

# Internet-Scale Content Mediation in Information-Centric Networks

George Pavlou<sup>1</sup>, Ning Wang<sup>2</sup>, Wei Koong Chai<sup>1</sup>, Ioannis Psaras<sup>1</sup>

<sup>1</sup>Dept. of Electronic and Electrical Engineering, University College London, UK

<sup>2</sup>Dept. of Electronic Engineering, University of Surrey, UK

## ABSTRACT

Given that the vast majority of Internet interactions relate to content access and delivery, recent research has pointed to a potential paradigm shift from the current host-centric Internet model to an information-centric one. In information-centric networks named content is accessed directly, with the best content copy delivered to the requesting user given content caching within the network. Here we present an Internet-scale mediation approach for content access and delivery that supports content and network mediation. Content characteristics, server load and network distance are taken into account in order to locate the best content copy and optimize network utilization while maximizing the user quality of experience. The content mediation infrastructure is provided by ISPs in a cooperative fashion, with both decoupled/two-phase and coupled/one-phase modes of operation. We present in detail the coupled mode of operation which is used for popular content and follows a domain-level hop-by-hop content resolution approach to optimally identify the best content copy. We also discuss key aspects of our content mediation approach, including incremental deployment issues and scalability. While presenting our approach we also take the opportunity to explain key information-centric networking concepts.

## 1. INTRODUCTION

The Internet has been enormously successful, with IP simplicity being a key factor that allowed it to reach an impressive scale. The original Internet model focused on interconnecting hosts for resource sharing purposes, but after significant evolution over the last two decades, the Internet is currently being used for a wide variety of applications and services. On the other hand, the vast majority of interactions relate to content access and delivery. This is evident from the proliferation of user-generated content, e.g., photos and videos made available through social networking sites such as Facebook and Myspace, through video aggregators such as YouTube, etc., and also through overlay content distribution infrastructures, for example peer-to-peer (P2P) systems such as BitTorrent and eMule, and content delivery networks (CDNs) such as Akamai and Limelight.

A key aspect related to today's content access is *fragmentation*: users need to know the content location *a priori* in order to access it and content has to be searched through specific intermediaries, e.g., Youtube, BitTorrent, etc. As a result, a lot of content tends to be accessible only by particular user communities. Given the continuing exponential increase in content generation (both amateur and professional), a converged architecture for unified content access and delivery is necessary, providing *name-based content access*. In this context, recent research has pointed to a paradigm shift from the current host-centric Internet model to an information-centric one, with various architectural approaches proposed [1][2][3][4][5]. The key aspect behind all these approaches is to address *named content* directly, with the best content copy delivered to the requesting user given that caching will take place within the network. In such an information-centric paradigm, content resolution and delivery functions will be natively realized by the network, enabling network operators to play a more active role in the future content-oriented Internet marketplace.

In this paper, we present an Internet-scale mediation infrastructure for content access and delivery in information-centric networks. Our mediation approach is evolutionary, operating initially as a tightly-coupled overlay over the current IP infrastructure, but it could eventually be supported natively within the network. Name-based content access is achieved through collaborative content resolution and delivery functions among Internet Service Providers (ISPs) who operate this content mediation infrastructure in a collaborative manner. Content providers

and consumers publish or consume content through a set of *unified* content primitives, via which they interact with their local ISP. This is quite different from existing loosely-coupled overlay approaches, in which a content provider or consumer may have to interact with multiple content overlays in order to maximize accessibility of the published content, or in order to locate a specific piece of content.

Our content mediation infrastructure can be instantiated through two complementary approaches: a *decoupled* approach for the majority of Internet content, and a *coupled* approach for widely-accessed popular content which can benefit from in-network caching. In the decoupled approach, content resolution takes place first, followed by content access using the server identified through the resolution process. In the coupled approach, content resolution and access are combined in a single phase, with content resolution following a gossip-like communication model, routing content consumption requests in a specific manner within the mediation plane in order to locate the targeted content source. In both approaches, if multiple content copies are available at different servers, the one with good availability (e.g., with low or medium server load) in combination with the least network distance is selected. In addition, monitored end-to-end path quality may be used together with the network distance. The approach aims to optimize both network utilization and user quality of experience (QoE).

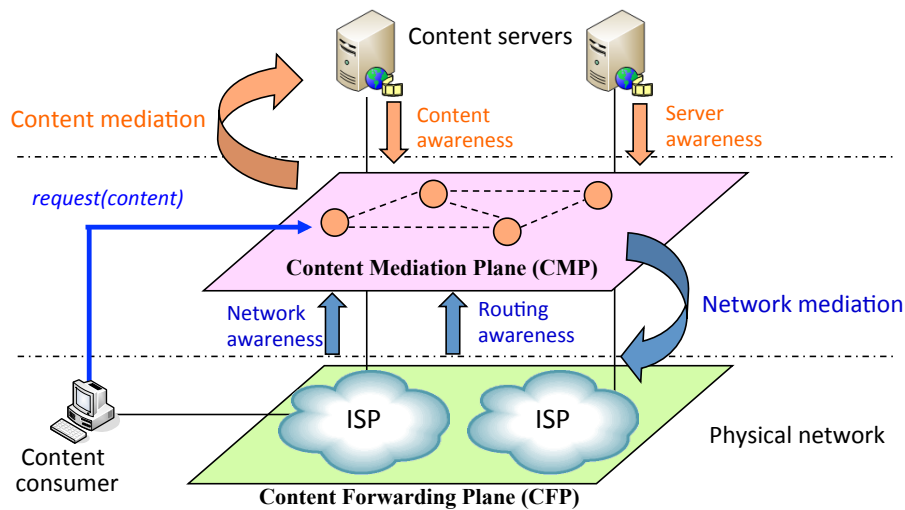
The rest of the paper is organized as follows. In section 2, we first introduce the concept of *content mediation*, which is a fundamental aspect of our ISP-operated tightly-coupled overlay approach. In section 3 we examine the functional aspects of mediation, presenting an associated functional architecture with components and their interactions. In section 4 we present an overview of our coupled approach, followed by a detailed description of content handling operations, i.e., publication, resolution and delivery. In section 5 we discuss various key aspects of our design, including incremental deployment, scalability and integration with search engines. We finally conclude the paper in section 6.

## 2. CONTENT MEDIATION

The proposed information-centric ecosystem is based on the concepts of *content* and *network mediation*. As depicted in Figure 1, a mediation plane operates between the “content cloud” and the underlying network infrastructure, providing content access and delivery in a holistic manner. This mediation plane works as a tightly coupled overlay, being collaboratively provisioned and operated by ISPs. Current information-centric architectures are either native, requiring fundamental changes in the Internet fabric through content-aware network protocols [1][2][3], or evolutionary, operating as tightly coupled overlays [4][5]. In both cases, there is intimate knowledge of the network characteristics and this is in contrast to current loosely-coupled network-agnostic overlays such as content delivery networks (CDNs) and peer-to-peer (P2P) systems. In fact, realizing the relevant limitations, it has been recently proposed to pass network usage information to overlays for enhancing both overlay and network performance through optimized content source/peer selections, e.g., the IETF ALTO framework [6][7] and the P4P paradigm [8]. On the other hand, an ISP-operated tightly-coupled overlay has intrinsic access to such information and can use it for selecting the best content source.

Our approach provides the following complementary mediation functions:

- *Content mediation* – the content mediation function gains awareness on both the content characteristics (e.g., quality requirements) and the content source conditions (e.g., server load). Based on this awareness, it is able to locate the best content copy in a relatively intelligent manner.
- *Network mediation* – the network mediation function gains necessary routing and network awareness for supporting content delivery through the best transport strategy in order to improve both user QoE and effective bandwidth utilization.



**Figure 1: Content mediation plane**

The content mediation plane is realized through Content Mediation Servers (CMSs) which communicate with each other in order to provide inter-domain mediation. Each domain or Autonomous System (AS) must operate at least one CMS, although it may operate more based on non-functional requirements such as availability, response time etc. In fact, CMSs are similar to today's Domain Name System (DNS) servers and every content publishing or consuming application should know its local CMS (through local configuration, in a similar fashion to today's local DNS server).

There exist two different approaches for the mediation plane to resolve and access content. In the first approach, CMSs resolve the content name or ID to a set of sources that hold that content, given that the content may be replicated. This list may also include routers with content caching capability, if there is capability for content caching *within* the network. This resolution can be achieved through suitable organization of content records in the CMSs, e.g. through a hierarchical Distributed Hash Table (DHT) approach or even through hierarchical content naming, in a similar fashion to the DNS, although in the latter case it is difficult to cope with dynamic content caching in the routers. A list of content sources is returned through the resolve operation and the best possible source is then selected based on network distance, server load and other relevant information available in the mediation plane, for example average network load along the path. The content is finally requested from the selected source. We call this the *decoupled approach* as it decouples content resolution from content request and delivery, in a similar fashion to the resolution of host names to IP addresses through the DNS before establishing a session to a remote host in the current Internet.

In the second approach, information in the content name / ID together with information in the CMSs about the "network direction" in which a particular piece content can be found can guide the resolution message in a hop-by-hop fashion to the content source. This information is used together with information on network distance and server load in order to locate the best possible content source. The reasoning behind this approach is that it can cope better with in-network caching and, given it emulates the function of native in-network approaches in the mediation plane, it can constitute an interim migration step towards full native deployment. We call this the *coupled approach* as it couples content resolution and access with content delivery: the content request message is routed through a chain of CMSs across domains to the content source. This is in line with native information-centric approaches such as [1][2][3] and in contrast with the decoupled approach which separates resolution from content access and delivery. These approaches are also commonly called *two-phase* (the decoupled one) and *one-phase* (the coupled one) in information-centric architectures.

One key aspect in all information-centric architectures is naming, given that it is very important for the resolution process. In fact, in native information-centric architectures, packets are routed based on content names or IDs instead of host addresses. The same is the case in the mediation plane for the coupled approach described above. These names are not necessarily "human user friendly" but they maybe opaque; they may also be self-certified in terms of security, given that it is the content itself and not the communication channel that needs to be secure.

Human users could find such a name or ID through search based on the content object properties, in a similar manner to today’s web search engines. In fact the mediation plane will also provide “hooks” to external search engines in order for them to index content based on meta-data.

The mediation plane can support information-centric operation over the current IP-based Internet, unifying content resolution, access and delivery for all types of content. In the decoupled approach, it can simply act as content-oriented enhanced “DNS” that could locate the best possible content source although network support may be optionally used for better-than-best-effort content delivery. But in the coupled approach network support is necessary, in the form of Content-Aware Routers (CARs) which have the capability to natively handle the delivery of content objects according to their IDs, and possibly with in-network caching functions for achieving localized content access. In the decoupled approach CARs are not necessary if content is to be streamed back to the consumer on a best-effort basis, following default BGP paths as in the current Internet. But they may be also used as overlay routing nodes in order to circumvent the shortest end-to-end path based on network load information that is available in the mediation plane through monitoring.

Content-aware routers may also cache popular content that is traversing them guided by the local domain CMSs; caching in this case relates to whole content objects and not to individual packets as in [2] or to chunks as in P2P systems. While [2] proposes indiscriminate caching everywhere along the path, our initial work on modeling and evaluating caching trees has shown that caching is more beneficial in specific network locations [9][10] and the CMSs could guide particular CARs to cache specific content objects. CARs are necessary in the domain edge (i.e. border routers *have* to be content-aware) but could also start being gradually deployed within the network for advanced information-centric network operation. In a target scenario, the mediation plane for the coupled approach will be collapsed completely into the network, with ubiquitous native content-aware routers routing packets based on names / IDs, as in recently proposed radical approaches [2][3].

### 3. FUNCTIONAL VIEW

We now present the functional view of this system in terms of the contained functional blocks in the Content Mediation Server and Content-Aware Router; this presentation is general, covering both the decoupled and coupled approaches. The overall functional architecture consists of two distinct planes as introduced in Figure 2: the content mediation plane (CMP) and the content forwarding plane (CFP). The CMP is responsible for content resolution, i.e., for the optimal identification of the best content source according to the specific requirements of the content consumer, while the CFP deals with end-to-end content delivery.

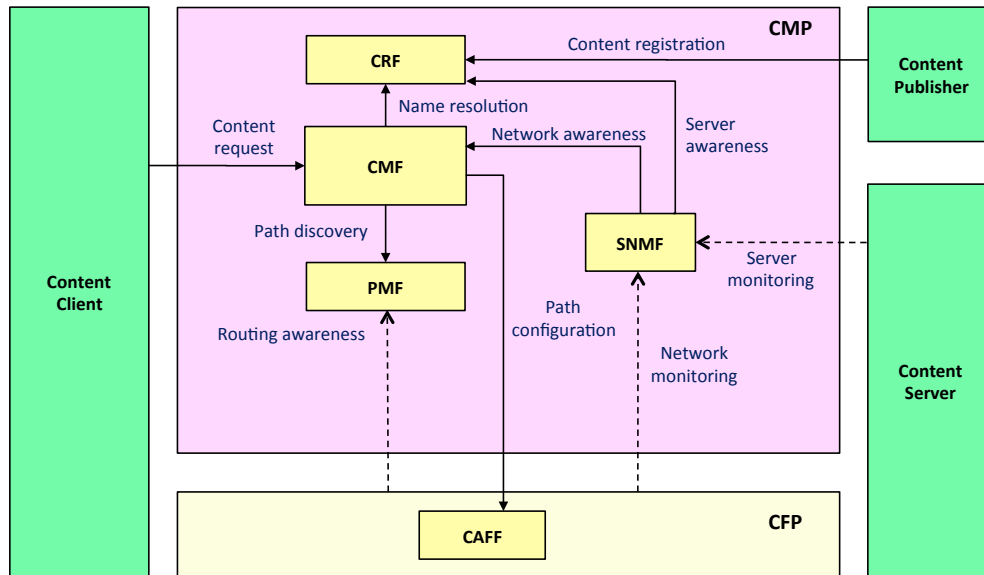


Figure 2: Content mediation functional architecture

The functional architecture is depicted in Figure 2 and encompasses the following functional blocks:

- Content Resolution Function (CRF): It is invoked during content publication and content consumption. Its main tasks are to maintain content records and to resolve content requests (i.e., resolve content names to content properties included in the content records and include content metadata). A content record provides the mapping information from the content name to the physical content sources in the Internet, and it may also include content metadata (e.g., alias, media type) and server load condition(s) to be used during the content resolution operation.
- Path Management Function (PMF): It interacts with the underlying network to gather necessary network reachability information across domains through eBGP, and also information concerning quality of service (QoS) capabilities and characteristics in QoS-capable networks, e.g., quality of service classes that are supported along the route. It is important to note that the information dealt with by the PMF is *long-term* information.
- Server and Network Monitoring Function (SNMF): It is responsible for gathering “just-in-time” information about both the server load conditions and also for potentially monitoring underlying path quality (e.g., bandwidth availability) at relatively short timescales for supporting optimized content resolution and delivery operations. The monitoring operation is typically time-driven, i.e., server and network conditions are periodically measured independently of the incoming content requests, although server load could also be event-driven.
- Content Mediation Function (CMF): Being the core function as “decision maker” in the CMP, it gets necessary input from the CRF, SNMF and PMF, interacts with content clients during content consumption and configures the CARs in the CFP for setting up delivery paths during content consumption. Its main functionality is to make decisions regarding the selection of the best content copy based on information regarding server and network conditions received from SNMF and the information about the available paths from the PMF. Based on this information, it then determines the content delivery paths and performs necessary configurations.
- Content-aware Forwarding Function (CAFF): It is the main function in the CFP and is responsible for forwarding content through end-to-end paths determined by the CMF. It is actually concerned with data-plane aspects for handling content delivery, including appropriate forwarding behaviors.

The overall content mediation system works as follows. Individual content publishers register their content to the CRF that is responsible for maintaining a global distributed content repository with records of all registered content items. When a content client requests a specific item, it uses a unified interface to contact the CMF for triggering the content resolution process. To start with, the CMF interacts with the CRF to identify the possible content locations according to the requested content name. Name resolution also takes into account the actual server load condition maintained by the CRF, which is effectively reported from the SNMF. Meanwhile, the CMF also receives both long-term and dynamic information as input from the PMF and SNMF respectively regarding route reachability and most recently monitored path conditions. An optimized decision based on the available information gathered is then made to identify the best server candidate together with the delivery path. Thereafter, the CMF configures the underlying content delivery path (i.e., instructing the CAFF at the CFP) to be ready for delivering the content flow.

It is worth mentioning that the functional components of Figure 2 are all *logical* ones, and hence, can be realized through different physical instantiations in practice. In addition, not all functions are necessary in every instantiation. For example, in the decoupled approach the CRF becomes a separate physical entity in a similar fashion to a DNS server while in the coupled approach it is in fact tightly integrated with the CMF.

In the remainder of this paper we describe in detail the operation of the mediation plane using the coupled approach for inter-domain content access with server load awareness, providing substance to the information-centric concepts introduced so far.

## 4. SERVER LOAD AWARE CONTENT ACCESS

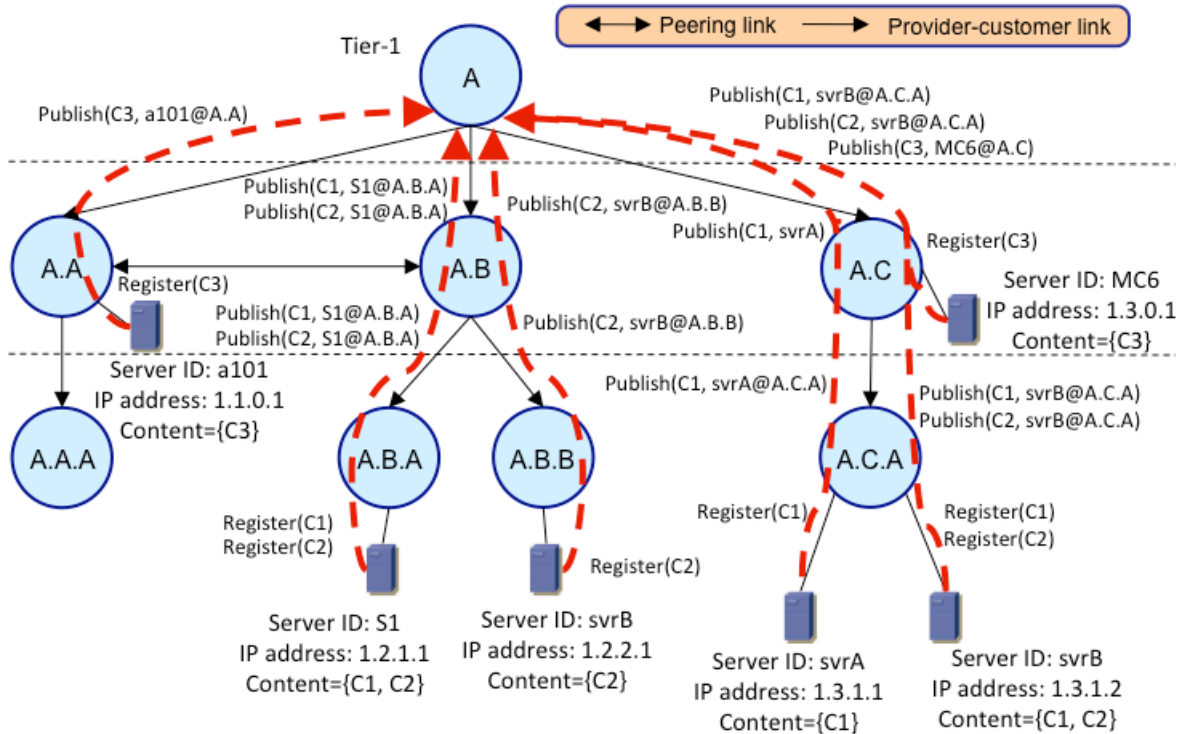
Our coupled content mediation approach is based on a gossip-style communication model between CMSs upon a content request. The content resolution process is also *server-load aware* given that when multiple copies of the same content are discovered, the best one is selected based on both server load and network distance from the content consumer. Such a feature has been investigated in replicated web servers [11] and overlay CDNs [12], but we are the first to consider it in the context of an information-centric architecture here.

The CMSs for each domain are equivalent to the resource handler in [1] and the rendezvous points in [4]. For simplicity, we will assume the existence of a single CMS per domain which we will call “resolver” in the remainder of this paper. Each resolver maintains a content record repository for all content in the local domain and all its “subordinate/customer” domains. This repository maintains an entry for each content item, mapping a content identifier (ID) to information related to all the sources hosting the content in this domain and in all subordinate/customer domains in terms of domain hierarchy. Scale can be huge, especially in tier-1 domains at the top of the domain hierarchy, but this approach applies only to “popular” content, as we also discuss in section 5, and this can circumvent the potential scalability problems. Content resolution is achieved through the routing of content consumption request messages between resolvers residing in each domain, according to their business relationships i.e., provider → customer or peer → peer. In a similar manner, a separate dissemination mechanism of up-to-date load information of individual content servers is also required in order to assist optimized content resolution.

Content publication, server load updates and content resolution messages follow the provider route forwarding rule: each message is forwarded hierarchically to the provider domain(s) until the first tier-1 domain is reached. Content and server load records are shared among all tier-1 domains while lower tier domains only hold this information for themselves and their subordinate domains in the hierarchy. The rationale for this hierarchical structure is scale, given that higher-tier domains are more resourceful and able to support high-capacity CMSs that will be able to cope with the full repository of Internet-wide content. Lower-tier domains are considered less resourceful and hence the amount of content and state information they keep relates to their position in the domain hierarchy. In strict consequence, lowest-tier leaf domains need only hold information pertaining to them. An additional reason for this organization in the resolution process is valley-free inter-domain routing [13]. Resolution messages reach tier-1 domains going upwards and then are forwarded downwards *only once*, based on information on server load and network distance that will identify the best possible source. This mode of operation is detailed in the following two subsections on content publication and resolution.

### 4.1 Content Publication

We illustrate the content publication process via Figure 3 which depicts the *domain-level* network topology, with each circle logically representing a domain with a resolver entity as explained. A content provider/owner first registers a new content item by issuing a *Register* message to its local resolver. As a result, a globally unique content ID is assigned to the registered item. The resolver is then responsible for advertising it outside its domain in order to allow discovery by interested consumers anywhere in the Internet. The resolver does this by sending *Publish* messages, following the domain-level *provider route forwarding rule* (i.e., each resolver forwards the *Publish* message to its counterpart in the *provider domain* until it reaches a tier-1 domain resolver). A typical *Publish* message carries the content ID and server location where the content item is actually hosted. The information on server location can be represented as “server-ID@domain”, but not through an explicit IP address. We illustrate the *Publish* message path on Figure 3 (the dashed lines). Each resolver receiving this message updates its content record repository by creating a new entry containing the content ID, the server location information and next-hop domain information, i.e. the neighboring domain from where the *Publish* message has been received. Note that *Publish* messages do not carry server load information which is separately disseminated among resolvers.



**Figure 3: Content publication forwarding process**

Each resolver will update its content record repository upon the reception of *Register* and *Publish* messages. If a *Publish* message is attempting to register a content item already existing in the repository, the resolver will update the existing content entry by adding the new server location information obtained from the newly received *Publish* message. Thus, each registered / published content item has a unique record entry in each resolver repository at any time, pointing potentially to multiple server locations. As an example, Table 1 shows how server location and other information is organized in the content record repository regarding content C1 at the resolver of domain A. Note also that server names need only be unique within their domain.

**Table 1: Content record repository structure**

Content ID	Server location	Next-hop	Server load
C1	S1@A.B.A	A.B	–
	svrA@A.C.A	A.C	–
	svrB@A.C.A	A.C	–

## 4.2 Content resolution

Here we describe how optimized content resolution can be achieved by taking into account dynamic server load conditions. As already mentioned, necessary information on up-to-date server load conditions needs to be disseminated across domains. This means the underlying information-centric network infrastructure is also responsible for scalable dissemination of server load information in order to achieve optimality in locating content servers. This dissemination task is performed by the SNMF functional block described in section 3.

### 4.2.1 Disseminating Server Load Information

We use *Update* messages that can be processed by the resolver in individual domains for disseminating server load condition information. The *Update* messages can be either issued *periodically* (e.g., every 10 min per update) or in an *event-driven* fashion, meaning that the message is disseminated only when a significant server condition change occurs (e.g., sudden surge in server utilization or server approaching overloaded state). In the former case, which is the focus of this work, we assume limited discrete levels of server load conditions, e.g., *Low (L)*, *Medium (M)*, *High (H)*, which are used uniformly across domains. The frequency of *Update* message propagation from

each server can vary between servers and relevant messages from different servers do not have to be synchronized.

The propagation of the *Update* message follows the same rules as the content publication operation, i.e., the *provider route forwarding rule* until a tier-1 domain. The message includes two pieces of information: (1) the server name together with implicit location information and (2) the current load condition. The inclusion of the server’s location (i.e., the name of the domain it is attached to) ensures that each content server is uniquely identified in the repository of any resolver. Continuing with the previous example, we illustrate in Figure 4 the path the *Update* messages follow (i.e., the dotted lines) in order to update their server load conditions. As shown, the server load information is only disseminated up to the root tier-1 ISP network, but not necessarily across the entire Internet.

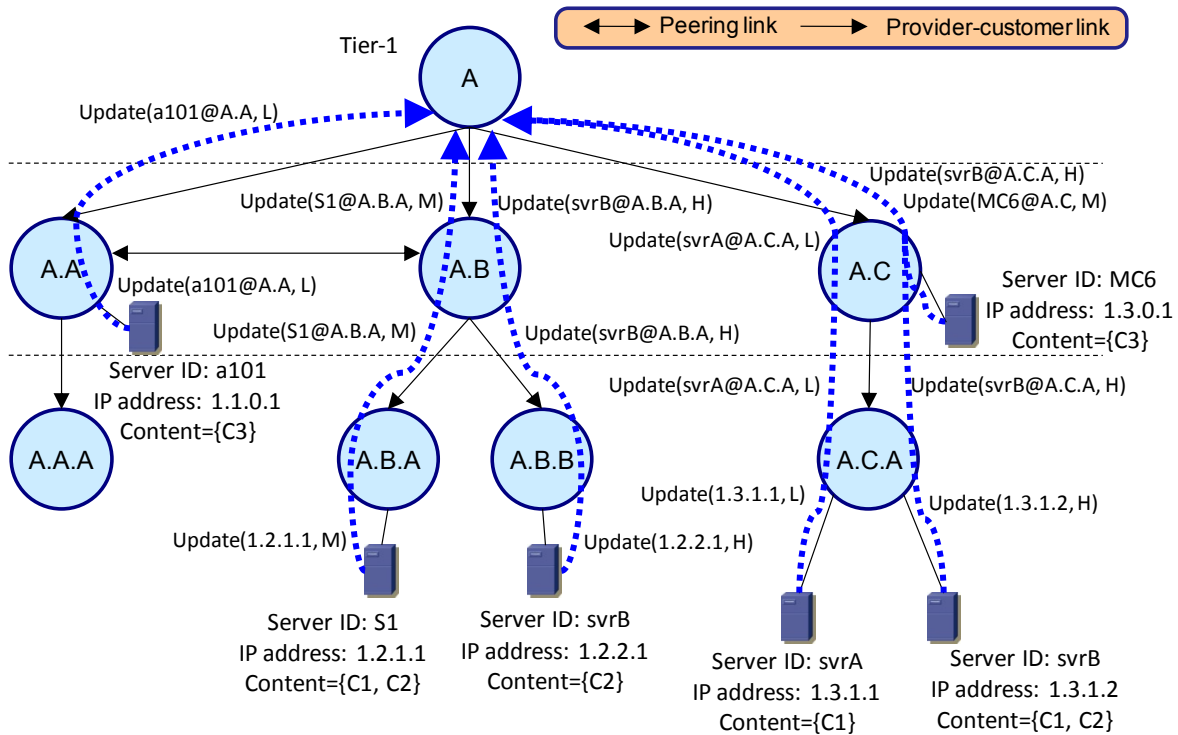


Figure 4: Server-load Update forwarding process

The update of the content record repository is on per content ID, but according to the received server load associated with the server location as the index. In Table 2 we show as example the content record repository of the resolver in domains A and A.C.A after the illustrated publication and a round of updates.

Table 2: Content record repository in tier-1 domain A (top) and stub domain A.C.A (bottom)

Content ID	Server location	Next-hop	Server load
C1	S1@A.B.A	A.B	M
	svrA@A.C.A	A.C	L
	svrB@A.C.A	A.C	H
C2	S1@A.B.A	A.B	M
	svrB@A.B.B	A.B	H
	svrB@A.C.A	A.C	H
C3	a101@A.A	A.A	L
	MC6@A.C	A.C	M
Content ID	Server location	Next-hop	Server load
C1	svrA@A.C.A	1.3.1.1	L
	svrB@A.C.A	1.3.1.2	H
C2	svrB@A.C.A	1.3.1.2	H



#### 4.2.2 Server-load Aware Resolution

Server-load aware resolution is performed jointly by the CRF and CMF blocks of the functional architecture. Specifically, the CRF is responsible for identifying potential content server candidates, while the CMF is the actual decision maker to determine the actual target server by taking into account specific conditions such as server load information and network distance. In general, most server selection policies fall into one of the following two: (1) selection aims at achieving equal load distribution amongst servers and (2) selection strives to achieve equal average access delay at the group of servers. It has been shown in [12] that, in terms of average delay, both approaches perform in increasingly similar fashion when the load increases. In this paper, we use server load as the metric to determine the “best” content server.

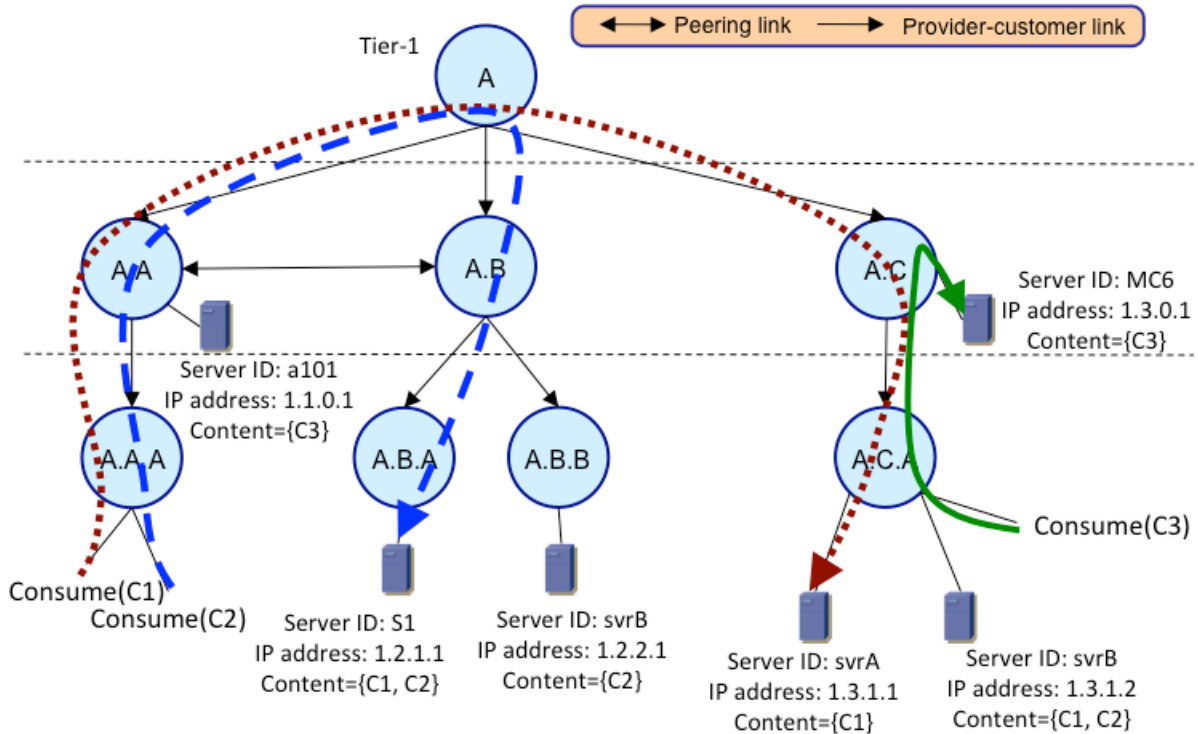
The user initiates the resolution process by sending a content consumption request containing the content ID (through a *Consume* message) to its local resolver. We define two distinct content resolution stages.

- **Uphill** – the forwarding of a *Consume* message from the local resolver “up” along the domain-level provider route until the requested content is first found. In case a domain is multi-homed with more than one provider, the *Consume* message will only be sent to one of the provider domains based on its local policy.
- **Downhill** – the forwarding of the *Consume* message “down” to the domain where the chosen content source resides. Here each domain decides the next-hop “downhill” customer domain as from now on the resolver should already have location information about the target content server.

Following the *provider route forwarding* rule for both the *Publish* and *Update* messages, each resolver has always a bird’s eye view of the content hosted and of the server load condition within its *own* domain as well as in its *customer* domains. When a *Consume* message is received, the resolver checks its content record repository for a matching content record. If none is found, the *Consume* message is forwarded to its provider resolver (uphill). If multiple matching records are found with different (downhill) locations, then the server with the lowest load condition that hosts the requested content is chosen to serve the request. The *Consume* message is then forwarded onwards to the next resolver based on the next-hop domain information recorded. If there is more than one candidate server having relatively low load condition, the resolver may apply additional criteria (e.g., domain-level hop count information inferred via the BGP route). The next-hop resolver follows the same procedure until the *Consume* message reaches the actual server from which case the content flow can be delivered back to the requesting consumer.

We continue to illustrate the resolution process based on our existing example in Figure 5, which shows the resolution path from the user to the selected content source. The routing of the *Consume* message on the downhill path follows the next-hop domain information.

- Request for content **C1** from a consumer in domain **A.A.A** (dotted line in Figure 5): The first matching content record is found in domain **A**. According to **A**’s content record repository, **svrA** in domain **A.C.A** has low server load while the other two candidate servers both have higher load conditions. Thus, the *Consume* message is passed downhill towards **svrA** via **A.C** (indicated as the next-hop domain in the repository (cf. Table 2)). This continues from domain **A.C** to **A.C.A** and finally to **svrA**.
- Request for content **C2** from a consumer in domain **A.A.A** (dashed line in Figure 5): The same operation is executed and in this case, **S1** in domain **A.B.A** is the “best” candidate. Note that there is no conflict even though two different servers shared the same name (i.e., **svrB** in **A.B.B** and **A.C.A**) in different domains.
- Request for content **C3** from a consumer in domain **A.C.A** (solid line in Figure 5): For this request, the first matching entry is found in domain **A.C** and only **MC6** (which has medium load) hosts this content. In this case, despite the availability of a remote server with low load (**a101**) in domain **A.A**, **MC6** is still chosen as the content source to satisfy this request. From this example we can see that it is not the objective to achieve *global optimality* for content server selection, as this requires that all the decision making processes to be made at tier-1 domains. Effectively, there is also a trade-off between optimality of server load conditions and the content delivery paths between the server and the consumer.



**Figure 5: Server-load aware content resolution process**

If a requested content is not found at the tier-1 resolver, then the *Consume* message will be propagated to all the tier-1 peers. An *Error* message is returned to the user indicating a resolution failure if none of the tier-1 resolvers has the requested content entry in their content record repository i.e., this content does not exist anymore.

### 4.3 Content delivery

For content delivery, we require CARs deployed at the network boundary in order to be able to natively process content packets according to their IDs. In fact, the content resolution process described above is only responsible for identifying the best content source through content and network mediation. The actual enforcement of the content delivery paths from the identified source back to the consumer can be achieved in different ways in general. In the simplest possible case, the default shortest end-to-end path could be used. However, in order to support scalability in the distribution of *popular* content to a large group of consumers, we propose to use a state-based approach for enabling inter-domain content multicasting. Specifically, content states can be maintained at CARs to form dedicated content delivery trees to reach individual consumers in the Internet. Such content states are installed during the content resolution phase through the communication between the resolver and the involved CARs. Detailed description on the state-based content delivery can be found in our previous work [5].

## 5. DISCUSSION

Having described first the concept of content mediation and the functional view of the mediation architecture, we presented in some detail the coupled approach for content access and delivery with server load awareness. Both the decoupled and coupled approaches are investigated in detail in the context of the EU COMET project – Content Mediation Architecture for Content-Aware Networks [14]. The viability of both approaches has been initially validated through testbed implementation while measurements and additional simulation experiments are investigating scalability and other performance aspects and are feeding back to the design for the next cycle. Based on our experience so far, we discuss here deployment and other issues regarding our content mediation approach and extrapolate to information-centric architectures in general.

### 5.1 Deployment Aspects

The minimum possible deployment of an information-centric approach is to unify content resolution for all Internet content and provide access to the best copy in anycast fashion. This can be done through a “web” of content resolution servers through which all content will be first published and resolved when requested. In fact, this is what we had first thought when we started thinking of a content-resolution layer over the current Internet a few years ago, codenaming this approach DNS++, as it is essentially a content-oriented DNS-like system. This is exactly what the content mediation servers do when resolving content in the decoupled approach: in this case, the content resolution function of Figure 2 is implemented through separate physical servers administered by the mediation servers, with the latter choosing the best possible copy in case of replication. The decision can take into account network distance, server load and possibly end-to-end path quality based on average load monitoring. If the content is delivered through default BGP routing, this infrastructure performs simply intelligent content resolution. Going a step further, if content-aware routers are deployed in domain edges, a different end-to-end delivery path than the default BGP one may be configured by the CMSs for better quality; in this case, the CMSs simply override the end-to-end BGP routing based on path quality monitoring which is available in the mediation plane. But the “vanilla” functionality of the decoupled approach does not require CARs or any network modification when using the default the end-to-end BGP routing, only end systems need to be modified with the relevant protocols to interact with the content mediation plane for both content publication and consumption.

Coming to deployment issues, such an infrastructure does not need to be ubiquitously deployed at once but it may be deployed incrementally by some domains/providers who will want to offer CDN-like services to their customers. Content that needs to benefit from this infrastructure will need to be registered through the local CMS but also content in “non-compliant” domains will be able to register with a remote CMS of another domain; the same is also the case with end users as they could access content through a remote domain CMS if their local domain is non-compliant. If though only isolated “islands” of this infrastructure are deployed, anycast selection decisions will rely mainly on network distance. In case of full deployment, server load and path quality may be also taken into account, with delivery potentially influenced through domain-edge CARs when the latter will start being deployed.

The coupled approach which we presented in this paper in some detail requires CARs to be deployed in domain edges and exploits them for content caching in order to localize access in a similar manner to native information-centric architectures but in this case this operation takes place in the mediation plane. While replication in the decoupled approach is mainly proactive in a CDN-like manner, i.e., with the content provider deciding where to place popular content replicas and registering them accordingly, in the coupled approach there is mainly reactive caching in CARs, according to guidelines set by the CMSs. In fact, the coupled approach goes two steps further in terms of information-centricity, planting state information in the CMSs in order to aid resolution – an approach also known as “breadcrumbs” in the literature [15] – and combining routing with resolution in a hop-by-hop manner as in most native ICN approaches. The advantage of the tight coupling is that caching is more easily supported but the approach also requires the deployment of CARs in network edges.

In fact, we see the decoupled approach deployed first, in a pure content-resolution-only fashion. In the second stage, CARs will be deployed in network edges for supporting both optimized delivery in the decoupled approach and also for enabling the parallel deployment of the coupled approach for popular content. In the third stage, and as CARs start also being deployed within the network, there could be a move towards a fully information-centric

approach, with native in-network content resolution and access through a native information-centric future Internet protocol. In this case, the mediation plane will be fully collapsed into the network.

## 5.2 Scalability

Scalability in information-centric infrastructures is a key issue given that there exist already more than 1 trillion ( $10^{12}$ ) unique URLs and we expect possibly 1000 trillion content objects ( $10^{15}$ ). It has been shown that it is possible to operate DHTs with more than 2 million nodes [16], so a hierarchical DHT structure should be able to cope with scale in the decoupled approach given large amounts of dynamic random access memory in the content mediation servers or natively in the content aware routers. Scalability for content resolution is in fact inextricably related to content naming and the DHT approach applies to flat self-certified names and the hierarchical DHT one to semi-hierarchical ones. On the other hand, fully hierarchical approaches that can scale are also possible, although these inevitably include some location information and, as such, are not appropriate for dynamic approaches with caching anywhere. Given that our decoupled approach is mostly a CDN-like approach with proactive replication instead of caching, we have chosen a hierarchical naming approach that can scale well. On the other hand, the coupled approach uses flat naming and may also scale given that we apply it to only popular content.

The reason we apply the coupled approach to only popular content is that it advertises subordinate domain content all the way up to tier-1 domains. As already mentioned, we have in mind deploying the decoupled approach for the vast majority of content objects, with the coupled approach applied to only *popular* content. Content popularity could be monitored and as soon as content is deemed popular, it should revert from the decoupled to the coupled approach, with the relevant ID advertised all the way up to the tier-1 level. In addition, content providers expecting some content to be highly popular (e.g., Olympic games coverage, etc.) could choose to publish their content through the coupled approach in the first place. The content mediation plane needs to also keep globally track of which content has become popular, so that resolution and delivery for a requested content follow the appropriate approach. Content may also lose its popularity status, with access reverting back to the decoupled approach. We intend to investigate aspects related to content popularity and to switching between the decoupled and coupled modes of operation in our future work.

Another aspect related to scalability is that of multi-homed servers. In this case, content access can take place from different “access” networks depending on the location of the consumers and the relevant end-to-end path. If no server load is advertised, there is absolutely no scalability implication for a server to be multi-homed. But with server load information advertised through more than one networks, this server load information will be present in different branches of intermediate ASes, in addition of course to the tier-1 ones, and may attract additional requests if the server load appears low and popular content is hosted. In this case, and disregarding in-network caching which will alleviate the situation, server overload may occur until the new increased server load is advertised through the global network. The solution in this case is to enforce more frequent server load updates for multi-homed servers and to also possibly exercise additional admission control policies. We intend to investigate relevant issues in more detail in our future work.

Finally, a final aspect related to scalability is delivery path optimization. Given that state-based reverse path forwarding is used in the coupled approach, the actual delivery path maybe suboptimal. In fact, there may be a tromboning effect, with a delivery path going up all the way to a tier-1 domain when a shortcut through a peer domain is possible. Given that the coupled approach applies to popular content which could be massively accessed, suboptimal delivery paths may have an influence on network performance and scalability, despite the fact that caching may help due to localized access. This problem may be alleviated through routing optimization as soon as delivery starts given that domains know from BGP routing that there exists a shorter path through a peering route towards the IP prefix to which content is delivered. Use of the shorter path can be “forced” through an explicit *Consume* message from an interim domain but relevant details are outside the scope of this paper.

## 5.3 Integration with Search Engines

Given that such an information-centric architecture will be gradually deployed, it will coexist for a considerable time with the current Internet and with content accessed through location-dependent URLs, with some CDN-style redirection “behind the scene” as is the case today. Content is currently searched through web search engines

which index web pages, e.g. google, and through intermediary-specific search engines which index intermediary-specific content, e.g. youtube. The problem with this approach is fragmentation of the search space, as discussed in the beginning, and a holistic information-centric architecture should attempt to unify all content. The content mediation plane could provide content search functionality for its content but, given that “external” content will coexist with mediation plane content for a considerable time, mediation plane administered content should be also indexed through external search engines.

Indexing of mediation plane administered content could be done in two different ways. First, in a proactive manner, with a dedicated web page created for every new published piece of content in an automated fashion. In this case, the web page will contain the content name/ID and the latter could be displayed with the search results and the page URL, so that client software can pick up the name/ID without having to access the web page first. In fact, if a similar approach was followed by today’s intermediaries, it could result in a unified content search space, with all content being accessible through web search engines. Second, in a reactive manner, with hooks provided by the CMSs allowing external search engines to index content administered through the content mediation plane and providing the name/ID to consumers, in a similar fashion to today’s URL. This approach is better as it aligns with the long-term strategy in which the operator-provided content mediation plane will integrate all content and will also provide hooks for external search engines.

## **6. SUMMARY**

With the vast majority of interactions over the Internet being related to content access, information-centric networking approaches propose a paradigm shift from a host-oriented to an information-oriented Internet. In this context, the content mediation approach presented here represents an evolutionary path towards the eventual realization of native information-centric network operation. Starting from a unifying content resolution infrastructure over the current Internet, it can gradually extend to information-centric network operation, initially through content-aware routers in domain borders and eventually through ubiquitous deployment. The decoupled approach is used for the majority of Internet content with the coupled approach used for popular content and benefiting from caching in content-aware routers. This infrastructure uses server load status to select the best content copy in an anycast fashion, taking also into account network distance and possibly end-to-end path conditions. This infrastructure is provided by the ISPs in a collaborative manner as a tightly-coupled overlay with the network, benefiting from intimate network information which can be used for network mediation. Server load information is also used for content mediation and this requires the cooperation of content providers.

As already mentioned, there are distinct differences between conventional loosely coupled overlay content infrastructures (e.g., CDNs) and tightly-coupled ISP-operated ones like the one presented here. In a CDN environment, the CDN provider is able to have complete control over the overlay infrastructure and of the content servers. As such, based on the knowledge of the content location and of server load conditions, incoming consumption requests can be strategically directed to the best content sources. On the other hand, because the overlay infrastructure is network-agnostic, the resolution/delivery processes cannot take into account network-level routing information. In ISP-operated content platforms, the key technical challenge is that there is no central entity to perform global content diffusion, but individual participating ISP networks need to collaborate with each other in order to accomplish this on an Internet-wide basis. With such an information-centric infrastructure, the global Internet becomes a native CDN, localizing traffic due to content caching, avoiding flash-crowd situations, enhancing end used QoE and allowing almost everybody to become a content provider without the cost or the supporting the redirection infrastructure required today. Last but not least, network providers will be able to take part in the content market through suitable business models, increasing their revenues and being able to invest more in their networks which is not possible today given their constantly eroding profit margins.

## REFERENCES

- [1] T. Koponen et al., "A Data-Oriented (and Beyond) Network Architecture" *Proc. ACM SIGCOMM 2007*, Aug. 2007.
- [2] V. Jacobson et al., "Networking Named Content," *Proc. of ACM CoNEXT 2009*, pp. 1-12, 2009.
- [3] P. Jokela et al., "LIPSIN: Line Speed Publish/Subscribe Inter-networking," *Proc. ACM SIGCOMM '09*, Aug. 2009.
- [4] B. Ahlgren et al., "Design Considerations for a Network of Information," *Proc. of ReArch 2008: Re-Architecting the Internet*, Madrid, Spain, Dec. 2008.
- [5] W. K. Chai et al., "CURLING: Content-Ubiquitous Resolution and Delivery Infrastructure for Next-Generation Services," *IEEE Communications*, pp. 112-120, March 2011.
- [6] J. Seedorf and E. Burger, "Application-Layer Traffic Optimization (ALTO) Problem Statement", IETF RFC 5693, October 2009.
- [7] The IETF Application Layer Traffic Optimization (ALTO) Working Group, <http://datatracker.ietf.org/wg/alto/charter/>.
- [8] H. Xie, Y. Yang, A. Krishnamurthy, Y. Liu and A. Silberschatz, "P4P: Provider Portal for Applications", *Proc. of ACM SIGCOMM 08*, Seattle, WA, USA, August 2008.
- [9] I. Psaras, R. Clegg, R. Landa, W. K. Chai, G. Pavlou, "Modeling and Evaluation of CCN Caching Trees", *Proc. of IFIP Networking 2011*, Valencia, Spain, May 2011.
- [10] W. K. Chai, D. He, I. Psaras, and G. Pavlou, "Cache 'less for more' in information-centric networks," *Proc. of IFIP Networking 2012*, Prague, Czech Republic, May 2012.
- [11] E. Zegura et al, "Application-layer Anycasting: A Server Selection Architecture and Use in a Replicated Web Service," *IEEE/ACM Transactions on Networking*, vol. 8, no. 4, pp. 455-466, Aug. 2000.
- [12] T. Wu and D. Starobinski, "A Comparative Analysis of Server Selection in Content Replication Networks," *IEEE/ACM Transactions on Networking*, vol. 16, no. 6, pp. 1461-1474, Dec. 2008.
- [13] Lixin Gao and Jennifer Rexford, "Stable Internet routing without global coordination," *IEEE/ACM Transactions on Networking*, December 2001, pp. 681-692.
- [14] G. Garcia, A. Beben, F. J. Ramon, A. Maeso, I. Psaras, G. Pavlou, N. Wang, J. Sliwinski, S. Spirou, S. Soursos and E. Hadjioannou, "COMET: Content Mediator Architecture for Content-aware Networks", *Proc. of Future Network and Mobile Summit 2011*, Warsaw, Poland, 15-17 June 2011.
- [15] E. Renshaw and J. Kurose, "Breadcrumbs: efficient, best-effort content location in cache networks," *Proc. of IEEE INFOCOM Mini-Conference 2009*, Rio de Janeiro, Brazil, 20 April 2009.
- [16] M. Steiner, T. En-Najjary, and E. W. Biersack, "A Global View of KAD", *Proc. of ACM Internet Measurement Conference (IMC 2007)*, San Diego, CA, USA, October 2007.