García-Reinoso, Jaime; Fernández, Norberto; Vidal, Iván; Arias Fisteus, Jesús (2015). Scalable Data Replication in Content-Centric Networking based on Alias Names. *Journal of Network and Computer Applications*, (2015), v. 47, pp.: 85-98.

DOI: https://doi.org/10.1016/j.jnca.2014.10.003

Scalable Data Replication in Content-Centric Networking based on Alias Names

Jaime Garcia-Reinoso[a,*], Norberto Fernández[a], Ivan Vidal[a], Jesús Arias Fisteus[a]

[a] *Universidad Carlos III de Madrid. Avda. de la Universidad 30 28911 Leganés - Madrid (Spain)*

Abstract

Content-Centric Networking (CCN) is a clean-slate proposal to redesign the current Internet by focusing on the content itself, instead of the classical computer-to-computer communication. In this paper we address scalability issues of the Forwarding Information Base (FIB) in CCN. Our solution proposes both the use of hierarchical names assigned by access providers and a novel *alias name* architecture. With the former, we allow the aggregation of entries at the routing tables of CCN content routers, while the latter reduces the processing load at those routers when replicas exist in different parts of the network. With some minor changes to the original proposal, we provide a scalable solution for data replication in CCN, which inherently supports content mobility at the same time. We validate our scheme by (1) comparing the scalability of CCN against our proposal and by (2) implementing and testing a proof-of-concept software based on CCNx, to prove the viability of this approach.

Keywords: Content-Centric Networking, Data Replication, Alias Name

1. Introduction

Nowadays, the Internet users have a plethora of applications to download content, like web browsers, file transport applications, peer-to-peer (P2P), etc. After the users provide the identifier of the content, how the application downloads it is transparent to them: the content may be stored in a single server

*Corresponding author
Email addresses: jgr@it.uc3m.es (Jaime Garcia-Reinoso), berto@it.uc3m.es (Norberto Fernández), ividal@it.uc3m.es (Ivan Vidal), jaf@it.uc3m.es (Jesús Arias Fisteus)

located “far away” from the user; it may be replicated in several servers if the service provider is using a CDN (Content Delivery Network); or, in the case of P2P, different parts of a file could be downloaded from different peers. Independently of how the application downloads content, there is an end-to-end communication where intermediate routers are just used to forward packets. In other words, the Internet (as opposite to users) cares about end-to-end communications, not content.

This model focused on computers instead of content has several drawbacks. First of all, it is necessary to secure the content exchange to guarantee confidentiality, integrity, availability, authenticity and non-repudiation of the content. Second, as communications are usually point-to-point (multicast is only available in certain networks) in the current Internet, it is not possible to use replicas of the objects in case they exist, and overlay networks (CDNs or P2P for example) have to be built on top of it to use these distributed replicas. Third, the Internet protocols (both IPv4 and IPv6) use one single locator/identifier value to route and identify computers. This is the main problem we have to face for mobility communications, as when a device changes its point of attachment to the network it must change its locator address. In IP, when a mobile node changes its IP address it is changing its locator as well as its identifier, which is not the desired behavior.

In recent years, several projects and organizations have proposed minor and major changes to Internet protocols to minimise or eliminate the aforementioned problems. IPsec (IP security) [1], TLS (Transport Layer Security) [2] and MIP (Mobile IP) [3] are examples of standards proposed to overcome some of those problems. Other initiatives are still under discussion or with little penetration like LISP (Locator/ID Separation Protocol) [4], RELOAD (REsource LOcation And Discovery) [5] and PPSP (P2P Streaming Protocol) [6] for example.

All the previous initiatives are focused on modifying or enhancing existing protocols, but other works try to go beyond that by proposing clean-slate approaches. Among them, we will focus on Information-Centric Networks (ICN), where everything is built around the content itself, independently of where it is stored, and based on *Publish/Subscribe* messages. Although there are different proposals around the ICN concept [7, 8, 9, 10], almost all of them have to

solve four main problems [11]: (1) the naming structure of the content, (2) the mechanism to find the content, (3) how to deliver the content to the requester and (4) caching the content inside the network. The authors in [12] present a survey describing the most important ICN proposals, including a comparison of the key functionalities described before. For example, and very related with our paper, the survey presents a comparison between hierarchical and flat naming, where the authors conclude that the former allows scalability when aggregation is possible while the latter avoids the location-identity binding. In other words, both approaches have their advantages and disadvantages, so other alternatives are necessary.

One significant initiative in the ICN paradigm is Content-Centric Networking (CCN) [10], which uses a hierarchical name scheme similar to Uniform Resource Identifiers (URIs), such as `/es/uc3m/it/joe/documents/paper.tex`. In addition, every individual Content Router (CR), which is a router with caching capabilities, has to know how to forward Subscribe messages (or Interest in CCN terminology) using its own Forwarding Information Base (FIB) table. With potentially billions of objects, the scalability of the FIB is a clear issue, and further study is necessary in this particular point. This can be even more problematic as content can be replicated, and replicas with the same content name can be distributed among different parts of the network domains [12]. Replication has also impact in CCN routing, which has to use a *Strategy Layer* in order to retrieve the content from the best source (with the lowest delay or the highest bandwidth, for example). In the case of core CRs receiving millions of packets, it would be advisable to do the source selection at the network end points, reducing the complexity of those intermediate nodes.

To solve the issues related to the scalability of the FIB and the processing overhead at the CRs, in this paper we propose a scalable data replication scheme for CCN. Firstly, our solution uses hierarchical provider-assigned names to facilitate aggregation, as it has been suggested in [13]. This aggregation comes at the cost of requiring different CCN names for the same content replicated in different provider networks. Secondly, a novel *alias name* architecture is introduced, so that replicas in different parts of the network with different names can be identified as objects with the same content. We extend the Interest packet

format, including a new field to transport an alias name as well as the content name. The advantage of our proposal is that CCN routers can check if the requested content is stored in its cache or not, as the content name is carried in the Interest. This way, consumers can select the alias name they want to use to retrieve each individual piece of data, or even try different alias in parallel, maintaing the benefits of using caching in the routers. Altogether, the proposed mechanisms achieve a scalable data replication, as FIB tables do not have to include entries for replicas.

To accomplish these goals, we introduce a new functional entity in the network called the *Alias Name Manager* (ANM), which could be placed by the access service provider inside its own domain network (but it can be anywhere in the network). Apart from the ANM, we extend the basic CCN proposal by introducing the notion of *Alias Routing Name* (ARN), which can be included as an optional field in CCN Interest packets, as stated before.

Besides the scalability advantages of our data replication scheme, it inherently supports mobility of content. When a content is moved to a different network it is assigned an alias name, which in turn has to be registered in the ANM. This entity allows accessing the content by using its original name.

The rest of this paper is structured as follows. In section 2 we present a survey of CCN to provide some background for the rest of the paper. In section 3 we describe our proposal explaining in detail all the modifications introduced to CCN. Section 4 compares regular CCN against our scheme in terms of scalability, by means of simulations. Section 5 presents a proof-of-concept software, implemented to show the feasibility of our proposal, and to evaluate its performance in terms of delay, when multiple copies of an object are replicated in several domains. Finally, section 6 closes the article with the main conclusions and the future work.

2. Background on Content-Centric Networking

Content-Centric Networking (CCN) [10] is a novel clean-slate design of the Internet, based on the concept of named content. Like other ICN initiatives, it focuses on retrieving the content by its name, instead of locating and establishing a communication with the end host that holds the content (as in the current

Internet). In CCN, names are hierarchically structured into a set of components, and applications can choose any naming convention for an appropriate operation. As an example, the CCN name `/es/uc3m/it/research/papers/paper.pdf/_v2/_s1`, could be used by an application to retrieve the first segment of version 2 of this paper. In this example, the convention followed by the applications dictates to use the marker *_v* to indicate the version number and the marker *_s* to identify the file segment. CCN names with subsequent segment numbers would allow the application to retrieve the whole paper for display. On the other hand, from the perspective of the CCN transport, names are opaque (i.e. the transport does not need to understand name semantics) and are composed by a set of binary encoded components.

CCN defines two types of packets, *Interest* and *Data*. When a receiver decides to retrieve a given content, it generates and sends an Interest packet that includes the CCN name of the desired content in a field that, for convenience, we will name Content Name Identifier (CNI) from now on. The Interest packet is routed by CCN routers towards a source of the specified content. If this packet reaches a node (i.e. a source or an intermediate CCN router) that holds some content that matches the Interest, this node can directly answer back with a Data packet including the desired content. A content matches an Interest packet if the content name includes the CCN name indicated in the Interest.

Figure 1 illustrates the forwarding model of a CCN node. Whenever an Interest packet is received on a face (network or logical interface), if the CCN node cannot satisfy the Interest, it stores the CCN name included in the Interest and its incoming face in a Pending Interest Table (*PIT*). Then, the Interest is forwarded to the next hop towards a source of the content, according to the information stored in a Forwarding Information Base (*FIB*). In case several faces to the content exist, a router has to select the proper one by running the algorithm implemented by its *Strategy Layer*, which selects the optimal next hop to use. This face selection implies extra processing at the content router.

The FIB can be built at each node by the execution of a routing protocol that, similarly to current IP networks, would be used to propagate content name prefixes between CCN routers (routing protocols such as OSPF or BGP could be adapted to this end [10]). Apart from content name prefixes, it is also

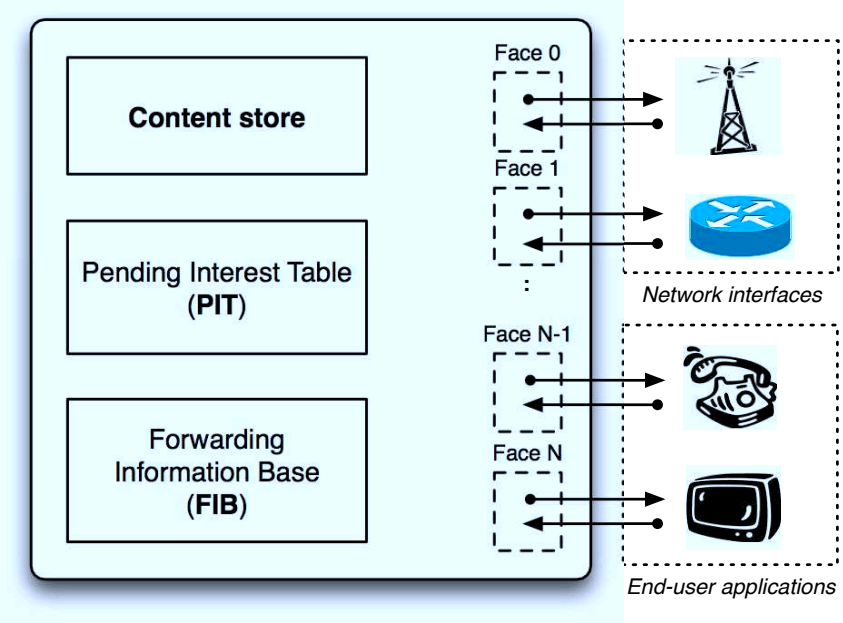


Figure 1: Forwarding model in CCN

possible to register content in a domain different from the original provider (i.e., when the prefix of the home provider does not match the names published by the visiting entity). In those cases, when the routing protocol is executed by the content routers, some of those CCN routers will not be able to aggregate names at their FIB. This occurs when the home domain of the content, and the visited domain where the content is located, are reachable from different faces of a given CCN router. This may lead to scalability issues in the FIB of the CCN routers as will be shown in Section 4.

When an Interest reaches a node that maintains a matching content, that Data packet is transmitted in response. This Data packet is routed back to the receiver via the reverse path followed by the Interest, based on the information stored in the PITs of the traversed CCN routers. To reduce bandwidth consumption and network delays of subsequent requests of the same content, each CCN node that receives the Data packet caches it into a Content Store (*CS*). To take into account the limited storage capacity of the CS, the replacement of Data packets can be done attending to LRU (Least Recently Used) or LFU (Least Frequently Used) policies, with the aim of preserving the most demanded

content in the cache.

Some works have studied the performance of CCN, mainly due to the caching system for a single path [14] and multi-path [15]. Other authors have provided a comparison between IP-based solutions and ICN, in particular between CDN and CCN [16, 17], where the results reflect that the benefits of CCN depend on the particular scenario under analysis. Furthermore, CCN presents other benefits compared to IP, like integrated consumer mobility, content authentication, content replication, etc.

3. Scalable Data Replication in CCN

In [18], the authors study the viability of deploying CCN in nowadays Internet, concluding that there are still several challenges to adopt it globally. One of these challenges is the scalability of the FIB and the routing in general. As we stated in the introduction, our work extends the CCN proposal to enhance the routing process, allowing a scalable data replication. The main issues in CCN routing are:

1. Although CCN proposes hierarchical names, the routing lacks this hierarchical structure, which implies that two names with common prefixes (or even the same name) could be located in different parts of the network. This means that, in general, entries in the FIB cannot be aggregated, and several faces per entry could be necessary, as it was explained in section 2.
2. In case the same content is available in different parts of the network, a CCN node may include this information in its FIB. In such cases, when a node has to forward an Interest, it has to use the *Strategy Layer* to choose the best face to send it to. This mechanism forces CCN content routers to maintain more information about the state of their faces, store the delays for different replicas and perform more operations to properly route Interests.

Regarding the lack of hierarchical structure in routing, the NDN (Named Data Networking) project (which is based on the CCN proposal) proposes in [13] to use ISP-based aggregation naming as a first approach. Our initiative

departs from that idea, dividing a CCN name in two parts: a globally-routable prefix assigned by the ISP and a name suffix. The former is a prefix that enables aggregation and is used to populate the FIBs of inter-domain routers. The latter is used for intra-domain routing purposes. With this solution, the FIB table of core content routers is scalable, independently of the number of replicas. Inside ISP networks, providers can choose to use regular CCN in case the number of objects is under a certain limit, or they can add further levels of hierarchy.

With respect to data replication, we have defined a novel scheme whose main concepts and components are described in the next sections. To illustrate this description, we will use the scenario that is depicted in Figure 2, where a content from `/es/uc3m/` is replicated in another domain (`/com.example/`).

3.1. Alias Name

In CCN, authors simply state that a content has a name, but in this paper we have to differentiate between several options. When an object is created, the user has to assign it a *Content Name* (CN). ISPs have to delegate unique prefixes to each of their users, so they can generate unique CNs too. If the object has to be replicated by the user in other places of the network or in case the object is downloaded by other users, the new copy has to be assigned an *Alias Name* (AN), which is a valid name in the network where the new copy is located at. The AN has to be generated as a normal name, using the prefix delegated by the ISP to the user and the suffix the user wants. Notice that in the original content, the AN and the CN are exactly the same.

3.2. Alias Name Manager

In case several copies of the same content exist, it is necessary to give this information to consumers so they can use them to accelerate the download process. The *Alias Name Manager* (ANM) is in charge of storing the mapping between ANs and CNs. The concept of the ANM is similar to the concept of *tracker* [19] in P2P networks, where peers publish or register their own contents while other peers can retrieve a list of peers serving a given content. ANMs can be deployed using different strategies: at the own client, at the access CR, as a single server or using an un-centralized and scalable approach as DHTs (Distributed Hash Tables). How this entity will be deployed depends on its

estimated load (number of registrations and requests per unit of time) and can be decided by each provider or end user, at different domain levels.

In case organizations want to provide access to replicas of their own domain, or even to the objects of their itinerating users, they could provide ANMs using *well-known names*. Given an object CN, an ISP can implicitly indicate the well known ANM by inserting an empty component `//` separator in the delegated prefix name. The name of the well known ANM can be algorithmically derived by inserting the string `mgmt/replicas/` after the empty component. For example, as shown in Figure 2, for the content with name `/es/uc3m//it/joe/video/a.mpg` the default ANM can be accessed to look for replicas of that content with the name `/es/uc3m//mgmt/replicas/it/joe/video/a.mpg`.

Alternatively, the same organization may have several ANMs for different sub-organizations. For example, the sub-organization `/es/uc3m/it/` may decide to use its own ANM by registering the entry `/es/uc3m/it//mgmt/replicas/` in the FIBs of the appropriate CRs and publishing all its content under the prefix `/es/uc3m/it//`.

3.3. Content Metadata

We define a set of information items describing a content, which we call *metadata*. As a naming convention, a client application can retrieve the metadata about a content by concatenating the *_meta* special component to the name of that content. For example, for the video `/es/uc3m//it/joe/video/a.mpg` nodes will also answer to Interests for `/es/uc3m//it/joe/video/a.mpg/_meta` with the corresponding metadata. The Data packet containing the metadata is signed by the publisher serving the content. Metadata includes the following information:

- Name of the content item or Alias Name (AN).
- Original name of the content or Content Name (CN).
- Name of an ANM, in case the publisher wants to use a manager different to the well-known ANM.
- List of other alias names for this content, if the node is also acting as the ANM for this content.

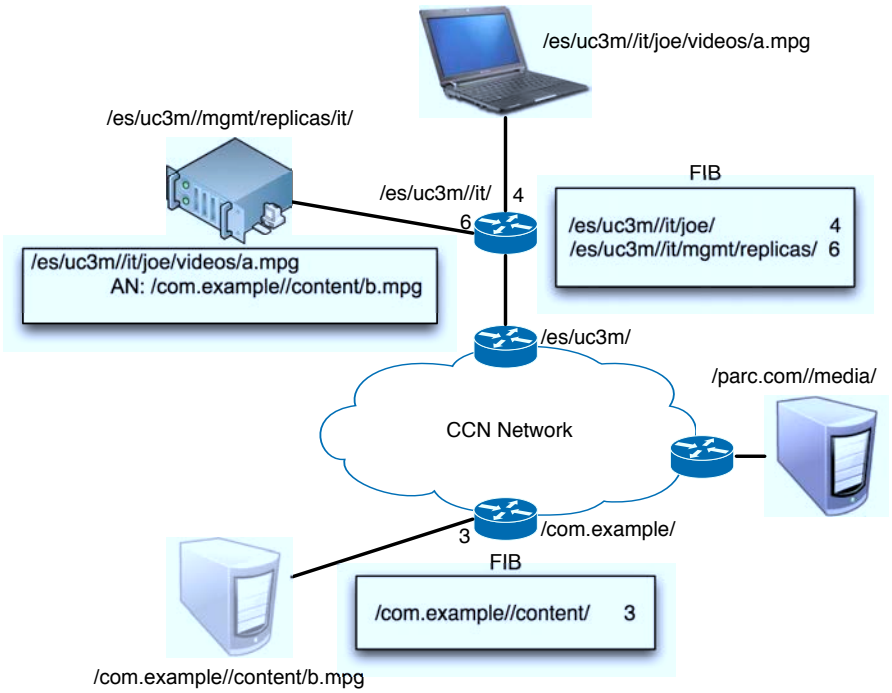


Figure 2: Scenario with one replica

- Indication about whether all or only some pieces of this content are available locally. In the latter case, bitmap encoding the specific pieces that are available through this replica.

This information will be used for the management of replicas and mobility, and is also used to authenticate the provider of this content. Interests for metadata packets must be configured with the appropriate values to prevent them from being served from the content store of intermediate nodes, to get an up to date list of replicas. This behavior can be forced by using the do-not-answer-from-content-store combination in the *AnswerOriginKind* field of the Interest packet, for example (see `https://www.ccnx.org/releases/latest/doc/technical/InterestMessage.html` -July 2014-).

3.4. Registering Alias Names

Nodes that have a full or partial copy of a content item may choose to become providers (replicas) for that content. Note that those nodes need to store

not only the content itself but the whole data packets as they were received, including the original signatures, because they are expected to send them to other consumers. This storage function can be done by a middleware at the replica source itself.

In our proposal, nodes containing replicas have to register their AN in the ANM associated to the original CN. The ANM name is obtained by using the one provided by the original source in the metadata information. When it is not provided there, or when the original source is down, the well-known ANM name is used instead. In order to maintain the flexibility of nodes to decide which ANMs to use for their contents, first of all consumers have to send an Interest to the original content name, plus the *_meta* suffix, to get the metadata. If no response is obtained, they have to send an Interest to the well-known ANM name. Both Interest packets may be sent in parallel if the client wants to reduce the time needed to get the list of replicas.

To register an AN, the replica source has to obtain a CCN name generated by the ANM for the content name, as in step (1) in Figure 3 (in the next flow diagrams we enclose inside parentheses a comma separated list with the most important fields in a Data packet). That name has to be treated as opaque by the replica. For example, it may be `/es/uc3m//mgmt/replicas/it/a6ff510fd`. The replica source has to append the alias name of the replica to that opaque name, sending an Interest to it, with instructions to not being served from the content store of intermediate nodes. For instance, if the name of the replica is `/com.example//content/b.mpg`, the Interest would be sent to `/es/uc3m//mgmt/replicas/it/a6ff510fd/com.example//content/b.mpg`. The ANM receiving this Interest will answer back with a Data packet including an identifier the ANM will use in the next step and the time period the replica is valid, as in step (2) in Figure 3.

Using mechanisms outside the scope of this work, the ANM will decide whether it accepts the request or not. However, before accepting any replica, it must send an Interest packet to authenticate the replica source by using the identifier supplied to the replica source in the previous step. In the example above, the Interest would be sent directly to `/com.example//content/b.mpg/e1f310ab8`, as presented in Figure 3 in step (3). The replica source must reply back with

a data object, which includes the string challenge signed with its own private key, to properly authenticate itself against the ANM.

The ANM keeps an up to date list of replicas by using a soft state procedure. The replica source has to refresh its replica information in the ANM by repeating the replica registration steps (last two steps of Figure 3 which are the same as steps (2)-(3)). If no message is received after a given timeout (provided by the ANM in the second step) the replica is discarded by the ANM.

3.5. Routing

In regular CCN, when a node has to forward an Interest it has to use the incoming content name identifier (CNI) to perform a longest prefix match with its FIB. In our proposal, the Interest message may include an optional field called the *Alias Routing Name* (ARN). In case a consumer receives a list of ANs, for instance after contacting the ANM, it has to construct one Interest per piece of the data object. It is up to the consumer to decide which AN to use for a given data object:

- If the original copy is selected, the regular CCN procedure is followed. In this case, the consumer has to include the CN of the content in the CNI field of the Interest.
- If the consumer selects a replica, it has to include the AN of the replica in the optional ARN field of the Interest. The CN of the content is also included in the CNI field.

At CCN CRs, the forwarding scheme is exactly the same as in regular CCN for Interests with one field name, but it is slightly different when the new ARN field is present. In those cases, the CCN node uses the CNI field for the local Content Store lookup. If there is a local match, the Interest is consumed by sending the data object, thus finishing the process. If the data object is not stored locally, the ARN, contrary to regular CCN, is used to perform a longest prefix match in the FIB, returning the proper outbound face. Lastly, independently of the presence of the ARN field, the tuple <CN, incoming face> will be added to the PIT. Summarizing the routing behavior, the only change introduced in the routing process at a CR is for the outbound face selection, where the ARN has to be used instead of the CNI, if present.

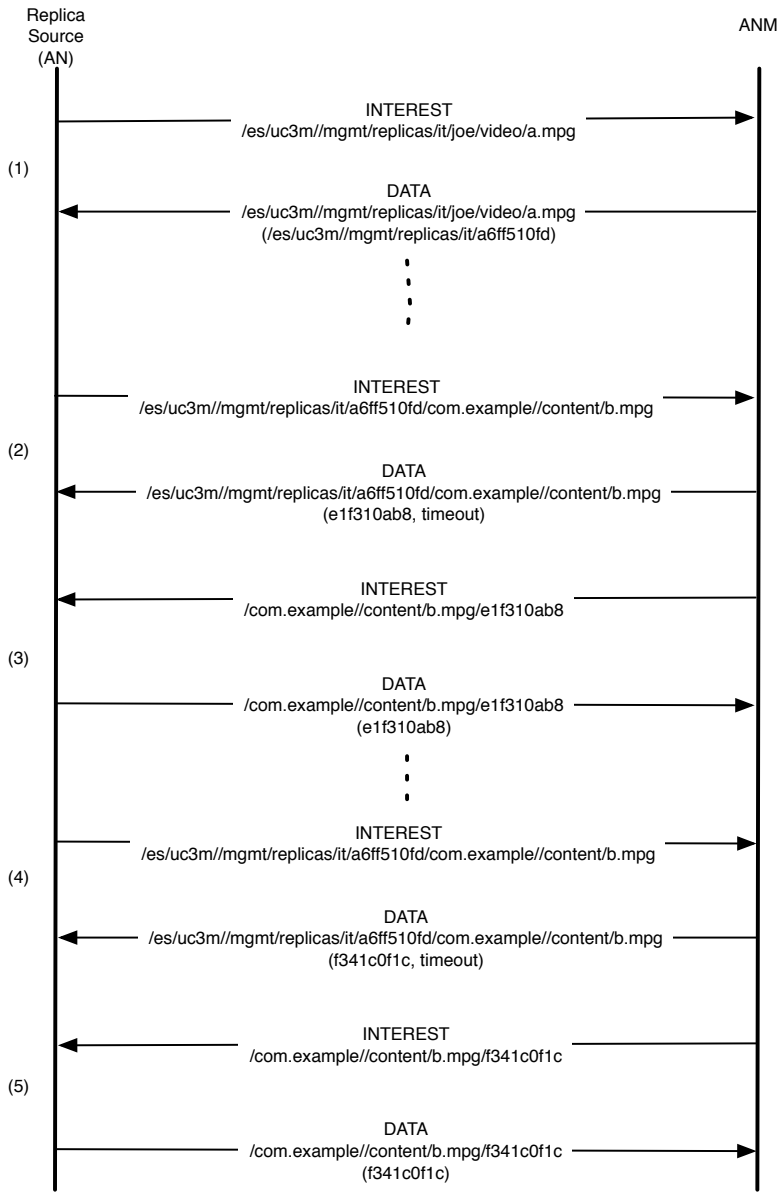


Figure 3: Flow diagram for a replica registration

3.6. Use Case

Figure 4 presents a flow diagram of a request when the original source delegates replicas to its domain server and there exists one replica source. The consumer must execute the following process:

1. A consumer sends an Interest to get the metadata of the content item (e.g.

`/es/uc3m//it/joe/video/a.mpg/_meta`). The response Data packet may include the ANM name, although in this example we assume that the well-known name `/es/uc3m//mgmt/replicas/it/joe/video/a.mpg` is used. Optionally, metadata may also include a list of replicas (when the node behaves as the ANM for its own content). However, we assume that this list is not provided in this example.

2. As soon as the consumer receives the metadata, it can start requesting actual segments with the original name CN.
3. With the ANM name, the consumer may request the list of replicas using that name. In our example, the consumer sends an Interest with the name `/es/uc3m//mgmt/replicas/it/joe/video/a.mpg`. The Data packet with the answer includes the name the consumer has to use in case it wants to register itself as a replica (as it was explained in section 3.4) and the list of replicas, if any. In our example, just one replica source has registered in the system with name `/com.example//content/b.mpg`. Please notice that, although Figure 4 presents a scenario where the consumer sends the Interest to the ANM after beginning the actual information request, it could be possible to do both requests in parallel.
4. As there is at least one replica (one entry in the AN list), the consumer can ask for the metadata of that replica, to authenticate it and to know which pieces of the content it serves, by means of its bitmap. The Interest will be sent to `/com.example//content/b.mpg/_meta` and the Data response must be signed by the replica source itself.
5. Now, the node can start sending Interests to pieces of the replica source by using its bitmap. In order to do that, Interest packets will carry the name of the original content (CN) (e.g. `/es/uc3m//it/joe/video/a.mpg/_si`) in the CNI field, where *_si* represents segment number *i*. However, those packets have also to include the *ARN* field with the AN (e.g. `/com.example//content/b.mpg/_si`) used to route the Interest packet. Different pieces of the content may be requested to different replicas, to balance load and to improve download speed. A replica receiving an Interest where the *ARN* field is present, must return the data packet with the CN of the content and the signature of the original provider of the data. In our

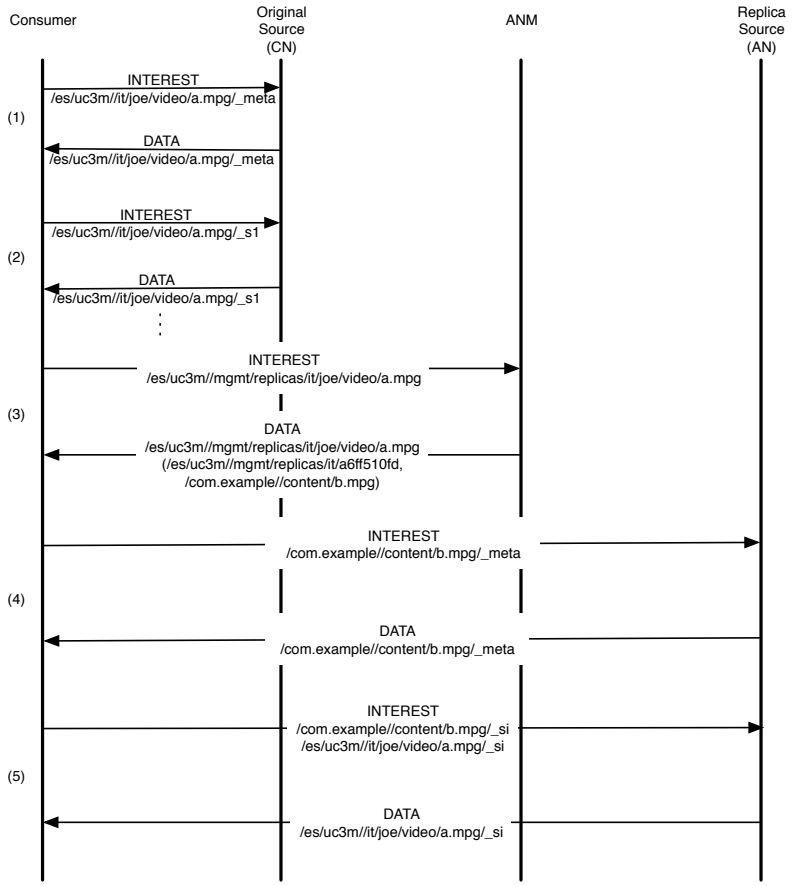


Figure 4: Flow diagram when there is one replica

example, we assume that the Interest reaches the replica, but it is also possible for a CR with an object `/es/uc3m//it/joe/video/a.mpg/_si` in its Content Store to reply it back, thus consuming the Interest.

The scenario presented in Figure 4 assumes that the original source is available. If this is not the case, the consumer will not receive a reply back with the metadata of the object as in step (1) in Figure 4, so the transfer of data cannot progress. In that situation, a consumer must send an Interest message to the well-known ANM. Sending an Interest to this name will retrieve a list of replicas for that content from the ANM, just as in step (3) in Figure 4. With the name of the replicas, the transfer can progress as in steps (4) and (5) in Figure 4.

In the original CCN proposal, as well as in our solution, a consumer may benefit when different copies of the same content are spread in the end nodes of the network, as copies with a lower delay from the customer can be used to retrieve such content. Opposite to our solution, in [10] the authors suggest that content routers with multiple faces towards the same prefix have to decide the *best* face to use, by using a strategy layer. Our solution moves this decision to the end nodes, distributing the processing load of source selection, thus providing a better scalability.

3.7. Inherent Mobility Support

Once the mechanism for managing alias names (ANs) has been explained, mobility of nodes can be achieved naturally. A node moving from its usual access point of the network to another access point must, after having received its new prefix name delegated by the visited network, register itself in the ANM. With this information, the moving node can start the process explained in section 3.4 to register all (or a subset of) its content using its home prefix name as the CN and the just assigned visited prefix name as the AN. The ANM will include this AN in the lists of alias names associated to all registered contents with CN as prefixes. It would be possible to dynamically generate ANs for non registered CNs, i.e. to answer to all Interests requesting contents served by the moving node. A possible scenario is presented in Figure 5, where a node moves from `/es/uc3m//it/` to `/gz/provider/`.

Figure 6 presents the steps the original source in Figure 5 has to do to register itself as a replica source, and what a consumer has to do to access that replica. These are:

1. The original source should register itself at the ANM just after it obtains its new domain name in the visited network (`/gz/provider//mobile/user42`). The first step is to obtain a unique identifier from its ANM to register its objects. In this example we assume that the original source wants to register all its objects, using its prefix name (`/es/uc3m//it/joe`). As there is no entry for such name, the ANM generates a new random identifier for such request (`/es/uc3m//mgmt/replicas/it/b3eg313af`).

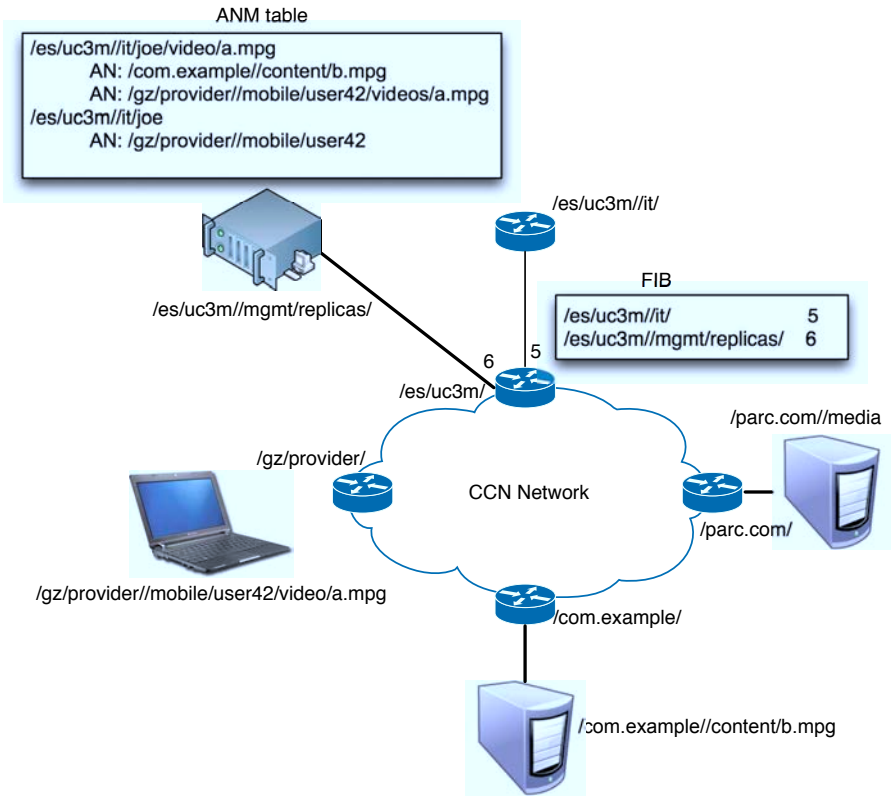


Figure 5: Scenario where the original source is in a different domain

2. The original source registers itself as an alias name for all its content. Please notice that it is still necessary to authenticate the user, so this entry is in a non authenticated state.
3. The ANM sends an Interest with a challenge to the registered alias to check the authenticity of the user. The original source must sign the data packet using its private key so the ANM can check its authenticity. After this step, the ANM can answer back all requests for all objects with the registered prefix by appending to this prefix the suffix included in the Interest request, as explained in the next steps.
4. Later on, a consumer sends an Interest using the original name (CN) `/es/uc3m//it/joe/video/a.mpg` of the object. This message arrives to the access CR of the original domain, which does not receive any answer as the original source is not accesible in that domain.

5. After a timeout or in parallel, the consumer sends an Interest to the well known ANM of the object. The ANM has to concatenate the specific object name (`video/a.mpg`) to the registered alias (`/gz/provider//mobile/user42`) to generate the alias name. The Data packet includes a list with all alias names of the requested object. In particular, it contains the alias of the original source `/gz/provider//mobile/user42/video/a.mpg` generated as explained before. Next steps are identical to the ones explained in section 3.5.
6. The consumer sends an Interest to `/gz/provider//mobile/user42/video/a.mpg/_meta` to obtain the metadata information and to authenticate the replica source. Notice that each CCN Data packet includes a *key locator*, which indicates where the key used to sign it can be located.
7. Finally, the consumer starts sending Interest for each individual segment including both the *CNI* (`/es/uc3m//it/joe/video/a.mpg/_si`) and the *ARN* field (`/gz/provider//mobile/user42/video/a.mpg/_si`) used to route the request.

4. Analysis of scalability

With the aim to compare our solution with the CCN proposal, this section presents a study were we have evaluated the total number of extra FIB entries and faces a router has, when multiple copies of a single object are registered at different domains. To simplify the analysis, we consider a single content object that will be replicated in different network domains, although our results can easily be extrapolated to more general scenarios considering multiple content objects. This evaluation was conducted by means of simulations, using a simulator implemented in Matlab[1] for this purpose. This simulator allows us to define network topologies with routers, links and a set of domains.

For each simulation, we define a network topology following a procedure that will be described later in this section. This topology allows interconnecting a set of network domains, each of them represented by a name prefix (e.g. `/es/uc3m` can be the prefix corresponding to University Carlos III of Madrid). We start

[1] http://www.mathworks.es/products/matlab/ (July 2014)

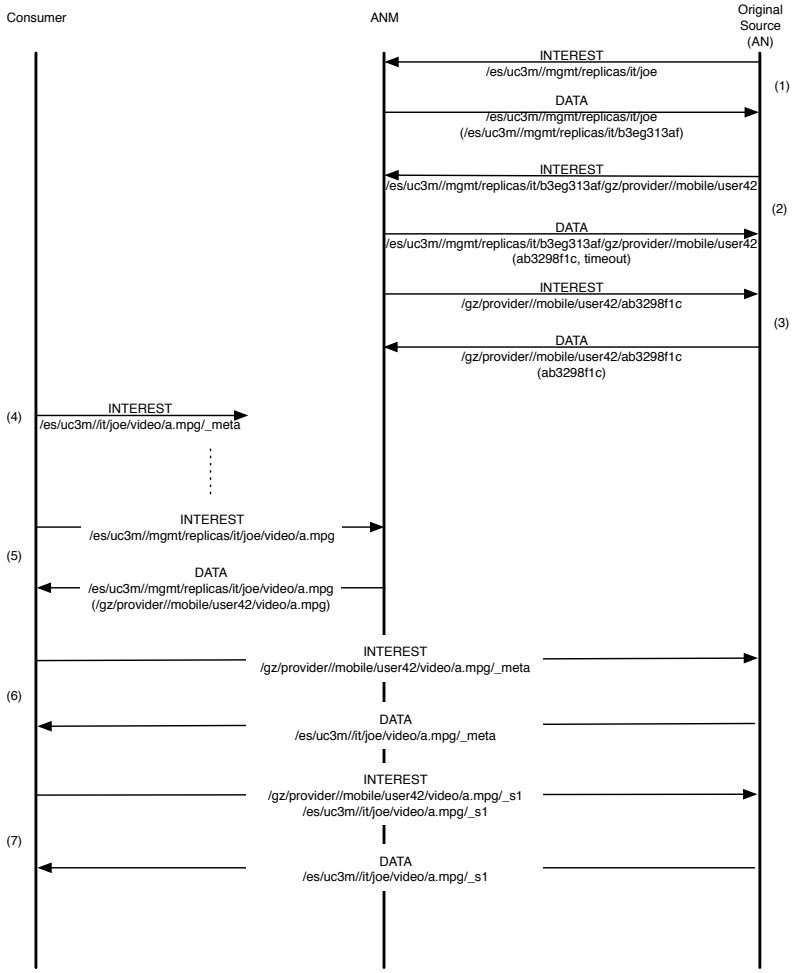


Figure 6: Flow diagram when the original source is in a different domain

the simulation from a stable state, where there is a single content object (i.e. the object with CCN name `/es/uc3m/a.mpg`) which is located in the network domain under the name prefix `/es/uc3m`. Additionally, each CCN router in the topology has an entry in its FIB to reach every network domain. When the simulation starts, a replica of the content object `/es/uc3m/a.mpg` is copied to another network domain, which is randomly selected. Afterwards, the FIB of each content router is updated if necessary, to include a path to the new replica. The state required in the FIB of the CCN routers is then evaluated, and the simulation proceeds by appending a new replica of the object to another network domain and repeating the aforementioned procedure. The simulation finishes

when a replica of the content object is placed in every network domain.

In the case of regular CCN, when the first replica of the content object is copied to a new network domain, a registration request is propagated to the network from the access router of the domain. Each router receiving this request updates its FIB table according to the following procedure:

1. If the incoming face of the registration request is different from the face that is currently configured in the FIB to reach the name prefix of the domain (i.e., `/es/uc3m`), then a new entry is appended to the FIB. For example, if a router has an entry in its FIB for the name prefix `/es/uc3m` through face 1, and a registration request is received for `/es/uc3m/a.mpg` via face 2, then a new entry <`/es/uc3m/a.mpg, face 1, face 2`> is appended to the FIB. Note that the new entry includes two faces, to indicate that `/es/uc3m/a.mpg` can be reached from both of them.
2. On the other hand, if the incoming face of the registration request is the same that is included in the FIB to reach the content object, then no changes are required to the FIB of the content router. In the previous example, if the registration request for `/es/uc3m/a.mpg` arrives from face 1, then a new entry is not appended to the FIB, as the existing entry <`/es/uc3m, face 1`> can be used to reach the content object `/es/uc3m/a.mpg` in all the locations it is available.
3. Once a router has two entries in the FIB for the content object, subsequent registration messages corresponding to new replicas of the object only add an extra face to the second entry, when the incoming face of the registration request is different from the faces already configured in the FIB.

In the case of our proposal, Interest packets are routed using alias names, and not content names as opposed to regular CCN. An alias name identifying a replica is constructed using a name prefix that is specific to the network domain holding the replica. Note that, as we commented, the simulation starts from an stable state where each CCN router in the topology has an entry in its FIB to reach each network domain. In consequence, adding a new replica in a different network domain does not require to generate registration messages, as the Interest can be routed using its alias name with the information that is

already available in the FIBs of the CCN routers. For the sake of clarity, we have omitted the results obtained with our proposal in the following figures. In all the considered scenarios, independently of the number of replicas in the system, our proposal maintains the expected behavior and holds a single entry in the FIB of each router per network domain. Thus, all figures in this section represent the extra number of entries and faces in the FIB of regular CCN routers, compared to our proposal.

Although the authors of [15] have stated that the chosen topology has not a big impact on their simulation results, for this particular study we use two different topologies: a real one extracted from the Rocketfuel project[2] and a random topology generated from an analytical model. The real topology uses the data obtained by the Rocketfuel project from the Sprintlink provider (AS=1239) with 315 routers and 43 different regions or cities. We have decided to instantiate one name domain per region, by randomly connecting one router of a region to its corresponding domain. The delay for such links is uniformly selected from the set of values {1ms, 2ms}. For each simulation, a domain is randomly selected to store the first copy of the object of interest. The name of the object is generated using the domain's name as a prefix. To reduce the effect of the selected access delay, 100 simulations were ran to calculate the average number of extra entries, as well as the extra faces for the second entry for all network routers. Figure 7 shows these average values, as well as the 95% confidence interval for each of these values. The x-axis represents the number of domains that have registered a copy of the studied object during the simulation.

Two conclusions can be extracted from the number of FIB entries plotted in Figure 7: (1) the second copy of an object has a significant impact on the average number of extra entries in the FIB of the routers, and (2) the number of extra entries rapidly increases with the number of domains with copies, to asymptotically reach a value near 1. On the other hand, the number of extra faces in the second entry of the FIB (dashed line in Figure 7) logarithmically increases with the number of replicas.

In Figure 8 we present the average number of extra entries and faces, cov-

[2]http://www.cs.washington.edu/research/networking/rocketfuel/

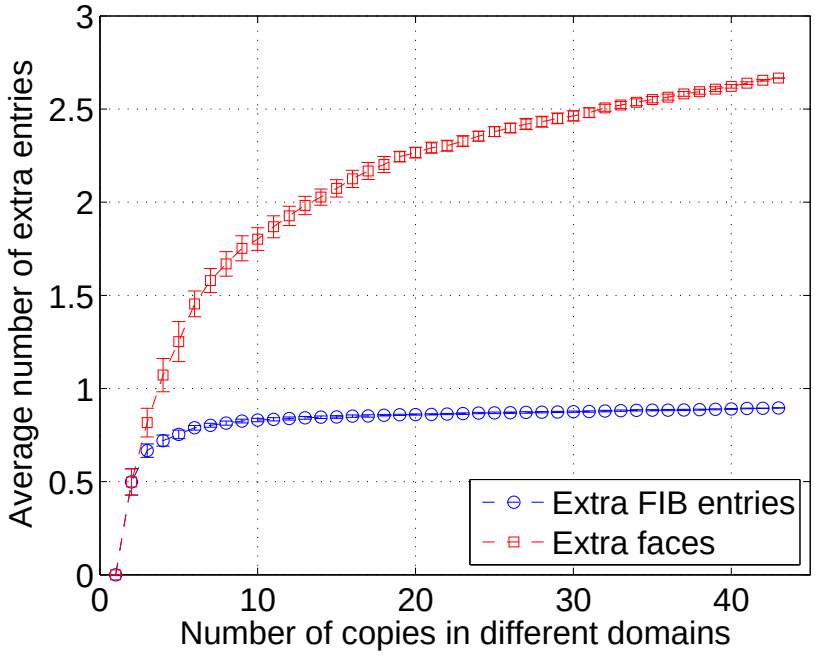


Figure 7: FIB entries in the real topology using regular CCN.

ering only the subset of routers that interconnect different regions (i.e., border routers). As it can be observed from this figure, the state that is required in border routers is higher with respect to the number of extra faces.

The second topology uses an analytical model to generate the position of the routers, the probability to have a link between routers, the delay of the links and the identification of routers that will sit between network domains. The placement of the routers follows a heavy-tailed approach (Zipf in our model with parameter s) as suggested by other topology generators (such as Brite[3]). Two routers, u and v are connected with a probability that depends on the distance between them, following the Waxman model [20]. This edge probability is given by:

$$P(u,v) = \beta \exp \frac{-d(u,v)}{L\alpha} \tag{1}$$

In (1), $d(u,v)$ is the distance between routers u and v, L is the maximum

[3]http://www.cs.bu.edu/brite/ (July 2014)

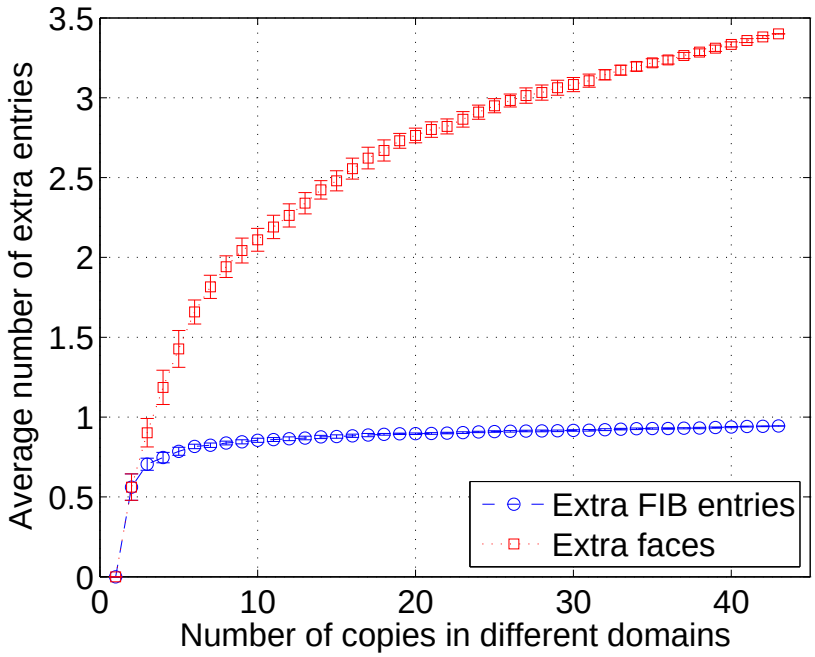


Figure 8: FIB entries, in border routers, in the real topology using regular CCN.

distance between routers, and α and β are two real parameters in the range $(0, 1]$ that control the ratio between short and long links and the degree of the routers, respectively. In our simulations we chose all these parameters following a uniform distribution ($s \in \{0.5, 1, 1.5\}$, α and β in the range suggested in [20]). As with the previous topology, 100 simulations were ran to evaluate the results taking into account different values of the input parameters. The number of routers and domains considered was the same that in the case of the real topology: 315 routers and 43 domains. Figure 9 shows the results for this random topology. Two main conclusions can be extracted from these results: (1) the number of extra entries rapidly increases with the number of copies in domains and, when there are six copies in the system, all routers have exactly one extra entry in their FIBs, and (2) the average number of extra faces is higher than in the real topology. These results are mainly due to the grade of connectivity in this topology.

Figure 10 shows the *betweenness* for both topologies. The betweenness of a router, as defined in graph theory, is the total number of short paths that pass

through it, which gives an idea of how well connected the network (graph) is. In the real topology (Figure 10a), the betweenness rapidly goes to zero (only 40% of routers are in the shortest path of any other two routers) while in the random topology (Figure 10b), all routers have a betweenness greater than zero.

The main conclusion that can be extracted from these results is that copying a single object from one domain to a different one has a significant impact on regular CCN. The number of extra faces increases when the object is replicated at different domains. On the other hand, if we consider multiple objects with non aggregatable names, replicated in different domains, the number of entries in CCN content routers linearly increase with the number of replicas. This illustrates the scalability issues that need to be addressed in regular CCN.

Our solution permits to deploy a scalable CCN, introducing an extra overhead to (1) register new replicas and (2) to use those replicas, if necessary. For the former, the extra overhead comes from the registration of the AN in the ANM, and the authentication of the replica source, as shown in Fig. 3. For the later, the cause of the overhead is the signalling to download the meta-file with alias-names, as shown in Fig. 4. It is important to notice that this overhead is per file, and not per chunk, and that the registration and the usage of replicas are optional. It is up to the customers and ANM managers to decide which contents have to be replicated and the number of replicas per content. Thus, different replication scenarios are possible (from no replication at all to full replication of every content).

5. Implementation and experimental validation

Apart from the validation through simulations shown in Section 4, we have also implemented a proof-of-concept of our approach to show its feasibility. Furthermore, this implementation allows us to evaluate its performance in terms of download delay in two scenarios: (1) only one copy of an object is available at its original domain, and (2) multiple replicas exist in different domains. These experiments include all the proposed signaling necessary to discover, register and contact replicas. This will be useful to evaluate the extra delay introduced by our scheme compared with regular CCN.

We have used the CCN software provided by the *CCNx* (version 0.4.1) open

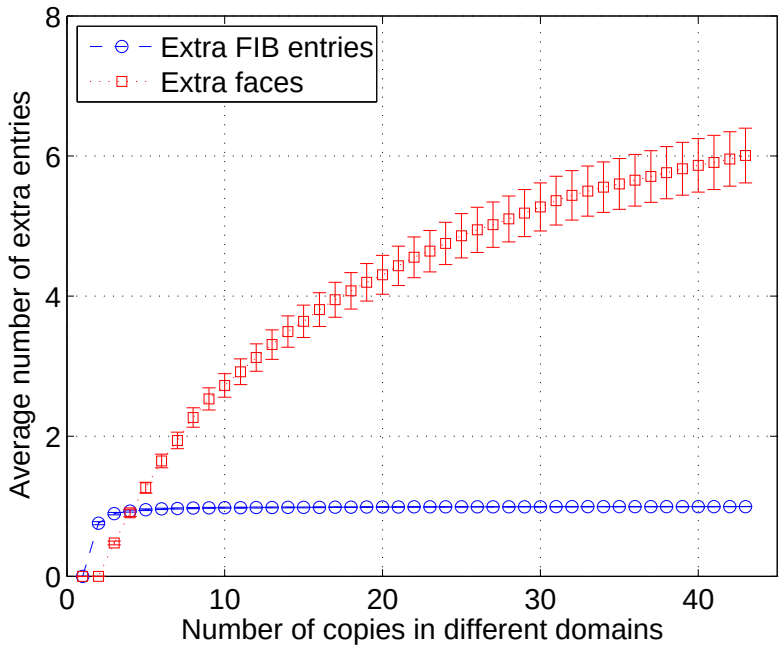
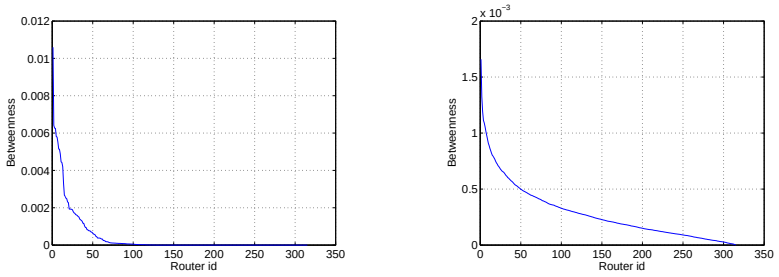


Figure 9: FIB entries in the random topology using regular CCN.



(a) Real topology.

(b) Random topology.

Figure 10: Betweenness for the studied topologies.

source project[4] for a fast deployment of our testbed. The modifications necessary to extend this software with our proposal are minimal, with less than 100 new lines. Although we are just validating the ideas presented here, we had to develop new applications (or to modify existing ones) by using the APIs provided by CCNx. These applications are prototypes using the replica system described in previous sections. The changes to CCNx and all the new and

[4]http://www.ccnx.org/ (July 2014)

modified applications will be explained in section 5.1.

Although very simple, the testbed used in this validation stage, presented in section 5.2, has all the elements involved in our proposal. It will be used in two different scenarios designed to demonstrate the proper functioning of the ideas presented in this paper. Later, in section 5.3 we gather all results from both scenarios and for different representative parameters.

5.1. Changes to CCNx

The CCNx open software implements much of the ideas presented in [10] to enable the collaboration with the research community. There are two differentiated parts in the code: one part is written in C/POSIX, including the content router, while the second one is composed of Java libraries. Next, we will describe the modifications we have made to the content router (C/POSIX part) and in some Java libraries, to implement all the changes proposed in this article.

5.1.1. Changes to Content Routers

The C implementation of the CCN daemon, *ccnd*, which plays the role of a content router, was modified to support the required protocol changes. In particular, the main modification involved the processing of Interest messages by the router. When a new Interest message is received, the router checks whether it contains an *Alias Routing Name* or not. If a routing name is present, the router relies on this name for routing purposes. If not, the content name is used.

5.1.2. Changes to clients

The main change to the client side is to add the *Alias Routing Name* optional field to the Interest messages. Apart from the CCNx core files, other applications were changed to run all the tests. In our case we have selected the *ccngetfile* application, which retrieves a given content (using the Java CCNx API) storing the content in a local file. With our modifications, the user could provide an extra parameter with the Alias Name of the retrieved content, which will be used after the download finishes to register itself as a replica in the ANM of the CN. Two key changes were introduced in the code: (1) first, the application

has to get the list of possible ANs by using the well-known name of the ANM extracted from the CN and (2) it has to decide from whom it will request each piece of data. As this validation is just a proof of concept, the decision described in the second point was preconfigured in the code. Notice that selecting the best source when a content is distributed around the network is another challenge of CCN, and we leave this study for future work.

To generate background traffic in the experiments, two new applications were developed: the *CCNClient*, which generates Interests with a random time between requests following an exponential distribution and a content server called *SimpleServer*, which answers back with a random generated Data packet consuming Interests. The *SimpleServer* receives as input the prefix name it is serving while *CCNClient* receives the prefix name it has to use to generate Interests as well as the parameter λ with the average rate of requests per second.

5.2. Testbed Description

Figure 11 presents the whole testbed deployed to run all tests involved in this validation part. We have used six Linux based modern PCs to deploy such testbed, one per content router (CR). In Figure 11 the end terminals (both consumers and providers) are deployed as a single machine together with a CR, which includes the Content Store of the client. All links are shaped to a symmetric 3 Mbps connection by using the *Traffic Control* (tc) command, which is part of the *iproute2* suite[5]. Some of those links have been configured to have an extra delay by using the same command. As it will be described in section 5.3, two scenarios, with a high difference in terms of delay, are considered here. In the first one, the delay introduced by the access link of C1 is $5ms$, while in the second one, this delay is $25ms$.

All CRs run the *ccnd* daemon, which allows Interest forwarding and provides a local Content Store cache. The Content Stores in $CR1$ and $CR2$ were limited to 1000 objects[6] while the remaining ones were started using the default size

[5]http://www.linuxfoundation.org/collaborate/workgroups/networking/iproute2 (July 2014)

[6]CCNx allows configuring a desired value for the content store size in terms of *objects*, although this is not a strict upper bound.

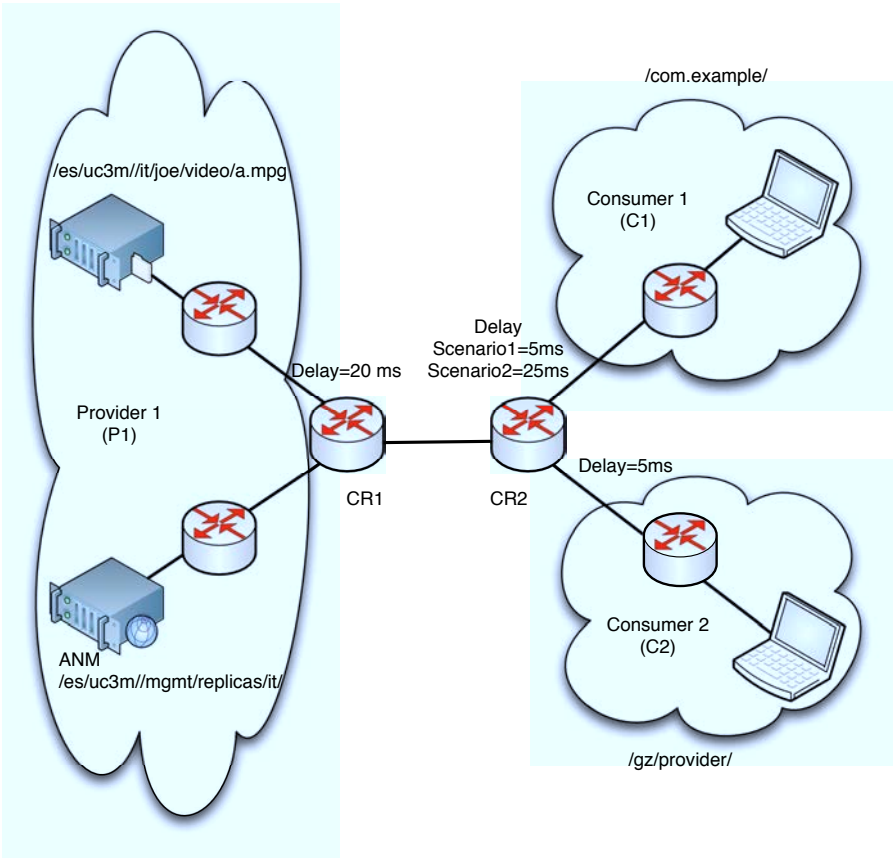


Figure 11: Testbed used for the validation

value. The entries of the FIB were added manually in each individual CR using the *ccndc* command. To refresh the caches at $CR1$ and $CR2$, a background traffic was injected in both CRs using our applications *SimpleServer* at $CR1$ and *CCNClient* in $CR2$, providing different values for the time between requests, which follows an exponential distribution.

Provider 1 (P1) runs a CCN repository (*ccn_repo*) which allows to serve files using the command *ccnputfile*. This way, a file is fragmented in objects including all signatures and metadata packets. In our tests, the content `/es/uc3m//it/joe/video/a.mpg` is a 2 Mbytes (500 objects of data approximately) file generated with random content. It will be requested by the consumers at domains `/com.example/` and `/gz/provider/`. P1 offers an ANM server to their customers so they may manage the alias name of their own objects. In our

testbed, the ANM of P1 manages all objects of the domain `/es/uc3m//it`, so its well-known name is `/es/uc3m//mgmt/replicas/it`.

5.3. Results

As introduced in the previous section, there are two sets of results for two different scenarios: in the first one we show the benefits of using a replica that is closer to the downloading customer than the original source, while in the second one we present the opposite scenario, to measure the extra delay suffered when a replica has a greater round trip time compared with the original source.

In both scenarios we ran two set of tests. They differ in background traffic: an exponential distribution between requests generating $\lambda = 50$ requests/s (600kbps) in one case and $\lambda = 100$ requests/s (1.2Mbps) in the other, to simulate video streaming with different bitrates. In each experiment we perform the following steps:

1. The testbed is reset to the default values.
2. All CRs are configured with their corresponding FIB entries to reach all domains.
3. The *SimpleServer* at $CR1$ and the *CCNClient* at $CR2$ (either with $\lambda = 50$ or $\lambda = 100$) are initialized to start the background traffic between both CRs.
4. With the background traffic filling the Content Store in $CR1$ and $CR2$, C1 starts the *ccngetfile* application to retrieve the content `/es/uc3m//it/joe/video/a.mpg`, which implies contacting the ANM. As there are no replicas yet, C1 only uses the original source (we call this *single source* in the rest of the section).
5. After the download finishes, C1 registers itself at the ANM with the alias name `/com.example//content/b.mpg`.
6. Later on, C2 proceeds to start the download of `/es/uc3m//it/joe/video/a.mpg`. To collect different results, we ran several tests modifying the time C2 has to wait to start its download process after C1 has completely finished the download. We call waiting time, T_w, to this time, which lets us modify the content popularity in the experiments: popular content has a high access frequency, which implies a small value of T_w. In our tests, after T_w

seconds, C2 starts *ccngetfile* to download the same content as C1. After contacting the ANM, C2 receives two different ANs for the same content: `/es/uc3m//it/joe/video/a.mpg` and `/com.example//content/b.mpg` that will be used to download the content (we call this *multi-source* in the rest of the section).

As this is just a proof-of-concept and there is no strategy layer algorithm implemented at the moment, *ccngetfile* has been preconfigured to know which is the best source to download content from[7]. For example, in the first scenario, where C1 has a lower delay than the original source, C2 starts downloading the content from the original source (10 objects) and all the remaining ones from C1. In the second scenario, C2 downloads 10 objects from the original source, the next 10 objects from C1, and the remaining ones again from the original source, which provides a lower delay.

Figure 12 presents the results obtained for the total download time vs T_w in the scenario 1, where the best choice for C2 is to use C1, and for different values of λ. For each T_w we ran 100 executions, which allows us to obtain an acceptable 95% confidence interval that is also shown in all figures. In this figure we present the results for multi and single source and, as it was expected, using multi-source is better than using a single source. In Figure 12, the multi-source scenario with $\lambda = 100$ requests/s has three different range of values for different values of T_w. For $T_w = 1$, all objects of the file are stored in the $CR2$ cache, while for T_w in the range of 5-15 seconds approximately, some objects are in $CR2$ but others have to be downloaded from C1. Lastly, when T_w is above $20s$, due to the background traffic, there are no objects of the file in $CR1$ and $CR2$ caches and both sources (the original source and the replica) have to be used to download it.

When $\lambda = 50$ requests/s, Figure 12 shows that there are just two different ranges of values for the total download time with multi-source: for T_w in 1-15 seconds all objects are in $CR2$, while when T_w is in 20-30 seconds there are no objects in $CR1$ and $CR2$, so both sources have to be used.

[7]Notice that in CCN the same requirement exists, although in that case the routers, instead of the end nodes, have to run those algorithms.

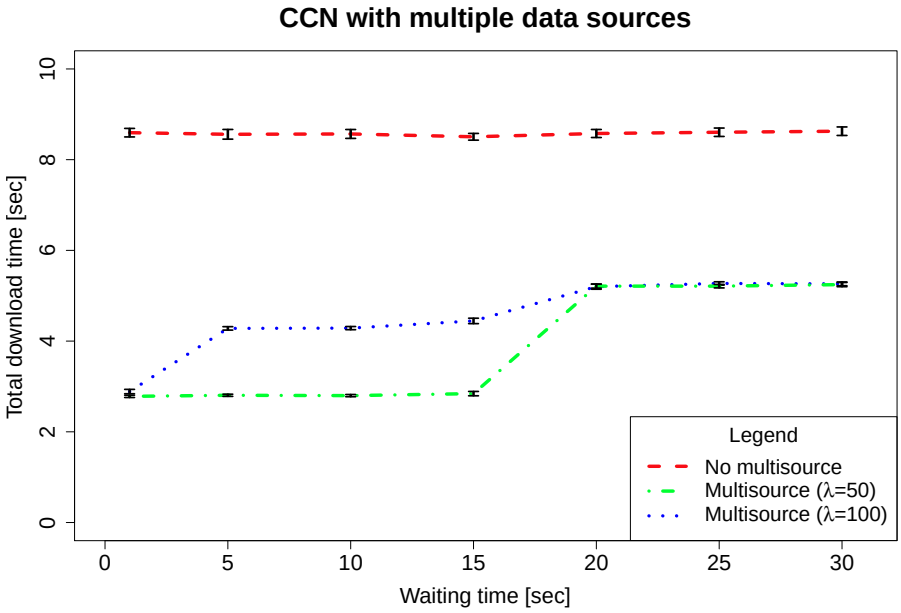


Figure 12: Results scenario 1 when C1 has a delay of $5ms$

Figure 13 gathers the result for scenario 2, when the replica has a bigger delay than the original source. The explanation for all ranges of the total download time is similar to that presented in the first scenario. In this graph we want to show that, when both sources are used in the multi-source test, the total download time is a slightly worse than the single source scenario. This is because 10 objects are downloaded from a replica with a bigger delay, so multi-source has no benefit in such kind of scenarios. How to detect such kind of situations is out of the scope of this paper and we leave it for further study.

6. Conclusions

This paper presents an enhancement to the Content-Centric Networking proposal, focused on the management of replicas to improve the overall routing process, and to solve the scalability issues of the FIB that are present in the original CCN proposal. We have introduced two changes: (1) to structure a CCN name in two parts: a globally-routable prefix assigned by the ISP, amenable to aggregation, and a name suffix, and (2) to define a novel *alias name* architecture, so replicas in different parts of the network with different names can be bound

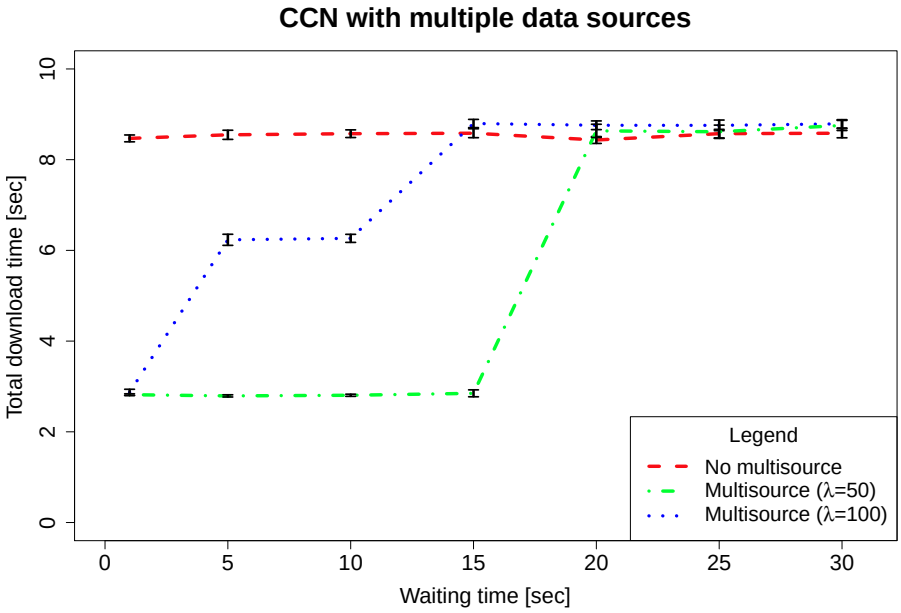


Figure 13: Results scenario 2 when C1 has a delay of $25ms$

to the same content. The former change does not affect the protocols proposed in the original CCN, as this is just a method to maximize the aggregation at the routing tables of the content routers. For the latter, some minor modifications are necessary to include a new optional field in the Interest message and to use that field, if present, at the content router for the selection of the outbound face. Furthermore, our proposal inherently supports mobility, as this is just a special case of alias name registration. This way, it is not necessary to add any new entry to the FIB of the routers when a content is moved from one place to another, or replicated in another network domain. Thus, our solution improves FIB scalability.

Two main conclusions can be extracted from this work: (1) our proposal is scalable in terms of the number of entries in the FIB, while regular CCN is not and, (2) it is feasible to implement our solution based on CCNx with some minor changes.

Apart from validating the modified protocols, we have shown that it is still necessary to face other challenges of CCN like the design of an optimal strategy layer at the consumer ends when replicas exist, that we leave for further work.

Acknowledgments

The authors would like to thank to all people working on the CCNx project, which was extremely useful for the development of this work. This article has been partially supported by the Spanish Ministry of Economy and Competitiveness (MINECO) by means of the project MASSES (TEC2012-35443) and by the Comunidad de Madrid E-Madrid (S2009/TIC-1650) project.

References

References

[1] S. Kent, K. Seo, Security Architecture for the Internet Protocol, RFC 4301, Internet Engineering Task Force (Dec. 2005).
URL http://www.rfc-editor.org/rfc/rfc4301.txt

[2] T. Dierks, E. Rescorla, The Transport Layer Security (TLS) Protocol Version 1.2, RFC 5246, Internet Engineering Task Force (Aug. 2008).
URL http://www.rfc-editor.org/rfc/rfc5246.txt

[3] C. Perkins, IP Mobility Support for IPv4, Revised IP Mobility Support for IPv4, Revised, RFC 5944, Internet Engineering Task Force (2010).

[4] D. Farinacci, V. Fuller, D. Meyer, D. Lewis, The Locator/ID Separation Protocol (LISP), RFC 6830 (Experimental) (Jan. 2013).
URL http://www.ietf.org/rfc/rfc6830.txt

[5] C. Jennings, B. Lowekamp, E. Rescorla, S. Baset, H. Schulzrinne, REsource LOcation And Discovery (RELOAD) Base Protocol, RFC 6940 (Proposed Standard) (Jan. 2014).
URL http://www.ietf.org/rfc/rfc6940.txt

[6] A. Bakker, Peer-to-Peer Streaming Peer Protocol, Internet-draft, Internet Engineering Task Force, version 1 expires on August 2, 2012 (2011).

[7] D. Lagutin, K. Visala, S. Tarkoma, Publish/Subscribe for internet: PSIRP perspective, Towards the Future Internet Emerging Trends from European Research 4 (2010) 75–84.

[8] T. Koponen, M. Chawla, B. Chun, A. Ermolinskiy, K. Kim, S. Shenker, I. Stoica, A data-oriented (and beyond) network architecture, in: ACM SIGCOMM Computer Communication Review, Vol. 37(4), ACM, 2007, pp. 181–192.

[9] M. Gritter, D. Cheriton, An architecture for content routing support in the internet, in: Proceedings of the 3rd conference on USENIX Symposium on Internet Technologies and Systems-Volume 3, USENIX Association, 2001, pp. 4–4.

[10] V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs, R. Braynard, Networking Named Content, Communications of the ACM 55 (1) (2012) 117–124.

[11] J. Choi, J. Han, E. Cho, T. Kwon, Y. Choi, A survey on content-oriented networking for efficient content delivery, Communications Magazine, IEEE 49 (3) (2011) 121–127.

[12] G. Xylomenos, C. Ververidis, V. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. Katsaros, G. Polyzos, A survey of information-centric networking research, Communications Surveys and Tutorials 16 (2) (2013) 1024–1049.

[13] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J. Thornton, D. Smetters, B. Zhang, G. Tsudik, D. Massey, C. Papadopoulos, et al., Named data networking (ndn) project, Tech. rep., Tech. report ndn-0001, PARC (2010).

[14] I. Psaras, W. K. Chai, G. Pavlou, Probabilistic in-network caching for Information-Centric Networks, in: Proceedings of the second edition of the ICN workshop on Information-centric networking, ACM, New York, NY, USA, 2012, pp. 55–60.

[15] D. Rossi, G. Rossini, Caching performance of content centric networks under multi-path routing (and more), Tech. rep., Telecom ParisTech (2011).

[16] M. Mangili, F. Martignon, A. Capone, A comparative study of content-centric and content-distribution networks: Performance and bounds, in:

Proceedings of IEEE Global Communications Conference (Globecom 2013), Atlanta, GA, USA, 2013, pp. 1403–1409.

[17] G. Ma, Z. Chen, Comparative study on ccn and cdn, in: Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference on, IEEE, 2014, pp. 169–170.

[18] D. Perino, M. Varvello, A reality check for content centric networking, in: Proceedings of the ACM SIGCOMM workshop on Information-centric networking, ACM, 2011, pp. 44–49.

[19] R. Cruz, N. M.S., Y. Gu, J. Xia, D. Bryan, J. Taveira, D. Lingli, PPSP Tracker Protocol (PPSP-TP), Internet-draft, Internet Engineering Task Force, version 7 expires on August 27, 2012 (2012).

[20] B. M. Waxman, Routing of multipoint connections, Selected Areas in Communications, IEEE Journal on 6 (9) (1988) 1617–1622.