the reverse paths, which would be used for content delivery. Therefore, in Path Discovery process we use Reverse Path Retrieval mechanism, which retrieves information about the reverse paths to the client domain.

Routing Awareness process

The Routing Awareness is an offline process operating in long time scale. It reacts only to changes in the inter-domain topology or in the re-provisioning of domains. The Routing Awareness computes long term properties of paths going towards a given network prefix based on information provided by particular domains. These properties should cover:

- COMET CoSs supported along the path,
- path length expressed in terms of number of domains,
- the list of domains on the path, e.g., the list of AS numbers,
- vector of QoS parameters characterizing the path, i.e., values of maximum packet losses or maximum delay, or optionally, some metric defining the "overload probability" (relation between offered load and path capacity). Note that those values should be long term parameters that do not depend on the carried traffic. For example, they should come from the provisioning process of particular domain.

Note that the path properties cannot be confidential as they are advertised through entire Internet [26].

The Routing Awareness should be performed by the RAEs operating in each domain. The RAEs exchange update and withdraw messages between peering domains. Each RAE selects a set of preferred paths, updates path properties and propagates selected paths to peering domains. Note that the RAEs require close cooperation with domain's administrator because:

- there is a mapping between COMET CoSs and CoSs supported inside a given domain,
- the calculation of the path properties requires knowledge about the domain properties, e.g., the AS number and the QoS parameters that characterize given domain from ingress to egress,
- each domain could have its local policies and agreements with peering domains that may impact the path selection process.

All details of the proposed routing awareness process are provided in deliverable D4.1 [49]

Reverse Path Retrieval mechanism

[6] The COMET CoSs are described in details in D4.1

The Reverse Path Retrieval (RPR) mechanism provides information about paths going in reverse direction, i.e., from the server domain to the client domain. The RPR mechanism is performed by a component of the CME called Path Storage. It is invoked upon each consumption request in the client's domain CME when the list of servers becomes available. It follows the next steps:

1. Whenever the Path Storage component is asked about the reverse paths, it checks whether information about reverse paths is available in its local cache. If this is so, it returns the list of paths and the process is finalized.
2. If the path information is not present, then the Path storage component will ask the Path Storage component located in servers' domains for available paths to the client's domain.
3. Path Storage component stores the received paths in its local cache and returns the list of paths to the client's domain CME. Then, the process finishes.

The reverse path caching will reduce the inter-CME interaction while maintaining information about the paths from frequently used servers.

Smart flooding approach

This procedure is initiated upon every content request and finds all available paths from the CME adjacent to the content server storing the content towards the CME adjacent to content consumer. It actually calculates all available domain paths, from which content could be delivered, while taking into account the content consumer's preferences or capabilities (e.g. CoS).

The main idea behind this mechanism is that paths over the Internet are typically asymmetric, thus discovered paths from client towards content server might be different or offer different CoS at a specific time from the ones towards the opposite direction (which is the actual content consumption path). The proposed mechanism performs "smart flooding" of the network with find_path messages, where every node forwards them to its neighbours that meet the content consumer's CoS requirements, until client's respective CME is reached. This node will be then responsible to select the most relevant path and content server for content consumption, through the COMET decision process.

The COMET functions that are involved in the Path Discovery process are the Content Mediation Function (CMF) and Path Management Function (PMF), located both in the edge (client and content server) and intermediate domains.

After Name resolution, the CMF, upon receiving the requested content record from the CRF, has the list of content servers that contain copies of the content, their adjacent CMEs, as well as relevant content properties and initiates reactive and reverse path discovery process. The steps that are presented in Figure 19 and are going to be described below present the whole process for one single content server, but we assume that the same procedure is going to be performed for every content server in the content server list:

1. The CMF adjacent to the content client sends a discover_paths message to the CMF adjacent to the content server, in order to initiate path discovery process. The message will contain the CN, the content server's host identifier, the requested CoS and other relevant properties.
2. The CMF adjacent to the content client also notifies its respective PMF to start a timer and wait for find_path messages for the specific content request, by sending a wait message which includes the requested CN, the host identifier of the CME adjacent to the content server and requested CoS.
3. The CMF adjacent to the content server stores the received discover_paths message in its request table, in order to be able to initiate content consumption if it is requested.
4. It also forwards the received discover_paths message to its respective PMF, which will initiate path discovery process towards the PMF adjacent to content client.

5. Then the PMF starts sending find_path messages to its outgoing links. The interactions and other messages exchanged between intermediate PMFs will be presented in the execution of the path discovery algorithm.
6. Finally, the PMF adjacent to content client receives find_path messages, waits for the timer, initially set, to expire and returns all discovered paths to its respective CMF, in order to select the best pair of content server and discovered path.

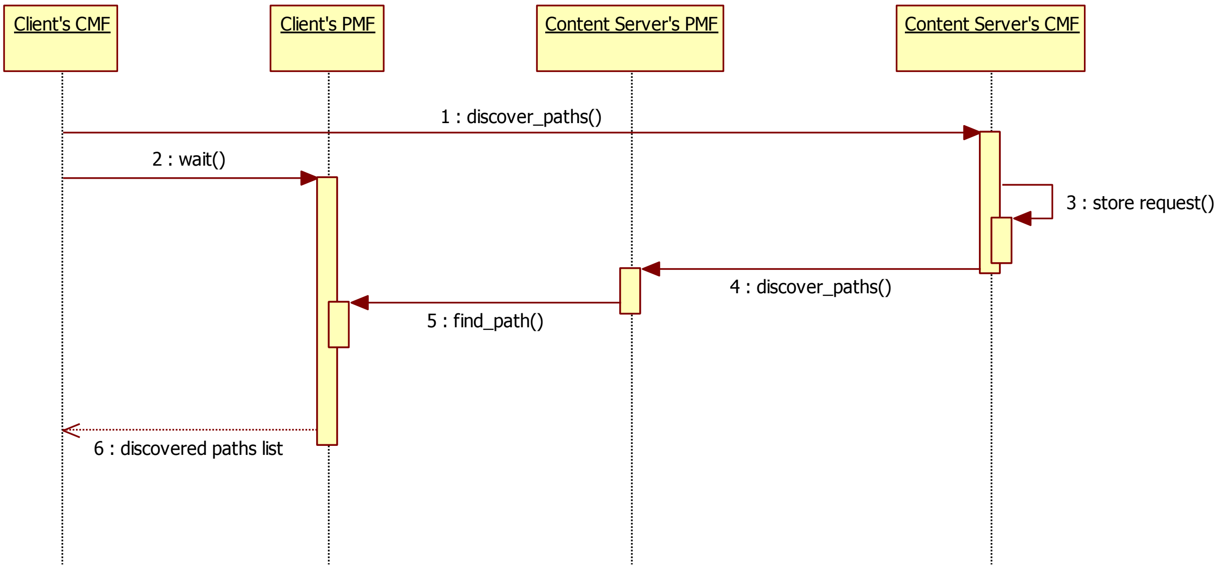


Figure 19: Basic interactions between CMFs and PMFs during reactive path discovery

Upon the smart flooding discovery process, the CMF will select the best path and content server with the decision process criteria defined in Section 5.1.5.3, thus will also need to configure PMFs accordingly (delete path discovery table entries for non-selected paths). This process is described in Section 5.3 of Deliverable 4.1 [49].

In the description of our mechanism, we define the following parameters, messages and tables:

The parameters are the following:

- <CN, Client CME, Content Server CME>: This triplet contains the content name, the host identifier of the CME adjacent to content client and the host identifier of the CME adjacent to content server and is used to identify a specific content request from a content server adjacent to its respective CME towards a content client.
- CS is the host identifier of the content server storing the content.
- Hop_count is included in find_path messages to identify the number of hops of the path already constructed.
- Previous- and Next-Hop are the host identifiers of the previous- and next-hop CMEs.
- CoS_link is the maximum QoS class that 2 connected CMEs can offer at a given time period. This information is updated regularly.
- CoS_req is the QoS class requested by the content consumer. It is included in the content consumer's initial request for a specific content and is used throughout the whole process to include or exclude certain paths in the list of possible content consumption paths.
- CoS_path is the QoS class of the path between source CME and current CME that has already been constructed.

The messages exchanged are the following:

- `discover_paths` message consists of:
    - <CN, Client CME, Content Server CME>
    - CS
    - CoS_req
- `wait` message consists of:
    - <CN, Client CME, Content Server CME>
    - CoS_req
- `find_path` messages consist of:
    - <CN, Client CME, Content Server CME>
    - CoS_path
    - CoS_req
    - Hop_count
- `ACK/NACK configure_path` messages consist of:
    - <CN, Client CME, Content Server CME>
    - CoS_path
    - Hop_count
    - ACK/NACK identifier

The tables are the following:

- `Path Discovery Tables` (PDT) keep records of all find_path messages sent by a PMF to its neighbour PMFs and store the following parameters:
    - <CN, Client CME, Content Server CME>
    - Previous-Hop
    - Next-Hop
    - CoS_path
    - Hop_count
- `Request Tables` (RT) are used to keep records of the specific content request when initiating path discovery mechanism. For a specific content request only the CMF adjacent to content server holds a request table, in order to be able to start content consumption when the content delivery path is selected.
    - <CN, Client CME, Content Server CME>
    - CS
    - CoS_req

We identify 2 main phases while reactive path discovery algorithm executes, the Initialization phase when the PMF adjacent to content server initiates the process and the Execution phase when any PMF forwards find_path or configure_path messages to its neighbour or previous-hop PMFs, according to specified criteria.

During the Initialization phase, the PMF adjacent to content server initiates path discovery process, finds its neighbour outgoing links and for each of those links checks the available CoS that can be served. More specifically, it checks if every outgoing link is provisioned with a CoS higher or equal the CoS_req (CoS_link>=CoS_req). If it does so, sends a find_path message to the outgoing link, where CoS_path is the known CoS_link of the outgoing link in this case and hop_count is equal to 0 for the initialization phase.

It also adds a record of the sent find_path message in its path discovery table and repeats the process for the next neighbour outgoing link, until there is no other.

The algorithm of Initialization phase is presented in Figure 20:

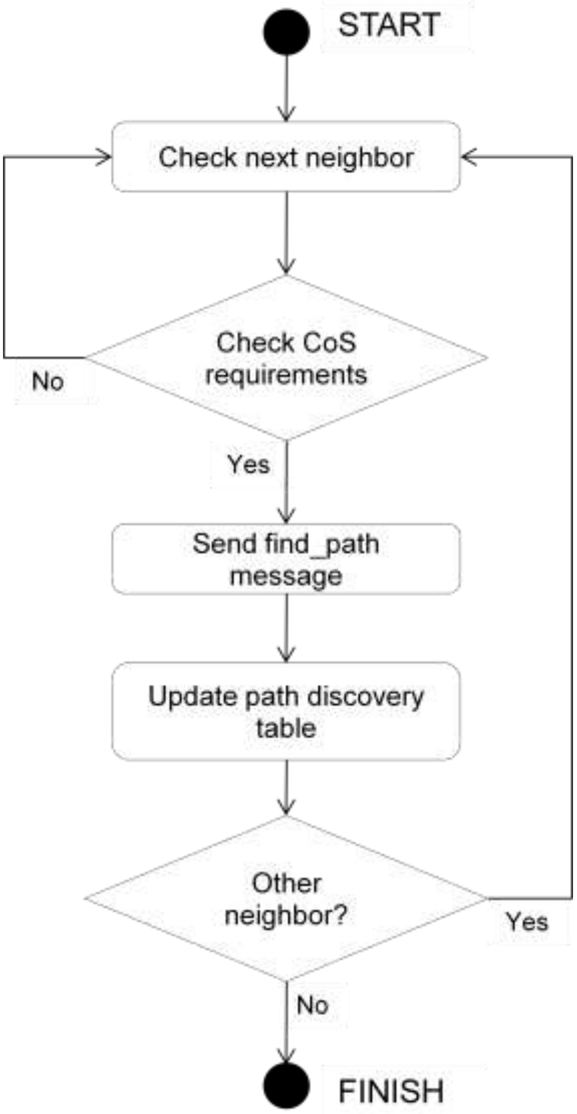


Figure 20: Initialization Phase of reactive path discovery

During Execution phase, any PMF, upon receiving a find_path message, finds its neighbour outgoing links (the link from which find_path message was received is excluded) and for each one checks if it is provisioned with a CoS higher or equal the CoS_req (CoS_link>=CoS_req). If that is true, proceeds to hop count check. If not, then proceeds to the next neighbour.

PMF also checks for every neighbour outgoing link if find_path message was sent towards the opposite direction in previous steps of the path discovery process. This is done by checking the following 3 parameters (we exclude the other ones, since we assume that is for the same content request from the same content server): Next-Hop, Previous-Hop, Hop_count.

Let's assume that in a specific step of the path discovery process, PMF B has a stored record in its path discovery table for a find_path message sent to PMF C during Hop_count x, as presented in Figure 21 below, and that now B receives a find_path message from PMF C, with Hop_count y.

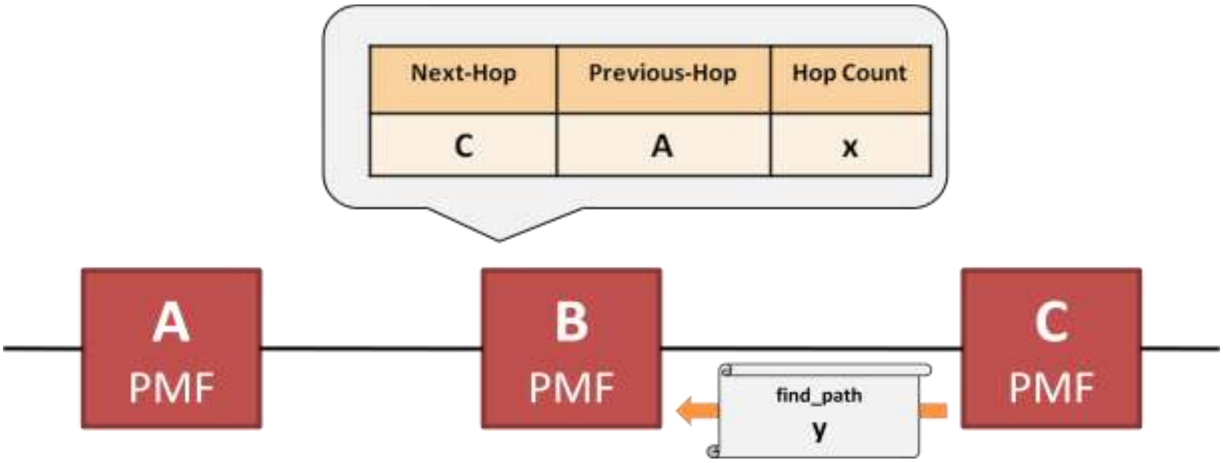


Figure 21: Example of reactive path discovery

The PMF B knows that its only outgoing link is towards A, checks CoS requirements for BA link (let's assume that it is fulfilled) and finally has to check if messages were sent in the opposite direction in previous steps of the path discovery process. Thus, PMF B checks its path discovery table and identifies a record {C, A, x}, while now it would have to add a record {A, C, y} for sending a find_path message to A:

- If y>x, then it will not send a find_path message towards A and will also send a NACK configure_path message to C.
- If y<=x, then it will send a find_path message to A and also update its path discovery table with a record for {A, C, y}.

If no outgoing link fulfil CoS and hop count requirements, then the PMF must send NACK configure_path messages to its previous-hop PMFs to delete path discovery table records, related to the specific <CN, Client CME, Content Server CME> triplet and Hop_count.

In case that a find_path message is sent, then CoS_path is the minimum of the next hop CoS_link and the CoS_path received, i.e., CoS_path = min {CoS_link, CoS_path}. This way, the CoS_path parameter holds the maximum CoS that can be supported throughout the currently constructed path.

Finally, the PMF updates its path discovery table accordingly with record for find_path message sent to a neighbour and proceeds to next neighbour outgoing link.

The algorithm of Execution phase is presented in Figure 22.

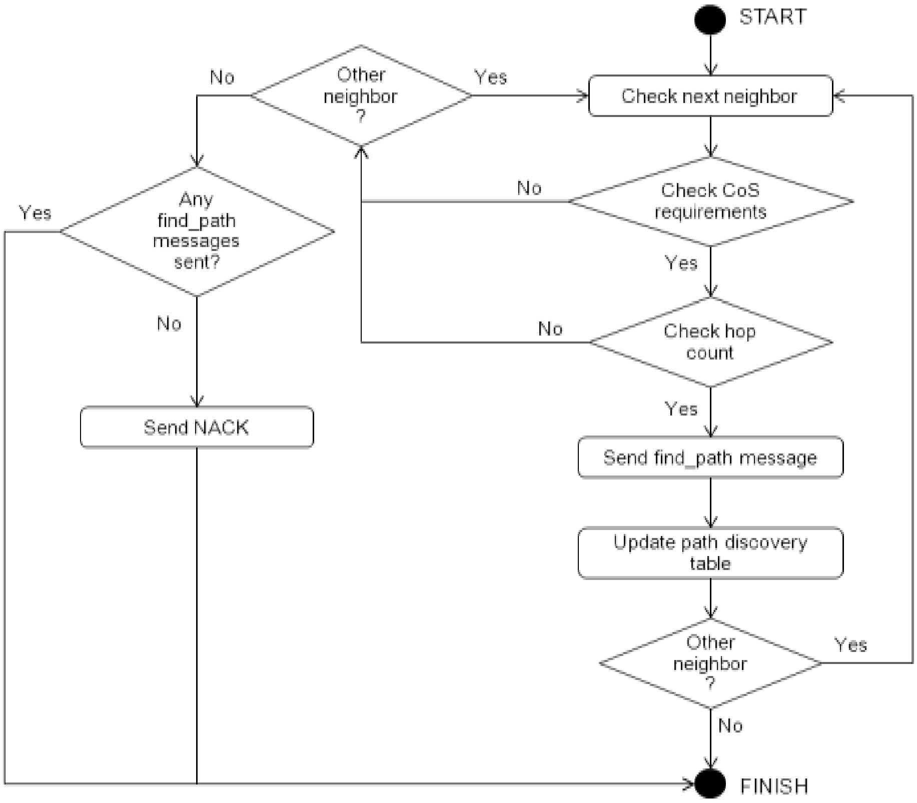


Figure 22: Execution phase of Path Discovery Algorithm

Records in path discovery tables may also be cached and reused in future content requests for the same requested CoS, eliminating the time needed to discover all possible paths from content servers to a content client each time, as well as improving the scalability of the system and reducing the number of messages exchanged in the network. Thus when a CMF receives the essential parameters for a specific content request, instead of initiating the path discovery process, may ask its respective PMF to check if there is any record in its path discovery table, related to the requested content and CoS.

5.1.5.3 Decision Process

The decision process is performed by the CME situated at the client's domain. It aims to select the best server and path (from the available candidates) that may deliver the content to the client that has requested it. This process is invoked upon each consumption request, when information about candidate servers and paths becomes available.

The process starts when the CME of the client domain receives the Content Record as a result of the Name Resolution process. The Content Record includes the list of servers that may transmit the content as well as the information about the servers' condition, e.g. server's load. Once retrieved the list of servers, the CME obtains a list of paths invoking the Path Discovery process. In Table 2, we present an example list of paths with characteristics that may be taken as an input for the Decision Process.

Table 2: Example list of paths provided to Decision Process

Server/path	COMET CoSs	Path length [num of AS]	List of AS numbers	vector of QoS parameters, e.g. [mean IPDT, IPLR...]
Server 1:				
Path 1	Best effort	2	AS X, AS Y	[40, 0.02]
Path 2	Better than Best Effort	3	AS J, AS K, AS Y	[10, 0.005]
Path 3	Premium Service	3	AS J, AS K, AS Y	[10, 0.001,]
Server 2:				
Path 1	Best effort	2	AS A, AS B	[80, 0.008,]
Path 2	Better than Best Effort	3	AS A, AS C, AS D	[40, 0.001,]

Moreover, the CME may collect some short term information from its local SNME related to e.g. the current load, some statistics about last requests between its clients and servers, etc. In general, the Decision process finishes with the selection of the best server and path. Whenever the decision is taken, then the CME initiates the preparation of the content delivery process.

The complex set of parameters used in the COMET Decision Process requires a multi-criteria decision algorithm. These algorithms are widely studied in the Multiple Criteria Decision Analysis (MCDA) [28][29]. MCDA splits into single and multiple objective optimization. In general, the Decision Process in COMET requires Multi-objective optimization [31]. It assumes that the CMF is able to build a model of the problem, which supplies a set of effective solutions.

The basic model of Multi-objective optimization defines the decision space $\Re^i$, which consists of the decision vectors $x=(x_1, x_2, \ldots, x_i)$. Each decision vector contains i decision variables. Any decision variable may have bounded amount of feasible solutions by given constraints. Therefore, the space of decision vectors may be also bounded.

Multi-objective optimization focuses of optimizing a set of k objective functions $\Pi_1(x), \Pi_2(x), \ldots, \Pi_k(x)$, which are called aggregate objective function[27]. These functions can be either minimized or maximized[7]. So, the aggregate objective function is simply a vector of objective functions:

$$\Pi(x) = (\Pi_1(x), \Pi_2(x), \ldots, \Pi_k(x))$$

To each decision vector $x \in X$, exists one unique objective vector $y \in Y,\ where\ \Pi : X \rightarrow Y$ with,

$$y = (y_1, y_2, \ldots, y_k) = \Pi(x) = (\Pi_1(x), \Pi_2(x), \ldots, \Pi_k(x)).$$

In Multi-objective optimization, a solution *x"* dominates the solution *x'* if and only if

$$\forall k^* \in \{1, \ldots, k\}:\ \Pi_{k^*}(x'') \leq \Pi_{k^*}(x')\ \ and\ \exists k^- \in \{1, \ldots, k\}:\ \Pi_{k^-}(x'') < \Pi_{k^-}(x')$$

and a solution *x'* is called efficient if and only if there is not another solution *x"*, which dominates x'.

The set of efficient solutions is the Pareto optimal set and the set of all the outcome vectors *y* resulting from $y=\Pi(x)$ where *x* is an efficient solution, is the Pareto Frontier.

Whenever the Pareto optimal set contains more than one efficient solution, the Decision Process should choose one of them. In fact, the Decision Process could (1) provide a priori some knowledge

[7] The problem does not loose generality by the fact that we consider uniquely minimization.

about the problem in order to ensure that the effective solution outgoing from the model is unique or (2) consider a posteriori the whole set of effective solutions and choose one unique solution. In COMET, we focus on designing appropriate aggregate objective function that allows getting unique solution that optimizes the efficiency of content delivery.

In the Decision Process, the decision space consists of the decision vectors with the following set of possible decision variables:

1. Decision variables related to the content (obtained from Content Record):
    - COMET Class of Service (C_{CoS}),
    - traffic descriptor – set of parameters describing content transfer, like Peak Bit Rate (*PBR*), Sustainable Bit Rate (*SBR*) of the Content
    - content duration (seconds) or length (bytes).
2. Decision variables related to the servers (obtained from SNME):
    - the server load (ρ_s) - a quality parameter of the load of the server, e.g., high, medium or low load. We may quantify the qualitative parameters by associating the parameters to the range of values,
    - the server bandwidth (BW_S).
3. Decision variables related to the client (obtained from client registration):
    - Client maximum subscribed BW (BW_c).
4. Decision variables related to the path properties (obtained from Path Discovery process):
    - COMET Class of Service supported on the path (P_{CoS}), understood as the COMET CoS available on the path from the server domain towards the client,
    - path length (d_p) expressed in terms of the number of domains, or of the number of required content aware forwarders,
    - the list of domains on the path, e.g., the list of AS numbers,
    - vector of QoS parameters (path weight) characterising the path, e.g., values of maximum packet losses, maximum delay, bandwidth,
5. Decision variables related to local information at the client's side:
    - network load conditions in the client's domain, e.g. load on the peering ingress and egress links (ρ_p),
    - successful rate of preceding path enforcement processes (μ_p), where consumers at the client's domain where involved. This information is attached to the path,
    - number of connections served by the path (N_p), which could help for load balancing decisions,

It is important to point out that for the Decision Process, we have considered a number of decision algorithms that differ in a set of decision variables as well as in the aggregate objective functions. Our objective is to evaluate the different algorithms and select the most effective one, which assures an effective content delivery. In our studies we will consider two factors. The first one is a set of decision variables, which should be taken into account and, the second one is the appropriate form of the aggregate objective functions. In Table 3, we briefly present some currently considered decision algorithms. Note, that the list of candidate algorithms is still open and will be updated during our studies.

Table 3: Considered decision algorithms

Decision algorithm	Decision variables	Constraints	Aggregate objective functions	Comments
Algorithm 1 *Path length*	*C_{CoS}, P_{CoS}, d_p*	$C_{CoS} - P_{CoS} = 0$	$\min(d_P)$	This algorithm considers only the path length. It selects the closest server.
Algorithm 2 *Server load*	*C_{CoS}, P_{CoS}, ρ_s, SBR, BW_s*	$C_{CoS} - P_{CoS} = 0$ $SBR - BW_S \leq 0$	$\min(\rho_S)$ $\min(BW_S^{-1})$	This algorithm considers only the server load. It selects the least loaded server.
Algorithm 3 *Server and path length*	*C_{CoS}, P_{CoS}, ρ_s, d_p, SBR, BW_s*	$C_{CoS} - P_{CoS} = 0$ $SBR - BW_S \leq 0$	$\min(\rho_S)$ $\min(BW_S^{-1})$ $\min(d_P)$	This algorithm considers both the server load and path length. It selects the least loaded server and the closest server.
Algorithm 4 *Server and path characteristics*	*C_{CoS}, P_{CoS}, ρ_s, d_p, SBR, BW_s, p_{weight}*	$C_{CoS} - P_{CoS} = 0$ $SBR - BW_S \leq 0$	$\min(\rho_S)$ $\min(BW_S^{-1})$ $\min(P_{weig}$	This algorithm considers both the server load and path characteristics. It selects the least loaded server with the path of the best characteristics.
Algorithm 5 *Server, path characteristics and local information*	*C_{CoS}, P_{CoS}, ρ_s, d_p, SBR, BW_s, p_{weight}, μ_p, N_p*	$C_{CoS} - P_{CoS} = 0$ $SBR - BW_S \leq 0$	$\min(\rho_S)$ $\min(BW_S^{-1})$ $\min(\mu_P^{-1})$ $\min(N_P)$ $\min(d_P)$ $\min(P_{weig}$	This algorithm considers the server load, path characteristics and local information about constraints. It selects the least loaded server with the path of the best characteristics that meet local constraints.

Besides these presented algorithms and in order to increase the effectiveness of content delivery, we will analyze the recently investigated approaches that introduce a reference point [30] and [32]. By using this method, the aggregate objective function finds the effective solutions of the Pareto optimal set, which are nearest from the given reference solution (generally the reference solution may be not into the Pareto optimal set). As an example we can think about an algorithm, which looks for a medium server load and medium path load solution avoiding other solutions as low server load and high path load.

The considered aggregate objective functions do not avoid having more than one effective solution into the Pareto optimal set, so, we should define tie-breaking rules, which will allow us to select only one solution, e.g. fewer server load is preferred. Let us remark that whenever two effective solutions of the Pareto optimal set define exactly the same outcome vector of the Pareto frontier, then the ordering is not enough and then, the process should randomly choose one of them.

5.1.6 Preparation for Content Delivery

As a result of the Decision Process, the CME knows the server and path that will be used for the delivery of the content. The following step is the preparation of the underlying network in order to deliver the content from the Content Server to the Content Client through the chosen path with the required CoS. For this process, the CME enforces some rules in the CFP on a per consumption basis. Before the path configuration starts, the following properties should be known:

- Path information:
    - Path identification – the sequence of the involved domains in form of AS number list,
    - COMET Class of Service – the name (or enumerated value),
- Content information:
    - Content identifier – in case it is used for forwarding,
    - Traffic descriptor – the set of parameters for double token bucket; the meaning of the values may be different for each COMET Class of Service,
    - Duration (seconds) or length (bytes),
- Transport information:
    - Transport protocol (UDP/TCP),
    - Server network address, server port number
    - Client network address (it could be a gateway's address when server's address is hidden), optionally consumer port number.
- Address of COMET entity in the server's domain

Moreover, we assume that all local policy checks in client domain are already performed, e.g., available bandwidth at the domain's ingress is sufficient to fulfil the consumption.

The process of path enforcement follows the steps (we consider only the positive scenario):

1. Client's CME (cCME) sends the request to the server's CME (sCME). It includes: path, content and transport information.
2. sCME performs sanity check
    a. Does the server's address belong to sCME's domain?
    b. Is the path information valid?
3. sCME performs policy check
    a. Should sCME do anything for this COMET Class of Service? If it is Best Effort service, then it should do nothing.
    b. Does sCME have the resources to handle this consumption? (traffic description vs. available resources → admission control)
    c. Other options, e.g., inter COMET trust management
4. sCME maps the path information to the "forwarding rules"
    a. Path information (AS list + COMET Class of Service) → forwarding rules (binary data)
    b. Forwarding rules will be used in CAFE forwarding. They could be encoded or encrypted in a way that only "next hop" CAFE knows how to use them. Details about forwarding rules' structure will be provided in deliverable D4.1 [49].
    c. Forwarding rules are usually known for a given path. They can be predefined (statically) during provisioning, or they can be obtained on demand (dynamically) following COMET entities along the AS path. It would be reasonable to store the

result of "on demand operation" in sCME for further use (in a cache, which can be invalidated by duration or by path changes).

d. There are multiple optional features as, e.g., particular paths may require following always on-demand operation and performing admission control in mid domains.

5. sCME finds the CAFE, which handles the content server
    a. Network address (IP) → network prefix → CAFE
6. sCME configures the flow classifier in the CAFE
    a. The selected CAFE receives "transport information" and uses it for packet classification using multi-field classification filter.
    b. Each classified packet is modified with COMET forwarding header, which contains forwarding rules and other data. Details about COMET forwarding header will be provided in WP4 documentation.
    c. The validity of classification rules may be defined by different means, e.g., deadline in the future, maximum duration of silence (refreshed by traffic itself).
7. sCME responds to the cCME with positive result

Figure 23 presents the message sequence chart for the path configuration process.

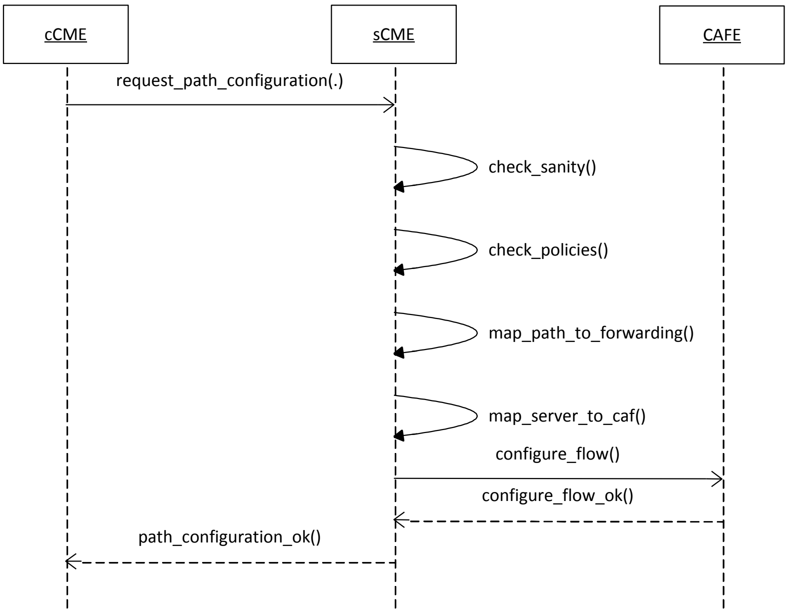


Figure 23: Path configuration message sequence chart.

5.1.7 Interim Study

The architecture of the decoupled approach for the COMET System described in the previous sections has been designed with the aim of meeting certain aspects as scalability, deployability, feasibility, security and sustainability.

In our study, we have focused on the Content Mediation Plane of the COMET architecture and analysed each of the different operations that it performs by separate. These are the following:

5.1.7.1 Content Naming Scheme

The Content Naming Scheme proposed for the Content Record-based Resolution approach has been designed with the purpose of being able to handle efficiently large amounts of content, being able to support significantly more objects than those handled currently.

Given that the proposed Content Names are character strings and can have a variable length, they can be arbitrarily long to be able to deal with billions of content objects.

5.1.7.2 Content Publication

The Content Record-based resolution approach offers to Content Providers and Content owners a unified interface in order to facilitate them the publication and distribution of their contents.

In addition, the approach achieves the publication through trusted interfaces with the CRF, so safe and trusted transactions with the COMET system are guaranteed.

5.1.7.3 Name resolution

The scalability of the name resolution operation of the COMET System has not been proven yet. Further study is required for both approaches for the resolution (DNS-based and The Handle System-based).

Regarding the DNS-based resolution approach, it is supposed to have similar behavior to the DNS System with respect to the aspects of the study. This system is able to achieve content resolution in a range between 50 and 150 ms depending on whether cache techniques are applied or not, so the studies should focus on the possible deterioration that these times could experiment because of the retrieval of the Content Records instead of the current DNS response.

5.1.7.4 Path discovery

The path discovery approaches that have been described in section 5.1.5.2 require further study in order to revise the aspects of scalability, duration of the different approaches and deployability.

5.1.7.5 Decision process

The decision algorithms that have been described in section 5.1.5.3 require further study in order to revise the aspects of scalability, duration of the process and deployability.

It is important to point out that for the Decision Process, the decision algorithms that have been considered differ in a set of decision variables as well as in the aggregate objective functions and the future studies will focus on the evaluation of the different algorithms and select the most effective one, which assures an effective content delivery.

5.2 Coupled Resolution and Delivery Approach

5.2.1 Overview

Various researchers are starting to advocate the transition of the Internet model from *host-centric* to *content-centric*, with various different architectural designs proposed [11][39][12][2][4][8]. Many of these proposals support the key feature of *location independence*, where content consumers do not need to obtain explicit location information (e.g., the IP address) of the targeted content source *a priori*, before issuing the consumption request [4][11][12][39][40]. Nevertheless, under some circumstances location requirements are still demanded not only by content consumers but also by content providers. On the one hand, content providers may want their content accessed only by content consumers from a specific region (which is known as *geo-blocking*), for example BBC iPlayer, Amazon Video-on-Demand, Apple iTunes Store and Sina video services. On the other hand, content consumers may prefer content originated from specific regions in the Internet, for instance, a US-based shopper might only like to check the price of an item through the user's explicit input in the URL (e.g. Amazon.com and Amazon.ca), and supported by name resolution through the standard Domain Name System (DNS) [15], with the relevant requests directed towards the specific regional web server. Similar practice can be observed in multimedia-based content access (e.g. in video on demand services), where consumers have specific requirements regarding the location/area of content sources.

We introduce in this section a coupled content resolution and delivery infrastructure. The objective is to both accurately and efficiently "hit" (or "not hit") content objects in *specific regions/areas of the Internet*, based on specific user requirements and preferences. Such a paradigm, deployed by ISPs, will allow both content providers and consumers to express their location requirements when publishing/requesting content, thanks to the embedded content *scoping/filtering* functions. In particular, instead of following the conventional DNS-like approach, where a content URL is translated into an explicit IP address pointing to the targeted content server, our proposed content resolution scheme is based on hop-by-hop "gossip"-like communication between dedicated content mediation entities residing in individual ISP networks. Content resolution operations can be driven by a variety of factors, including the business relationships among ISPs (provider/customer/peer), content consumer preferences and local ISP policies. This resolution approach is natively coupled with content delivery processes (e.g., path setup), supporting both unicast and multicast functions. More specifically, a content consumer simply issues a single *content consumption request* message (capable of carrying his/her location preferences on the content source candidate(s)), and then individual content mediation entities collaboratively resolve the content identifier in the request, in a *hop-by-hop manner*, towards the desired source. Upon receiving the content consumption request, the selected content source starts transmitting the requested content to the consumer. During this content resolution operation, "multicast-like" content states are installed along the resolution path so that the content flows back immediately upon the completion of the resolution process. This is in contrast to the current IP-based content delivery services where name resolution and content delivery are separate processes. By exploiting multicast delivery techniques, we increase the sustainability of the system in view of the expected explosion of content in the Internet.

5.2.2 Content Naming Scheme

The coupled approach requires a form of aggregatable labels for content capable of being sequentially ordered. They are referred to as *content identifiers* (IDs). A content item to be published and accessed is allocated a globally unique content ID. Multiple copies of the same content that are physically stored at different sites in the Internet share one exclusive ID. We assume a new ID space which is sufficiently large (e.g. 40 bytes space similar to [12]) to ensure system sustainability.

There will be no specific information encoded into these IDs. Thus they do not contain any semantic and thus independent of structure. This is an important feature. Firstly, without encoding information into the IDs, there will be no danger of security breaches as there is simply nothing to exploit from. Second, without encoding information into the IDs, they are then more flexible. For instance, IP addresses reveal location information. If a content is moved from one server to another, the original server must re-direct the request for this content to the new server or simply return an error message. In our approach, such situation will not arise. Another example is when the information on the owner of the content is encoded into the content label. The management and distribution of these content IDs are not enforced within COMET but may rely on external trusted organizations such as what happened in the current IP networks where Internet Assigned Numbers Authority (IANA) [51] is responsible for global coordination of the IP addresses. The specific policies or rules in determining the exact allocation of the IDs are not within the scope of this project. In real world, this is highly dependent on various social and political factors that may have nothing to do with its technical viability.

Technically, this coupled approach only requires that each domain is allocated a set (or multiple separate sets) of sequentially connected IDs.

5.2.3 Unification of Content Access Method

COMET aims to achieve a unification of content access by supporting global search and defragmenting the Internet since in the current Internet, it is often that some content are only accessible (or discoverable) within specific user communities or groups. A content consumer may have to install various applications / clients to find specific piece of content. In this approach, we introduce the Content Resolution and Mediation Entity (CRME) that unifies the access method.

We present the three most basic primitives for global content publishing and consumption for this approach.

Command:	`Register`
Description:	This primitive for content publisher to initiate a content publication process to the COMET system. The content publisher sends this command to its local or immediate CRME. The CRME will create a new content record for this content and then start propagating the record based on the specific publication rules within the coupled approach framework (see section 5.2.5for details).

Usage Format(s):	`Register(OPTION(Domain), Content ID)`
	`Register(OPTION(Domain))`

Options:	`INCLUDE, EXCLUDE`

Command:	`Publish`
Description:	This primitive is used by CRMEs to propagate a new content record based on the coupled approach. Typically, this is the next step after receiving a Register command from a content publisher (see section 5.2.5 for details).
Usage Format(s):	`Publish(OPTION(Domain), Content ID)`
Options:	`INCLUDE, EXCLUDE`

Command:	`Consume`
Description:	This primitive is issued by content consumer via a content client to request for a content. It is sent to the local or immediate content mediation server.
Usage Format(s):	`Consume(OPTION(Domain), Content ID)`
Options:	`INCLUDE, EXCLUDE`

These are the three basic primitives necessary for the working of the coupled framework. Other primitives are also envisioned (e.g. `Update`). The finalized unified access interface will be presented in D3.2.

5.2.4 Overview of Basic Operations

Currently, the Internet is a collection of more than 30,000 domains (or Autonomous Systems (ASes)) structured in a hierarchical manner with each AS classified into different tiers. Tier-1 ASes are typically worldwide ISPs (e.g. AT&T, Sprint etc) while the stub ASes mostly comprised of enterprises and universities. The scale and connectivity level decrease from top of the hierarchy to the stub domains. The reachability of a domain depends on the existence of business relationships with its neighbouring domains. Specifically, there are two types of relationships:

- Transit (or customer – provider relationship) – the transit services are commonly form between low- and high-tier domains. The lower tiered domain effectively purchases the transit service from the higher-tiered domain for Internet connectivity.
- Peering – this relationship usually exists between two domains of the same tier. It allows both domains to exchange traffic flowing through their networks possibly without formal payments to each other.

The coupled approach sets its foundation upon this hierarchical nature of the current Internet domains.

In this approach, the content manipulation operations rely on the Content Resolution and Mediation Entity (CRME) at the Content Mediation Plane (CMP) and the Content-aware Forwarding Entity (CAFE) at the Content Forwarding Plane (CFP).

A CRME primarily handles content publication requests, discovers the requested content and supports the delivery of content while the CAFE collaborates with its local CRME(s) to enforce receiver driven content delivery paths.

The CRME is envisioned to be present in each domain for (1) handling *local* publication requests and content consumption requests and (2) for interacting with their neighbouring counterparts for content publication/resolution across domains. Both content servers and clients are configured to know their local CRME(s). The number of CRME within each domain depends on performance and resilience considerations. For example, having more extra CRME(s) within a domain will provide higher redundancy and thus increase resilience in case of CRME failures.

Figure 24 depicts the functional view of the overall infrastructure; we explain the operational properties of each functional block below. The internal structure of the CRME consists of four logical components. It is responsible for dealing with requests from both content providers and consumers (via CRME-CS and CRME-CC interfaces respectively).

The *Content Resolution Function* (CRF) acts as a content record repository which contains a database of content records that have been propagated or registered within a domain. The CRF allocates content IDs and creates the new content record upon reception of a new content registration instruction. The specific rules on how the content record is created and disseminated are explained in the next sections.

The *Content Mediation Function* (CMF) coordinates and mediates the various content-related operations. It also holds the responsibility of content ID lookup upon each content consumption request from a content client. When a content consumer via the content client sends a content request, the request first reaches the CMF which triggers a communication between CMF and CRF to resolve the request. The CMF will contact its local CRF to verify if the requested content exists within the local index before proceeding to determine the next resolution step.

The *Path Management Function* (PMF) maintains a record of ingress and egress Content-aware Forwarding Entity (CAFE) within the local domain for each active content session being delivered in the network. CMF communicates with PMF for path discovery purposes whereby PMF collects offline routing information from the network (e.g. BGP reachability information) and then enables the CMF to find the possible path(s) for the delivery of a content. To compute the best path, CMF uses additional monitoring information gathered from the *Server and Network Monitoring Function* (SNMF). SNMF gathers necessary “near real-time” information on content server and underlying network conditions for supporting optimized content resolution and delivery configuration operations. These information are then fed to the CMF to enable the mediation of content delivery path.

CMEs communicate with other entities via specialized interfaces as described below:

- Inter-CRME interface – it enables interaction amongst CRMEs in neighbouring domains especially when they cooperate in content publication and searching for a requested content across domains.

- CRME-CS interface – it connects content servers owned by content providers with CRMEs, and allows content providers to publish content, optionally with scoping (see next section) requirements on potential content consumers. This interface is also responsible for passing information on server load conditions to a CRME for enabling optimized content resolution operations.
- CRME-CC interface – it connects content clients with the CRMEs and allows consumers requesting and receiving content with scoping/filtering preferences on candidate content sources.
- CRME-CAFE interface – This bi-directional interface allows a CRME to actively configure relevant CAFEs for each content session (e.g. content state maintenance). It also gathers necessary information from the underlying network that will be used for optimized content resolution processes.

A CAFE is the network element that is able to natively process content packets according to their IDs. In general, it is not necessary for every router in the network to be a CAFE, and typically CAFEs are planted at the network boundary as ingress and egress points for content delivery across ISP networks. The function of CAFEs will be specified later with the description of content delivery process.

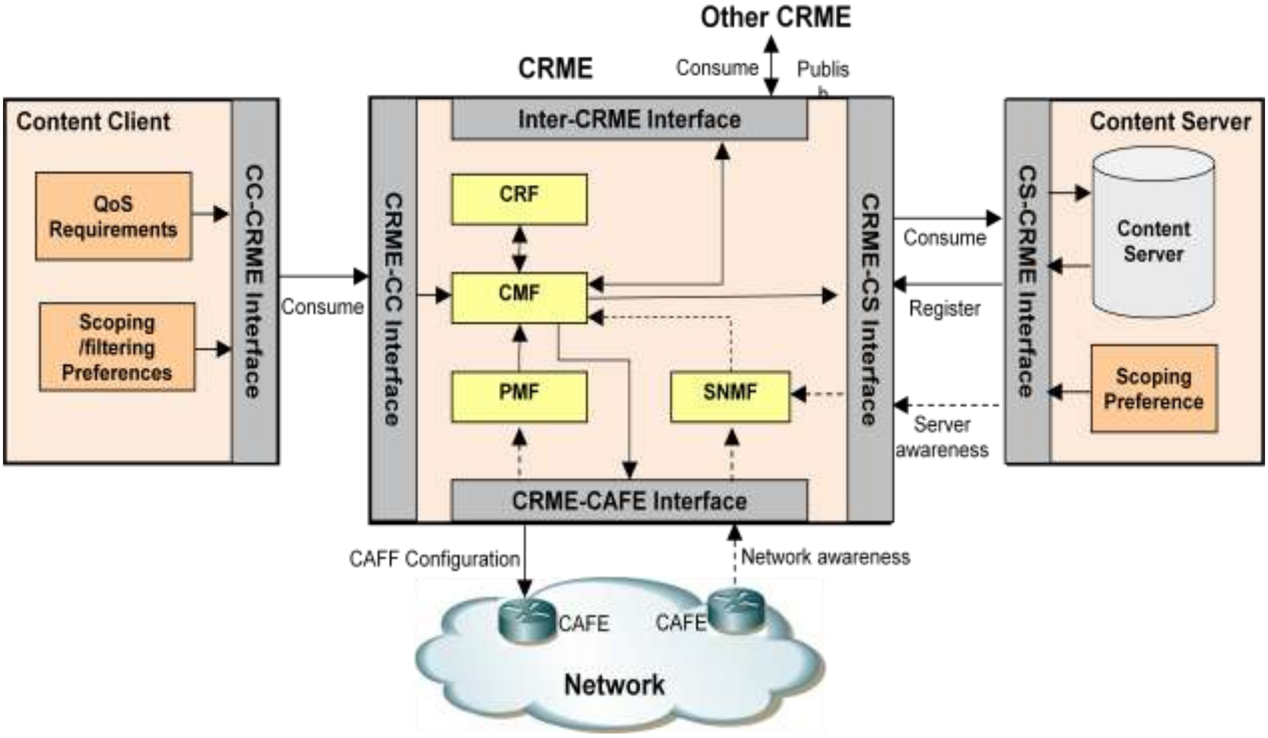


Figure 24: High-level functional view of the hop-by-hop hierarchical content resolution approach

Fundamentally, we envision the following three-stage content operation lifecycle: *publication, resolution* and *delivery*. The task of content resolution is to (1) identify the desired content source in the Internet according to the requested content ID and optionally content consumer preferences, and (2) actually trigger the content transmission by the selected content server. Once the content server starts the transmission of the content upon receiving the content consumption request, the content delivery function is responsible for enforcing the actual delivery path back to the requesting content client. Content publication and resolution can be operated based on various factors, including business relationships between domains (ISP-ISP SLA), inter-domain BGP routing configurations, local ISP policies and content provider/consumer preferences. In the following, we detail the procedure of these stages according to our approach.

5.2.5 Content Publication

Content publication is the process of making content available to interested content consumers across the Internet. It consists of two stages.

Stage 1: Content Registration – It begins with the content provider notifying the local CRME that a new content is now available via a `Register` message. This message will reach the CRF via the CS-CRME interface. In the case where multiple copies of the same content are available at different locations, the content provider is responsible for informing the local CRMEs of each content server hosting that specific content copy. Upon reception of the `Register` message, the CRF registers this content by creating a new record entry in its local content management repository containing at least the following (1) a globally unique content ID assigned to that content, and (2) the *explicit location* of the content (i.e. IP address of the content server). Additional information (metadata) can be also recorded in the content entry for assisting future content resolution operations.

Stage 2: Content Publication Dissemination – Once the content is registered to a CRME (specifically within the CRF), this CRME is responsible for publishing it across the global Internet to ensure successful discovery of the content by potential consumers. This is achieved through the dissemination of the `Publish` message across CRMEs in individual domains according to their business relationships. A Publish message is created by the CRF where the content is actually registered by the content provider. By default, each CMF disseminates a new Publish message towards its counterpart in the *provider domain(s)* until it reaches a tier-1 ISP network. Each CRF receiving a new `Publish` message updates its content management repository with a new record entry containing the content ID and the *implicit location* of the content (i.e. the IP prefix associated with the neighbouring domain from where the `Publish` message has been forwarded). Following this rule, each CRF effectively knows the locations of all the content within its own domain (explicitly) and those under it (its customer domains, implicitly). Peer domains, however, will not know the content records of each other.

We introduce the concept of *scoped publication* to allow publication of content only to specific areas in the Internet as designated by the content provider. This feature is able to natively support regionally-restricted content broadcasting services such as BBC iPlayer and Amazon VoD that are only available within the UK and the US respectively. We achieve this through the INCLUDE option embedded in the `Register`/`Publish` messages where the content provider specifies a scoped area in the Internet, e.g. only the IP prefix associated with the local ISP network where the content is registered. A special case of *scoped publication* is the *wildcard mode* (denoted by asterisk "*" symbolizing *all* domains) for which the content provider has no restrictions on the geographical location of potential consumers in the Internet.

Figure 25 illustrates different scenarios in the publication process. The figure depicts the *domain-level* network topology with each circle logically representing a domain containing a CRME entity. For simplicity and clarity, we refer the content provider and content server as one entity in this section. We first assume that content provider **S1** registers a content item (assigned with ID **X1** by the local CRF in the stub domain **A.A.A**) to the entire Internet by issuing a `Register` message with a wildcard. Each intermediate CRF along the publication path creates a content entry for **X1** associated with the IP prefix of its customer domain from where the Publish message has been forwarded. For clarity, the `Publish` messages are omitted in the figure for other scenarios. Our approach also allows local domain policies to influence the publication process (e.g., domain **B.A** has the policy of NOT propagating content **X2** originated from the multi-homed domain **A.B.B** to its own provider). **S3** illustrates the *scoped* registration by only registering content **X3** to tier-2 domain **A.A** from this content provider. This effectively limits the access of content **X3** to domain **A.A** and its customer domain **A.A.A**. Finally, records for different copies of the same content can also be aggregated in content publication processes. In the figure, both **S4** and **S5** host one copy of content **X4** respectively, but the two content publication messages from **B.B.A** and **B.B.B** are merged at **B.B**, in which case domain **B** only holds one record with an aggregated location information (**X4**→**B.B**). Upon the actual content consumption request for **X4** from a content client, **B.B** can forward it to either **B.B.A** or **B.B.B** based on performance conditions gathered from SNMF such as content delivery path quality or server load, as will be discussed later.

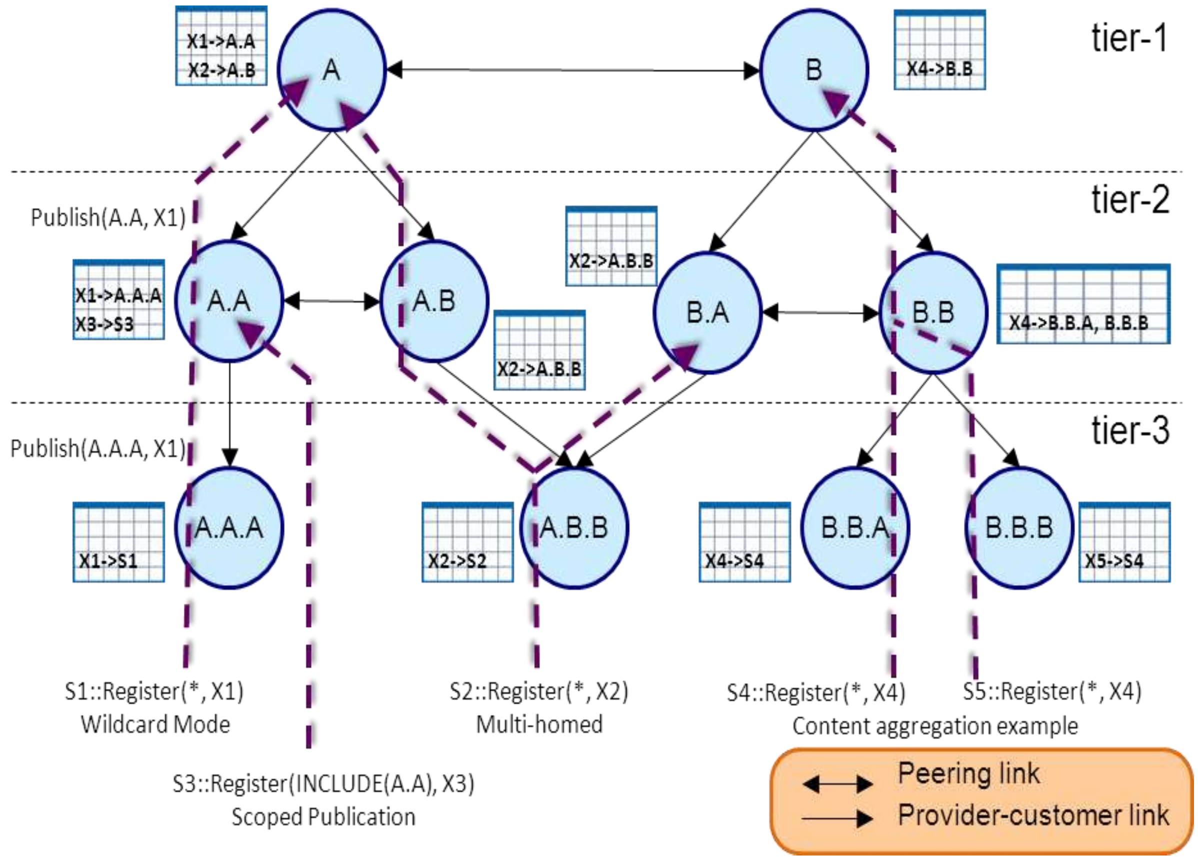


Figure 25: Content publication process

5.2.6 Content Resolution

In the content resolution process, a content consumption request issued by a content client is resolved by discovering the location of the requested content and is finally delivered to the actual content source to trigger the content transmission. A content client initiates the resolution process via a `Consume` message containing the ID of the desired content. The primary resolution procedure follows the same "*provider route forwarding*" rule in the publication process (i.e. the Consume message will be further forwarded to its provider(s) if the CMF cannot find the content entry in its local CRF repository). In case a tier-1 domain is not aware of the content location, then the request is forwarded to all its neighbouring tier-1 domains until the content consumption request is delivered to the identified content source. If the content is not found after the entire resolution process, an `Error` message is returned to the requesting content client indicating a resolution failure.

We define two distinct content resolution stages:

- **Uphill** – the forwarding of a content consumption request from the local CMF "up" along the provider route until it reaches a domain whose CMF has the record entry for the requested content ID.
- **Downhill** – the forwarding of the content consumption request from the domain whose CMF has the record entry of the requested content ID "down" to the explicit content server that hosts the content.

Similar to the publication process, *scoping functions* can also be applied in the resolution process, either embedded in the request from a content client or actively issued by a CRME for route optimization purposes during the content delivery phase (see next section). Such a function allows a content consumer to indicate preferred ISP network(s) as the source domain of the requested content. Specifically, a content client may use the INCLUDE option in Consume messages, which carry one or multiple IP prefixes (or domain names) to indicate where he/she would like to receive the content[8] from. Since a set of explicit IP prefixes for candidate content source is carried in the Consume message, the corresponding resolution process becomes straightforward: each intermediate CRME only needs to forward the request (splitting required in the presence of multiple non-adjacent IP prefixes) towards the targeted IP prefix(es) directly according to the underlying BGP routes. In case multiple inter-domain routes are available towards a specific prefix, the most explicit one will be followed, as is consistent with today's inter-domain routing policy. In Figure 26, content client **C1** issued a Consume message for content **X1** indicating its preference for content source in domain **A**. This Consume message is then explicitly forwarded towards **A** from **B** following the underlying BGP routing, but without splitting it to **C** despite that a copy of **X1** is also accessible from **C**'s customer domain **C.A**. This scoping-based content resolution path is illustrated with the solid line in the figure.

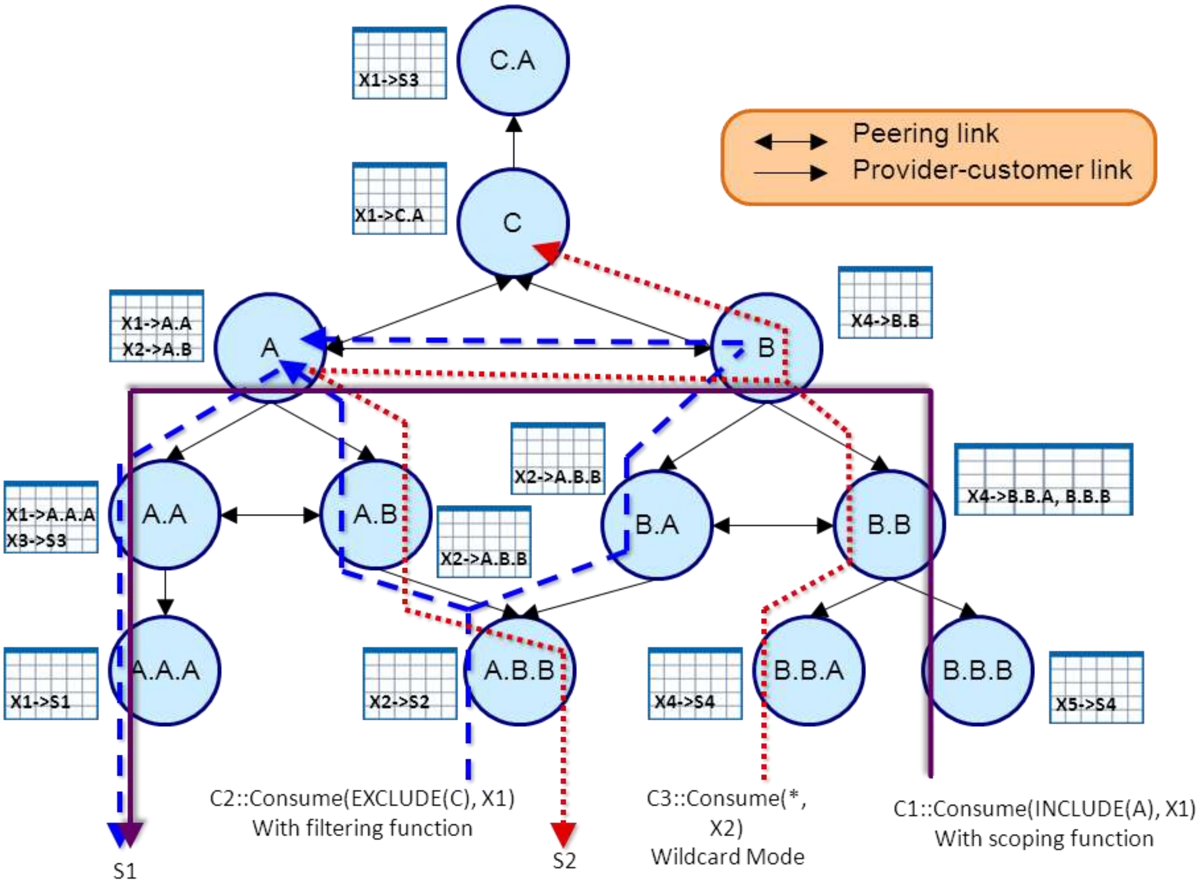


Figure 26: Content resolution in scoping, filtering and wildcard modes

[8] Strictly speaking, it is not always required that ordinary content consumers know the actual IP prefix of the domains they prefer but their local CRMEs may be responsible for translating the "region information" (e.g. domain names) into IP prefixes through standard DNS services.

The *filtering* function in content resolution operations has complementary effect to *scoping*. Instead of specifying the preferred networks, the content consumer has the opportunity to indicate unwanted domains as possible sources of the desired content. The filtering function is enabled via the `EXCLUDE` option in Consume messages. It is important to note the fundamental difference in resolving content consumption requests with *scoping* and *filtering* functions. In contrast to the *scoping* scenario in which a content consumption request is explicitly routed towards the desired IP prefix(es) according to the BGP route, in the *filtering* case, each request is routed based on the business relationship between domains (similar to content publication operations). Consider again Figure 26 with content client **C2** requesting content **X1** with the exclusion of domain **C**. Since it is multi-homed, the request is sent to both its providers **A.B** and **B.A** (see the dashed line in the figure). However, at the tier-1 level, domain **C** is excluded when resolving this request even though a copy of content **X1** can be found in the customer domain of **C**.

A wildcard in a content consumption request can be regarded as a special case whereby the content client does not have any preference on the geographical location of the content source. The wildcard-based resolution is illustrated in Figure 26 via the request from content client **C3** for content **X2** (dotted lines). We see that **B** splits the request to both **A** and **C** at the tier-1 level. Since only domain **A** has the record entry for content **X2**, the request is resolved downhill to **S2**.

Through the illustrations above, we show that bi-directional location-independence can be achieved in the sense that neither content clients nor servers need to know a priori the explicit location (i.e. the IP address of the actual content server and the consumer) of each other for content consuming. In particular, content clients may include *implicit* content scoping / filtering information when requesting content. The content resolution system then automatically identifies the server in the desired “area” that hosts the content. On the content provider side, when a content is published, scoping can be applied such that the content can only be accessed by content clients in the designated area in the Internet. As we will show in the following section, thanks to the multicast-oriented content delivery mechanism, the content server is not aware of the explicit location of the active content clients of that content.

5.2.7 Preparation for Content Delivery

As the name of the approach implied, the content delivery process is tightly coupled with the content resolution process. The content delivery paths are enforced in a receiver-driven multicast approach. During the content resolution procedure, requests sent by CCs are processed at the mediation plane and are forwarded from one domain to another based on the resolution procedure described above and the CCs *scoping* and *filtering* preferences. While traversing these domains, soft states are installed along the resolution path. Specifically, this is done by configuring the local CAFEs that will be involved in the delivery of the content back from the server to the CC. It is worth mentioning that, in case of splitting of "resolution path” (e.g. at multi-homed domains), soft states will be only maintained for one single path for the actual content delivery. States on the other paths will be torn down through either soft- or hard- state signaling messages. This feature will be further investigated during the rest of the project.

When the content is discovered, the delivery path is simply the reverse of the resolution path. The content is delivered following the soft states already installed during the resolution stage. This implies that there is a small requirement on maintaining soft states in the COMET system.

The details of the content delivery procedure for this approach are described in D4.1 [49].

5.2.8 Interim Study

5.2.8.1 Scalability

The fundamental domain-level hop-by-hop content resolution strategy presented follows a similar style to that proposed in [12]. However, through the new *scoping* and *filtering* functions, our approach provides the necessary flexibility for both content providers and consumers to publish/request content at/from their desired area(s). The scalability of the system, thus, is dependent on the amount of content and the popularity of the content recorded within each CRME, with the most "vulnerable" CRMEs being those that maintain the highest number of popular content entries. This is in contrast with intuition that the most strained CRMEs will be the tier-1 ones, since content publications and requests may often not reach the tier-1 level based on our approach. Again, we take BBC iPlayer as an example where both the content publication and consumption requests are restricted only to IP prefixes from within the UK. In addition to that, local domain policies may also override the default publication route (see **S2** in Figure 25).

Business incentives also present a natural load distribution mechanism for our system. We foresee ISPs charging higher publication tariffs for popular content published at higher tier domains (with tier-1 domains being the most expensive) which are able to be potentially accessed by a higher number of consumers in the Internet. This mechanism forms a business tussle from the content providers' point of view when provision of wider access is coupled with higher monetary cost. Instead, a content provider may strategically replicate content to multiple lower-tier regional ISPs (by applying scoping functions there) in which they believe their content will be locally popular.

Finally, our system also allows aggregation in two distinct ways. First, as illustrated in Figure 25 for **S4** and **S5**, the record for copies of the same content can be merged during the publication process among CMEs. Second, a block of sequential content IDs should be allocated to *inter-related* content so that they can be published in one single process. This rule exploits the fact that a specific content provider usually offers content with some relationship with each other (e.g. all episodes of a television series, or all football matches in a World Cup event). This allows for coarser granularity in the publication process whereby the content provider can send only one Publish message to publish all the related content. The local CRME still assigns a unique content ID for each content, but the IDs are sequentially connected. The onwards publication process will only involve the entire block of the IDs rather than the individual content records. This alleviates higher tier ISPs from the need to know each content hosted within and under their domain. Now, instead of matching explicitly the content ID in the Consume message, the CRME simply checks if the content ID is within a specific range of published content. The final location of the specifically requested content is actually handled at the *last-hop* domain where inter-related content entries are locally de-aggregated.

5.2.8.2 Security

There are different facets in terms of security considerations. First, the communication amongst the CRMEs that form the backbone of the framework must be secured. We assume that each CRME is capable of authenticating their neighbouring counterparts following the exchange of public keys during the establishment of inter-domain relationship (ISP-ISP SLA). Standard third-party security infrastructure may also be used (e.g. using digital certificate) to further ensure the identity of CRMEs.

The current Internet uses IP addresses as both identifier and locator. In all communications, host location is always exposed and thus opens to potential malicious attacks (e.g. Denial-of-Service attack). Location independence is deemed necessary to improve security. As detailed, our architecture achieves bi-directional location independence by pointing to the "implicit" next hop information (e.g. IP prefix) towards the target rather than using explicit addresses associated with content servers.

Furthermore, new proposals on content-centric networks focus on securing the content via self-certified content names with cryptographic features [9][11][39][12]. These measures can be seen as complementary to our architecture and can be accommodated without conflict.

5.2.8.3 Network-awareness

It has been recently proposed that network information to be passed up to the application layer for enhancing both service and network performance, typically through optimized content source/peer selections e.g. the IETF ALTO framework [41][42] and the P4P paradigm [43]. In this approach, network-awareness during content resolution is achieved via CRME-CAFE interactions within each local domain. RAEs provide PMFs with network information regarding inter- and intra-domain routing for optimised content delivery. The SNMF in each CRME also ensures timely monitoring of the network conditions and the detection of anomalies causing drastic degradation of network performance, such as network failures or server congestions. In addition, content servers may also benefit from this facility by providing server load information to CRMEs. Such information, thus, enables optimized anycast-based delivery operations when multiple copies of the same content are found during the resolution process. For illustration, consider Figure 26 with a wildcard consumption request for content **X4** received at domain **B.B**. With network-awareness, **B.B** can then forward the request down to the customer domain that is capable of providing better performance rather than randomly choosing one. Alternatively, a content consumer may request a content item with specific end-to-end QoS requirements. In such a case, only those domains advertising the right capabilities (e.g. edge-to-edge bandwidth support) will be forwarded the requests during the content resolution phase. As a result, upon a successful resolution procedure, a feasible end-to-end content delivery path has also been identified, thanks to the coupled content resolution and path setup operations.

6 Awareness in the Content Mediation Plane

In the current Internet, copies of the same file can be found at multiple servers for many reasons. For instance, some content providers maintain the same file at different servers to provide fast delivery to each server's local area. Another example is peer-to-peer applications where multiple end users both receive and transmit, acting as servers, the same file to other users. It is not practical however for users to specify which server they prefer to access a file from since they do not have any relevant information to help them do so.

Furthermore, content is currently routed using statically configured paths based on how the network has been engineered without taking into consideration the path characteristics or the runtime state of the network.

As described in D2.2 [48], the CMF is in charge of locating the most appropriate path and server for delivering the content. For this to be done, the CMF uses the awareness provided by the different functional blocks in the COMET system.

Content awareness relates to the knowledge in the CMF function about content characteristics such as the QoS requirements or the available content sources. This awareness is usually provided by the CRF function, but some characteristics could be implicit in the Content Identifier used in the content request from the Content Client (e.g. QoS requirements for the content travel in the content request in the coupled approach).

Content server awareness, or just **server awareness**, refers to the knowledge provided to the CMF about the server conditions (CPU usage, memory usage, or even bandwidth usage). This awareness is provided by the SNMF block.

Routing awareness deals with routing information visible at the inter-domain level and shared between domains. In the COMET system, routing awareness is focused on finding the inter-domain paths and the inter-domain network topology. For that purpose we defined in D2.2 [48] the PMF, which collects information about available paths/routes and provides it to the CMF. Information about individual paths includes a set of path properties per each destination network, such as:

- COMET CoS supported along the path
- Path length expressed in terms of number of domains
- The list of domains on the path
- A vector of QoS parameters characterizing the path (e.g. maximum packet loss probability, maximum delay)

The values of path properties are calculated based on the parameters shared by a particular domain. Therefore, path properties are limited only to global information, which operators intend to share. More details about the Routing Awareness process are presented in D4.1 [49].

Network awareness refers to the knowledge about network conditions that are made available to the CMF. This information relates to short/medium-term information about network condition depending on the traffic demands. It must be noticed that the path conditions provided by the routing awareness process are long-term information, "governed" by the frequency of changes in peering SLAs and the intra-domain QoS parameters established by the network planning.

Next sections present how server and network awareness can be achieved in the COMET system so that content can be delivered to users from the optimal server and via the optimal path in the network, thus improving their quality of experience.

6.1 Content Server awareness

In this section, we describe how COMET achieves content server awareness in the two approaches of the COMET system, and specifically in the two content resolution approaches in chapter 5. In the first approach, the content server conditions are disseminated by exploiting the fact that each content consumption operation requires the access of its content record in CREs whereby a mechanism is designed to report up-to-date server conditions to the CREs. On the other hand, under the coupled resolution and delivery approach, the content server-awareness is achieved following the similar paradigm in the publication and resolution processes whereby the information is diffused in a hop-by-hop manner.

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

6.1.1 Achieving Content Server-awareness in the coupled approach

Content server resource awareness can be enabled by disseminating the relevant information in the COMET system. In COMET, all registered content servers that wish to publish their contents should report their server load information to the CMP, i.e. to their local CRMEs. A content server may actively report its load condition, by appending the information to other messages such as still-alive messages or content publishing messages. The information will be stored within the CMP and will not be accessible by individual content consumers. Dissemination of this information across all CRMEs within the CMP is not a trivial task given scalability issues.

Combined with the state-based content delivery process described in deliverable D4.1 [49], we propose two solutions for the dissemination of server load information within the CMP. In the first solution, for each published Content-ID, a stub CRME periodically reports the best load condition among all local servers (that host the content) to its counterpart in the provider domain. Each intermediate CRME compares the received server load conditions (both local and remote ones) and reports the best one, coupled with its own IP prefix, to its provider domain until reaching the Tier-1 domain. This operation is on per Content-ID basis, and only server condition values are reported, but not necessarily the actual server ID since this information has already been disseminated in the initial publication operation.

For each Content-ID, the CRME in each domain is aware of the load conditions of (1) its locally attached servers and (2) the best server condition reported from each of its direct customer domains. During the content resolution process for a Content-ID, in case multiple content server candidates are available, the CRME may determine in which (downhill) direction to forward the content request. This is an anycast-like server selection according to load conditions. The example in Figure 29 demonstrates how this solution works. Servers **S1** and **S2** both have copies of content **X** and both register with their CRMEs (indicated by circles in the figure). The CRME in domain with prefix 1.2.1/24 periodically polls the local content server **S1** for its load condition, and reports to the CRME in domain with prefix 1.2/16 which is its provider domain, shown as blue dot-dash arrow. This process is also carried out by the CRME in domain with prefix 1.2.2/24 regarding server **S2**.

The CRME in domain with prefix 1.2/16 selects the server with the best condition (e.g. assuming **S1** in this case) and reports to the CRME in domain with prefix 1/8 which is its provider domain, indicated by a blue dot-dash arrow. The report is associated with its own prefix 1.2/16, and not with the actual ID of **S1**. In this case, the domain with prefix 1/8 is a Tier 1 domain. It has a local content server which also has a copy of content **X** and registers to its CRME. Its CRME, as with CRMEs in other domains, periodically polls the local content server **S0** for its load condition.

Upon the reception of a content request from domain with prefix 1.1/16 for content **X**, the CRME in the domain with prefix 1/8 compares the server load between **S0** and **S1**. If **S1** is better than **S0**, then the CRME will forward the request to its counterpart in the domain with prefix 1.2/16, shown in Figure 29 with a dotted red arrow. The CRME in the domain with prefix 1.2/16 is aware that **S1** is attached to its customer domain with prefix 1.2.1/24 and then forwards the request to the corresponding CRME (dotted red arrow).

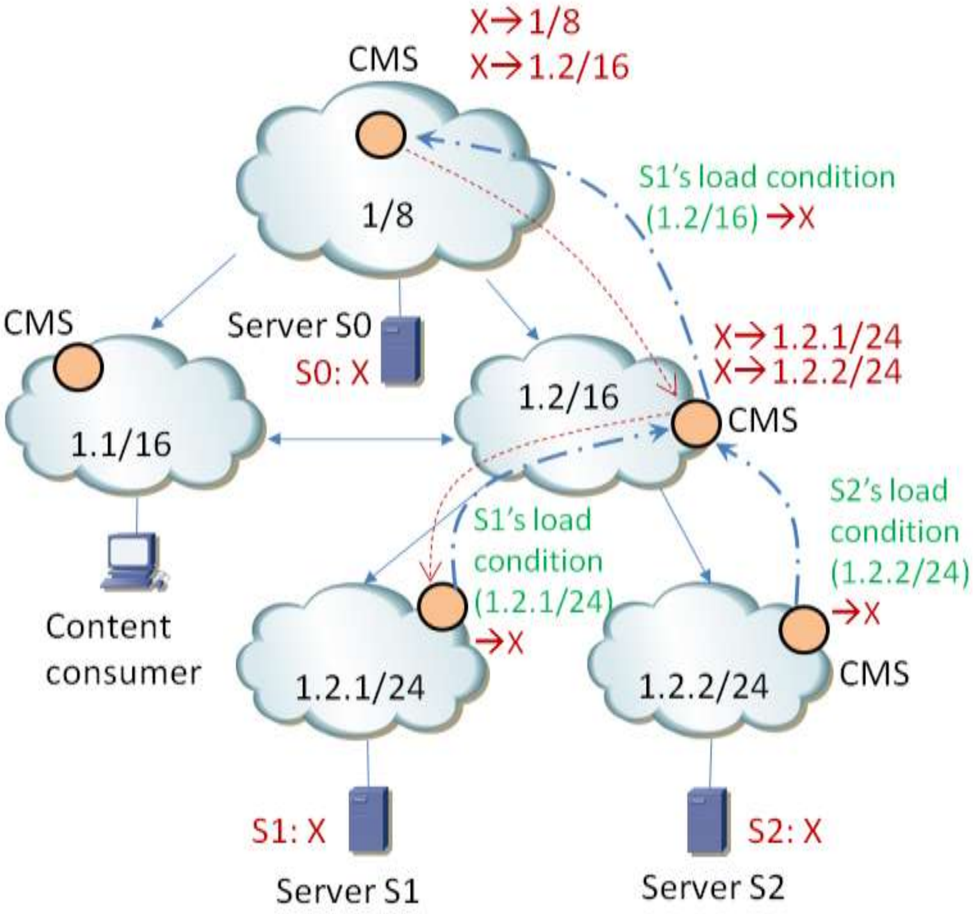


Figure 29: Example of solution 1 for content server load information dissemination

The second solution is to allow the CRME to report all the attached servers' load information rather than only the best one to the CRME in its provider domain. This means that each stub CRME periodically reports the load conditions (coupled with server IDs) of **all** the local servers to its counterpart in the provider domain. Each intermediate CMS reports **all** of the received conditions together with its local server conditions to its provider domain until reaching the Tier-1 domain. This type of server load dissemination operations is content ID independent. With this approach, the CRME in each domain knows the load conditions of (1) its locally attached servers and (2) **all** the servers attached to its direct and remote customer domains. To enable this operation, the publication of each content ID should include the ID of the server that actually holds the content. Figure 30 shows an example of how this solution works.

In Figure 30, the CRME in the domain with prefix 1.2.1/24 periodically polls the local content server **S1** for its load condition, and reports this information along with the server ID (**S1**) to the CRME in the domain with prefix 1.2/16. This process is also carried out by the CRME in the domain with prefix 1.2.2/24 regarding server **S2**. The CRME in the domain with prefix 1.2/16 then reports the load conditions and IDs of both **S1** and **S2** to the CRME in its provider domain with prefix 1/8, which is a Tier1 domain in this case. This process is depicted with a blue dash arrow in Figure 30. The CRME in the Tier1 domain periodically polls its local content server **S0** for its load condition and stores this information in its database together with the server load reported by the CRME in the domain with prefix 1.2/16. Therefore, it maintains the load information of all three servers' which have copies of content **X**.

Upon a content request from the domain with prefix 1.1/16 for content **X**, shown as dotted red arrow in Figure 30, the CRME in the domain with prefix 1/8, being aware of the location of content **X**, it compares the server load between **S0**, **S1** and **S2**. If **S1** is the best option, then the CRME in the Tier1 domain will forward the request to its counterpart in the domain with prefix 1.2/16, which is aware that **S1** is attached to the domain with prefix 1.2.1/24 and forwards the request accordingly, as shown with the dotted red arrow.

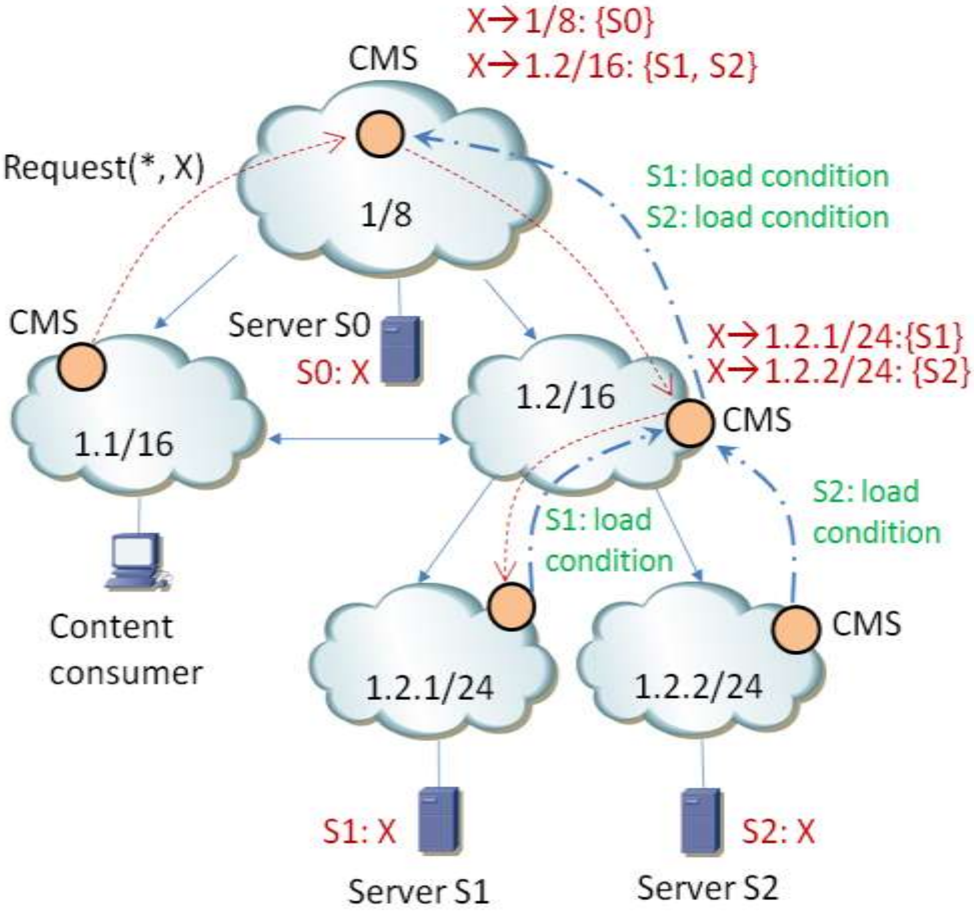


Figure 30: Example of solution 2 for content server load information dissemination

6.2 Network-awareness

In this section we describe how the COMET system, and specifically the entities hosting the CMF, achieves network awareness in the decoupled and coupled approach. Network awareness allows the COMET system to improve the efficiency of content delivery. In particular, information about network state is used in the decision process for selecting the appropriate server and the content delivery path. In addition, the information about the network state could be used in the path configuration process e.g. rejecting to configure a path if there is not available BW in the peering link to deal with the content efficiently.

6.2.1 Scope of network state information

We consider three different kinds of network information that can be passed to the CMF: local network condition information, global network condition information and content consumption information.

6.2.1.1 Local Network awareness

Local network condition information relates to network operators' confidential data gathered during the OAM processes (Operations, Administration and Maintenance). It theoretically includes multitude of information about traffic and network topology, e.g., traffic volumes and demands, failure indications, QoS monitoring and security incidents.

This data, either gathered or locally known within a domain, is not shared with other parties. Although the above information is confidential to individual domains, the COMET system may use it locally during the decision process and during the path configuration. Such local measurements are provided by the SNMF.

In COMET we will restrict our studies to the ingress and egress load of inter-domain peering links per CoS. This information, together with the information from the PMF about maximum BW per CoS guaranteed by the peering SLA, could be used to obtain the available bandwidth per CoS. Thus, it would be possible to reject content consumption requests based on unavailable bandwidth, or even decide to use a different CoS in the network for the content delivery, knowing that the performance could be lower than the necessary.

In addition, we will investigate the use of other local metrics such as the load on intra-domain network links, that could be used e.g. to reject content consumption requests based on the available users' access bandwidth.

6.2.1.2 Global Network awareness

Global network awareness deals with information visible at the inter-domain level and shared between domains. It relates to short-term information about network condition, thus differing from the global QoS path conditions provided by the routing awareness process, which are long-term.

The coupled approach foresees the possibility of performing local measurements of metrics, e.g. delays, between a local domain and other domains, and sharing these measurements with other domains following a similar dissemination as the one in the content publication.

6.2.1.3 Content consumption awareness

Finally, network awareness could be indirectly obtained from the content consumption statistics. These statistics may cover:

1. The path status, e.g. the aggregate rate of running flows, the ratio of served content requests, etc.
2. The traffic demand, which could be estimated based on content requests
3. Negative results of past consumption requests, e.g., rejection of path configuration should be recorded in the system

It should be noted that the above information is available and visible in the COMET system. As such, interaction with other systems is not required.

It must be noted that this information is not provided by the SNMF, but is available in the CMF.

6.2.2 Achieving Network-awareness in the Decoupled Approach

In the decoupled approach, we foresee the utilization of the local network information and the content consumption statistics as parameters to be used by the multi-criteria decision algorithms in the CME. For more information on this process, see 5.1.5.3.

Regarding the local network information, we will exploit the information about the load on ingress and egress peering links per CoS. Agents in the network routers will notify periodically the load to the SNME or to a NMS which will forward it to the SNME. In order to measure network load, sampling can be activated in the network routers, although this sampling must be high enough to capture traffic amounts relevant for the maximum BW per CoS

The notification from the network routers will be done e.g. via SNMP traps. The SNME will gather this information and will inform periodically to the CME following a push model. The notification intervals should be low (in the order of tens of seconds or minutes).

Regarding the content consumption statistics, they are directly available in the CME. However, it is possible that several CMEs are present in one domain, being the information in one CME partial with respect to the whole domain. It has not foreseen any mechanism in the decoupled approach to share this information between CMEs, so we will restrict to the usage of the local information in the CME.

6.2.3 Achieving Network-awareness in the Coupled Approach

To enable network-awareness in the COMET system, the relevant information needs be disseminated within the CMP. We firstly assume that long-term static paths have been pre-provisioned across domains in an offline manner. According to the cascaded QoS model through provider-level Service Level Specifications (SLS), each domain knows the **provisioned** end-to-end QoS capability towards individual destination domains under **normal** traffic conditions, which can be used as the static path condition. Due to traffic dynamics however, the actual path condition may vary and may sometimes violate the agreed values in the provider SLS.

Server selection operations should be combined with path conditions selection. Therefore, we have the server selection criteria from a Tier1 domain point of view: optimized server load and optimized downhill path condition.

We can have two options:

1. A dynamic and greedy approach which always tries to achieve the optimal overall “cost”, possibly with weighting between the two objectives (server load / path condition).
2. An approach that always takes the server load as the main factor and selects the least loaded server, as long as there is no path congestion/failure, i.e. the path condition being a “constraint” rather than an “objective”.

To obtain the path condition, there are three possible strategies. The first one is to use end-to-end measurements, which involves periodical monitoring of the end-to-end inter-domain paths, a process carried out by either end users or CRMEs. With this approach, end users will actively measure the quality of the path that is currently carrying their content flows. It should be noted that end users are not expected to be involved in the path condition acquirement procedure due to the high overhead. Here, end users are not trying to determine path conditions of all the possible paths but only of the ones they are using instead. This means that end users have been allocated servers and paths before they perform measurements on the paths they are using. The implication here is that the resource selection function may not choose the optimal path but leave the task for end users. If the resource selection function is to choose the optimal paths for the end users, it should carry out measurements for all possible paths. The “path” here is not end-to-end path from the end user to the content server where a content candidate is hosted but one between the end user’s domain and the content server’s domain. These measurements will also cause overhead although not as much as in the case where the end users carry out the measurements since the mediation paths can be stored in CRMEs to serve multiple end users.

The second strategy is to disseminate the path condition information in a hop-by-hop manner in the CMP. Each CRME can periodically measure the actual domain-to-domain QoS performance between each neighbour domain and itself, and report to its counterpart in neighbouring domains along the original provider-level SLS chain.

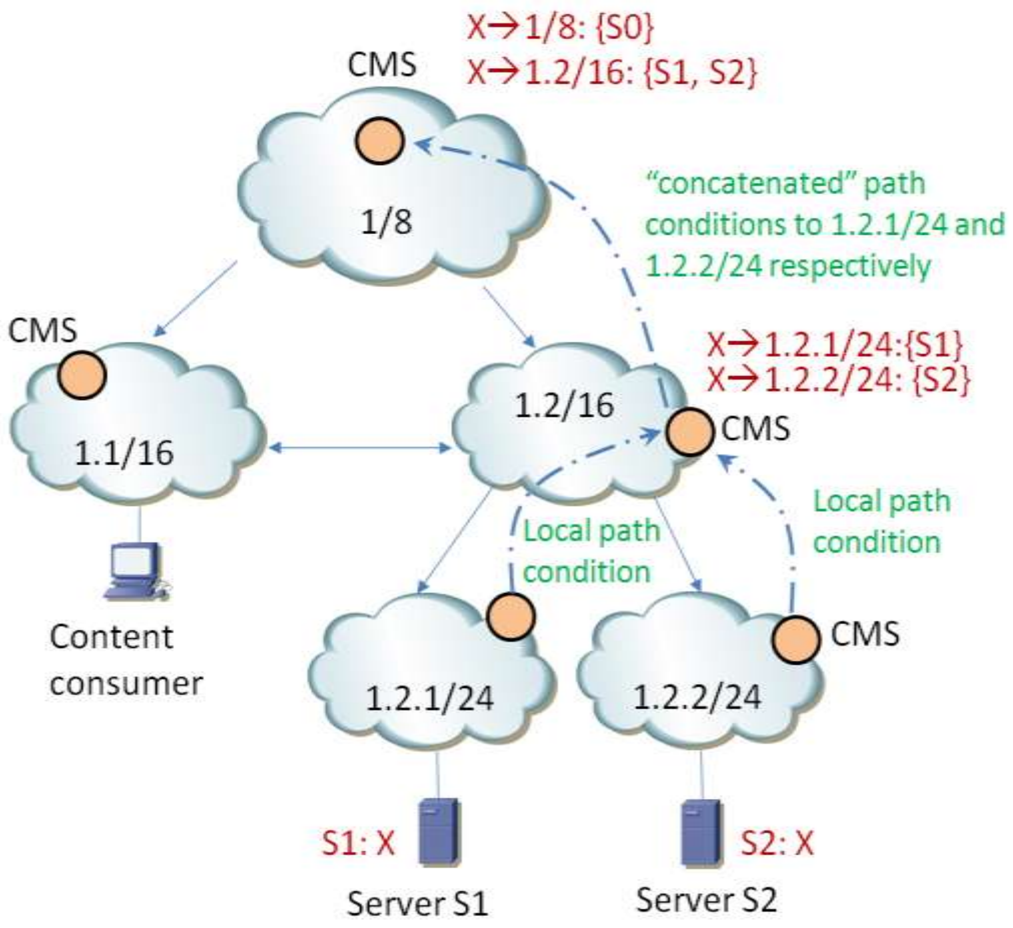


Figure 31: Example of solution 1 for path condition information dissemination

In the case where QoS information is disseminated along the original provider-level SLS chain, the procedure is similar to the one regarding server load dissemination described in the previous section. As shown in Figure 31, the CRME of domain 1.2.1/24 periodically measures the actual edge-to-edge QoS (e.g. delay) between its domain and its neighbour domain with prefix 1.2/16 and reports to the CRME of domain 1.2/16. So does the CRME in the domain with prefix 1.2.2/24. Both are shown as blue dash-dot arrows.

The CRME of domain 1.2/16 periodically measures it's the edge-to-edge performance between itself and its provider domain and concatenates it with what has been reported by the two customer domains (1.2.1/24 and 1.2.2/24) respectively. It then reports the result to its provider domain with prefix 1/8.

The CRME of domain 1/8 periodically measures its own edge-to-edge performance and concatenates it with what has been reported by the CRME of domain 1.2/16. The information should be stored by the CRME of domain 1/8.

In this case, Tier 1 CRMEs, such as the one of domain 1/8 in Figure 31, can be aware of the end-to-end performance from its own domain towards all stub domains, i.e. the ones with prefix 1.2/16, 1.2.1/24 and 1.2.2/24 in Figure 31. Therefore, the resource selection function will always rely on Tier 1 CRMEs to provide path condition information and the chosen optimized path will be via Tier 1 domain initially. For privacy concerns, each the CRME in each domain should only disseminate their network/QoS conditions to their **own providers**, but not peering or customer domains. Therefore, in case the CRME in a provider domain knows there are multiple content servers for the targeted content in its different customer domains, it is able to determine which one to use based on the collected (concatenated) path information.

7 COMET-ENVISION Interface

In this section, we outline the plan for the joint COMET-ENVISION collaboration. We identify the adaptations needed in the COMET's CMP to interface with ENVISION and provide the potential network metrics to be fed to ENVISION with the aim to augment its network monitoring functions.

While both projects focus on various aspects of digital data content in the Internet (content access, dissemination, delivery etc), the high-level approaches employed are different. As explained, in the COMET project, we try to solve the problem via an overlay at the network level resulting in a 2-plane approach aiming to mediate the delivery of Internet content via native COMET network entities. On the other hand, the ENVISION project [50] deals with the problem by developing techniques for the content delivery at the application layer and by fostering the collaboration between the applications and the underlying ISP networks to achieve the co-optimisation of the often misaligned application and network performance objectives.

To enable collaboration between the two projects, we envisage the creation of a COMET-ENVISION interface that enables communication between the relevant entities from both sides. Via this interface, we foresee that COMET can supply some network performance information monitored within the COMET system to the ENVISION system, thus allowing the related ENVISION functions to perform their optimization in a more timely and informed manner.

Specifically, the COMET-ENVISION interface is expected to be located between the SNMF of COMET, and possibly the PMF, and the Network Optimization Function (NOF) of ENVISION.

The candidate information that the SNMF (and PMF) can provide to NOF through this interface includes:

- Long-term inter-domain topological information (e.g. IP addresses of the routers in peering links, peering AS numbers)
- Routing information per destination network, specifically a set of paths per destination network prefix, each path consisting of the following information:
  - COMET CoS supported along the path
  - Path length expressed in terms of number of domains
  - The list of domains on the path
  - A vector of QoS parameters characterizing the path (maximum packet loss probability, maximum delay)
- Load of inter-domain peering links per CoS and the maximum bandwidth per CoS (according to the agreed peering SLAs)
- Long-term intra-domain edge-to-edge QoS parameters (according to the network planning and provisioning). These parameters are used in the routing awareness process in COMET to exchange network reachability information with other domains.

Due to the sensitivity of some of this information, we do not presume that ISPs by default agree to share them with other operators. However, note that both the SNMF and the NOF are owned by the same ISP. The NOF would use this information directly to improve the network optimization itself and/or to modify the information exposed to the application, without necessarily revealing the raw information received by SNMF.

Other specifications of the interface (e.g. the messaging format, the interface technology (push / pull)) are dependent on the implementation of both projects and thus will be defined later.

8 Summary and Conclusions

The interim specifications of the two content resolution approaches that are based under the high-level COMET architecture are provided in this deliverable. They were both designed and developed after careful study of current practical systems in use in real world along with newer research proposals that attempt to advance the content access and dissemination of the future Internet.

The starting point of the design of the first approach (also known as the decoupled approach in this deliverable) is the current DNS system with analogous input from the Handle system. Combining the two systems with considerable inputs from the literature, we developed a resolution framework that revolves around content records. We derive two ways to publish content and resolve content name within this decoupled approach; one based on the DNS system while the other from the Handle system. The core resolution paradigm resembles the table look-up based methodology used in the DNS servers with hierarchical structure and record caching facilities. We introduced several new entities (e.g., CRE, CME, SME etc) that interlink and communicate with each other to realize the functions identified in the high-level COMET architecture (cf. D2.2 [48]). The underlying rationale of this line of approach is that without tearing down the very foundation of the current resolution infrastructure and reusing at least the operation fabric of it, the new COMET system can be more readily deployable to the real world; cutting down the deployment cycle time and minimizing the disruption to the current Internet. The naming of content in this approach follows the example of the URL and Handle system we have today. We also detail the preparation of the content delivery in this deliverable which serves as the bridging point to the corresponding deliverable for WP4 (i.e., D4.1 [49]).

The second approach (also known as the coupled approach in this deliverable) takes on a very different design path. Instead of building on top of existing deployed system, it follows a clean slate approach and thus, its design is not constrained by what infrastructure and equipment have been set up. The argument is that the fundamental design principles of the current Internet is already outdated and incompatible with the primary use of the Internet today and the future. Attempting to "patch" the whole system may be simpler and cost-efficient but the long-term benefits may also be limited. The coupled approach exploits the hierarchical nature of the Internet topology and the business relationships amongst the different domains to create a hop-by-hop resolution solution that at the same time prepares the content delivery path (and hence the name of the coupled approach). The contents are identified via aggregatable Content-IDs that can be sequentially ordered. To realize this coupling, the content publication and resolution operations are designed to follow specific dissemination rules across the Internet with added flexibilities for both content providers and consumers via the scoping and filtering capabilities embedded within the solution. We introduced the CRME as the main entity in this approach in achieving this. The basic idea was to install multicast-like states at the edges of the domains that the resolution process involved and the delivery of the content simply follows these "breadcrumbs" back to the content consumer. We also investigate a resolution optimization techniques where we find better delivery path through peering domains rather than completely dependent on provider routes. Similar to the first approach, we also detail the preparation of the content delivery in this deliverable that constructs the domain-level delivery path (i.e., the reverse of the resolution path). The actual delivery mechanisms are detailed in D4.1 [49].

A key aspect of the COMET system is its awareness with respect to the performance of both the network and hosting servers. We detailed how these two types of awareness are achieved for the two approaches. Server awareness mainly spanned from the fact that nowadays multiple copies of the same content can be found in the Internet and ideally, content consumer would always like to access the content via the best server (e.g., the nearest server, the lowest loaded server etc) while the requirements (e.g., user QoS/QoE requirements) drive the need to know the network capabilities / conditions. The decoupled approach seeks to gain this awareness through update on the content records from the server and network respectively. The coupled approach, on the other hand, follows the same idea of the content publication and resolution rules in disseminating the server and network condition information and thus, forming a uniform methodology in diffusing content related information.

Finally, this deliverable includes the initial work on the COMET-ENVISION interface. Potential performance metrics to be communicated from COMET to ENVISION have also been identified and discussed. The preliminary specification (e.g., the interfacing functional blocks between the two projects) will be described in more details in the next deliverable.

9 References

[1] M. Gritter and D. R. Cheriton, "TRIAD: A New Next-Generation Internet Architecture," http://www-dsg.stanford.edu/triad, July 2000.

[2] P. Francis and R. Gummadi, "IPNL: A NAT-extended Internet Architecture," in Proc. of ACM SIGCOMM '01, pp. 69-80, San Diego, CA, USA, Aug. 2001.

[3] A. Jonsson, M. Folke and B. Ahlgren, "The Split Naming/Forwarding Network Architecture," in Proc. Swedish National Computer Networking Workshop (SNCNW), Sept. 2003.

[4] I. Stoica, D. Adkins, S. Zhuang, S. Shenker and S. Surana, "Internet Indirection Infrastructure," in Proc. of ACM SIGCOMM '02, pp. 73-86, Pittsburgh, PA, USA, Aug. 2002.

[5] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan, "Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications," ACM SIGCOMM 2001, San Diego, CA, August 2001, pp. 149-160.

[6] P. Eugster, P. A. Felber, R. Guerraoui and A.-M. Kermarrec, "The Many Faces of Publish/Subscribe," ACM Computing Surveys, Vol. 35, no. 2, pp. 114-131, June 2003.

[7] A. Zahemszky, A. Csaszar, P. Nikander and C. E. Rothenberg, "Exploring the Pub/Sub Routing & Forwarding Space," in International Workshop on the Network of the Future, 2009.

[8] D. Clark, R. Braden, A. Falk and V. Pingali, "FARA: Reorganizing the Addressing Architecture," in Proc. of ACM SIGCOMM FDNA Workshop, Aug. 2003.

[9] R. Moskowitz and P. Nikander, "Host Identity Protocol Architecture," RFC 4423, IETF, May 2006.

[10] H. Balakrishnan, K. Lakshminarayanan, S. Ratnasamy, S. Shenker, I. Stoica and M. Walfish, "A Layered Naming Architecture for the Internet," in Proc. of ACM SIGCOMM '04, pp. 343-352, Portland, OR, USA, Aug. 2004.

[11] M. Walfish, J. Stribling, M. Hrohn, H. Balakrishnan, R. Morris, and S. Shenker, "Middleboxes No Longer Considered Harmful," in Proc. of OSDI '04, pp. 215-230, San Francisco, CA, USA, Dec. 2004.

[12] T. Koponen, M. Chawla, B-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker and I. Stoica, "A Data-oriented (and Beyond) Network Architecture," in Proc. ACM SIGCOMM '07, Kyoto, Japan, Aug. 2007.

[13] M. D'Ambrosio et. al., "D-6.2 Second NetInf architecture description," 4WARD project deliverable. http://www.4ward-project.eu/index.php?s=Deliverables

[14] M. Caesar, T. Condie, J. Kannan, K. Lakshminarayanan, S. Shenker and I. Stoica, "Routing on Flat Labels," in Proc. of ACM SIGCOMM '06, pp. 363-374, Pisa, Italy, Sept. 2006.

[15] P. Mockapetris, "Domain Names – Concepts and Facilities". Internet Engineering Task Force (IETF) Request for Comments (RFC), RFC 1034, November 1987. Available: http://www.ietf.org/rfc/rfc1034.txt

[16] P. Mockapetris, "Domain Names – Implementation and Specification". Internet Engineering Task Force (IETF) Request for Comments (RFC), RFC 1035, November 1987. Available: http://www.ietf.org/rfc/rfc1035.txt

[17] T. Berners-Lee, "Uniform Resource Locators (URL)". Internet Engineering Task Force (IETF) Request for Comments (RFC), RFC 1738, December 1994. Available: http://www.ietf.org/rfc/rfc1738.txt

[18] P. Vixie, "Dynamic Updates in the Domain Name System (DNS UPDATE)". Internet Engineering Task Force (IETF) Request for Comments (RFC), RFC 2136, April 1997. Available: http://www.ietf.org/rfc/rfc2136.txt

[19] Sam Sun, Larry Lannom, Brian Boesch, "Handle System Overview". Internet Engineering Task Force (IETF) Request for Comments (RFC), RFC 3650, November 2003. Available: http://www.ietf.org/rfc/rfc3650.txt.

[20] Sam Sun, Sean Reilly, Larry Lannom, "Handle System Namespace and Service Definition". Internet Engineering Task Force (IETF) Request for Comments (RFC), RFC 3651, November 2003. Available: http://www.ietf.org/rfc/rfc3651.txt

[21] Domain Name System (DNS) Parameters — http://www.iana.org/assignments/dns-parameters.

[22] N. Freed, N. Borenstein, "MIME Media Types". Internet Engineering Task Force (IETF) Request for Comments (RFC), RFC 2046, November 1996. Available: http://www.ietf.org/rfc/rfc2046.txt.

[23] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1". Internet Engineering Task Force (IETF) Request for Comments (RFC), RFC 2616, June 1999. Available: http://www.ietf.org/rfc/rfc2616.txt

[24] J. Peterson, R. Sparks, M. Handley, E. Schooler, "SIP: Session Initiation Protocol". Internet Engineering Task Force (IETF) Request for Comments (RFC), RFC 3261. June 2002. Available: http://www.ietf.org/rfc/rfc3261.txt.

[25] M. Handley, V. Jacobson, C. Perkins, "SDP: Session Description Protocol", Internet Engineering Task Force (IETF) Request for Comments (RFC), RFC 4566. July 2006. Available: http://www.ietf.org/rfc/rfc4566.txt

[26] GSMA IR.34, Inter-Service Provider IP Backbone Guidelines, GSM Association ver. 4.9, March 2010, avaialbe at http://www.gsmworld.com/documents/IR3449.pdf

[27] A. Messac, E. Melachrinoudis, and C.P. Sukam, "Aggregate Objective Functions and Pareto Frontiers: Required Relationships and Practical Implications", Optimization and Engineering Journal, Kluwer Publishers, vol. 1, issue 2, June 2000, pp. 171-188.

[28] K.S. Pradyumn, "On gradient based local search methods in unconstrained evolutionary multi-objective optimization", in proc. of the 4th international conference on Evolutionary multi-criterion optimization, March 05-08, 2007, Matsushima, Japan.

[29] M. Ehrgott, "Multicriteria Optimization", Springer-Verlag New York, Inc. 2005. ISBN: 3540213988

[30] A.P. Wierzbicki, M. Makowski, J. Wessels, "Model-based decision support methodology with environmental applications", Kluwer Academic Publishers. Dordrecht, NL. 2000. ISBN: 0-7923-6327-2

[31] Y. Sawaragi; H. Nakayama and T. Tanino, "Theory of Multiobjective Optimization", Mathematics in Science and Engineering, vol. 176 Academic Press Inc., 1985 ISBN 0126203709.

[32] A. Wierzbicki, "The use of reference objectives in multiobjective optimization", Lecture Notes in Economics and Mathematical Systems, vol. 177. Springer-Verlag, pp. 468–486

[33] T. Berners-Lee, R. Fielding and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", RFC 3986, January 2005

[34] Mealling, M. and R. Denenberg, "Report from the Joint W3C/IETF URI Planning Interest Group: Uniform Resource Identifiers (URIs), URLs, and Uniform Resource Names (URNs): Clarifications and Recommendations", RFC 3305, August 2002.

[35] Sollins, K. and L. Masinter, "Functional Requirements for Uniform Resource Names," RFC 1737, December 1994.

[36] Moats, R., "URN Syntax", RFC 2141, January 1997.

[37] Mealling, M. and R. Daniel, Jr., "URI Resolution Services Necessary for URN Resolution", RFC 2483, January 1999.

[38] Sollins, K., "Architectural Principles of Uniform Resource Name Resolution", RFC 2276, January 1998.

[39] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking Named Content," in CoNEXT '09. New York, NY, USA: ACM, 2009, pp. 1–12.

[40] P. Jokela, A. Zahemszky, C. E. Rothenberg, S. Arianfar and P. Nikander, "LIPSIN: Line Speed Publish/Subscribe Inter-networking", in Proc. ACM SIGCOMM '09, Barcelona, Spain, August 2009.

[41] J. Seedorf and E. Burger, "Application-Layer Traffic Optimization (ALTO) Problem Statement", IETF RFC 5693, October 2009.

[42] The IETF Application Layer Traffic Optimization (ALTO) Working Group, http://datatracker.ietf.org/wg/alto/charter/

[43] H. Xie, Y. Yang, A. Krishnamurthy, Y. Liu and A. Silberschatz, "P4P: Provider Portal for Applications", in Proc. ACM SIGCOMM '08, Seattle, WA, USA, August 2008.

[44] Jacobson, Van, et al., "VoCCN: Voice-over Content-Centric Networks", Palo Alto Research Center, Palo Alto, CA, USA: ACM, 2009.

[45] Alduan, Maria, et al., "Architecture for Future Media Internet", nextMedia and COAST EU projects.

[46] Tarkuma, Sasu et al., "The Publish/Subscribe Internet Routing Paradigm (PSIRP): Designing the Future Internet Architecture", IOS Press, 2009.

[47] COMET Deliverable, "D2.1: Business Models and System Requirements for the COMET System", The COMET Consortium, July 30th 2010.

[48] COMET Deliverable, "D2.2: High-Level Architecture of the COMET System", The COMET Consortium, November 30th 2010.

[49] COMET Deliverable, "D4.1: Interim Specification of Mechanisms, Protocols and Algorithms for Enhanced Network Platforms", The COMET Consortium, November 30th 2010.

[50] The EU FP7 ENVISION Project, http://www.envision-project.org

[51] The Internet Assigned Numbers Authority (IANA), http://www.iana.org/.

10 Abbreviations

AS	Autonomous System
BE	Best Effort
BGP	Border Gateway Protocol
BTBE	Better Than Best Effort
CAFE	Content-aware Forwarding Entity
CAFF	Content-aware Forwarding Function
CC	Content Client
CCN	Content-centric Network
CDN	Content Distribution Network
CID	Content Identifier
CFP	Content Forwarding Plane
CME	Content Mediation Entity
CMF	Content Mediation Function
CMP	Content Mediation Plane
CNRI	Corporation for National Research Initiatives
COMET	COntent Mediator architecture for content-aware nETworks
CoS	Class of Service
CN	Content Name
CNAME	Canonical Name
CNS	Content Name Server
CP	Content Provider
CPU	Central Processing Unit
CRE	Content Resolution Entity
CRF	Content Resolution Function
CRME	Content Resolution and Mediation Entity
CS	Content Server
DARPA	Defense Advanced Research Projects Agency
DDoS	Distributed Denial of Service
DHT	Distributed Hash Table
DNS	Domain Name System
DOA	Delegation-oriented architecture
DOI	Digital Object Identifier
DONA	Data-oriented Network Architecture
DoS	Denial-of-service
DPI	Deep Packet Inspection
FARA	Forwarding directive, Association and Rendezvous Architecture

FIB	Forwarding Information Base
FQDN	Fully Qualified Domain Name
GHR	Global Handle Registry
GS	Guaranteed Service
HIP	Host Identity Protocol
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
i3	Internet Indirection Infrastructure
IANA	Internet Assigned Numbers Authority
ID	Identifier
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPNL	IP-Next Layer
ISP	Internet Service Provider
LHS	Local Handle Service
MCDA	Multiple Criteria Decision Analysis
MIME	Multipurpose Internet Mail Extensions
MPEG	Moving Picture Experts Group
NAT	Network Address Translation
NLRI	Network Layer Reachability Information
OCLC	Online Computer Library Center
P2P	Peer-to-peer
PBR	Peak Bit Rate
PIT	Pending Interest Table
PMF	Path Management Function
PURL	Persistent Uniform Resource Locator
QoE	Quality of Experience
QoS	Quality of Service
RAE	Routing Awareness Entity
RH	Resolution Handler
RPR	Reverse Path Retrieval
SBR	Sustained Bit Rate
SDP	Session Description Protocol
SIP	Session Initiation Protocol
SME	Server Monitoring Entity
SNF	Split Naming / Forwarding
SNMF	Server and Network Monitoring Function
TCP	Transmission Control Protocol

TLD	Top Level Domains
TTL	Time To Live
UDP	User Datagram Protocol
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
URN	Uniform Resource Name
VoIP	Voice over Internet Protocol

11 Acknowledgements

This deliverable was made possible due to the large and open help of the WP3 team of the COMET project within this STREP, which includes besides the deliverable authors as indicated in the document control. Many thanks to all of them.