COntent Mediator architecture for content-aware nETworks

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

Deliverable D4.1
Interim Specification of Mechanisms, Protocols and Algorithms for Enhanced Network Platforms

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:

Dr. Ning Wang
University of Surrey
Guildford
Surrey GU2 7XH
UK

Document Control

Title: Interim Specification of Mechanisms, Protocols and Algorithms for Enhanced Network Platforms

Type: Public

Editor(s): Ning Wang

E-mail: n.wang@surrey.ac.uk

Author(s): Ning Wang, Lei Liang, Chaojiong Wang (UniS)

Wei Koong Chai, Ioannis Psaras, Marinos Charalambides (UCL)

Gerardo García de Blas, Francisco Javier Ramón Salguero (TID)

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

Sergios Soursos, George Petropoulos, Spiros Spirou (INTRACOM TELECOM)

Eleftheria Hadjioannou (PRIMETEL)

Doc ID: D4.1-1.0.docx

AMENDMENT HISTORY

Version	Date	Author	Description/Comments
V0.0	May 20th, 2010	Ning Wang	Initial ToC
V0.1	September 27th, 2010	Ning Wang	Revised ToC
V0.2	October 15th, 2010	Ning Wang	Received contribution before the first integration deadline
V0.3	November 15th, 2010	Ning Wang	New contributions from WUT/IntraCom/PrimeTel
V0.4	December 5th, 2010	Ning Wang	Updated contributions from WUT
V0.5	December 27th, 2010	Ning Wang	Updated contributions from UniS
V0.6	January 2nd, 2011	Ning Wang	Updated contributions from WUT and UniS
V0.7	January 28th, 2011	Ning Wang	Updated contributions from TID. Draft version to be reviewed
V0.8	February 2th, 2011	Gerardo García de Blas, Ning Wang	Modifications and comments from the internal reviewers
V0.9	February 9th, 2011	Ning Wang	Last version before final submission
V1.0	February 14th, 2011	Ning Wang	Final version ready for submission

Table of Contents

1 Executive Summary 5
2 Introduction 7
3 State-of-the-art of Networking Technologies 10
3.1 Multi-paths routing 10
3.1.1 Making extensions to BGP route attributes and selection process 10
3.1.2 Tag-based multi-path routing 10
3.1.3 Learning alternate routes from non-adjacent ASes through subscription 11
3.1.4 Pull-based path exploration 11
3.1.5 Structured Inter-AS Overlay 12
3.2 Multi-criteria routing 12
3.2.1 Methods based on single metric 13
3.2.2 Methods based on path cost function 13
3.3 Content caching techniques 14
3.3.1 What needs to be cached 14
3.3.2 Cooperative Caching 15
3.3.3 Cache Algorithms 15
3.3.4 Cache location 16
3.3.5 Caching in CCN 16
3.3.6 Related Research Areas: Macro- vs. Micro-Caching 17
3.4 Content Delivery Networks 18
3.4.1 Introduction 18
3.4.2 What is a content delivery network? 19
3.4.3 CDN state of the art 19
3.4.4 Stakeholders 19
3.4.5 Architecture 20
3.4.6 Most popular CDN providers 21
4 Quality of Service Engineering for content delivery 22
4.1 Overview 22
4.2 COMET Classes of Service 23
4.3 Content delivery in different network environments 25
4.3.1 Best effort network (Current Internet) 25
4.3.2 Multi-service network environments 25
5 Mechanisms and algorithms for routing awareness 28
5.1 Introduction 28
5.2 Specification of routing awareness process 29

5.2.1 Messages *30*
5.2.2 Basic operations *31*
5.2.3 Path ranking algorithm *31*
5.3 Consideration on deployment 32
6 Mechanisms and algorithms for basic delivery process 35
6.1 Overview 35
6.2 Stateless content delivery process 35
6.2.1 Introduction *35*
6.2.2 Path provisioning: collecting forwarding information for paths *36*
6.2.3 Path configuration: preparation for content delivery *38*
6.2.4 Content forwarding *42*
6.3 State-based content delivery process 43
6.3.1 Introduction *43*
6.3.2 Content Delivery Path Configuration During Content Resolution *43*
6.3.3 Content forwarding *45*
6.3.4 Inter-domain Content Delivery Path Optimisation *45*
7 Advanced Content Delivery Features 47
7.1 Content caching 47
7.1.1 Introductory Notes *47*
7.1.2 Motivation and Assumptions *47*
7.1.3 Initial thought on Caching in COMET *48*
7.2 Supporting point-to-multipoint content delivery 48
7.3 Edge controlled routing 50
8 Summary and Conclusions 53
9 References 54
10 Abbreviations 58
11 Acknowledgements 59
12 Annex – Study of CCN caching 60

1 Executive Summary

This deliverable presents the *interim* specifications on the protocols and algorithms associated with content forwarding plane (CFP) functionalities for end-to-end content delivery in multi-domain environments.

The document begins with a comprehensive literature review on relevant networking technologies associated with multimedia-based content delivery services. In particular, (inter-domain) routing optimization techniques have been specifically analyzed, including multi-path routing and multi-constraint routing, both of which play an important role in content delivery with Quality of Service (QoS) awareness. In addition, an overview on existing content distribution networks (CDNs) has also been presented in the literature review. Given that content caching technologies have been more and more used for enhanced content delivery services across the Internet, we have also performed a comprehensive literature review and analysis on the recently developed content caching techniques in this deliverable.

As far as the COMET approach is concerned for content delivery in the CFP, we classify the proposed mechanisms into two major categories. First of all, we specify "offline" operations used in preparing for the actual content delivery in sections 4 and 5. Section 4 basically presents the overall definition of COMET Class of Services associated with content delivery and their mappings onto existing multi-service aware network platforms such as DiffServ and MPLS. Top-level discussions on practical deployment on top of both Best-Effort (BE) based Internet and QoS-aware platforms have also been provided in this section. Section 5 deals with providing routing awareness in the path management function (PMF) to the actual content delivery operations. Specifically, some lightweight protocol enhancements have been specified on top of today's inter-domain routing protocols (BGP and its extensions in the literature). Fundamentally, this refers to the provisioning of dedicated paths across domains that will be available for content delivery with different QoS requirements.

Section 6 provides a comprehensive description on the "online" operations of the actual content delivery upon each content request. The key task is to identify and configure optimized content delivery paths from the targeted content server back to the content consumer. Broadly speaking, two distinct content delivery approaches in the CFP have been proposed in WP4. In the *stateless* content delivery approach, which is tied with the content record-based resolution approach specified in D3.1 [62], optimized end-to-end content delivery paths are identified after the possible content sources in a content record. Upon this, the Content Mediation Entities (CMEs) located in the source and consumer domains are responsible for exploring optimized paths for carrying the content traffic back towards consumer. The determination of the actual path (and content source) is based on the routing awareness as well as server and network conditions captured from network monitoring techniques. On the other hand, the *stateful* (a.k.a. *state-based*) content delivery approach is natively linked with the coupled content resolution technique described in D3.1, in the sense that the actual path configuration/enforcement is performed during the content resolution phase. As a result, the actual domain-level content delivery path will be in the reverse direction of the original content resolution paths hop by hop across intermediate domains.

In Section 7 we introduce a set of advanced content delivery techniques that are also being investigated in WP4. In order to enhance QoS (e.g. delay) to end content consumers as well as network resource optimization (e.g. content traffic reduction), intelligent in-network content caching techniques have been initially considered based on the content-centric network (CCN) model. Given that many of today's content delivery applications are point-to-multipoint, we also investigate how point-to-multipoint functionalities can be enabled for content delivery. Last but not least, we also introduce our preliminary design of the edge-controlled routing (ECR) where individual content consumers are allowed to actively make requests to the COMET system at the ISP side in order to switch content delivery paths in different contexts such as perceived QoE deterioration and change of user preferences.

All the interim techniques specified in this deliverable will be further enhanced in WP4 during the rest period of the corresponding tasks, and their final version will be presented in D4.2 in Month 21.

2 Introduction

This deliverable mainly addresses research issues that are related to network enhancements for supporting content-awareness in the future content-centric networks.

Towards this ultimate objective, in WP4 various networking mechanisms and algorithms will be designed and implemented, following either evolutionary or revolutionary/radical approaches. On one hand, evolutionary approaches aim at fast/incremental deployment of enhanced networking techniques for efficient Internet-wide content delivery without fundamentally changing the underlying legacy systems. In this case, existing platforms, such as DNS-like content resolution and IP based addressing/routing/forwarding architectures will be considered. On the other hand, in order to support long-term evolution of future Internet, clean-slate design of future content-centric network architectures, mechanisms and protocols will also be investigated in this work-package, which will not necessarily rely on existing network platforms. This deliverable will present the interim specification of our proposed approaches in both categories, namely the *stateless approach* and the *stateful approach*.

Nevertheless, in both design strategies a set of common design objectives and requirements will be considered for supporting future large-scale content-delivery, which is summarised below:

- *End-to-end Quality of Services (QoS) Support*: delivery of real-time multimedia based content (e.g. IPTV, video streaming applications) across the global Internet requires enhanced, sometimes stringent end-to-end QoS support. Therefore, one of the ultimate goals to be concerned is to enable QoS-awareness for supporting future content delivery services in multi-provider environments. Towards this end, individual network operators may apply appropriate networking techniques to enable such functionalities. These may include both intra- and inter-domain QoS-aware routing and path selection, service differentiations in the forwarding plane (e.g. DiffServ), as well as other mechanisms where appropriate. For instance, content caching at intermediate network devices (routers) can be deemed as yet an alternative approach dimension (to routing and forwarding) for improving end-to-end QoS such as reduced delay in content delivery.
- *Network Efficiency*: How to make efficient use of the underlying network resources for supporting QoS-aware content delivery services across the global Internet is a critical issue to be concerned by individual network operators. In particular, specific operators may have distinct policies in provisioning own resources, including bandwidth allocation, routing decisions etc. Hence, any policy/decision conflict between network providers may potentially lead to adverse impact on the outcome of end-to-end QoS provisioning efforts. As such, how providers can efficiently collaborate with each other in order to achieve a win-win situation in terms of cost-efficiency in global content delivery will become a key issue to be considered in the project.
- *Scalability and complexity*: In both evolutionary and revolutionary approaches, system scalability and complexity will always be the top design requirements to be concerned. Given potentially billions of pieces of content to be delivered across the entire Internet, the proposed network platform(s) should exhibit high scalability and low processing complexity in order to accommodate such massive amount of data. How such features can be further enhanced on top of our proposed approaches will be further investigated during the rest of WP4.

As far as the contributions from this Deliverable are concerned, we first define in section 4 the global COMET Class of Services (CoS) that can be used for supporting differentiated traffic treatment in end-to-end content delivery. How end-to-end CoS can be achieved through the mapping of specific internal QoS capabilities within participating ISP networks has been detailed in this section. On the other hand, we also consider offline and long timescale network provisioning operations for preparing for the actual content delivery, specifically the provisioning of (multi-paths) inter-domain routes across domains. On top of this, how such information can be

"disseminated" to the actual COMET content delivery decision making engine is another important issue to be addressed. Towards this end, in section 5, we present proposed mechanisms and algorithms for routing awareness process. This process is responsible for gathering information about inter-domain routing paths that are used for content delivery. In principle, this process is similar to the inter-domain routing but we introduce two main enhancements. First, the routing awareness process aims to find inter-domain paths that optimize delivery of content. Therefore, this process should match transfer requirements of specific type of content with the QoS capabilities offered by domains. The second enhancement corresponds to propagation and maintenance of multiple paths between the source and destination domain. This feature allows for applying smart traffic engineering algorithms, load balancing in the network as well as improving reliability of content delivery. The first specification included in section 5 covers format of messages exchanged between routing awareness entities (RAE), basic operations performed by RAE as well as the considerations about different strategies of RAE deployment.

Section 6 presents the two alternatives for content delivery. The stateless content delivery process presented in section 6.2 is related with decoupled content resolution process described in D3.1. This approach assumes that Content aware forwarding entities (CAFEs) forwards content based on information about the selected path that is stored in the COMET header attached to the original packet containing bits of content. As a result, CAFEs maintain only the neighbourhood (local) information, i.e., how to forward packet to the next CAFE instead of keeping routing table with all prefixes. In addition, the stateless forwarding allows the Content Mediation Plane (CMP) to select adequate path for each consumption request (similar as in case of source routing). The stateless content delivery process covers three actions that are: (1) collecting of forwarding information for selected paths, (2) preparation for content delivery and (3) content forwarding. The collecting action is performed at semi-long time scale, i.e., when new routing path is used. Its result is valid for a long time (hours, days) and may be used by multiple content deliveries as long as the selected path is the same. The result of collecting is the list of forwarding rules required to forward content along the selected path. The preparation for content delivery is invoked per each consumption request. This action is responsible for configuration of edge CAFEs and assigning appropriate COMET header containing the list of forwarding rules. Finally, content forwarding relates directly with packet handling in CAFEs.

Concerning the revolutionary/radical option, we have proposed a stateful (or *state-based*) content delivery process presented in section 6.3, which is to be tied to the coupled content resolution approach specified in D3.1. According to this approach, specific content states are required to be maintained at involved CAFEs as the ingress and egress nodes of individual domains. This type of state maintenance is similar to the principle of IP multicast, but there are some key differences. A fundamental one is that the state configuration is effectively done by the local Content Resolution and Mediation Entity (CRME) in the CMP. More specifically, during the content resolution phase, once the CRME has determined to forward the content resolution request to its counterpart in the next hop domain towards the targeted source, it is responsible for configuring the corresponding CAFEs in its local domain for preparing for the actual delivery of the content flow back to the consumer. Due to the fact that the content resolution and delivery processes are tightly coupled, the actual *domain-level* delivery path is effectively in the reverse direction of the corresponding resolution path. Detailed specification of relevant operations is provided in section 6.3.

Towards the end of this deliverable, in section 7, we provide preliminary design specification of a set of advanced content delivery techniques that are also being investigated in WP4. First of all, in order to enhance QoS performance to end content consumers as well as network resource optimization, *in-network* content caching techniques have been initially considered based on the content-centric network (CCN) model. We started this research item through some detailed performance evaluation and analysis on the CCN-based caching techniques (see Annex A). Given that many of today's content delivery applications are point-to-multipoint, we also investigate how inter-domain multicast functionalities can be enabled for global content delivery across carrier and ISP networks. Last but not least, we also introduce our preliminary design of the edge-controlled

routing (ECR) where individual content consumers are allowed to actively make requests to the COMET system at the ISP side in order to switch content delivery paths in different contexts such as perceived QoE deterioration and change of user preferences.

3 State-of-the-art of Networking Technologies

3.1 Multi-paths routing

As the current BGP has the limitation of single path selection, one may naturally consider extending its capabilities to enable path diversity. A fundamental idea of this approach is to allow an autonomous system (AS) to discover alternate routes either actively or passively, and enforce them into the routing tables. Table 1 presents several mechanisms under this strategy, and we will discuss in this section how they can increase AS-level path diversity.

Mechanism	*Path Exploration Mode*	*Description*	*Packet Forwarding method*
R-BGP	Push-based	Each AS advertises both best and the most disjoint alternate route to downstream ASes	Failover virtual interface
BGP Splicing	Push-based	Exploit candidate routes that stored in the Adj_RIB_In table	Splicing bits and tunnels
Source Selectable Routing	Push-based	Deflect traffic from the default egress points to other egress points, which could take a different inter-AS route	Tags
G-ISP	Push-based (based on subscription)	Learn alternate routes from remote (non-neighbouring) ASes	Tunnels
MIRO	Pull-based	Enquiry to any other ASes for specific routes that meets AS's requirements	Tunnels

Table 1: Inter-domain multi-paths routing approaches

3.1.1 Making extensions to BGP route attributes and selection process

One reason why BGP only allows single path selection and advertisement can be explained by its route update process: a new route for a given destination prefix replaces a previous route for the given prefix. Thus, destination prefix becomes the key to filter out duplicate route advertisements. In order to enable multiple path selection, some extensions to BGP route attributes and selection process have been proposed. In [1] , the author proposed a new attribute, called Path Identifier, in the UPDATE message. Each new route advertisement can be associated with a unique path identifier. As a result, a path is identified by a combination of the destination prefix and the path identifier. This makes BGP possible to enable multiple path selection and advertisement towards the same prefix, as long as the paths have different path identifier. Furthermore, the authors in [2] proposed a new attribute *attr_set* to ensure that the selection and enforcement of multiple paths performed independently at each BGP router does not cause any routing inconsistency.

3.1.2 Tag-based multi-path routing

Although BGP only selects one single best route, it may have learnt multiple distinct routes towards a prefix, which can be found in the BGP *Adj_RIB_In* table. This creates the opportunity to achieve path diversity by exploiting these routes that have not been finally selected. BGP splicing [3] takes an advantage of the implementation of modern IP routers, which allows the existence of multiple line cards, in order to enable installation of multiple routes, which are obtained from the

Adj_RIB_In table. In the data plane, a few extra bits, called splicing bits, in the packet header indicates which routing table to use. In this scenario, in addition to the default route towards each specific destination prefix, packets can be also tunnelled to alternate egress routers in which case a different AS path will be followed towards the same destination. An advantage of this solution is that it only utilizes the locally stored routing information without explicitly requiring neighbouring ASes to advertise additional routes. However, it requires some modifications to the IP header in order to for IP routers to recognise the routing bits from the packet header and use them to select the corresponding path.

A similar approach to BGP path slicing, which uses extra bits to control packet forwarding over multiple routes, but do not require making extensions to BGP is to deflect traffic off the default inter-AS route. From an AS point of view, this can be done by deflecting the traffic to alternate egress points in which their outgoing inter-AS paths may traverse different ASes. This is referred as source selectable routing [4] . More specifically, a small set of potential egress routers, which can provide loop-free inter-AS route diversity is firstly identified. End systems can measure the quality of these diverse routes and select the best one that meets their service requirements. In the data plane, similar to the splicing bit, tags are carried by each packet to indicate which egress inter-AS path to use (i.e. either the default or the deflected one). One concern of this solution is that deflecting traffic to alternate egress points by an ISP without global cooperation may interfere with some performance goals in intermediate ISPs.

3.1.3 Learning alternate routes from non-adjacent ASes through subscription

It can be observed that if an AS learns routes only from its neighbouring ASes, the achievable path diversity may be limited, as either the neighbouring ASes may also have only limited knowledge of diverse paths, or they may choose not to advertise alternate paths they are aware of. A possible solution to this limitation is to learn alternate routes directly from *remote* (i.e. non-neighbouring) ASes in the Internet, as they may have learnt routes that are different from the ones advertised by the neighbouring ASes.

A Global-ISP (G-ISP) [5] serves as an additional (virtual) ISP that provides transit services to its remote customer ASes. A customer AS subscribes to G-ISPs, which are not necessarily directly connected, and maintains BGP connections with them in order to learn the routes they advertise. With G-ISPs, customer ASes can learn more routes than solely relying on their neighbouring ASes. If the route provided by a remote G-ISP is selected, traffic will be delivered to the G-ISP via inter-AS tunnels and from there the traffic is forwarded to the destination over the standard BGP route.

3.1.4 Pull-based path exploration

For all the approaches previously presented, their path exploration methods can be regarded as *push-based* mode, which follows a similar fashion to the current BGP in which downstream ASes take a proactive role in advertising alternate routes to upstream ASes. As opposite to the push-based mode, in the *pull-based* route discovery, an upstream AS actively requests alternate routes from the downstream ASes when necessary. This pull-based route discovery is generally more scalable than the push-based mode since an AS requests alternate routes (often with specific requirements) only when it reckons necessary, thus avoiding unnecessary BGP route messages to be advertised over the Internet.

A notable pull-based path exploration approach, MIRO (Multi-path Inter-AS Routing) [6] allows a customer AS to make enquires to its neighbouring ASes for alternate routes according to its own specific requirements, in addition to the single default route that would normally provided by BGP. An example of route enquiry can be "*Any route to AS X avoiding AS Y (which is included in the current default path)?*". Then, the responding ASes can advertise any feasible route that they have learnt from other ASes. Finally, tunnelling is used to deliver traffic along the negotiated alternate route.

An advantage of MIRO is that, through negotiations, responding ASes can make a decision on which alternate routes to advertise based on their local network policies, thereby giving them more control over the flows of traffic entering into their networks. On the other hand, customer ASes are offered greater flexibility by the ability of expressing their preferences on route enquires. MIRO also avoids explosion in disseminating reachability information and is backwards compatible with BGP for incremental deployment. One concern of this solution is that additional routing table is required to store the states of incoming and outgoing tunnels.

3.1.5 Structured Inter-AS Overlay

Instead of making extensions to the current routing protocols like BGP, path diversity can also be increased by creating a virtual routing layer over the underlying Internet routing infrastructure. Such approaches are often used for enhancing reliability and QoS purposes. An overlay network can be formed among several nodes and it is able to bypass BGP policy-driven routing decisions in order to find paths between nodes that cannot be explored by BGP [7] , thereby increasing path diversity. QoS-awareness can be also introduced to overlay routing, even across multiple autonomous systems [8]

The basic principle of overlay network is that overlay nodes are installed in individual ASes and they exchange network information about the measured quality of paths via a dedicated signalling mechanism according to specific performance metrics such as throughput, delay, etc. Routing in the overlay networks makes use of intermediate nodes: the traffic traverse a path from the source node to the intermediate node followed by the path from the intermediate node to the destination node, all along their underlying BGP routes. As the intermediate node is not on the default BGP path from the source to the destination, the end-to-end overlay paths will not be overlapping with the default one. The benefits of using overlay network are numerous, including enhanced route availability, performance, reliability, etc. There are some concerns on using overlay, however. A major issue is the violation of legacy inter-AS policy. As overlay routing can choose arbitrary paths between ASes, it is possible that an overlay route will violate some fundamental guidelines of inter-AS routing, such as *valley-free* and *customer-route-preferred* routing, in which case routing disputes between ASes and instability problems could result. Other concerns include increased network overheads due to the measurement and dissemination of overlay path quality as well as encapsulation for packet forwarding.

3.2 Multi-criteria routing

The multi-constraints routing, also known as QoS routing, aims to select routing paths that satisfy a given set of constraints [9] [10] . Let us consider the network as a directed graph $G(N, E)$, where N represents the set of nodes, while E is the set of links. Each link $u \rightarrow v$, $u,v \in N$, $u \rightarrow v \in E$, is characterised by an m-dimensional vector of weights $w(u \rightarrow v) = [w_1, w_2, ..., w_m]$. The multi-constraints routing finds a path P going from a source node to a destination node that should satisfy given vector of constraints $L=[l_1, l_2, ..., l_m]$. This means that $w_i(P) < l_i$, $i=1, ..., m$, where $w_i(P)$ is the weight i of the path P calculated as the concatenation of the weights w_i of all the links in the path. Usually, there are a number of feasible paths that satisfy constraints between source and destination. Therefore, the routing should select from the set of feasible paths at least one preferred path or, in case of multi-paths routing, a set of preferred paths. For this purpose, the multi-constraints routing uses additional optimisation criteria that allow to rank feasible paths. The optimization may have different objectives as e.g., minimising path length, improving QoS characteristics, providing load balancing, etc.

Multi-constraints routing belongs to the class of Multi-Constraint Problems (MCP) that, in principle, are NP-complete. Since it is difficult to find exact solutions for the NP-complete problems, the commonly investigated approach is to apply heuristics methods. Below, we review the most recently investigated approaches.

3.2.1 Methods based on single metric

The simplest heuristic method, proposed in [11] to tackle with multi-constraints routing, is to convert the m-dimensional vector of weights $w(u \rightarrow v)$ into a single scalar value w by using an appropriate cost function $f(.)$, $w_{u \rightarrow v} = f(w(u \rightarrow v))$. It allows to solve the multi-constraints routing by the Dijkstra or Bellman-Ford algorithm. On the other hand, this approach does not guarantee that selected paths will satisfy the constraints because the scalar value w does not contain enough information about the particular constraints. Therefore, we do not recommend this method for the COMET path discovery process.

3.2.2 Methods based on path cost function

These methods assume that routing concatenates vectors of link weights along the path. Thanks to this, each node may check whether the considered paths meet constraints and then it may select the preferred path from the set of feasible paths. The key element having impact on the effectiveness of these methods, is an appropriate selection of cost function *cost_f(.),* which is used to rank feasible paths and then to select the preferred path. Among others, this problem was studied in [12] [13] [14] [15] . These studies pointed out that the most effective cost functions are nonlinear, strict monotonic and convex functions. The commonly recognised approach is to apply functions based on the Minkowski norm of order p, see equation (1). The Minkowski norm ranks feasible paths based on the distance from the point zero.

$$cost\_f(.) = \begin{cases} \left(\sum_i \left(\frac{w_i}{l_i} \right)^p \right)^{1/p} & , \quad w_i \le l_i \, , i = 1,\ldots,m \\ \infty & , \quad w_i > l_i \, , i = 1,\ldots,m \end{cases} \tag{1}$$

One interesting extension analyses a number of preferred paths instead of a single one [15] . This approach allows to consider a set of solutions, which are indistinguishable from the ranking provided by the cost function. As a consequence, such k-path algorithms may find better solutions comparing to single path algorithms, especially in the case of hop-by-hop routing where decision taken by one node strongly influences the decision space of following nodes.

On the other hand, there is a number of proposals for extension of the BGP (Border Gateway Protocol) protocol towards the multi-constraints routing. These approaches, investigated in [16] [17] [18] [19] [20] define new QoS_NLRI attribute for the BGP to carry information about Classes of Services (CoS) supported along the inter-domain paths. As CoS are usually characterised by a number of parameters, these proposals use different cost functions to evaluate candidate paths and to select the preferred path. The objective of cost functions is to maximise the availability of CoS, so they take into account specific features and constraints of particular CoS. The investigated functions cover: lexicographic comparison of path weight vectors, delay and bandwidth index and different types of norms, like e.g. the above mentioned Minkowski distance or the inversed distance from the target values.

Special interest is brought by the approach of the EuQoS project [19] . It presents a cascade QoS interconnection model, where each Autonomous System (AS) establishes end-to-end SLAs only with immediately adjacent ISPs. Besides, end-to-end traffic classes of service (CoS) are globally defined. The key point is that provisioning and route advertisement is performed independently per CoS. Firstly, pSLSs are provisioned per CoS and then each AS advertises for each CoS, through an extension of the BGP protocol (named EQ-BGP), the end-to-end QoS capabilities from its ingress point towards each IP destination prefix. End-to-end QoS capabilities are calculated per CoS by “concatenating” the local QoS capabilities per CoS and the end-to-end capabilities per CoS offered by the adjacent AS. This approach allows having different paths for each IP destination prefix, one for each particular CoS, while also leaves room for different concatenation methods for

each CoS. However, the EQ-BGP speakers run in practice one BGP instance per CoS, which generates more volume of routing signalling traffic (n times more than the required by normal BGP, where n is the number of CoSs).

For the COMET routing awareness and path discovery processes, the methods based on the path cost function and the k-path algorithm seems attractive. However, COMET should define the appropriate cost function, which takes into account the requirements for content delivery. Besides, the BGP extensions in EQ-BGP seem also interesting for the purpose of COMET.

3.3 Content caching techniques

One of the most common ways to overcome the scalability problem is with the introduction of redundancy, which in the case of content is done via caching. It allows the provision of a faster response to the users, since they are typically placed closer to the end user than the content server, while at the same time offloads the central content location as well as the links on the whole path.

In the course of the last two decades a number of similar approaches have been developed including transparent caches (often done by proxies) or complete mirroring of content. Content Distribution Networks (CDNs) use extensively these approaches jointly with different user redirection techniques (typically DNS or HTTP redirection). Due to their relevance, CDNs deserve a separate section (see section 3.4).

Irrespective of the actual caching approach or technology in use, the following questions need to be addressed:

- Identifying content that needs to be cached.
- Cache location — where in terms of physical and network topology caches should be placed.
- Cache replacement policies — how long we need to keep content before it is outdated or is not required any more.

In this section we will present classical as well as some of the more novel approaches to the above problems.

Some protocols, such as HTTP include support for caches, which allow both parties (server and client) to request non–transparent operations (W3C). This helps to identify when cache content should expire.

3.3.1 What needs to be cached

Content caching can be broadly split in two areas: *object* and *byte caching*.

One of the basic forms of caching is *object caching* which is the technique used by most of web caches. All objects requested by users and which passes through the proxies, are cached and stored until they expire by cache replacement policy. An object in this context constitutes complete pages or under some circumstances parts of pages, images or any other media objects.

While the above approach is the simplest and most effective when dealing with rather static content it has major drawbacks when content is changing often or is otherwise dynamic. Moreover it is protocol dependent. *Byte caching*, on the other hand, works on the TCP level and is only concerned with byte flows rather than the content that they constitute.

A number of technologies are specifically targeting caching dynamic web content and aiming to solve various problems, which occur when caching complete pages (e.g. complete page would invalidate if any of its dynamic parts change no matter how small it is). Some CDNs (Akamai) and web application servers use dynamic page assembly when individual fragments of the page are cached separately and the final page is assembled at the edge servers. This approach has been extended to dynamic proxy-based caching in [21] .

Caching multimedia is a special topic. As the files tend to have significant size, storing complete files might be rather complex, especially as users can randomly request different parts of audio/video streams and not all parts need to be cached. In [22] *segment-based caching* was presented and it was shown that it is effective in terms of hit ratio and average start-time and can be successfully utilized when the users' interest is volatile. This approach has seen further development in particular in [23] where lazy segmentation and active pre-fetching methods are discussed. In this context we would also like to mention a study presented in [24] which looks at the ecosystem of proxy and multiple media servers and shows that "the network performance improves significantly when the caching of media files is segment-based and segments are allowed to migrate among the content servers". The segment based approach to caching video content combined with hierarchical–tree proxies structure frequency-based policy is presented in [25] which shows how a tree topology helps to achieve higher hit-ratio.

3.3.2 Cooperative Caching

The most basic scenario of caching is the caching in consumer's premises, e.g. to cache web content in an office. In this scenario, a single cache server is used, typically a caching HTTP proxy.

However, in larger scenarios involving lots of users from different places (caching by CDNs or by large content providers), there is a need for multiple cache servers. This leads, in turn, to the requirement that these servers must communicate to select the most appropriate source of content. This concept is known as *cooperative caching* and was first formalized in application to the squid web cache ICP protocol [26] . This work introduces a lightweight messaging protocol, which is used to communicate between multiple squid servers. The protocol has become a widely used standard in modern web.

Alternative solution to ICP — a Cache Digest was presented in [27] which allows proxies to expose information about their content in a compact form, which would be consequently used to identify "peers" with required content.

An attempt to tackle the problem of excessive overhead lead to the Summary Cache protocol [28] where each cache server stores a summary of the content cached by other participating nodes to avoid unnecessary requests.

Cooperative Caching has also been integrated with application level multicasting in [29] for use particularly in CDNs for delivering multimedia content. The approach allows for close cooperation between caches in the same cluster while cooperation between servers in a different cluster can be rather limited.

3.3.3 Cache Algorithms

Cache algorithms, often referred to as replacement policies, try to optimize hit-ratio vs. latency balance constrained by given resources. The basic idea is to try and predict which content will be required based on the historical data. The optimal solution, which is to discard the information that will not be needed for the longest time in the future, is known as Belady's optimal algorithm or the clairvoyant algorithm.

The typical and most trivial algorithms are Least Recently Used and Least Frequently Used, which rely on access time and frequency respectively to select which objects are to be removed. An attempt to optimize performance by keeping track of both parameters is made in Adaptive Replacement Cache [30]

Various other parameters are taken into account by different more sophisticated caching algorithms; in particular cache items can have different retrieval and storage costs, and can have an associated expiration date. The later is particularly used in HTTP, which defines whole range of cache control headers as defined in RFC 2616 [31] such as *Expires*, which explicitly sets when the item in cache should expire.

3.3.4 Cache location

When dealing with multiple cache servers the question which arises is where they should be located to optimize network performance and user experience. The question was given particular importance in view of the increasing spread of CDNs, which need the best algorithms for locating multiple content replicas.

In the case of transparent caches (those that do not require any actions from users' side), the problem of cache location is discussed in [32] . This work attempts to find a solution of minimizing data flow and average delay given a certain number of caches and provides optimal solutions for two network architectures and touches upon some special cases. Furthermore it provides computationally efficient solution for practical problems.

In this context it is important to mention [33] which attempts to optimize performance under a given traffic pattern by best placing M proxies in N potential locations. The problem is modelled as a Dynamic Programming one and optimal solution for tree topology is presented which is $O(N^3M^2)$ complex.

The mirror (or replica) placement which is strongly related to cache location but is of particular importance to CDNs is discussed in [34] . In this work the problem is constrained by allowing mirrors to be placed in particular locations. The results of this work include best-performing algorithm to tackle the problem. Moreover it states that "*there is a rapidly diminishing return to placing more mirrors in terms of both client latency and server load balancing*".

3.3.5 Caching in CCN

Authors in [35] have proposed a Content-Centric Networking model, which is mainly based on *Networking Named Content*. According to that approach, data packets are uniquely named, according to the content they transfer. That said, content that is requested multiple times, i.e., by more than one end-user, can be delivered to all interested parties as it travels through the network.

In particular, instead of buffering a packet till it is forwarded to the interested user and then discarding it, as happens today, CCN first forwards the packet to the interested user and then *"remembers"* the packet, till this packet *"expires"*. The *"remembering"* and *"expiration"* of packets is accomplished using caching techniques at the *packet* level. The model requires every CCN-compatible router to be equipped with a cache, which holds packets for some amount of time. If subsequent requests for this same content arrive before the content expires, the packet is forwarded again to the new interested user, instead of having to travel back to the content server to retrieve it.

The sequence of events from the *content-request* to the *content-expiration*, according to is described below.

Content-request. A request for content is made, based on an *Interest* packet issued by the user. This *Interest* packet contains the name of the content and is propagated towards the content server.

Content-resolution. The *Interest* packet travels back towards the content server, in order to find the requested content and ship it back to the user. Each router on the way from the content-requester to the content server keeps a registry of the *Interest* packet, in order to form the return path from the server back to the user.

Content-retrieval. The requested content is sent back to the user, following the same path as the *Interest* packet. In this case, the content is said to *"consume"* the *Interest* packet. Note that *Interest* and content/data packets maintain a one-to-one relationship, resembling the TCP-ACK packet relation.

Content-remembering. The content-remembering procedure comprises the main difference from today's IP flow model. According to the authors in, *"CCN caches are the same as IP buffers, but have a different replacement policy"*.

A CCN router, after forwarding the packet to the outgoing interface, puts a copy of it *at the top of the cache*. Hence, as the content travels back from the server to the interested user, it is cached in all CCN-compatible routers along its way. Since CCN content can be identified by *name,* future requests for the same content can be served by intermediate routers, if the content is still in the cache.

The amount of time that this content is cached for depends mainly on i) the number of requests that a particular router is receiving for this specific content, ii) the number of requests for other contents, and iii) the size of the cache. Once a new request (i.e., *Interest* packet) for a *cached* content is received, this content is forwarded to the user and is also *put back at the top of the cache*.

Content-expiration. As new *Interest* packets requesting different content arrive, other cached packets move towards the bottom of the cache. When a packet reaches the bottom of the cache, it is discarded and *"expires"* from the cache (in other words, the router *"forgets"* about this packet).

Clearly, this will benefit *popular content*: content that is requested multiple times, *simultaneously*, or within the same time-window, will be distributed accordingly *by the intermediate router*. This way response time and network resource requirements are reduced, since the content does not have to be retrieved from the content server.

As the content travels back from the server to the requesting user, it consumes all requests interested in it and then it is stored to the router's buffer. According to the authors in [35] the router buffer is not different than what we have today, size-wise, i.e., it is not comprised of huge hard disc memories. Therefore, once the buffer is full due to more recent requests, content is discarded accordingly, following a *Least Recently Used* (LRU) approach.

3.3.6 Related Research Areas: Macro- vs. Micro-Caching

The above approach can be considered as closely related to i) multicast, ii) in-network caching [36] and iii) network-level pub/sub data dissemination [37] . For example, web-caching approaches depend heavily on coordination techniques, in order to initially decide *where to cache content* and later on *how to find the cached content*. In particular, coordination can be either explicit, or implicit. Explicit coordination implies that caches communicate with each other to exchange information about cached content and deal with incoming requests accordingly. Obviously, this approach suffers from the communication overhead needed in order to exchange information regarding the cached content. In contrast, implicit coordination does not rely on message exchanges to cache and locate content, but instead, local cache management policies take care of caching or not caching incoming content. Clearly, the CCN approach is closer to the implicit coordination caching techniques. For example, in [38] authors propose an implicit in-network web-caching approach, where cache management takes place in a distributed manner. They call their approach *Breadcrumbs*, to reflect its main operational property. That is, as the requested file is downloaded from source (server) to the requestor, the file leaves a *"trail of breadcrumbs"* on its way back. The *breadcrumb* is a small packet, so the authors assume that it can be stored in the intermediate servers indefinitely. Upon a request for the same file from another user, the request follows the trail of breadcrumbs, upwards, till it finds a cached copy of the file, or reaches the source. Although this is an interesting approach as for the content retrieval process and relates to the CCN approach, it does not apply to simultaneous requests (i.e., flash crowds), which is the main target of [35] .

In general, strategic content replication to reduce response time and the amount of network resources (e.g., bandwidth and server load) needed to send the content back to the user have

already been investigated in the form of i) Web Proxy Caching (e.g., [38] [39] [40] and ii) Content Delivery Networks (CDNs) (e.g., [41]).

In both cases, the issue of utmost importance is the choice of the best possible geographical location to (either statically or dynamically) replicate content. In case of Web-Proxy caches, the ISP is responsible for deciding where to replicate, while in case of CDNs, this role is taken by the corresponding company that owns the network of surrogate servers. In both cases, however, *the location of the proxy/surrogate server is always known and fixed in advance.*

Web Proxy Caching techniques deploy web-proxies in strategic locations in the network to store whole web-pages. Proxies are normally used and controlled by ISPs in order to offload parts of their networks and servers. There are many different web-caching techniques, e.g., static vs dynamic caching, or caching of web-pages that are foreseen to be popular etc., but *the location of the proxy is always agreed/fixed in advance*. This is in contrast to the CCN caching paradigm proposed in CCN.

A CDN is a network of surrogate servers which stores copies of contents that originally reside in the origin server of the content provider/publisher. There are different ways to distribute content efficiently into the surrogate servers, over a given geographic area, and different architectures on how to build a CDN (e.g., active network vs. overlay approach). CDNs are discussed in further detail in Section 3.4. The point of interest to us for now is that similarly to Web Proxy Caching, in CDNs too, *the location of the proxy is always agreed/fixed in advance.*

Furthermore, IP-multicast [42] [43] has been proposed and investigated in the past to serve multiple users that are simultaneously interested in the same content. However, IP-multicast serves users that belong to the same group *only* and are prepared to receive all of the content that the *group* wants to receive (i.e., not necessarily the parts of it that individual users are interested in).

CCNs, , constitute the conceptual marriage of the above technologies, *but in the "micro-level"*. We classify Web-Caching and CDNs as *"macro-caching"* approaches, since they target caching of entire objects, be it web-pages, or files; we consider CCNs as a *"micro-caching"* approach, since caching here is done at the packet-level. Moreover, in all the above technologies the setup is fixed and predefined. In contrast, *in CCNs content is cached and multicast "on-the-fly", wherever it is requested or is becoming popular*.

3.4 Content Delivery Networks

3.4.1 Introduction

The demand for multimedia content, such as online video, audio streaming, gaming and webcasting, over Internet is increasing rapidly. Traditional web sites are changing in order to support their customers' needs for multimedia content delivery. In addition to the traditional multimedia content, new kinds of content are emerging, such as high-definition TV, social media, interactive applications and multimedia over mobile networks. QoS and mainly QoE are now something expected by the content consumers since, the delivery of multimedia content over Internet is getting more and more popular.

Better network conditions such as better bandwidth availability, reduced network delay and consistent delivery times are required in order to satisfy the delivery of some highly demanding services (i.e. HD video streaming and online multiplayer gaming). For instance, an HD video requires about five times the bandwidth of the same video in SD format [44] . However, some multimedia applications might not work properly due to the flash crowd problem. The flash crowd problem refers to the phenomenon of large amounts of demands occurring simultaneously for the same content which causes network congestion or/and increases the levels of traffic flow [45] .

In order to overcome the problems caused by the high multimedia content demand and to provide the expected QoE to the end users, *Content Delivery Networks (CDNs)* have been proposed. These new generation networks achieve more efficient content delivery over the Web and improve content consumers' experiences. In addition, they allow content providers to offer higher quality experiences [44] [46] [47] . Next it is discussed how the CDNs achieve this.

3.4.2 What is a content delivery network?

A CDN is a distributed system of servers interconnected through the Internet cooperating together to satisfy users' requests for content delivery. In particular, a CDN replicates content from an origin server to a group of geographically distributed cache servers, called surrogate servers, and efficiently helps the delivery of multimedia content from content providers to a large community of content consumers [1][44]

The surrogate servers store copies of identical content found on the origin server (the server in which content provider's content is stored). In this way content is located closer to end-users. When a request for content is made by a user, the particular request is directly sent to the cache server "closest" to the user. The notion "close" could include geographical, topological, or latency considerations [46] [48] Then, the requested content is delivered from the cache server to the end user through a core network and one or more access networks. This delivery is done in a transparent way from the end user's point of view. Particularly, the end user does not observe any difference in the way that their requested content is delivered from [44]

From origin server's perspective, a tremendous amount of work is offloaded since a CDN is responsible for delivering content on its behalf [48] . The cache servers of a CDN may store some or the entire origin server's content. Therefore a CDN tries to reduce the direct delivery of content from the origin to an end-user as much as possible.

The key idea of the CDN mechanism is to serve each user's request by locating a cache server, including the requested content, close to the user. As a result, network performance is improved (bandwidth maximization, accessibility improvement and correctness maintenance through content replication are achieved) and thus, faster, more reliable and with higher quality applications and services are offered to the end-users. Content providers benefit from higher customer satisfaction, lower bandwidth consumption, reduced infrastructure costs etc. End-users benefit from the content delivery quality, speed and reliability improvement [44] [47]

Furthermore, a CDN addresses the flash crowd problem by offering scalability; the ability of a network to expand in order to handle increasing content demands and serve bandwidth-hungry applications without any significant reduction in performance [1][45]

However, the price of a CDN service is quite high [50] which sometimes does not fit with the budget of many small to medium enterprises (SME). Some of the factors that affect the cost of a CDN service are the bandwidth cost, the variation of traffic distribution, the size of content replicated over surrogate servers, the number of surrogate servers and reliability, stability and security issues [45]

3.4.3 CDN state of the art

Initially, CDNs were horizontal, providing the same delivery mechanism for all kinds of content. Nowadays, this approach does not meet the requirements of modern content delivery applications. Different services require different types of traffic and/or performance needs; therefore, CDNs should provide different solutions for each kind of services. New CDNs will be vertical in order to be able to support all content providers' specific needs [44]

3.4.4 Stakeholders

The major stakeholders of CDN architecture are: the *content provider*, the *CDN provider* and the *end-users*. A content provider is the person who supplies multimedia content to be distributed

throughout the Internet. The content provider's multimedia content is stored to their origin server. A CDN provider is a proprietary organization or company that provides infrastructure to deliver content in an efficient and reliable manner on behalf of content providers. CDN providers may cooperate or compete with each other. A content provider must sign up with a CDN provider and pay the specific fee in order to use the CDN service. End-users are the entities that access content from the content provider's website [45]

3.4.5 Architecture

Typically, a CDN consists of four subsystems: *content-delivery*, *request-routing*, *distribution* and *accounting* sub-systems [45] [49]

Content-delivery

The content-delivery subsystem consists of a set of edge servers (surrogates servers), responsible for delivering copies of content to end-users on behalf of the origin servers. These servers are located at the ISP's points of presence (POPs) or entry points which are located at ISP's backbone. There are two approaches about how a CDN is structured: the *overlay approach* and the *active network approach*. A CDN follows the first approach when core network components, such as routers and switches, do not assist the delivery of content. The second approach is adopted when routers and switches play an active role in content delivery [44] [45]

The placement of surrogate servers throughout the Internet is a critical issue and is directly related to the efficiency of the content delivery process [47] . Several placement algorithms have been proposed, trying to specify the optimal locations of surrogate servers [51] These algorithms achieve improved performance for the end users and minimize the infrastructure's cost [47]

Moreover, efficient techniques can be applied when a large amount of users request the same multimedia content. In this case, a CDN can use *IP multicast* or the *splitting technique*, in an effort to increase the efficiency of content delivery to the users[45]

Request-routing

This sub-system is responsible to redirect the users' requests to the appropriate, nearby edge server [49] The major mechanisms that are used for request-routing are: *DNS redirection* and *URL rewriting* [49] The request-routing is also responsible for interacting with the distribution sub-system and keep updated content stored in the cache servers [45]

Distribution

The distribution sub-system is responsible to move content from the origin server to the surrogate servers in an efficient way. It also ensures consistency of content stored in the surrogate servers. Due to the limited disk space of a cache server, it is not feasible for all the content available at the origin servers to be replicated to a single cache server. Therefore, cache servers replicate part of the content found in the origin servers and this selection of content has to be done efficiently [1][45] .

There are three approaches related to distribution: *cooperative push-based*, *uncooperative pull-based* and *cooperative pull-based* [45] [47] . Using the cooperative push-based approach, the content is loaded in the cache servers before the users' requests and the surrogate servers cooperate in order to reduce the replication and update cost. When a CDN receives a user's request, it is directed to the surrogate server that contains the requested content. If content cannot be found in anyone of the CDN's surrogate servers, the request is directed to the origin server. In [50] they proved that greedy-global heuristic algorithms are the most appropriate solution for push-based schemes. However, this approach has not been yet adopted by a CDN provider [52] [53] . In the uncooperative pull-based approach, a customer's request is served by the cache server if it has the content, otherwise, if a cache miss occurs, the cache server pulls content from the origin server and saves it for future use. In this case, the cache server does not need to pre-fetched content from origin server until it receives the first request for this particular content. However, the optimal cache server from which to serve the content is not always chosen. The last approach, cooperative

pulledobased approach, provides a smarter solution than the previous one. When a cache miss occurs, all the cache servers cooperate with each other in order to find nearby copies of the requested content and store them in their caches. Uncooperative pull-based approach is used by the most popular CDN provider such as Akamai and Mirror Image [47] .

Accounting

The responsibility of the accounting sub-system is to keep logs of client accesses to the CDN and other information such as the usage of the CDN servers and network statistics. This sub-system works in real time and cooperates with the content-delivery, request-routing and distribution sub-systems to gather the useful information. This information is used for accounting, billing and maintenance purposes [45]

3.4.6 Most popular CDN providers

Akamai Technologies (www.akamai.com). It is currently the market leader (more than 80% of the overall CDN market) in providing content delivery services. It owns 61000 servers deployed in 70 countries. Akamai was proposed by MIT researchers in their effort to solve the flash crowd problem and now delivers static, dynamic content and streaming audio and video. It follows the DNS-based request-routing technique, un-cooperative pull-based distribution approach and network and overlay content delivery structures.

Mirror Image Internet, Inc (www.mirror-image.com). Mirror Image supports surrogates servers located in 22 countries across America, Europe and Asia. It supports static content, streaming media, web computing and reporting services. Network and overlay approaches are followed as well as the un-cooperative pull-based approach [47]

LimeLight Network (www.limelightnetworks.com). It owns surrogate servers located in 72 locations around the world (America, Europe and Asia). LimeLight Networks delivers static content such as video, music, games and in general delivers huge amount of content to large audiences. It follows the overlay, un-cooperative pull-based and DNS-based request-routing approaches [45] [47]

Some other CDN providers are: *Accellion* (www.accellion.com), *AppStream* (www.appstream.com), *EdgeStream* (www.edgestream.com), *Globix* (www.globix.com) and *Mirror Image* (www.mirror-image.com).

4 Quality of Service Engineering for content delivery

4.1 Overview

The COMET approach for Quality of Service (QoS) engineering assumes that content is delivered using dedicated paths (content delivery paths), which support COMET Classes of Service (CoS). The COMET CoS assures adequate transfer of given type of content in each domain between server and client. The content delivery paths are organized and maintained by the Path Management Functional block (PMF) with the aid of routing awareness process. The PMF is responsible for matching the content delivery requirements with the transfer capabilities offered by particular domains. Figure 1 illustrates the basic idea of the investigated approach.

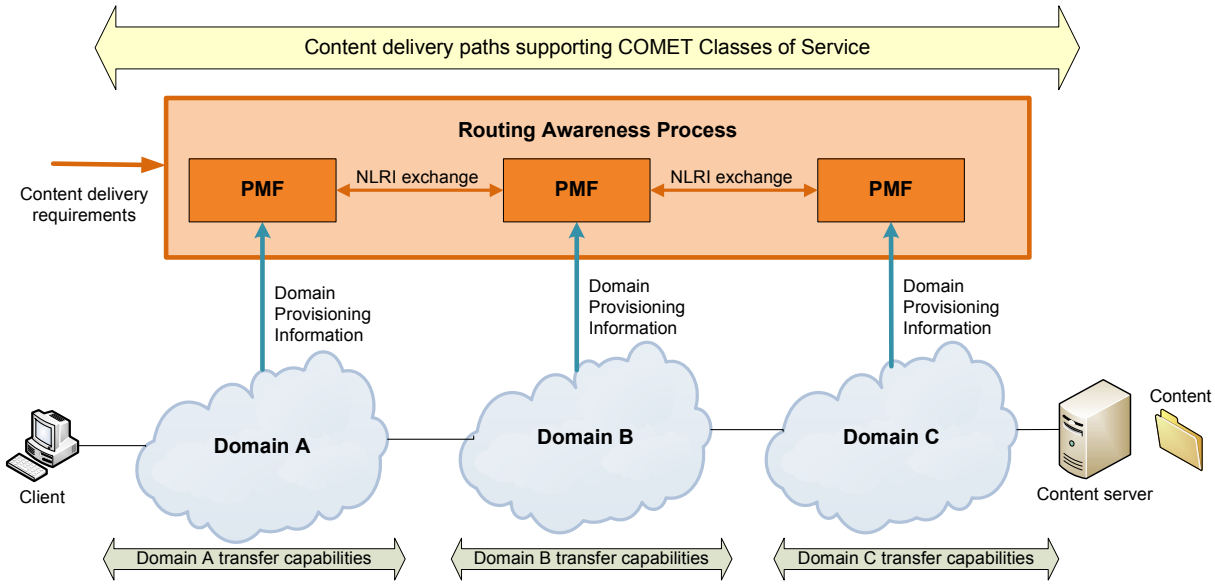

Figure 1: Approach for creating content delivery paths.

The domains usually differ in offered transfer capabilities, i.e., they support different intra-domain CoSs. In order to perform coherent transfer of the content from the server until the client, each domain should perform a mapping between COMET CoSs and the transfer capabilities offered by itself. This mapping should be performed as a part of configuration of COMET system, after the domain provisioning information is available, e.g., domain owner policies, available resources and transfer capabilities.

Other important elements enabling the QoS engineering in COMET are the Content Aware Forwarding Entities (CAFEs). CAFEs handle the packets transferring the content by using a dedicated COMET header appended to the packets as shown in Figure 2. This allows to take advantage of three main features: first, CAFEs may enforce the content delivery paths regardless of existing routing. This feature not only relaxes constraints of standard routing protocols where single shortest path is used (e.g., BGP), but it also allows to use content delivery path on demand for particular content consumption (using previously established paths). Second, CAFEs may perform adequate packet classification/marking for mapping between COMET CoSs into transfer capabilities used in particular domain. Although this function seems to be simple, it is essential to provide end-to-end QoS across the multi-domain network. Both RAEs and CAFEs allow the creation of an overlay network for content delivery, which in turn enables the setting of specific interconnection agreements for content delivery, independent of those currently used for Internet traffic, with autonomous systems potentially located at hop distances higher than one AS.

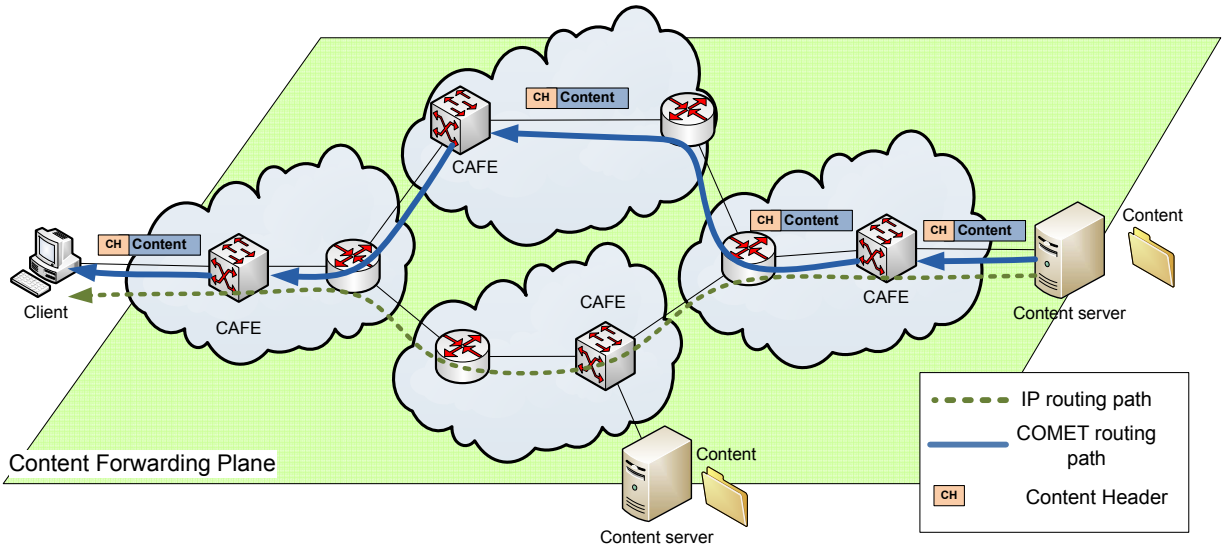

Figure 2: Approach for enforcement of content delivery path.

Summarising, even though the content delivery in COMET relies on transfer capabilities provided by particular domains, the PMF and CAFE allow to apply specific traffic control and engineering algorithms in order to optimise content delivery paths. Thanks to them, the COMET may improve both the quality perceived by content consumers and the efficiency of resource utilisation (network and server resources).

4.2 COMET Classes of Service

The COMET approach for delivering the content follows the commonly recognised approach of end-to-end Classes of Service (e2e CoS) that is widely investigated for providing QoS in a multi-domain network, see [54] [55] [56] The e2e CoS define the transfer capabilities required to meet QoS requirements of given type of traffic. The e2e CoSs are defined in the global scope, and then they are mapped into appropriate intra-domain network CoSs offered by particular domains. These intra-domain CoSs have only local meaning, because particular domains may differ in offered intra-domain CoSs and the assured QoS level, depending on service provider policies, technical constraints or business relations.

In COMET project, we follow the end-to-end CoS approach with two main extensions. First, we define the end-to-end CoS, called COMET CoS, that are strictly oriented to content delivery. They are designed taking into account characteristics of content types considered in COMET. Similar to end-to-end CoS, the COMET CoSs are defined in the end-to-end scope and they should be properly mapped into intra-domain CoSs offered by particular domain. The second extension is the COMET specific routing awareness function, which allows to fix inter-domain routing paths taking into account requirements of COMET CoS and transfer capabilities of intra-domain CoS offered by particular domain. Thanks to routing awareness function, we assure wide visibility of COMET CoS. The COMET CoS are used in several COMET processes. First, each content is assigned to COMET CoS during content registration based on required service level. Second, COMET CoS are used in routing awareness process to build content delivery paths based on intra-domains CoS. Finally, we use COMET CoS during path configuration process to assign appropriate forwarding rules to delivered content, e.g. marking, queuing.

We start the design of COMET CoS from the analysis of content characteristics. First of all, we classify the content considered in COMET in content types based on the content nature and the requirements for content transfer. Basically, we distinguish two content types that are: *live* and *pre-recorded*.

The main feature of the **live content** is that content producer prepares the content in an on-line manner, i.e., the content is captured, encoded and transmitted in a scale of fraction of a second. As a result, we do not know the size of the content, although the duration can be approximated. Good examples of this content type are live transmissions of sport or cultural events.

Even though we cannot accurately describe the duration or size of the content, the important characteristic is the setting of the codec. This allows for estimation of traffic profile emitted by servers, where we distinguish 2 main types of encoding: i) constant bit rate encoding and ii) variable bit rate encoding. From the perceptive point of view the variable encoding offers better QoS (better quality at lower bit rate), while at the same time it generates bursts of data that may temporarily overload the network. As a result the buffers may overflow and encoded stream may be damaged.

Notice that the content server usually streams the content for multiple clients at the same time. This naturally brings into attention the point-to-multipoint capabilities, e.g., multicast. While there are several overlay solutions that uses TCP transport protocol (e.g. Octoshape), the transmission over UDP is more natural. In addition, transmission over UDP is able to exploit natural features of multicast in IP networks. On the other hand, it requires more stringent transfer capabilities in the network, e.g. lower packets losses.

The **pre-recorded content** is prepared in an off-line manner, i.e., the processing of content is completed and then it is ready for transfer. All properties of the content and its meta-information are known beforehand. Typical examples of such content are movies or songs.

While both duration and bit rate requirements are known, the emitted traffic profile depends on the used transport protocol. The UDP protocol reduces the load at server side and simplifies the synchronization at the receiver. The problem with traffic bursts remains the same as for live traffic. Although using TCP protocol filters the bursts produced by application and in addition, allows for retransmissions of lost packets, it is often limited by, so called, *bandwidth delay product* which bounds the maximum achievable throughput when network delay becomes noticeable. This is important limitation when the content uses high resolution fidelity and thus requires large amounts of bandwidth.

The pre-recorded content is usually available in multiple network locations, which improves scalability of traffic sources and moreover it allows for load balancing among the content sources.

The COMET CoS take into account both types of content and relate them with potential transfer capabilities offered by particular domains. We consider 3 COMET CoS:

- **Premium (PR)** CoS – it offers guaranteed services by exploiting Service Level Specification agreements between peering domains. The guarantees define the service at the level of packet transfer characteristics, i.e., packet loss ratio, packet transfer delay metrics and guaranteed throughput. Note that for achieving absolute QoS guarantees, the domains may employ resource and admission control functions.
  Depending on the type of the content, live or pre-recorded, we may differentiate 2 Premium sub-CoS dedicated for each of them respectively. Premium CoS requires that all domains along the path provide the mapping to intra-domain CoSes, which fulfil the requirements.
- **Better than Best Effort (BTBE)** CoS – it offers the transfer of content over the COMET controlled paths in the network. This requires that the forwarding over a path is enforced by CAFEs. While this CoS does not introduce any kind of guarantees, it improves the ability to control the load over the paths in the network. Consequently, we enable new mechanism for traffic engineering that can be used for relative differentiation of the service. This service is intended for media-delivery applications, which traffic may be transferred without QoS guarantees, e.g., P2P based real-time applications. This CoS may be used also for delivery of live or pre-recorded content, but in this case without QoS guarantees. This may happen when domain cannot provide Premium CoS for particular content delivery.

- **Best Effort (BE)** CoS – it offers the transfer of content in best effort way using existing routing paths in the network.

Table 2 summarizes main features of considered COMET CoS.

	Best Effort	**Better than Best Effort**	**Premium CoS**
Requires CAFE	No	Yes	Yes
Ability to use COMET routing awareness information	No	Yes	Yes
Relative QoS (with traffic engineering)	No	Yes	Yes
Absolute QoS (with admission control)	No	No	Yes
Specialization for live and pre-recorded content	No	No	Yes
Multicast feature	No	Optional	Optional
Traffic description	None	Optional	Double token bucket

Table 2: Characteristics of COMET CoS

4.3 Content delivery in different network environments

4.3.1 Best effort network (Current Internet)

Current Internet offers only the best effort service, which does not provide QoS capabilities. As a consequence, the COMET system deployed over current Internet may support only two types of content delivery paths: BE and BTBE paths. The BE paths follow the paths available in the Internet, so once the server is selected, the content is delivered in the same way as in the current Internet. On the other hand, the BTBE paths are engineered and configured by the COMET system. Thanks to traffic handling mechanisms in CAFEs and traffic engineering algorithms in PMF, the BTBE paths may improve the content delivery when comparing with BE paths.

Obviously, content delivery in the current Internet can benefit from other features offered by COMET, e.g. anycast (server selection taking into account the server load).

4.3.2 Multi-service network environments

Given the existing paradigms in providing **multi-service network** environments, such as Differentiated Services (DiffServ) in packet forwarding, as well as new routing and path selection techniques for enabling service differentiation, we have investigated how the COMET content delivery platform can be gracefully built on top of these existing infrastructures, if necessary with lightweight adaptations or extensions. In addition to enabling service differentiation, it is also essential to consider overall network resource optimisation objectives in order to achieve end-to-end multi-service aware content delivery with cost-efficient network support. According to the common practice, Service Level Specifications (SLSs) are needed between the customer side and the network side which is responsible for specifying QoS parameters (such as delay and packet loss bounds). In case of end-to-end QoS across multiple autonomous networks, provider-level SLSs can be also established in order to "bind" the QoS capability of individual ISP networks. The outcome of such provider SLSs will also be used to drive the underlying (inter-domain) routing configuration for QoS enforcement. As far as the COMET paradigm is concerned, such operations

are performed in an offline manner. When the content delivery services have been activated, the pre-provisioned paths (possibly multiple end-to-end paths between each source-destination network pair, each corresponding to a specific QoS level) can be dynamically used for content delivery according to dynamic network conditions.

In section 4.2 we have defined three types of Class of Services (CoS) in the context of the COMET based approach, namely *Premium (PR), Better-than-best-effort (BTBE) and Best effort (BE)*. Now the key issue is how these COMET-defined CoS can be efficiently mapped onto the underlying differentiated network capabilities. In the data forwarding plane, it is natural to use dedicated DiffServ code points (DSCP) to enforce specific packet forwarding treatments inside the network, as is the common practice today. It is important to note that it is up to individual ISP's options to define the mapping between DSCP values and provisioned QoS classes inside their own networks, although the identification of the COMET CoSs should be globally unique, i.e. recognised by all COMET-participating ISPs. As such, another level of code point mapping is necessary when COMET media flows are being delivered across network boundaries – the globally recognised COMET code point carried by the content packets should be specifically marked to the locally understandable DSCP values when they are injected into the next hop domain. As far as routing is concerned, as indicated previously, each COMET CoS is supported by dedicated offline path provisioning processes in order to enable service differentiation. Even within one single COMET CoS, it is also possible to provision multiple equivalent content delivery paths so that content packets can be adaptively transmitted across multiple paths, depending on the path quality/conditions in dynamic environments. Fundamentally, the offline engineering of the content delivery capability across multiple COMET CoSs is performed individually CoS by CoS without interference with each other. Certainly, more advanced approaches can be envisioned for more advanced usage of network resources, for instance bandwidth sharing or path sharing between different COMET CoSs. Such features will be potentially investigated in the project at a later stage.

The handling of the packets by CAFEs allows us to map the COMET CoSs into different technologies used for content delivery. In Table 3, Table 4, and Table 5, we present the exemplary mappings for **selected network technologies**.

COMET CoS	**Priority Code Point (PCP) value**	**Notes**
Premium	3 (Critical Applications) 4 (Video)	1) Live content should use PCP value 4; 2) Pre-recorded content should use PCP value 3; 3) Resource and admission control function should be applied before accepting the content delivery.
Better than Best Effort	2 (Excellent Effort)	No SLA type resource assurance – other traffic control mechanisms may apply. This service is intended for media-delivery applications which do not require strict QoS guarantees, e.g., P2P.
Best Effort	0 (Best Effort)	No resource assurance

Table 3: Mapping of COMET CoSs into Ethernet priorities [54]

COMET CoS	Service Class name	DSCP name and value	Notes
Premium	Broadcast Video	CS3 (011000)	1) Live content should use CS3; 2) Pre-recorded content should use AF3x; 3) Resource and admission control function should be applied before accepting the content delivery.
	Multimedia Streaming	AF31 (011010) AF32 (011100) AF33 (011110)	
Better than Best Effort	High-throughput Data	AF11 (001010) AF12 (001100) AF13 (001110)	No SLA type resource assurance – other traffic control mechanisms may apply. This service is intended for media-delivery applications which do not require strict QoS guarantees, e.g. P2P.
Best Effort	Standard	DF (000000)	No resource assurance

Table 4 Mapping of COMET CoSs into DiffServ Service Classes [57]

COMET CoS	Treatment Aggregate (TA)	DSCP name and value	MPLS EXP value	Notes
Premium	Real Time	CS3 (011000)	100	1) Live content should use Real Time TA; 2) Pre-recorded content should use Assured Elastic TA; 3) Resource and admission control function should be applied before accepting the content delivery.
	Assured Elastic	AF31 (011010) AF32 (011100) AF33 (011110)	010 011	
Better than Best Effort	Assured Elastic	AF11 (001010) AF12 (001100) AF13 (001110)	010 011	1) No SLA type resource assurance; 2) Domain supporting both PR and BTBE CoSes must guarantee no overlaps between 010 and 011 MPLS EXP values.
Best Effort	Elastic	DF	000	No resource assurance

Table 5 Mapping of COMET CoSs into DiffServ Treatment Aggregates with MPLS option [58]

5 Mechanisms and algorithms for routing awareness

5.1 Introduction

The objective of routing awareness process is to find content delivery paths in multi-domain network. These paths should support COMET Classes of Service (CoS) basing on the transfer capabilities offered by particular domains. The routing awareness is an off-line process performed in long time scale. It reacts to changes in inter-domain network reachability or re-provisioning of particular domains. We assume that routing awareness process is performed by specialized Routing Awareness Entities (RAE) that should be implemented in each domain. In principle, the functionality of RAE is similar to the inter-domain routing entity, e.g., external BGP speaker [55] . It exchanges Network Layer Reachability Information (NLRI) with other RAEs located in peering domains to build inter-domain routing paths. However, the RAE differs from BGP speaker in two main features. First, the RAE is responsible for building content delivery paths that support COMET CoSs. Therefore, each RAE should perform appropriate mapping between COMET CoSs and intra-domain CoS offered within domains. For that purpose, each domain operator should provide information about offered intra-domain CoSs jointly with values of QoS parameters that are assured by particular intra-domain CoS between any ingress and egress points of the domain, see Figure 3. These values will be propagated between RAEs as a part of the NLRI information characterising the path's properties. Therefore, the QoS parameters should be valid for long term and cannot depend on the carried traffic. Typically, these values should come from provisioning of the domain.

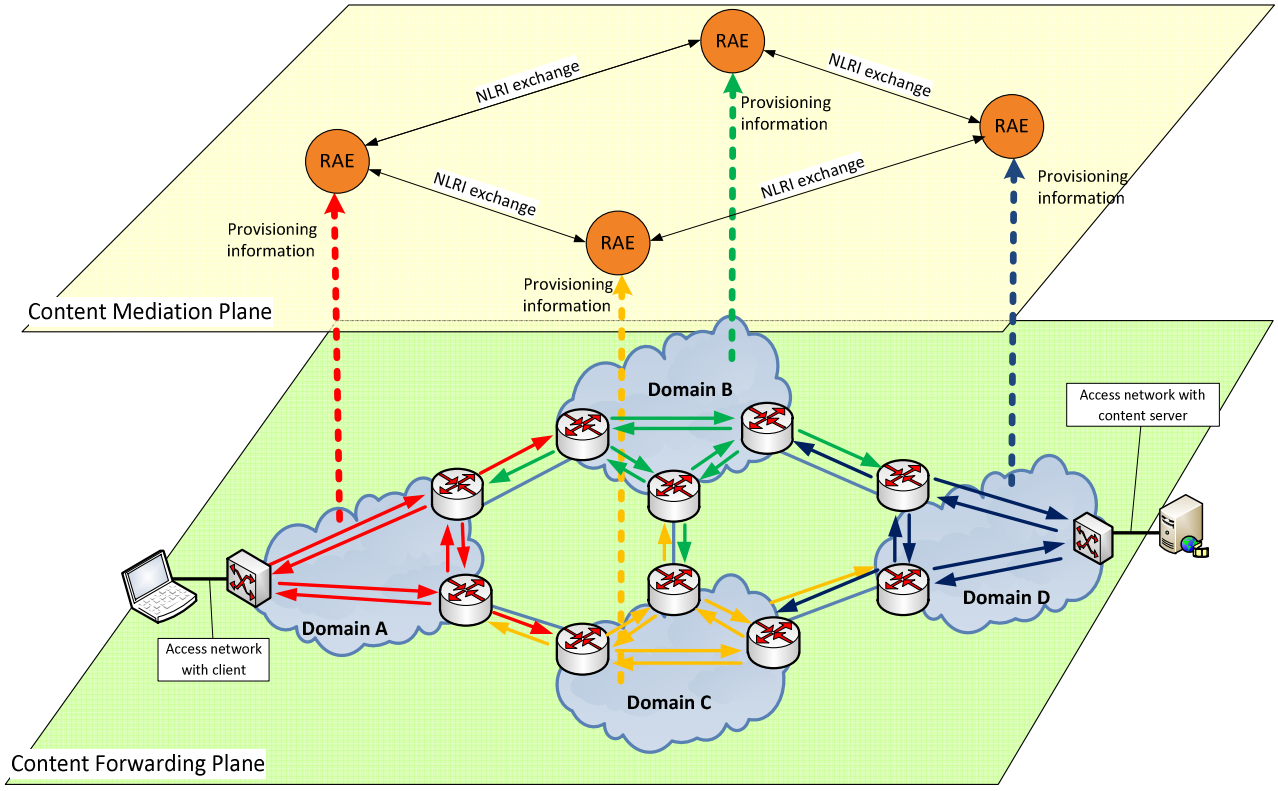

Figure 3: Routing awareness and provisioning.

Second, the RAEs propagate information about a number of alternative preferred paths. This multipath feature is a fundamental difference comparing to the BGP, which propagates only single preferred path usually the shortest one. It allows the Content Mediation Function to optimise content delivery by applying smart traffic engineering algorithms, load balancing in the network as

well as improving reliability of content delivery. In addition, the information about alternative paths jointly with the flexibility of path configuration provided by CAFEs allows COMET system to choose different path for each content consumption request. We believe that the multipath routing performed by RAE will allow to increase the number of satisfied content requests comparing to the single path routing exploited in the current Internet.

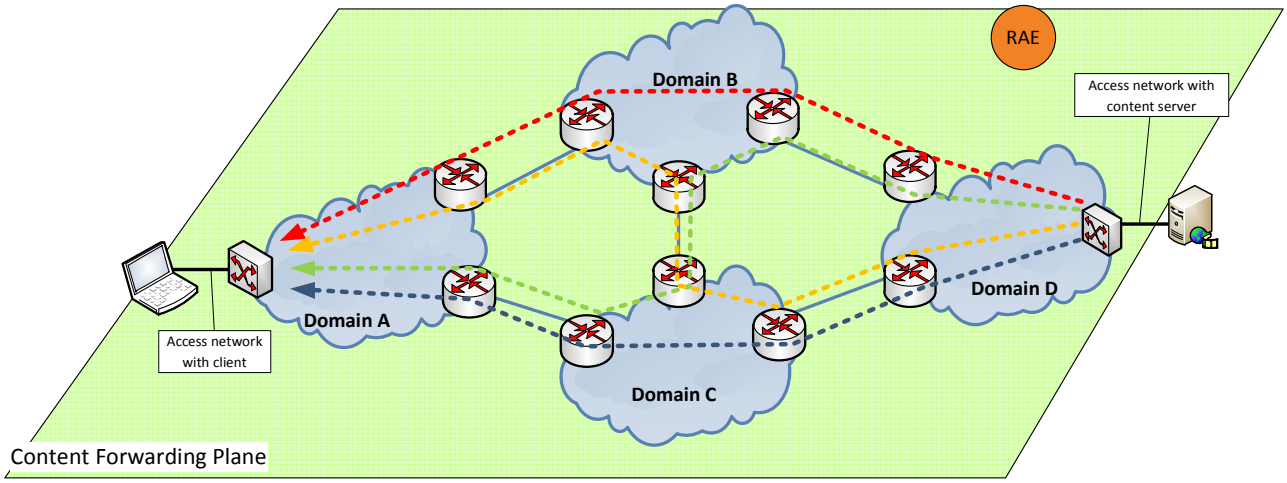

Figure 4: The example of multi-domain paths achieved with RAE.

After the routing awareness process is finished, the RAE provides Content Mediation Function (CMF) information about discovered paths and their properties. 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.

Taking into account the above requirements, we propose that routing awareness should be performed by multi-path, multi-criteria (QoS) inter-domain routing protocol. Although, there is a number of proposals that are investigated in the area of multipath routing and multi-criteria routing (see sections 3.1 and 3.2), we decide to define a generic protocol which merges both multipath and multi-criteria features.

This protocol will express the next hop in terms of next AS (the first AS in the list of ASs) and not in terms of next IPv4 or IPv6 address. That is, the routing awareness process is decoupled from the forwarding plane and the addressing scheme. In this way, the RAE can support both IPv4 and IPv6 or any other new addressing method.

It must be also noticed that the RAE does not configure the forwarding plane; it only provides information about potential paths to the CMF (more specifically, to the CME or CRME entities), which will perform the path configuration process to instruct CAFEs appropriately. The interaction between RAEs and CMEs/CRMEs has not been detailed in this deliverable. We foresee this interaction to be similar to the one between RAEs but at an intra-domain level. The description of this interaction will be included in D4.2.

5.2 Specification of routing awareness process

In this section we present first specification of proposed multi-path, multi-criteria routing protocol that we design for performing routing awareness. We design our protocol as an enhanced version of currently used BGP-4 protocol [55] taking into account achievements of EuQoS and AGAVE

projects related to QoS extensions of BGP [16] [18] [19] as well as recent proposals for multipath extensions [56] .

5.2.1 Messages

Since RAEs should exchange the NLRI information between peering domains, we define two types of messages that are: UPDATE and WITHDRAW. The UPDATE message is used to advertise: (1) new prefixes, when they appear within domain, (2) new paths going towards already known prefixes, as well as (3) updates of path's properties, e.g. values of QoS parameters after domain re-provisioning. In addition, the UPDATE message may be also used for implicit path removal.

We propose the following structure of UPDATE message:

UPDATE{
network prefix,
COMET CoS,
List of PATHs {...}.
}

Where a path has the following structure:

PATH {
List of domains {...}, // e.g., the ordered list of AS numbers,
metric [QoS parameters, length,...] ...
}

We assume that the first domain in the *list of domains* determines the next hop. Moreover, the *metric* is a vector of parameters describing path's properties (see the requirements presented above).

On the other hand, the WITHDRAW message is used to remove given prefix or a set of paths which becomes no longer available. The structure of WITHDRAW message is the following:

WITHDRAW{
network prefix
COMET CoS
List of PATHs {...}
}
PATH {
List of domains {...}, e.g., the list of AS numbers
}

We use WITHDRAW message in two cases:

1. to remove a given network prefix. In this case, the source domain sends WITHDRAW message to all peering domains,
2. to remove a set of paths belonging to a given branch of destination sink tree. In this case, the domain at the beginning of the branch sends WITHDRAW message uphill along the branch.

5.2.2 Basic operations

Each RAE maintains two tables of paths that are named *Known Paths Table* (KPT) and *Preferred Paths Table* (PPT).The former stores information about all paths that has been advertised by peering domains, while the latter stores the paths preferred by the domain. The RAE uses a path ranking algorithm to evaluate the known paths and to select a set of preferred paths. In multi-path approach, the path ranking algorithm selects a number of preferred paths going towards the same network prefix. The proposed path ranking algorithm is presented in section 5.2.3.

Let's focus now on basic RAE operations, that is, the actions performed by the RAE after reception of UPDATE and WITHDRAW messages:

1. When the RAE receives UPDATE message from peering domain:
    a. It performs sanity check of UPDATE message to remove paths that contain its AS number. This process allows us to avoid routing loops.
    b. It updates properties of paths received in UPDATE message, i.e., path length, list of domains, QoS parameters, with values corresponding to its domain and then it stores paths in KPT. It is worth to mention that this process may also remove the previously advertised paths, when received UPDATE has less paths then the previous one.
    c. It invokes the path ranking algorithm to select the new preferred paths and prepare new *PPT*.
    d. If *new PPT* is identical to the old one, the process is finalized. Otherwise, the RAE updates PPT and sends UPDATE message with the currently preferred paths to its peers. Then, the process is finalized.
2. When the RAE receives WITHDRAW message from peering domain:
    a. It removes paths from the KPT.
    b. It invokes the path ranking algorithm to prepare new *PPT*.
    c. If *new PPT* is identical to the old one, the process is finalized. Otherwise, the RAE updates PPT and sends WITHDRAW message to its peers with the set of paths, which should be removed. Then, the process is finalized.

5.2.3 Path ranking algorithm

The objective of the path ranking algorithm is to select the set of preferred paths from the list of known paths stored in the KPT. Note that in inter-domain network, there may exist a number of feasible paths that meet requirements of COMET CoSs and have similar properties, e.g., similar length of the paths, offering similar QoS parameters. The path ranking algorithm evaluates the feasible paths using a specific cost function, *cost_f(.)*. This function ranks the paths basing on (1) vector of QoS parameters characterizing the particular path, (2) set of constraints related to particular parameter as well as (3) specific policies of the domain. The set of preferred paths will be advertised to the peering RAE entities. These paths may be similar from the point of view of the assumed cost function but they could differ essentially in other criteria, e.g. the number of disjointed domains. We argue that the set of preferred paths should be limited to 2-5 paths in order to limit the path table size.

Note that the effectiveness of the routing awareness process is strongly influenced by choosing the appropriate path ranking algorithm, see [56] Therefore, in COMET, we will consider different proposals starting from the cost functions considered in multi-criteria routing (presented in section 3.2) and multi-criteria decision theory and we will evaluate them by simulations in order to select the most effective one.

The proposed path ranking algorithm follows these steps:

1. It splits the KPT into sets of paths going towards a given prefix or domain and belonging to the same COMET CoS,
2. For each set, the algorithm removes paths that are dominated by any remaining path.

3. Now, the algorithm ranks the paths in each set using the cost function.
4. Finally, the algorithm prepares the subset of preferred paths based on ranking and other optimisation criteria, like load balancing. This step finalises the algorithm.

5.3 Consideration on deployment

This section provides initial considerations for deployment of RAEs. It must be noticed that these considerations are preliminary and will be subject of review in deliverable D4.2 expected for Month 21.

The routing awareness process performed by RAEs is similar to the current inter-domain routing performed by BGP speakers in the Internet. However, there are some differences which have direct implications in the deployment. Next figure shows a typical deployment of BGP speakers in the current Internet. For the sake of clarity, we follow here a classical separation of Tier-1, Tier-2 and Tier-3 providers without a loss of generality, although we are aware that the current interconnection does not follow such a clear separation in tiers.

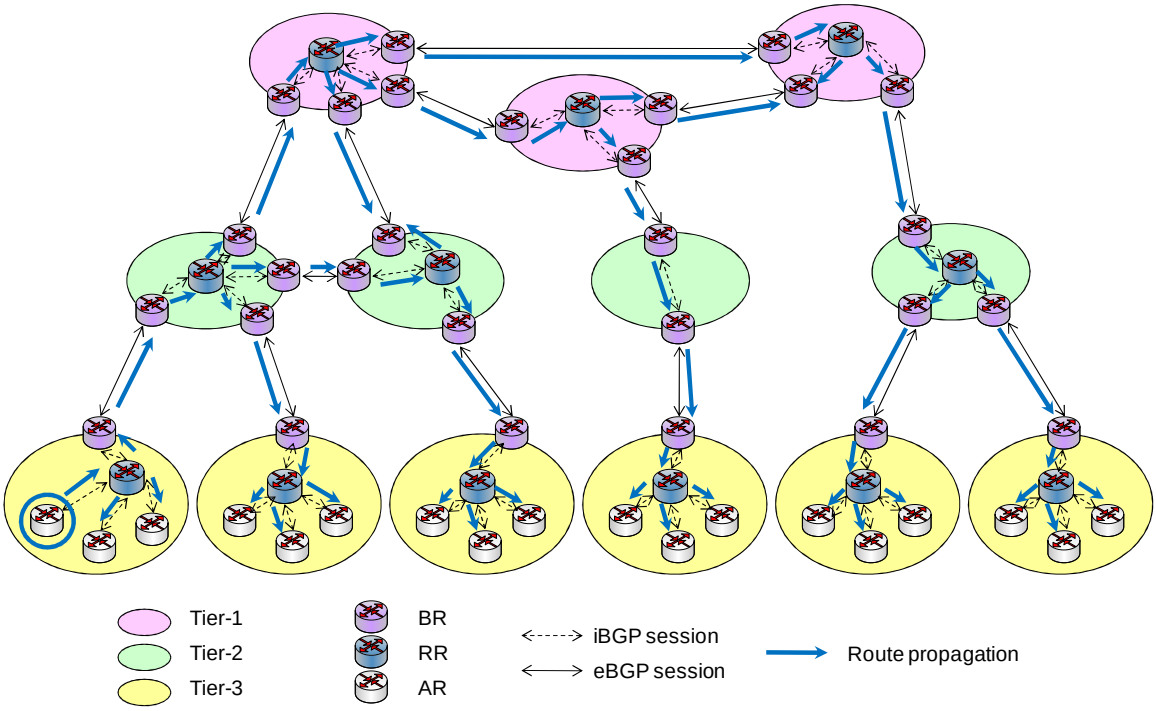

Figure 5: Inter-domain routing in the current Internet

We distinguish three different elements involved in the propagation of routes in the current Internet:

- Border Routers (BR), which are in charge of the actual interconnection between domains, exchanging NLRI information with peering domains. Border Routers have eBGP (external BGP) sessions with the Border Routers in the peering domains, receiving route advertisements (updates and withdrawals) from them, and propagating route changes uphill to the peering domains as depicted in the figure. These Border Routers are also responsible for the packet forwarding.
- Route Reflectors (RR) are used to propagate routes inside one domain. Border Routers inside one domain need to exchange NLRI information with other Border Routers in the same domain. For that purpose, they establish iBGP (internal BGP) sessions, similar to eBGP sessions but restricted to one domain (restricted route propagation towards peering domains). When the number of iBGP speakers is large, a Route Reflector is used, so that

iBGP sessions are established with the Route Reflector, thus improving scalability by reducing the number of BGP sessions in each Border Router. Besides, Route Reflectors usually apply policies to prioritize some routes over others according to operator's policies. Route Reflectors are not intended to perform packet forwarding.

- Access Routers (AR) are responsible of providing connectivity to a number of Internet end users. New IP prefixes are added to Access Routers and must be propagated to the Border Routers of the domain, so that they can summarize and advertise them to the peering domains and therefore to the whole Internet. Interior Gateway Protocols (IGP) can be used for the route propagation inside one domain, but it is very common to use iBGP for this purpose. Access Routers have iBGP sessions with Route Reflectors, which will propagate the route changes to the Border Routers in the same domain. On the other hand, new routes arriving to a Border Router of an end domain can be propagated to the Access Routers in order for them to know to which Border Router forward the packets. Again, Route Reflectors are used to reduce the number of iBGP sessions.

In the current Internet, forwarding and routing are coupled in the Border Routers so that they are in charge of the exchange of routes as well as the packet forwarding. However, in COMET the forwarding and routing processes are decoupled so that RAEs are in charge of the routing awareness process (exchange of NLRIs), while CAFEs are in charge of forwarding.

Moreover, since the routing awareness protocol expresses next hop in terms of next AS and not in terms of next router, only one inter-domain speaker will be needed (or two in order to provide resilience) for the exchange of routing information, whereas in the current Internet all Border Routers need to participate in the exchange of NLRI with eBGP[1]. RAEs are expected to interact with:

- Other RAEs in peering domains to exchange routing information (including QoS path characteristics and multiple paths) with other RAEs in COMET peering domains.
- CMEs/CRMEs in the same domain, in order to propagate the previous information. As mentioned previously, this interaction has not been covered in the deliverable, but we foresee it to be similar to the one between RAEs but at an intra-domain level. The description of this interaction will be included in D4.2.

Besides, RAEs could interface to local BGP Route Reflectors, in order to collect the BE routing information. This interface will be based on iBGP.

Regarding deployment of RAEs, it can be different depending on the specific approach for content resolution.

In the decoupled approach, which follows the Content Record-based resolution (see D3.1), the decision process is performed by the client's CME, which must be aware of any route changes. Therefore, route changes (updates and withdrawals) must be propagated to all CMEs (see Figure 6). On the other hand, not all domains need to deploy RAEs.

[1] In the current Internet, it is not strictly necessary that BRs speak eBGP. Instead, RRs or BGP routers different from BRs can speak eBGP with other domains. This is done by configuring "ebgp-multihop" field accordingly. However, this is not a common practice since it makes the configuration difficult and can cause instabilities.

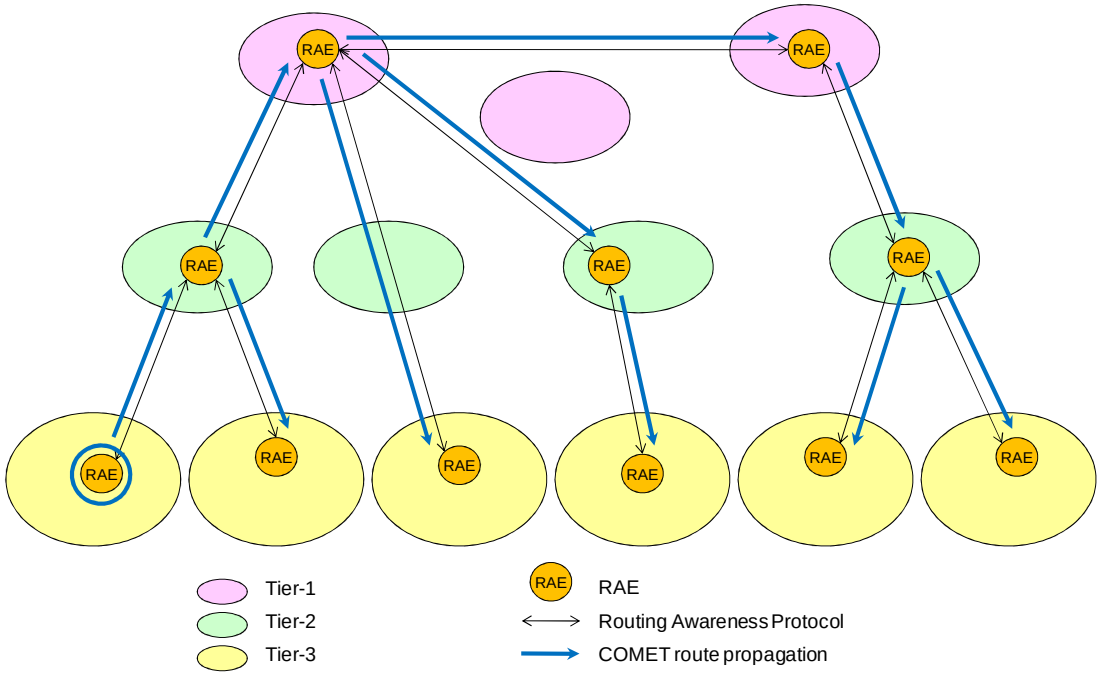

Figure 6: Deployment of RAEs in the decoupled approach

In the coupled approach, where content resolution is performed on a hop by hop basis, the decision process is performed in each CRME along the resolution path based on the information propagated during the content publication. In practice, this means that the decision is taken by upper tiers. Therefore, route changes should not reach all CRMEs, but only those in upper tiers (see Figure 7). On the other hand, due to the hop-by-hop nature of content resolution, it seems appropriate a priori that all domains deploy RAEs.

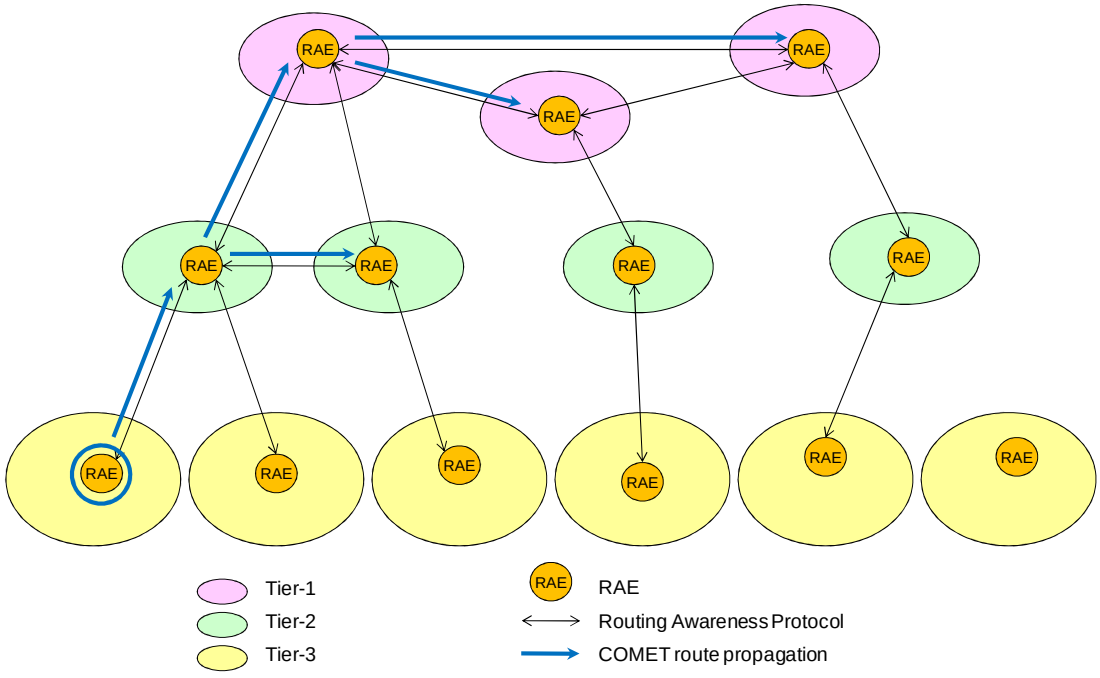

Figure 7: Deployment of RAEs in the coupled approach

6 Mechanisms and algorithms for basic delivery process

6.1 Overview

In the CFP, the content data should be delivered from a selected source to end users who requested the content via one or multiple selected paths. Within the COMET architecture, two content delivery processes have been proposed: stateless and state-based content delivery.

One of the approaches, the stateless content delivery, assumes that CAFEs maintain only the neighbourhood (local) information, i.e., how to forward packet to the next CAFE. All information about the selected path is stored in a COMET header attached to the original packet containing content payload. Since CAFEs store only local forwarding information, the packet must contain full information of the end-to-end path with a chain of CAFEs across domains. That specification of the path is configured in the first CAFE during the path configuration process.

The other "in-band" approach is to use state-based content delivery coupled with the hop-by-hop content resolution procedure described in section 5.2 of D3.1. It proposes to make the CMP and the CFP collaborate on establishing and maintain the delivery path.

6.2 Stateless content delivery process

6.2.1 Introduction

As shortly described previously, in stateless approach we assume that CAFEs maintain only the neighbourhood (local) information, i.e., how to forward packet to the next CAFE. All information about the selected path is stored in a COMET header attached to the original packet containing bits of content. We follow the idea of DiffServ architecture and we apply filtering and classification mechanisms at the edge of the network – at the content source side.

More specifically, after the decision process, a content server and path have been selected. The path is characterized by a list of ASs that will be traversed. This list of ASs is translated into a full specification of the path in a global scale: a list of CAFEs, together with a list of forwarding rule identifiers which will be used for content forwarding. Forwarding rule identifiers are stored in the COMET header, so that CAFEs use them to perform the packet forwarding.

We distinguish three different phases in the stateless content delivery, which run after the decision process:

- *Path provisioning (semi-long time scale).* "Tunnels" are created between neighbouring CAFEs and a label (forwarding rule identifier) is locally associated in each CAFE to the provisioned "tunnel". Although the "tunnel" is associated to a specific technology (IPv4, IPv6, MPLS, etc.), the label is technology-agnostic. When new routing path is going to be used, a path provisioning process is run. The purpose of this process is twofold:
    1) it confirms the availability of the path across all domains (possibly with resources)
    2) it gathers the forwarding information, namely forwarding rule identifiers.
- *Path configuration (on a per-consumption basis).* Once the path has been provisioned, the first CAFE must be instructed about the set of tunnels to be used, that is, the first CAFE in the path has to know the rules to perform filtering and flow classification of content packets, encapsulating them with a COMET header containing the list of forwarding rule identifiers. In order to instruct the first CAFE, the client CME must send a request of path configuration to the server CME. This process has been detailed in D3.1, but for the sake of completeness, we have also detailed it in D4.1.
- *Content forwarding (packet processing time scale).* It consists mainly in forwarding pieces of content with help of forwarding rule identifiers in the COMET header. The first CAFE in

the path will insert a header to each content packet containing the complete list of forwarding rule identifiers of the path. Next CAFEs only have to read the corresponding forwarding rule identifier to perform the content forwarding. Thus, this approach is stateless since it does not require CAFEs to keep state of the whole path.

6.2.2 Path provisioning: collecting forwarding information for paths

The collecting of forwarding information for selected paths is performed at semi-long time scale, i.e., when new routing path is used. The result of collecting is valid for a long time (hours, days) and may be used by multiple content deliveries as long as the selected path is the same.

The input of this action covers the *path information* expressed as list of domains (with network prefix) and a name (or identifier) of COMET Class of Service. Optionally, it may also contain service level parameters, which allows for allocating resources to paths in long time scale.

The output of this action is a *forwarding information* expressed as a list of forwarding rule identifiers. This list will be attached to each packet of content, which is transferred along the path.

The use of forwarding rules introduces following assumptions for the configuration and provisioning of the COMET:

- Each domain knows its peering domains (AS number or COMET number). Notice that peering in COMET may span across transit domains. (this is a topology assumption)
- Each domain knows identification of CAFEs (addresses) sending data to own CAFEs; this is a security feature (whitelisting inside own CAFEs). (this is an inbound traffic assumption)
- Domain knows how to forward packet from its CAFE to any other CAFE in peering domains, i.e., there is association: ($CAFE_x$, $CAFE_y$) → forwarding rule FR_{xy}. Forwarding rule defines details required for forwarding, e.g., protocol, technique (IP/IPv6/MPLS/other), source, destination and technique. The forwarding rule should be a result of cooperation between peering domains, but its scope is local to given domain. Moreover, each forwarding rule FR_{xy} can be uniquely assigned to the forwarding rule identifier $FRid_z$. The uniqueness constraint relates to given CAFE in given domain. (this is an outbound traffic assumption)

Those assumptions are demonstrated for a simple example in the Figure 8. We assume that domain B is peering with domains A and C. Domain B has 3 CAFEs, named B1, B2 and B3. During the provisioning,, the network administrators of the domains establish the following forwarding rules:

- Domain A allows the domain B to use CAFE A1 for traffic arriving from CAFEs B1 and B2, i.e., A1 will forward only the traffic from B1 and B3. Furthermore, the technology of forwarding is also determined at this step, e.g., it could be MPLS forwarding.
- Similar setting is also established for domain B and domain C. In this case, CAFEs B2 and B3 are allowed to transit traffic through C1.
- Note that only the preselected pairs of CAFEs may communicate. Traffic originating from other locations in the network will be dropped, e.g., traffic originating from B1 will not be handled in A2 or C1.

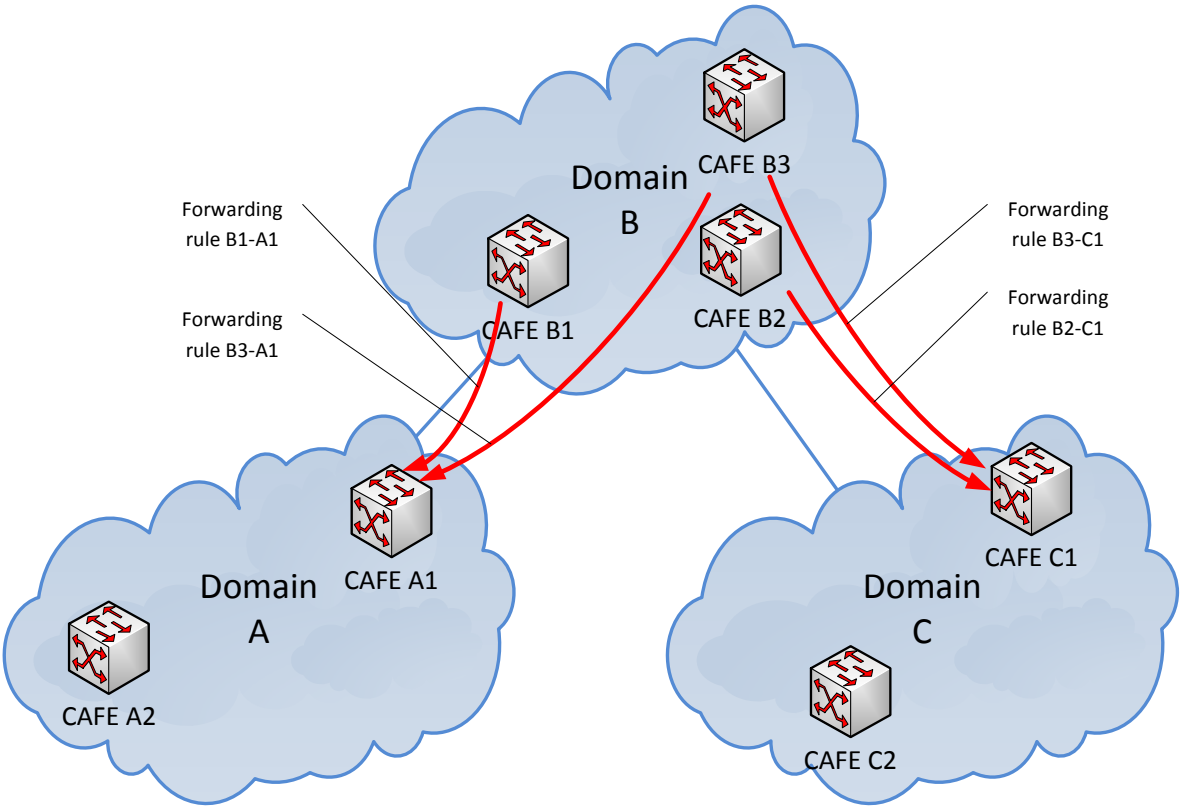

Figure 8: Exemplary configuration of domains for forwarding

The collecting is initiated by the CME in server's domain; it follows the direction of traffic in the content delivery. This is achieved in a domain-by-domain way using two messages: *collect_forwarding_rule_request(.)* and *collect_forwarding_rule_response(.)*.

The general algorithm for handling the *collect_forwarding_rule_request(.)* message in a given domain consists of the following steps:

1. Check policies. If result is negative send *collect_forwarding_rule_response(.)* message to the previous domain (on the path) with negative result.
2. Assign a local CAFE that will handle the requested path. The selected CAFE must accept traffic from previous CAFE from the list. Add this CAFE to the CAFE list in message.
3. If a given domain is the last domain, send the *collect_forwarding_rule_response(.)* message to the previous domain (on the path) and finish processing.
4. Send the *collect_ forwarding_rule_request(.)* message to the next domain (on the path).

The general algorithm for handling the *collect_ forwarding_rule_response(.)* message in a given domain consists of following steps:

1. Check policies. If result is negative, send *collect_ forwarding_rule_response(.)* message to the previous domain (on the path) with negative result.
2. Take the pair of last two elements from the CAFE list: {$CAFE_{n-1}$, $CAFE_n$}. Remove last element from the CAFE list.
3. Map a forwarding rule FR_{n-1} for the pair {$CAFE_{n-1}$, $CAFE_n$}. Append the identifier of forwarding rule $FRid_{n-1}$ to the list of forwarding rule identifiers.
4. If the given domain is the first (initiating) domain, the list of forwarding rule identifiers is complete. Finish the processing.
5. Send the *collect_ forwarding_rule_response(.)* message to the precious domain (on the path).

Figure 9 shows an exemplary message sequence chart for process of collecting the forwarding rules in path of 3 domains (numbered 1, 2 and 3). The entity responsible for message processing is the CME, although it may be initiated by other functional blocks.

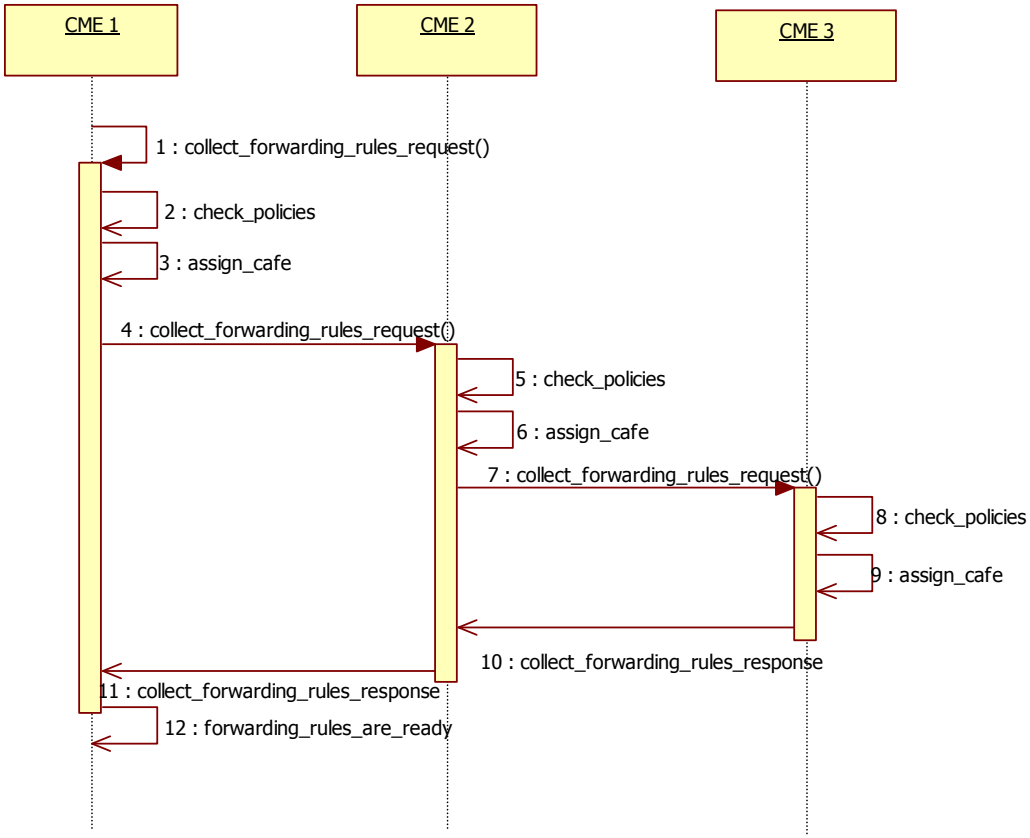

Figure 9: An exemplary message sequence chart for path configuration process.

Notice that the step named *check_policies* allows to apply different types of policies, e.g., end-to-end allocation of resources for particular paths, enforcement of business relations and silent redirection of selected path.

6.2.3 Path configuration: 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.

Two approaches for path configuration have been envisioned:

- Agnostic path configuration. Once the path has been provisioned, the first CAFE must be instructed about the set of tunnels to be used, that is, the first CAFE in the path has to know the rules to perform filtering and flow classification of content packets, encapsulating them with a COMET header containing the list of forwarding rule identifiers.
- Path configuration coupled with smart flooding path discovery. The path is configured following the reverse path of the discovered path.

The first one is independent from the specific path discovery process, so that it could be used regardless the path discovery mechanism. The second one is thought to be used only in conjunction with the smart flooding path discovery.

6.2.3.1 Agnostic path configuration

Before the agnostic path configuration starts, the following properties are known either from the Content Record obtained in the name resolution process or from the information obtained in the path discovery process:

- 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 CME 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 agnostic path configuration follows these 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.

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 optionally the "content 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 D4.2.
   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 10 presents the message sequence chart for the path configuration process.

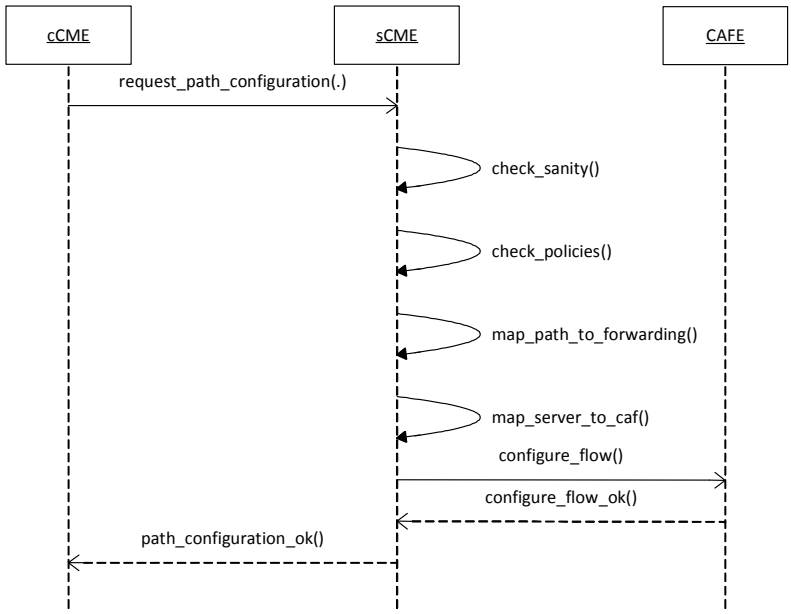

Figure 10: Agnostic path configuration message sequence chart

6.2.3.2 Path configuration upon smart flooding path discovery

As described in section 5.1.5.2 of D3.1, the *smart flooding path discovery* process is performed upon reception of a user request. The objective of the process is to calculate all the available content consumption paths from a given content server to a given content client. To achieve this, the Path Management Functional Block (PMF) of the content server's domain initiates the path discovery process and all intermediate PMFs exchange `find_path` messages for the specific content request. During this process, the CoS requirements set by the client and/or the content are considered. Additionally, the process avoids loops in discovered paths and tries to minimize the messaging overhead. The path discovery process is finalized when a timer, set in the PMF of the content client's domain, expires. Upon receiving the (multiple) `find_path` messages, the PMF of the content client's domain forwards the list of discovered paths to the client's CME. The latter one performs the *path decision* process in order to decide the most appropriate pair of Content Server and discovered path.

The next step after the path discovery and path decision processes is the *path configuration*. During the path configuration phase, messages flowing the opposite direction of that during the path discovery process are disseminated to all previously visited domains. More specifically, all the informed PMFs are informed whether the path(s) previously discovered were selected or not as the consumption path. This is accomplished via exchanging `ACK/NACK configure_path` messages, which were also introduced in D3.1. Upon receiving such messages (obviously one ACK and many NACK messages will be disseminated), each PMF updates the path discovery table.

The information enclosed in an `ACK/NACK configure_path message` includes:

- the tuple `<CN, Client CME, Content Server CME>`, which uniquely identifies the source and destination of a specific content
- the `QoS_path`, which denotes the CoS characterization of the a path
- the `hop_count`
- the `ACK/NACK` identifier, which denote whether the specific path was selected or not

In order to initialize the path configuration process, the client's PMF needs to send:

- A single `ACK configure_path` message to the previous-hop PMF in its path discovery table for the selected path and hop count
- One or more `NACK configure_path` messages to all previous-hop PMFs that belong to paths not selected.

Any intermediate PMFs between content client's PMF and content server's PMF:

- If it receives an `ACK configure_path message`, checks the matching record in its path discovery table and forwards the `ACK` message to the previous-hop PMF, defined in that record.
- If it receives a `NACK configure_path message`, finds and deletes the matching record in its path discovery table and if it is relevant, forwards the `NACK configure_path message` to the previous-hop PMF, defined in that record.

In any case, whenever the PMF sends or forwards an `ACK/NACK configure_path` message, it has to reduce by one the value of the hop count field, so that the PMF receiving the `configure_path` message could identify the matching record in its path discovery table.

The content server's PMF will certainly receive an ACK message for a specific content request, but it might also receive one or more NACK messages (in case it has more than one outgoing links, thus it has sent multiple `find_path` messages during path discovery process). In this case, it deletes all records in its path discovery table which are related to non-selected paths. In Figure 11, an example of path configuration after smart flooding path discovery is presented, where intermediate PMFs exchange `ACK/NACK configure_path` messages accordingly.

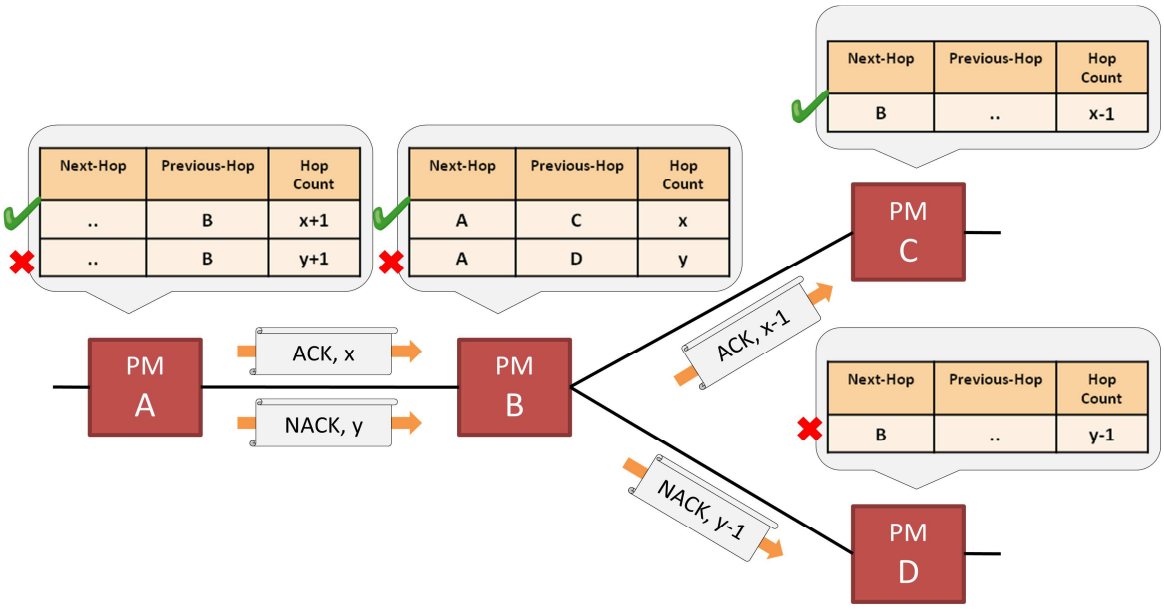

Figure 11: Message sequence chart of path configuration coupled with smart flooding path discovery

Additionally to the forwarding of ACK and NACK messages, the PMFs are also responsible to enforce the selected path to the CAFEs. Thus, upon the reception of an ACK message, the PMF needs to translate the respective entry of the discovery table to a forwarding rule and dispatch it to the CAFE.

6.2.4 Content forwarding

This process is performed at forwarding time scale (micro-/nanoseconds), i.e., it relates directly with packet handling in content aware forwarders (CAFE).

The input is twofold:

- Multi-field filter for the purpose of classifying the flow at the input of the first CAFE,
- List of forwarding rule identifiers {$FRid_1$, ...,$FRid_n$}.

The multi-field filter is only applicable for IP traffic. In case when non-IP protocols are used at the network layer, the classifying filter must be adapted.

The output is the forwarding on the enforced path (similar to virtual connection).

The processing of packet can be divided into 3 phases:

1. Classification of packets and encapsulation in COMET header in the first CAFE,
2. Processing of COMET header in each CAFE along the path,
3. Processing of packets in the last CAFE along the path.

The first phase requires configuration for each consumption, while configuration for second phase operates in long time scale. We assume that each CAFE knows its own association between identifier $FRid_k$ and corresponding forwarding rule.

The operations in the first phase:

1. CAFE receives a request for classifying the flow (consumption) with a multi-field filter. The request contains the list of forwarding rule identifiers. Optionally, it may contain traffic control rules according to needs, e.g., policing or shaping.
2. Classified packets are encapsulated in COMET header. The header includes:

3. List of forwarding rule identifiers {FRid1, ...,FRidn},
4. Other optional information
5. The classification rule is removed when there are no packets matching the rule for a given timeout value (rule is refreshed by content transfer).

The operations in the second phase:

1. CAFE receives a packet with COMET header.
2. CAFE confirms that packet arrives from whitelisted CAFEs, otherwise it drops the packet.
3. CAFE removes the current encapsulation from the packet (from previous CAFE).
4. If list of forwarding rule identifiers is empty, then go to next step. Remove the first identifier FRidk (COMET header modification) and map it into a forwarding rule. Apply the forwarding rule to the packet, e.g., encapsulation with MPLS, GRE or other protocol.
5. Forward the packet according to its current structure.

The operations in the third phase:

1. The list of forwarding rules is empty in the step number 4 of the second phase.
2. The local policies are applied, e.g., address translation in case of hiding the server's network address.
3. Note that the format of the COMET header is still under consideration. We expect that each forwarding rule will be encoded in 16 bit field (Tier 1 domains in the Internet establish peering with few thousand other domains). The encoding of the third phase policies is not yet decided as it requires a management of unique identifiers of content delivery streams. More details will be provided in the D4.2.

6.3 State-based content delivery process

6.3.1 Introduction

State-based content delivery process together with the proposed coupled content resolution scheme in D3.1 forms a hop-by-hop approach. Within this approach, the content delivery is coupled with content resolution procedure in a hop-by-hop manner. By "hop" we mean domain in this section. So basically every domain on the content delivery path will hold some sort of content state associated with each content delivery session. This is specifically done through the state configuration of the content mediation function (CMF) to content aware forwarding function (CAFF).During the content resolution procedure, requests sent by content clients are processed at mediation plane and are forwarded from the consumers' domains to the ones where content servers with the requested resources are hosted. The traces of these requests through different domains are used to provide paths for content deliveries.

The following sections describe how the content delivery path is established and optimized.

6.3.2 Content Delivery Path Configuration During Content Resolution

According to our design, content delivery paths are enforced in a receiver-driven multicast manner that needs state maintenance based on content identifiers. As described in D3.1, content consumption requests from clients are resolved through a sequence of content resolution and mediation elements (CRMEs) residing in individual domains according to either the business relationships between ISPs (in *wildcard* and *filtering* modes) or the BGP reachability information on the scoped source prefix (in *scoping* mode). In both cases, once a CRME has passed the content consumption request to its next hop counterpart in the neighboring domain, its content mediation function (CMF) needs to configure the local content aware forwarding elements (CAFEs) that will be involved in the delivery of content flows back from the potential server. Specifically, once a CRME receives a content consumption request from its counterpart in the previous hop domain

and forwards it towards the next hop CRME, it needs to correspondingly install the content ID state at the local egress and ingress border CAFEs connecting to the two neighboring domains[2]. The determination of ingress/egress CAFEs for each content consumption request is purely based on the BGP reachability information across networks. Within each domain, the communication between the non-physically connected ingress and egress CAFEs can be achieved either by establishing intra-domain tunnels that traverse non-content-aware core IP routers, or natively through the content-centric network routing protocols. As a result, the actual domain-level content delivery path is effectively the reverse path followed by the delivery of the original content consumption request. It is worth mentioning that CRMEs do not directly constitute the content delivery paths, in which case the configuration interaction between the CRME (specifically the content mediation function, CMF) and local ingress/egress CAFEs is necessary.

Let's take Figure 12 as an example for illustration. We assume that currently content client**C1** (attached to domain 2.1/16) is consuming a live streaming content **X** from server **S** (attached to domain 1.2.1/24). The content delivery path traverses a sequence of intermediate domains, and each of the corresponding ingress/egress CAFEs is associated with a star that indicates the content state maintained for content delivery. As mentioned previously, such content states are configured by the local CRMEs (the CMF function) during the content resolution phase. Now content client**C2** (attached domain 1.1/16) issues a consumption request for the same content. Upon receiving the content consumption request, the local CRME forwards it to its provider counterpart in domain 1/8 (uphill), as it is not aware of the content source location. Since the CRME in 1/8 knows that content flow for **X** is being injected into the local network via the originally configured ingress CAFE 1.0.0.2, it then updates its *outgoing next-hop CAFE list* by adding a new egress 1.0.0.3 leading towards content client **C2**. As a result a new branch is established from CAFE1.0.0.2 which is responsible for delivering the content back to the new client **C2** (the dash line), but without any further content resolution process.

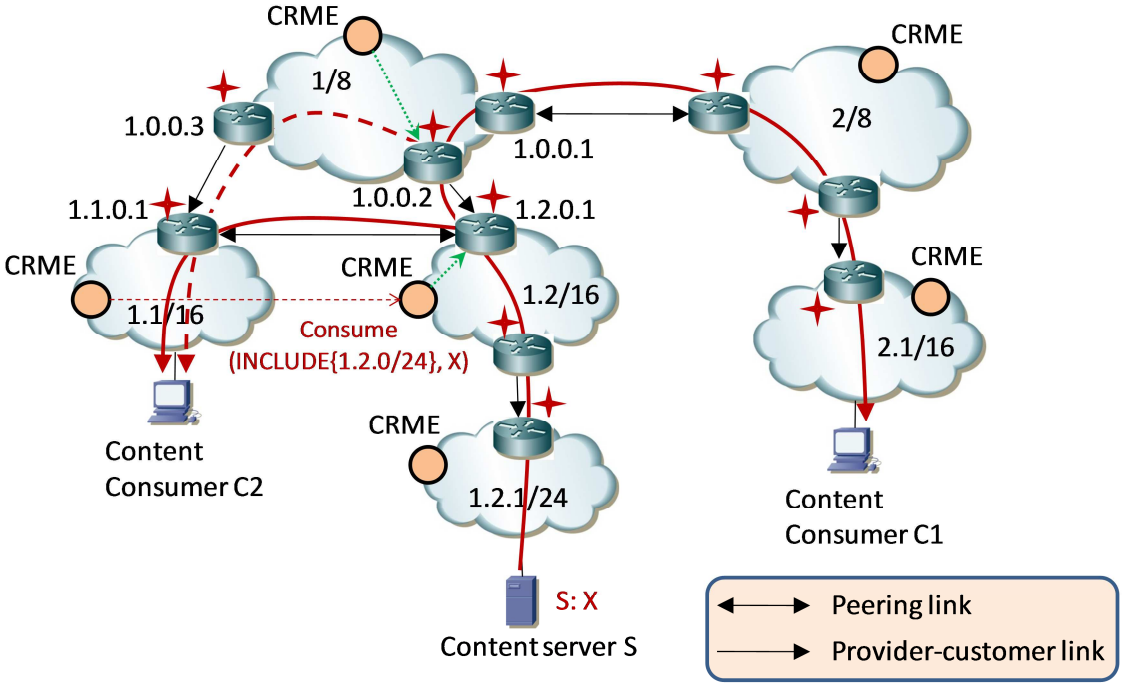

Figure 12: Multicast-based content delivery process

[2]In case of a failed content resolution, content states temporally maintained at CAFs can be either timed-out or explicitly torn down by the local CME.

6.3.3 Content forwarding

Since the content forwarding mechanism is state-based, our design follows a similar style to that of standard IP multicast, but only at individual CAFEs at the network edge.

According to our initial design, the basic forwarding state format maintained at each CAFE can be described as:

<Content ID, previous-hop CAFE Address, List of next-hop CAFE Addresses>

Then *content ID* entry specifies the identifier of a specific piece of content being delivered, which is similar to the concept of IP multicast address. *The previous-hop CAFE address* entry indicates the previous-hop of CAFEF from where the content flow will be injected to the local CAFE. This can be either an ingress CAFE belonging to the same domain (e.g. in Figure 12 the *previous-hop CAFE address* of 1.0.0.3 for *content ID* of X is 1.0.0.2 which is effectively the ingress CAFE in the network 1/8 for X), or an egress CAFE belonging to the previous hop domain (e.g. in the same figure the *previous-hop CAFE address* of 1.0.0.2 for *content ID* of X is 1.2.0.1 which is the egress CAFE in the network 1.2/16 for X). The entry *List of next-hop CAFE addresses* indicates the sequence of next hop CAFEs that the local CAFE should send the content to. For instance, regarding content X, the list of next-hop CAFEs maintained by 1.0.0.2 include 1.0.0.1 and 1.0.0.3 leading towards specific downstream content consumers. Again, a next-hop CAFE can be either a (egress) CAFE within the same domain or a (ingress) CAFE belonging to the next-hop domain.

Despite the similarity with the IP multicast paradigm, it is important to note a key difference: in IP multicast it is individual routers that are responsible for sending a group join request towards the targeted source based on their own knowledge about the location of the content source (or rendezvous point), typically following the default IP paths. In our case individual CAFEs are not responsible for forwarding the "join request" which is effectively the content consumption request. This is instead done in the CMP in a much more flexible manner, typically hop-by-hop among CRMEs according to specific rules or policies such as business relationship between ISPs, local ISP policies as well as network conditions. As such there is a vertical interaction between a CRME (in the CMP) and its local CAFEs (in the CFP) for installing content states.

It is worth mentioning that some additional features will be further developed during the rest period of the project and the specification of all the designed mechanisms/protocols will be provided in D4.2. These will mainly include:

- Interactions between CAFEs at the network edge and legacy IP routers at the core. Although using tunneling is an obvious option, we will also explore more native approaches without necessarily involving content packets encapsulation.
- Quality of Service (QoS) support –how content flows requiring different levels of treatments can be gracefully delivered with different paths capabilities and how this will have an impact on the content delivery mechanisms in the CFP.

6.3.4 Inter-domain Content Delivery Path Optimisation

The proposed content delivery operation is also supported by a routing optimization technique for path switching from *provider* routes to *peering* routes if identified. In the figure, once the CME(specifically its CMF) in domain 1.1/16 has noticed that the content flow with source address belonging to prefix 1.2.1/24 has been injected into the local domain via ingress CAFE 1.1.0.1 via the provider route, and it also knows from the local BGP routing information that there exists a peering route towards the content source, it then issues a new *scoping-based* content consumption request: Consume(INCLUDE{1.2.1/24},X) and sends it to the CRME in domain 1.2/16 in the peering route towards the source. Upon receiving the request, the CRME in 1.2/16 will update the local CAFE 1.2.0.1 by adding a new outgoing next-hop CAFE 1.1.0.1. As a result, a new branch via the peering route is established towards content client **C2**. Once the ingress CAFE 1.1.0.1 has received the content via the interface connecting to 1.2.0.1, it will prune the old branch via the provider route (the dash line). The purpose of such content delivery path optimization across domains is to

effectively reduce content traffic within top-tier ISP networks and also possibly reduce the content delivery cost for customer domains. Of course, this operation is not necessary if a CRME is allowed to send content consumption requests to its peering counterparts (in addition to the provider direction) during the resolution phase. However, such an option will incur unnecessarily higher communication overhead in disseminating content consumption requests, especially when the peering route does not lead to any source that holds the requested content.

7 Advanced Content Delivery Features

7.1 Content caching

7.1.1 Introductory Notes

Caching techniques within COMET have not been defined yet. Caching techniques are going to be integrated in the Content Forwarding Plane and will have to decide on dynamicity or not of i) the content to be cached and ii) the caching points within the network. That is, the content to be cached may come from specific providers only (e.g., the ones who pay extra in order to improve the quality of service provided to their customers), or may be of specific type only (e.g., real-time and media content). According to the CCN caching paradigm proposed in [35] all contents that travel through the Internet are cached for some time. Furthermore, there may be specific and predetermined caching points within the network, similar to web-caching technologies (e.g., at specific servers, or Content-aware Routers/Forwarders), or caching can be done on-the-fly, wherever content is requested. Again, according to CCN, caching is dynamic as for the points of presence of caching entities. This means that all network routers have the ability to cache content. In our case, caching will be done at the *Content-Aware Forwarding Elements* (CAFEs), implemented at the CFP. It is not clear yet where the decisions as to where to cache and what to cache are going to be made. That is, functions of the CMP may take care of such decisions, but the *Content-Aware Forwarding Function* (CAFF) may also be responsible for dealing with the caching properties of the COMET system.

The transition from today's host-centric model to the content-centric model proposed in [Jacobson09] will require a lot of effort, since (although an overlay) it touches many fundamental parts of today's Internet architecture, from naming and content resolution to queuing/caching management and flow control. In this study, we focus on the new flow model introduced in CCN and attempt to quantify the potential gains that the paradigm shift will bring with it. As stated before the CCN caching paradigm is dynamic both in terms of “content to be cached” and the caching points within the network. Therefore, a quantitative analysis of the gain of this paradigm will give us an insight of what is the best possible approach to caching in content networks.

To achieve this goal and in order not to violate the semantics in [35] we study the caching part of the CCN paradigm only, isolated from the rest of the proposed architecture. We carry out a *packet-level* analysis, instead of a flow-level one, since it is our opinion that given the totally unexplored research field, investigations on packet-level dynamics should precede flow-wide conclusions. This study attempts to determine how long a given packet, which is referred to as the Packet of Interest, *PoI*, remains in cache given a system topology and rates of requests.

7.1.2 Motivation and Assumptions

In this study the performance of the new CCN caching protocol is modelled and the performance gain quantified against today's end-to-end transmission paradigm. The purpose is to find out whether fully dynamic caching is “profitable” performance gain-wise in content-centric networks. Details of our analysis based on Markov Chain are included in the Annex of this Deliverable.

The model takes into account all the important parameters required to have a complete view of the new system. The model works from the point of view of estimating how long a particular content packet, *PoI*, remains in cache. This is a function of the rate with which it is requested; the rate with which contents other than *PoI* are requested; and the size of the cache. The model outputs the proportion of time that *PoI* is not in the cache, essentially reflecting the *cache miss ratio for PoI*. A cache miss for the *PoI* at the first cache means that the request has to travel upwards towards the server. If none of the caches/routers along the way *“remembers” PoI*, then the request will have to reach the content server to retrieve it.

The modelling here focuses solely on the caching part of the CCN proposal. It assumes that the content publication, registration, and identification are already in place and moreover that routing of both requests and contents is done according to [Jacobson09]. No attempt is made to address issues such as flow control or lost packets and the modelling is done in isolation from this.

7.1.3 Initial thought on Caching in COMET

Our findings indicate the following:

1. **Even extremely popular content is "forgotten" quickly by core Internet routers.** Given a tree topology, similar to the one in Fig. A2 shown in the Annex (a loose reflection of the topology of the Internet), core Internet routers receive large amounts of requests, much of which is for unpopular content.
2. **Leaf routers are more tolerant to popular content and remember it for longer time.** Leaf or edge routers receive fewer requests compared to core Internet routers and therefore, have the chance to cache content for longer.
3. **Larger cache sizes exaggerate point number 2 above, while leave point 1 untouched.** In Fig. A4 in the Annex, we see that the proportion of time the *PoI* spends in leaf router caches increases with the cache size. The same is not true for core Internet routers.

With these conclusions in mind, our immediate next step is to define a caching scheme for the COMET system. It is clear, for example, that applying the same caching policies for all contents and also caching at every router or CAFE within the network may not bring huge gains to the system. Therefore, discriminating between different kinds of contents (e.g., elastic vs. inelastic traffic) and applying different caching policies accordingly may increase the Quality of Service observed by the content-client.

Furthermore, as stated earlier, the CCN caching scheme caches content in all routers on the route from the content server to the content client. This may turn out to be a painful process, implementation-wise. As an alternative, we will investigate implementing caching at egress routers, or CAFEs of a domain and routing "popular" traffic accordingly. For example, streaming of a football game that is expected to become popular can be routed through specific domain-edge CAFEs, where it can be cached and streamed to the interested users that reside outside the domain in question. This will reduce the traffic to be carried by the local ISP and will still increase the Quality of Service for the content client.

There are still a few open issues with regard to the realisation of the above points. For example, as stated earlier, *"who is going to decide where to cache and what to cache?"*. We can see two different approaches here. The first one suggests that the CMP is held responsible for these decisions. In particular, the CMF, after gathering all the required information about the availability and load of the paths from the SNMF and the PMF, decides whether this particular content should be cached or not, and where it has to be cached, e.g., at the edge CAFE of the content server's domain, in order to keep traffic away from the core of its domain. The second approach is to let the network level CAFEs decide whether content has to be cached or not and where. This, however, will require a lot of intelligence to be added to the actual network elements, i.e., the CAFEs. A potential solution would be the addition of a *Content Caching Function* (CCF), possibly complemented with the corresponding *Content Caching Entity* (CCE), which similarly to the CAFF, makes decisions on content caching properties. These issues are yet to be investigated and will be reported at a later stage within the course of the project.

7.2 Supporting point-to-multipoint content delivery

This section describes the general support of point-to-multipoint content delivery in COMET. The scenarios cover the exploitation of multicast capabilities in the last-mile content client's domains. Inter-domain support of point-to-multipoint content delivery can be natively supported by the

coupled approach thanks to the content state maintenance at CAFEs (see section 6.3.3), and this will not be repeated here.

As mentioned in D2.1 [60], it is not the aim of the COMET project to re-invent the multicast technology, but to make use of the existing multicast technologies and provide an integrated solution where the COMET system can support point-to-multipoint content delivery thus achieving bandwidth savings due to traffic reduction and optimal resource utilization.

The support of point-to-multipoint content delivery in COMET consists in opening the multicast capabilities in the content client's domain to be used for content delivery. Moreover, the mechanism will allow non-multicast traffic (UDP unicast traffic) to be delivered through the content client's multicast network.

A content client's domain has two requirements in order to open their multicast capabilities through COMET:

- The content client's domain must manage a multicast network, no matter the specific multicast routing protocol (PIM-SM, PIM-SSM, MOSPF, etc.) used to build multicast tree internally and no matter the underlying technology (point-to-multipoint VLANs through VPLS, native Ethernet multicast, IP multicast).
- There must be dedicated CAFEs which must be directly attached to the multicast network (e.g. a designated router in PIM-SM or PIM-SSM).

The overall idea is that the last CAFE in the content delivery path becomes the source of the multicast content in the content client's domain. This CAFE, which will be called multicast-aware CAFE or mCAFE, has specific requirements besides those of normal CAFEs:

- It must be able to distinguish from a COMET header that the content packet is from a multicast content, in order to be able to forward it to the multicast network (e.g. to a designated router in PIM-SM or PIM-SSM multicast networks). For this purpose, a forwarding rule identifier will be provisioned in each mCAFE for identifying multicast content.

The support of point-to-multipoint content delivery has implications in the following processes of COMET.

- *Provisioning*. Although a provisioning process has not been specifically detailed in the architecture in D2.2 [61] or in D4.1, it is assumed that CMEs must be aware of the CAFEs inside its own domain. In the same way, CMEs are expected to be aware of the mCAFEs inside one domain, as well as the specific local forwarding rule identifier used for multicast purposes per mCAFE. Moreover, CMEs need to have a pool of private multicast group addresses to be assigned to each multicast content.
- *Content resolution*. Just after receiving a content resolution request from a content client and before starting the name resolution process, the CME must check the list of multicast contents currently being delivered inside its domain to check if the requested content is being delivered previously. If so, no name resolution is required and the content can be delivered from the mCAFE which receives the multicast content from the source. This requires the CME to manage a table of active multicast contents containing the content identifier (Content Name or Content ID) and the specific multicast group address used for that delivery. If the requested content is not in that table, a name resolution and decision process will follow.
- *Path configuration*. If the content is not in the table of active multicast content, the decision process will provide the server and path to be used for content delivery. Moreover, other properties such as the content information and transport information will be provided. The content information must include a flag indicating whether the content is suitable for multicast or not. If so, a lookup is made into the table of provisioned paths between server domain and client domain. If there is no provisioned path, a path is provisioned for the specific COMET CoS (see *path provisioning* in section 6.2.2). Once there is a path, the path

configuration process will start. This process will follow the same scheme as the *agnostic path configuration* (see section 6.2.3.1). There are two differences with respect to that process:
    - The list of forwarding rules to be configured in the first CAFE contains the list of forwarding rule identifiers of the provisioned path plus the forwarding rule identifier of the local mCAFE of the client domain.
    - After receiving the positive acknowledge of path configuration (step 7 in *agnostic path configuration*), the client CME will configure a flow classifier in the local mCAFE The mCAFE will receive "transport information", and optionally the "content information", and will use it for packet classification using multi-field classification filter. Besides, the mCAFE will receive the multicast group address which will be used to perform NAT of the packet destination address.
- *Content forwarding*. The content forwarding from the first CAFE to the last CAFE (the mCAFE) follows the scheme described in section 6.2.4. Next, the operations in the mCAFE are described:
    1. mCAFE receives a packet with CFP header.
    2. mCAFE confirms that packet arrives from whitelisted CAFEs, otherwise it drops the packet.
    3. CAFE removes the current encapsulation from the packet (from previous CAFE).
    4. If list of forwarding rule identifiers is not empty, remove the first identifier FRidk (COMET header modification). If that FRidk is the local forwarding rule associated to multicast content, then it removes the COMET header.
    5. Then, it will perform a flow classification to find the multicast group address associated to the flow and will perform NAT of the destination address.
    6. Finally, mCAFE will forward the packet according to its current structure towards the multicast network.

7.3 Edge controlled routing

Edge controlled routing (ECR) enables flexible path switching and adjustment functions that take into account both ISP network resource optimization requirements and user perceived quality of service assurance. To achieve this, the COMET agent intelligence residing at the content client side is responsible for monitoring the content delivery performance from the content consumer point of view. In addition, it interacts with the "main" COMET intelligence, specifically the Path Management Function (PMF) of its local content mediation element (CME) for coordinating ECR decisions when necessary. More specifically, if the COMET agent detects deteriorated QoS assurance for the content consumer, it may initiate a path switching request and send it towards its local CME (specifically the PMF component). Upon receiving the request, the CMEs (possibly across multiple domains) in the will coordinate with each other before an alternative route from the content source to the consumer can be determined. The basic requirement for the coordination is that the actual path switching will not be expected to affect the current network resources at the ISP side, and this is particularly the case of multiple simultaneous path switching requests are issued from individual content clients.

Once an alternative new path has been identified, there are two fundamental approaches to enforce it. In the conventional stateless-based (unicast) content delivery paradigm, the CME residing in the domain where the new path branches away from the current one is responsible for configuring its local network elements(such as the content-aware forwarders) in the CFP in order to perform content delivery path switching, possibly across multiple ISP networks.

Figure 13 below illustrates such an operation. The content client is currently receiving the content from the remote provider via the lower path (step 0). Once the COMET agent at the client side issues a path switching request to its local CME due to a deteriorated QoS performance (step 1), individual CMEs will coordinate with each other in order to determine an alternative content delivery path that is able to both enhance the content delivery performance at the consumer side,

as well as retaining (or even improving if possible) the network conditions in the involved ISP networks (step 2). As soon as such a path has been identified (e.g. the upper path shown in the figure), then the CME that is physically attached to the domain in which the new path breaks away from the old one is responsible for actively reconfigure the local content aware forwarders in the CFP in order to activate the actual switching (step 3(a)(b)). Finally, once such path switching reconfiguration has been actually activated, the content flow will follow the upper path towards the content consumer (step 4) with the old be being "torn down" for this content delivery session. It is worth mentioning that all such operations takes place in the underlying contend mediation plane and forwarding plane, leaving the running content consuming application unaware of these procedures.

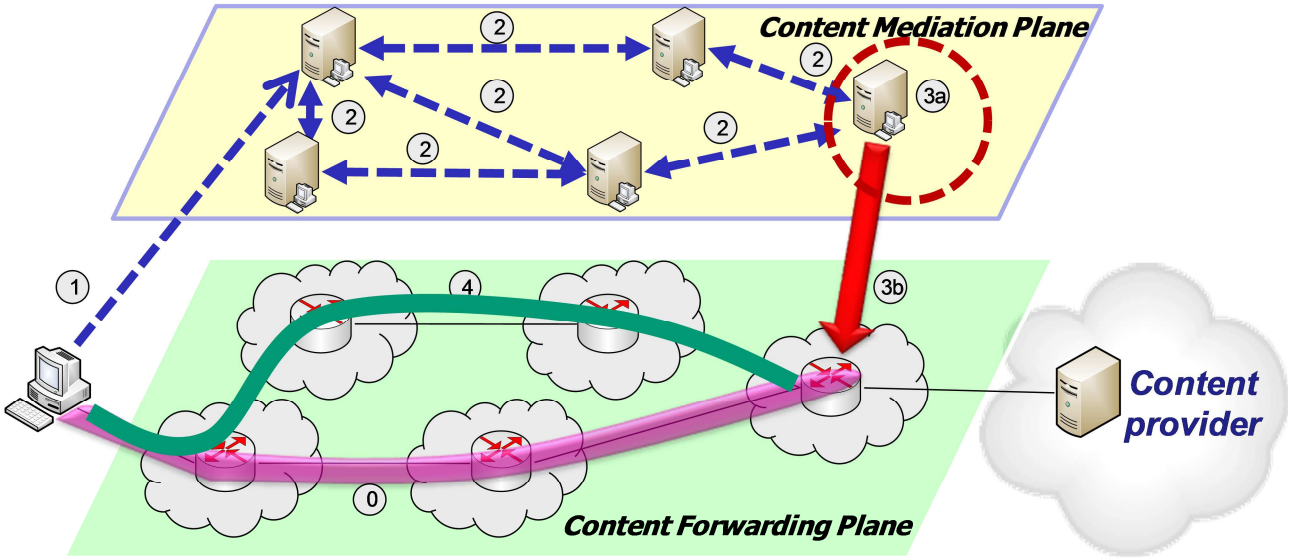

Figure 13: Stateless-based path switching example

An alternative path switching enforcement option is also being investigated in WP4, which is based on multicast-like stateless approach. Such a paradigm follows the same philosophy of IP multicast where the physical location (e.g. IP address) of the content consumer is not exposed. We take the example in Figure 14 for illustration. Upon the determination of the new alternative path by the collaboration of CRMEs (step 2), the local CRME at the content consumer side returns an acknowledgement (with the instruction for establishing the new path) to the COMET agent side (step 3). Thereafter, the COMET agent in the content client will send a new join request (similar to the PIM-SM group join in IP multicast) following the reversed direction of the new content delivery path, with the content state being installed along the way until the request reaches the branching point away from the old path (step 4). Finally the content from the source will follow the new path according to the states installed (step 5).

It is worth mentioning that such state based approach natively supports multicast based content delivery (see section 6.3) where in general a point-to-multi-point multicast tree is maintained toward multiple active content consumers. In case a particular content consumer has had its specific content delivery path switched, the other content consumers attached to the old path should not be affected. For instance in Figure 14 if the intermediate domain in the old lower path has active content consumers, then the tear-down operation of the old path triggered by the left-side content client should terminate there. Or alternatively, these intermediate content clients should be re-grafted onto the inter-domain multicast tree through other new paths. Of course such an operation needs additional intelligent coordination between involved CMEs across multiple domains.

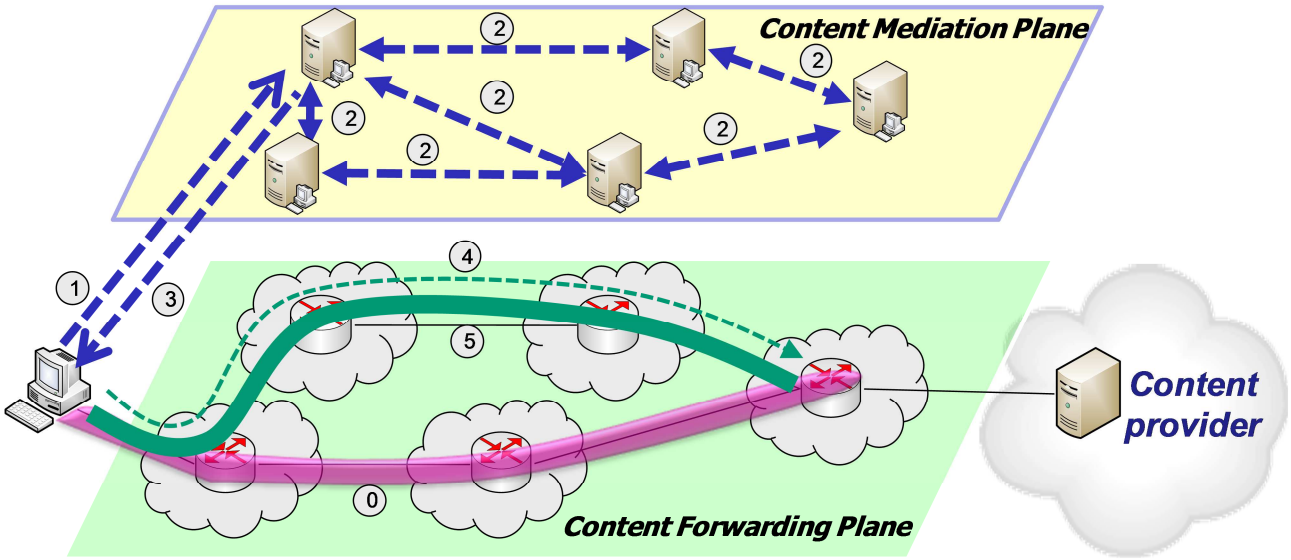

Figure 14: Stateful-based path switching example

8 Summary and Conclusions

This deliverable presents the interim specification of our proposed mechanisms, protocols and algorithms in the content forwarding plane (CFP) that are investigated in WP4 of the COMET Project.

First of all, we specify *offline operations* associated with content delivery operations according to the COMET approach. These mainly involve long timescale (e.g. monthly or even longer) network resource provisioning operations both within and across multiple ISP networks. As far as QoS differentiation is concerned, COMET has defined distinct Classes of Services (CoSes) for mapping today's popular content services such as live video streaming and pre-recorded content. For those with strict service level agreements (SLAs) with content consumers, the premium service class is needed in order to provide guaranteed service assurance in content delivery. In addition better-than-best-effort (BTBE) class can be also used for other real-time content applications without stringent QoS requirements. Finally the best-effort (BE) class is used for delivering content as it happens in today's Internet.

The provisioned paths across multiple domains for different CoSes need to be made aware to the CMP for the actual content delivery. Towards this end, the *routing awareness entity* (RAE) is used in COMET for disseminating such information up to the Content Mediation Entities (CME and CRME) in order to identify and enforce the optimal paths for delivering the content from the content source back to the content client. The logical function of path management function (PMF) embedded in RAE is responsible for such a purpose. In addition, some lightweight extension to the current BGP routing protocol has also been proposed in providing multi-path capabilities for delivering different classes of content across the Internet. In multi-service enabled networks (e.g. DiffServ), individual ISPs may implement different levels of QoS capabilities locally, but at the same time they should be responsible for mapping the global COMET CoS onto their locally configured QoS classes. How this can be achieved has also been discussed in this deliverable.

As far as the actual *online* content delivery is concerned, a dedicated element called content aware forwarding element (CAFE) is designed. Fundamentally, two distinct approaches have been described in this document, namely the *stateless* and the *stateful* approaches. In the stateless content delivery approach, which is coupled with the content based resolution approach specified in D3.1, optimized end-to-end content delivery paths are identified after the targeted content source has been found. Upon this, the Content Mediation Entities (CMEs) located in the source and consumer domains are responsible for exploring optimized paths for carrying the content traffic back towards consumer. The determination of the actual path is based on the routing awareness as well as some QoS conditions captured from network monitoring techniques. On the other hand, the *stateful* content delivery approach is natively linked with the coupled content resolution technique described in D3.1, in the sense that the actual path configuration/enforcement is performed during the content resolution phase.

Towards the end of this deliverable, we have also discussed some advanced techniques that will be further investigated during the rest period of the project, including content caching, support of point-to-multipoint content delivery and edge controlled routing.

All the interim techniques specified in this deliverable will be further enhanced in WP4 during the rest period of the corresponding tasks, and their final version will be presented in D4.2 in Month 21.

9 References

[1] D. Walton, A. Retana, E. Chen, J. Scudder, “Advertisement of Multiple Paths in BGP”, Internet Engineering Task Force (IETF),Internet Draft, draft-ietf-idr-add-paths-04.txt, August 2010

[2] P. Mohapatra, R. Fernando, C. Filsfils and R. Raszuk, “Fast Connectivity Restoration Using BGP Add-path,” Internet Draft draft-pmohapat-idr-fast-conn-restore-00, September 2008

[3] M. Motiwala et al, “Path Splicing,” Proc. ACM SIGCOMM, 2008

[4] X. Yang and D. Wetherall, “Source Selectable Path Diversity via Routing Deflections,” Proc. ACM SIGCOMM, 2006

[5] R. Cohen et al, “The Global-ISP paradigm,” Elsevier Computer Networks, Vol. 51, Issue 8, 2007

[6] W. Xu, et al., “MIRO: Multipath Inter-AS Routing,” ACM SIGCOMM 2006

[7] D. Anderson et al, “Resilient Overlay Networks”, Proc. 18th ACM Symposium on Operating Systems Principles

[8] Z. Li et al, “QRON: QoS-aware routing in overlay networks”, IEEE JSAC, January 2004, pp.29-40

[9] X. Masip-Bruin et al., “Research challenges in QoS routing”, Computer Communications, vol. 29, no. 5, March 2006, pp. 563-581.

[10] F. Kuipers et al., “Performance evaluation of constraint-based path selection algorithms," IEEE Network, vol.18, no.5, September-October. 2004, pp. 16- 23

[11] Z. Wang, J. Crowcroft, “Quality-of-service routing for supporting multimedia applications”, IEEE Journal on Selected Areas in Communications, vol. 14, September 1996, pp. 1228 –1234.

[12] G. Cheng, N. Ansari, “On selecting the cost function for source routing”, Computer Communications, vol. 29, issue 17, 2006, Elsavier, pp.3602-3608.

[13] G. Cheng, “The revisit of QoS routing based on non-linear Lagrange relaxation”, Int. J. Commun. Syst., vol. 20, 2007

[14] P. Khadivi, S. Samavi, and T. D. Todd, “Multi-constraint QoS routing using a new single mixed metrics”, *J. Netw. Comput. Appl., vol.* 31, no. 4, November 2008, pp. 656-676.

[15] P.Mieghem, F.A. Kuipers, “Concepts of exact QoS routing algorithms”, *IEEE/ACM Trans. Netw.,* vol. 12, no. 5, October 2004, pp.851-864.

[16] O.Bonaventure, “Using BGP to distribute flexible QoS information”, IETF draft draft-bonaventure-bgp-qos-00, February 2001.

[17] Li Xiao, Jun Wang, King-Shan Lui, K. Nahrstedt, “Advertising interdomainQoS routing information”, *IEEE Journal onSelected Areas in Communications,* vol.22, no.10, December 2004, pp. 1949- 1964.

[18] D. Griffin et al., “Interdomain Routing Through QoS-class Planes”. IEEE Communications Magazine, vol . 45 no 2, February 2007, pp.88–95.

[19] X. Masip-Bruin et al. “The EuQoS System: A solution for QoS Routing in Heterogeneous Networks”. IEEE Communications Magazine, vol . 45 no 2, February 2007, pp. 96-103.

[20] R. Prior, S. Sargento, “Towards Inter-Domain QoS Control.” 11th IEEE Symposium on Computers and Communications (ISCC-2006), 2006.

[21] Datta, A., Dutta, K., Thomas, H., Vandermeer, D., & Ramamritham, K., “Proxy-based acceleration of dynamically generated content on the world wide web: An approach and implementation”. *ACM Transactions on Database Systems, 29*(2), pp. 403-443 2004

[22] Wu, K.-L., Yu, P. S., & Wolf, J. L., "Segment-Based Proxy Caching of Multimedia Streams". *Proceedings of the 10th international conference on World Wide Web*, pp. 36-44

[23] Chen, S., Wang, H., Zhang, X., Shen, B., & Wee, S, "Segment-based proxy caching for Internet streaming media delivery". *Multimedia, IEEE , 12*(3), pp. 59-67, 2005

[24] Song, J. "Segment-based proxy caching for distributed cooperative media content servers", *ACM SIGOPS Operating Systems Review, 39*(1), pp. 22-33. 2005

[25] Satsiou, A., & Paterakis, M., "Efficient Caching of Video Content to an Architecture of Proxies according to a Frequency-Based Cache Management Policy" *ACM International Conference Proceeding Series, 2006*

[26] Wessels, D., & Claffy, K., "ICP and the Squid web cache", *IEEE Journal on Selected Areas in Communications, 16*(3), pp. 345-357, 1998

[27] Rousskov, A., & Wessels, D, "Cache digests. *Computer Networks and ISDN Systems", 30*(22-23), pp. 2155-2168, 1998

[28] Fan, L., Cao, P., Almeida, J., & Broder, A. Z, "Summary cache: a scalable wide-area web cache sharing protocol . *IEEE/ACM Transactions on Networking, 8*(3), pp. 281-293, 2000

[29] Ni, J., & Tsang, D. H., "Large-Scale Cooperative Caching and Application-Level Multicast in Multimedia Content Delivery Networks", *IEEE Communications Magazine, 43*(5), pp. 98-105, 2005

[30] Megiddo, N., & Modha, D. S., "ARC: A Self-Tuning, Low Overhead Replacement Cache", *Conference On File And Storage Technologies*, (pp. 115-130). San Francisco, 2003

[31] R. Fielding et al, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616

[32] Krishnan, P., Raz, D., & Shavitt, Y., "The cache location problem", *IEEE/ACM Transactions on Networking, 8*(5), pp. 568-582, 2000

[33] Li, B., Golin, M., Italiano, G., Deng, X., & Sohraby, K., "On the optimal placement of web proxies in the internet", *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings, 1999*

[34] Cronin, E., Jamin, S., Jin, C., Kurc, A. R., & Raz, D. (n.d.). Constrained Mirror Placement on the Internet. *IEEE Journal on Selected Areas in Communications*.

[35] V. Jacobson et al, "Networking Named Content", Proc. ACM CoNext 2009

[36] Ashok Anand, Archit Gupta, AdityaAkella, SrinivasanSeshan, Scott Shenker: Packet caches on routers: the implications of universal redundant traffic elimination. SIGCOMM 2008: 219-230

[37] Eugster, P. T., Felber, P. A., Guerraoui, R., and Kermarrec, A. 2003. The many faces of publish/subscribe. *ACM Comput. Surv.* 35, 2 (Jun. 2003), 114-131. DOI= http://doi.acm.org/10.1145/857076.857078

[38] Elisha J. Rosensweig, Jim Kurose: Breadcrumbs: Efficient, Best-Effort Content Location in Cache Networks. INFOCOM 2009: 2631-2635

[39] H. Che, Z. Wang, andY. Tung, "Analysis and Design of Hierarchical Web Caching Systems" In INFOCOM. IEEE, 2001, pp. 1416–1424.

[40] Norihito Fujita, Yuichi Ishikawa, Atsushi Iwata, RaufIzmailov, Coarse-grain replica management strategies for dynamic replication of Web contents, Computer Networks, Volume 45, Issue 1, The Global Internet, 15 May 2004, Pages 19-34

[41] Pallis, G. and Vakali, A. 2006. Insight and perspectives for content delivery networks. *Commun. ACM* 49, 1 (Jan. 2006), 101-106. DOI= http://doi.acm.org/10.1145/1107458.1107462

[42] Deering, S. “Host Extensions for IP Multicasting”, RFC 1112, 1989

[43] Ratnasamy, S., Ermolinskiy, A., and Shenker, S. 2006. Revisiting IP multicast. In *Proceedings of the 2006 Conference on Applications, Technologies, Architectures, and Protocols For Computer Communications* (Pisa, Italy, September 11 - 15, 2006). SIGCOMM '06. ACM, New York, NY, 15-26. DOI= http://doi.acm.org/10.1145/1159913.1159917

[44] D.Sjöberg, “Content Delivery Networks: Ensuring quality of experience in streaming media applications”, TeliaSonera International Carrier, CDN white paper 2008

[45] M. Pathan and R.Buyya, “A Taxonomy and Survey of Content Delivery Networks”, Technical Report, GRIDS-TR-2007-4, Grid Computing and Distributed Systems Laboratory, The University of Melbourne, Australia. 12 February, 2007

[46] J. Dilley, B. Maggs, J. Parikh, H. Prokop, R. Sitaraman and B. Weihl, “Globally Distributed Content Delivery”, IEEE Internet Computing, pp. 50-58, September/October 2002.

[47] G. Pallis and A. Vakali, “Insight and Perspectives for Content Delivery Networks”, Communications of the ACM, vol. 49, no. 1, ACM Press, NY, USA, pp. 101-106, January 2006.

[48] B. Krishnamurthy, C. Willis and Y. Zhang, “On the User and Performance of Content Distribution Network” In Proceedings of 1st International Internet Measurement Workshop, ACM Press, pp. 169-182, 2001.

[49] J. Ni, D. H. K. Tsang, I. S. H. Yeung and X. Hei, “Hierarchical Content Routing in Large-Scale Multimedia Content Delivery Network”, In Proceedings of IEEE International Conference on Communications, 2003 (ICC ’03), vol. 2, pp. 854-859, May 2003.

[50] J. Kangasharju, et al. “Object replication strategies in content distribution networks”, Computer Communications 25, 4 (Mar. 2002), 367–383.

[51] Qiu, L. et al. “On the placement of Web server replicas”, In Proceedings of the 20th IEEE INFOCOM Conference (Anchorage, Alaska, Apr. 2001), 1587–1596.

[52] Chen, Y. et al. “Efficient and adaptive Web replication using content clustering”, IEEE Journal on Selected Areas in Communications 21, 6 (Aug.2003), 979–994.

[53] Fujita, N. et al. “Coarse-grain replica management strategies for dynamic replication of Web contents”, Computer Networks 45, (2004), 19–34.

[54] IEEE Std. 802.1Q-2005, Virtual Bridged Local Area Networks, 19 May 2006

[55] Y. Rekhter, T. Li, and S. Hares, "A Border Gateway Protocol 4 (BGP-4)” Internet Engineering Task Force (IETF) Request for Comments (RFC), RFC 4271, January 2006.

[56] J. Uttaro et al., “Best Practices for Advertisement of Multiple Paths in BGP”, Internet Engineering Task Force (IETF), Internet Draft,draft-ietf-idr-add-paths-guidelines-00.txt, November 2010

[57] J. Babiarz, K. Chang, F. Baker, “Configuration guidelines for DiffServ service classes”, Internet Engineering Task Force (IETF) Request for Comments (RFC), RFC 4594, 2006

[58] K. Chan, J. Babiarz, F. Baker, “Aggregation of Diffserv Service Classes”, Internet Engineering Task Force (IETF) Request for Comments (RFC), RFC 5127, 2008

[59] Psaras I., Clegg R., Landa R., Chai W., Pavlou G., “Modelling and Evaluation of CCN-Caching”, UCL Technical Report, Available upon request.

[60] COMET Deliverable, “D2.1: Business Models and System Requirements for the COMET System”, July 30th 2010.

[61] COMET Deliverable, “D2.2: High-Level Architecture of the COMET System”, November 30th 2010.

[62] COMET Deliverable, "D3.1: Interim Specification of Mechanisms, Protocols and Algorithms for the Content Mediation System", November 30th 2010.

10 Abbreviations

AF	Assured Services
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
CFP	Content Forwarding Plane
CME	Content Mediation Entity
CMF	Content Mediation Function
CMP	Content Mediation Plane
COMET	COntent Mediator architecture for content-aware nETworks
CoS	Class of Services
CRE	Content Resolution Entity
CRF	Content Resolution Function
CRME	Content Resolution and Mediation Entity
CS	Content Server
DNS	Domain Name System
IANA	Internet Assigned Numbers Authority
ID	Identifier
IETF	Internet Engineering Task Force
IGP	Interior Gateway Protocol
IP	Internet Protocol
ISP	Internet Service Provider
NLRI	Network Layer Reachability Information
PMF	Path Management Function
QoE	Quality of Experience
QoS	Quality of Service
RAE	Routing Awareness Entity
SLA	Service Level Agreement
SNMF	Server and Network Monitoring Function
TTL	Time To Live

11 Acknowledgements

This deliverable was made possible due to the large and open help of the WP4 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.

12 Annex – Study of CCN caching

Modelling CCN Caching using Markov Chains

Consider the cache from the point of view of a given *PoI*, which may be cached. The question to be answered is: "what proportion of the time is that packet in the cache?". Initially, we consider only a single router and assume that the cache has room for exactly *N* packets. The modelling works from the perspective of a given *PoI* and asks questions about what proportion of time it is in the cache. Here we present the single-router analysis, while the interested reader can find the complete multi-router analysis in [59] .

A simple model for a single router

Assume that for a single router requests for the *PoI* arrive as a Poisson process with rate λ. Whenever such a request arrives, then this packet is moved to the top slot in the cache. Assume that requests that will move the *PoI* further down the cache (either requests for packets not in cache or requests for packets in cache, but further down than the packet of interest) also arrive as a Poisson process with rate μ.

This process can be simply modelled as a continuous time homogeneous Markov chain, where the state of the chain represents the exact slot that the packet currently occupies in the cache. Number the Markov chain states as follows. State 1 is when the *PoI* is at the "top" of the cache (just requested), state *N* is when our packet is at the bottom of the cache (any request for a packet not already in cache will push our packet out of the cache).State$N+1$ represents the state where our packet is *not* in the cache. The chain and cache are shown in Fig. A1.

All states (except state *1*) have a transition to state *1* with rate λ (another request for the packet arrives and the packet moves to the top of the cache). All states i in $1 \le i \le N$ have a transition to state $i+1$ with rate μ (a packet arrives which moves our packet lower in the cache).

$$\pi_{N+1} = \left[\frac{\mu}{\mu + \lambda}\right]^N$$

Eq. 1: Proportion of Time Not in the Cache for *PoI*

This is the proportion of the time that the *PoI* is not in cache (also the proportion of requests for the *PoI* that get a cache miss). Naturally, the proportion of the time our packet gets a "cache hit" is one minus this.

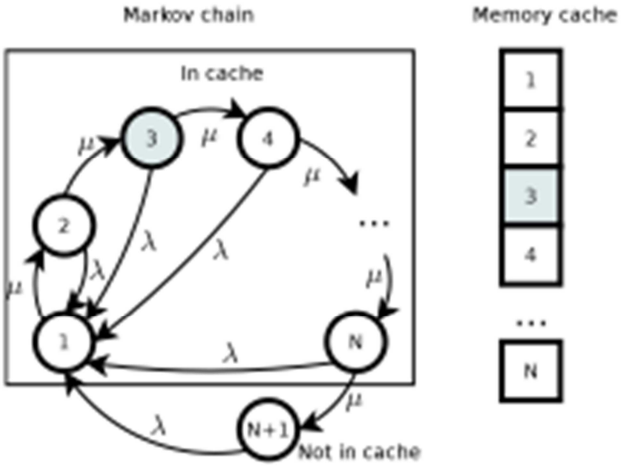

Fig. A1. The Markov chain and the cache it represents with the *PoI* at position 3 (shaded) in the chain

Simulation setup and initial results

Two simulation models are evaluated. The first is in Java and directly uses the analytical model introduced before. The second model is based on *ns-2*. The *ns-2* experiments are detailed in [59] .

The Java simulation takes as input a topology and, for each cache, the values of λ (request rate for *PoI*), μ (rate at which other requests move the *PoI* down the cache) and *N* (cache size in packets). For simplicity results are presented in terms of the ratio λ / μ, referred to as the *Content Popularity Ratio* (R_{CP}). The larger this is, the longer the *PoI* will stay in cache. For each router the simulator outputs the *Proportion of Time Not in the Cache* for the *PoI* from Eq. 1. The simulation topology used herein is depicted in Fig. A2.

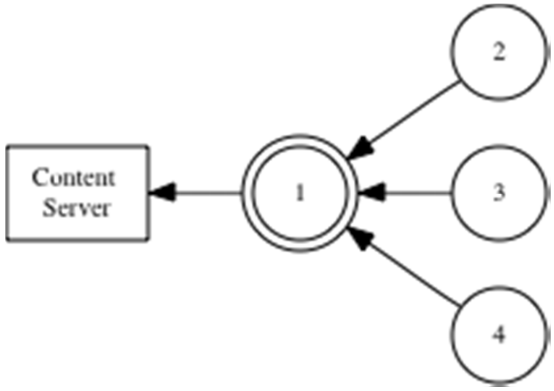

Fig. A2. Simulation Topology

Scenario 1: Content Popularity

We assess the properties of requested content with regard to its popularity. Content-Centric Networking was originally proposed to deal, among others, with popular content and flash crowds.

This scenario experiments with different values for R_{CP}. Values for R_{CP} range from 0.000125 to 0.01 -- obviously all of these values represent very popular content but the experiment shows how the popularity affects the time in cache. The buffer size N is set to 200 packets (approximate the size of the buffer of an IP gateway) and as discussed μ is set accordingly to the same value.

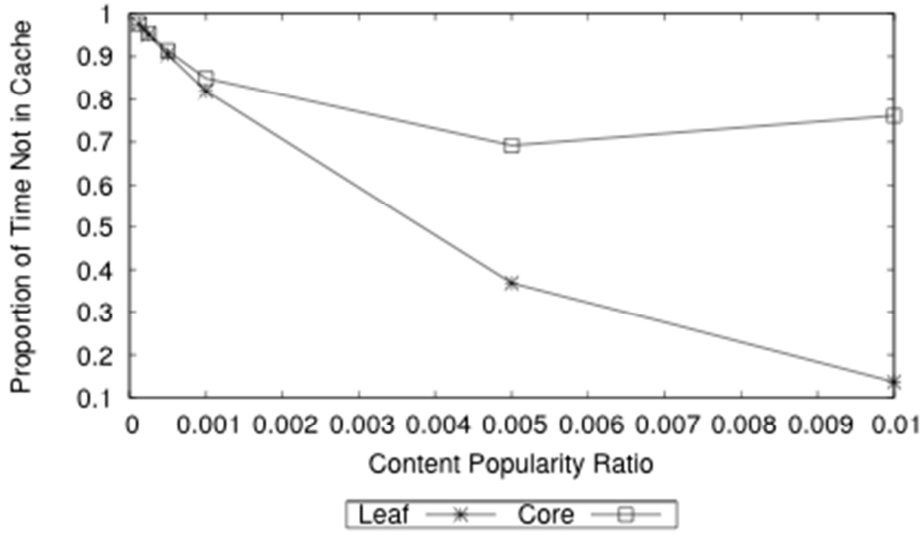

Fig. A3. Increasing Content Popularity

Fig. A3 presents the proportion of time the *PoI* is not in cache. Three conclusions can be drawn from this scenario, all of them more or less expected. First, it is clear that for more popular content

(larger R_{CP}) the content of interest stays in the cache for longer at the leaf. Secondly, unpopular content spends very little time in the cache. This time is comparable to the life-cycle of an IP packet in the IP buffer of today's networking model. This would be the case for most packets. Thirdly, there is a clear difference between caching times for popular content in the core and leaf routers. Caching at the leaf nodes can reverse the expected effect on popular content in the core. That is, more caching at the leaf leads to less caching in the core for very popular content.

Scenario 2: Cache Size

It is not yet clear how much space is needed, cache-size-wise, to implement the new CCN paradigm. Small IP router-like buffers can serve the purpose of implementing CCNs, but the gain against today's end-to-end model will be marginal (i.e., even popular content is going to be "forgotten" very quickly). Hence, bigger amounts of memory may have to be considered in order to obtain gains from the paradigm shift. Using the same settings as the previous simulation, and keeping $\mu = N$, the cache size N is varied from 100 to 64,000 packets (rounding packet sizes to 1KB this is 100KB to 64MB).

In Fig.A4, we see that for larger caches, even the less popular content has the chance to stay cached for longer times, as would be expected. Above 6MB of cache, even the less popular content is in cache at the leaf node for more than 50% of the time. Again, caching at leaf nodes has reduced the proportion of time spent in cache at the core.

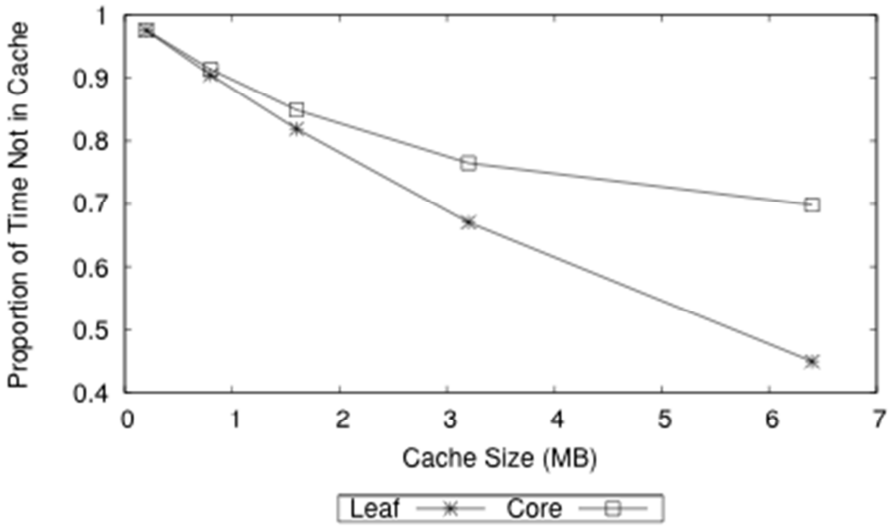

Fig. A4. Increasing Cache Sizes