Internames: a name-to-name principle for the future Internet

Nicola Blefari Melazzi*, Andrea Detti*, Mayutan Arumaithurai†, K.K.Ramakrishnan‡
* CNIT, University of Rome Tor Vergata, Italy., †University of Goettingen, Germany., ‡University of California, Riverside, USA.
blefari@uniroma2.it,andrea.detti@uniroma2.it,arumaithurai@cs.uni-goettingen.de,kk@cs.ucr.edu

***Abstract*—Information Centric Networking (ICN), a novel network paradigm, places the focus on the content instead of the end-hosts. ICN addresses content by names instead of locations and can ease content retrieval and improve network efficiency. Ongoing work attempts to extend the use of ICN to scenarios such as real time communication, group communication, push services, and addresses the issue of migration from the current network and coexistence of different network paradigms. In this work, we argue that by extending the current design of ICN from a ”host-to-name” to a ”name-to-name” architecture, the utility and efficiency of ICN could be further increased. We propose Internames, an architectural framework in which names are used to identify all entities involved in communication: content, users, devices, logical points, and services. Internames is envisioned to be an overarching name-to-name communication primitive that is fully compatible with ICN principles, accommodates the co-existence (or gradual migration) of different network realms (e.g., IP, ICN, VANET) and is suitable for application scenarios where ICN is somehow limited by its reliance on a ”host-to-name” approach. In this paper, we provide early insights into the Internames architecture by leveraging on the work done by the research community and identify components and challenges that require more detailed investigation.**

I. Introduction

Information Centric Networking (ICN) [1], [2], [11] is a new paradigm in which the network layer provides users with access to content by names, instead of providing communication channels between hosts. This is believed to offer several advantages [12], including: improved efficiency, thanks to in-network caching and content-based routing; simplified handling of mobile and multicast communication; ability to work in a non-infrastructure mode; a content-oriented security model; content-oriented access control/QoS; network awareness of transferred content. However, there are several issues of ICN[1] that still need to be addressed: i) the complexity and scalability of the proposed naming and routing functionality; ii) the cumbersome support for push services; iii) security and privacy concerns; iv) the need to devise a credible migration path from the communication paradigm of the current network infrastructure or, more practically, to allow the coexistence of different network operating modes, so that different sections of the network can operate according to different paradigms (ICN, IP, MAC-only). Due to these concerns, some have suggested implementing ICN functionality only in end-devices, supported by existing network infrastructures and protocols [3].

Our aim is to design an architecture that addresses such concerns and enlarges the scope of ICN beyond content retrieval, eases send-to-name and push services, and allows the use of names to route data also in the return path. In fact, while current ICN proposals use names instead of locations to address a remote end-point, the source of the communication is still identified by the end-point and its current location (although, these architectures do adapt to the receiver's mobility).

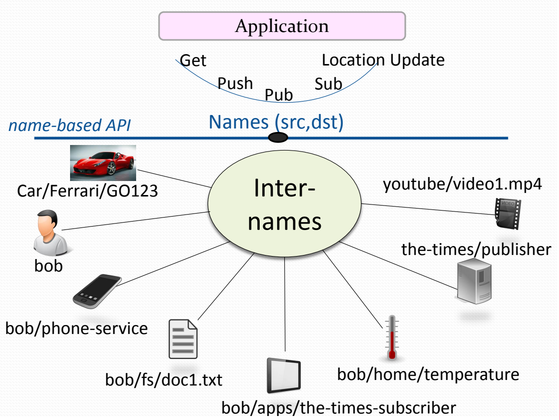


Fig. 1: Name-based Application Programming Interface

Internames evolves from ICNs host(s)-to-name model to a name-to-name principle in which names identify *both* source and destination entities, and the name of all communication entities is not statically bound to their physical location. In addition, names are used to identify *all* entities involved in communication: content, users, devices, logical points; also services are bound to a service-identifier (i.e. a name) rather than to an IP address, to easily enable re-location or duplication or anycast search of service (components), easing the support of emerging service-centric networking architecture (e.g. www.serval-arch.org). The basic Application Programming Interface (API) should accept names as identifiers of all requested content or services (see Figure 1). This principle should guide the whole network design, similar to what happened with the end-to-end principle [7]. Internames is supported by a Name Resolution Service (NRS) that maps names to network locations, as a function of time/location/context/service. The NRS could work either on-path and off-path and plays a key role in Internames by enabling the co-existence of multiple network domains, which we call network realms. In a manner similar to the role played by IP to unify different network technologies, Internames would reconcile and unify IP, cellular, sensor, IoT, ad-hoc, mobile, and new ICN networks. The NRS would map a name not only to a network location but also to the right protocol to be used to reach the current location of that name. In

[1] Some of these issues are applicable only to a particular set of ICN solutions

the same way unicast, multicast, broadcast, anycast type of communications would be a property bound to names, with the resolution service mapping names to the requested services. In this sense, the NRS is more powerful than the current DNS, and extends the functionality that has been suggested for the Global Name Resolution Service (GNRS) in [19]. Today's technology allows the implementation of a logically centralized NRS coupled with localized instantiations of its functionality. As a matter of fact, current search engines implement functions that are a superset of what the NRS would require, with some criticality remaining in the latency of the name resolution process, especially for mobile communications, and in mechanisms to mitigate the load on the NRS (e.g. caching). Of course scalability and performance studies are needed, but not considered in this workshop paper. The whole approach would be in line: i) with the recognized need for network abstractions, as theorized in Software Defined Networking approaches [13], since it relies on clear interfaces, including the name-based API; ii) with recent Network Function Virtualization [14] principles and more in general Cloud computing and Cloud networking techniques that could be exploited to implement the NRS and other network entities (described later in the paper); in particular, cloud networking, which extends the outsourcing of processing and storage services to the cloud also to networking services, could support efficient deployment of the resolution functionality.

What will Internames enable as a future network paradigm? It will allow named-entities to be mobile and be connected to the network infrastructure anywhere; enable them to be reached by using basic communication primitives; allow the pushing of information to a specific set of receivers based on name; allow to choose the return path independently from the forward path, supporting source/initiator mobility and choosing the path at the time the data is sent rather than at the time when a subscription was made (without needing to maintain state in the network for the reverse path, e.g., CCN PIT); allow communication to span networks with different technologies and allow for disconnected operation. Furthermore, with the ability to communicate between names, the communication path can be dynamically bound to any of a number of end-points, and the end-points themselves could change as needed. Key for the support of all such functionality is the NRS, which provides dynamic resolution of names to network locations, both for destination and for source names, thus allowing also the source to change point of attachment to the network during a communication session.

We believe that the zeitgeist or spirit of the times deems that such a vision is mature enough to be feasible, though pieces of it have been proposed several times in the past to no avail (e.g. the identifier/location split). Our contributions in this paper are:

- extending the host-to-name principle of ICN to a name-to-name principle and providing insights into the benefits of doing it;
- illustrating how such an architecture would allow different network domains to co-exist without having to migrate completely to one technology or the other;
- providing ideas of how a name-to-name architecture could look like and identifying its key components.

A. *Example Scenario*

To better motivate our proposed name-to-name communication framework, we describe a representative use case developed in the framework of the GreenICN joint EU-Japan project [15]: that of Disaster Management, to limit the effects of disasters on ICT infrastructures. When there is a single, static, sender of information to one or multiple recipients identified by a name, current CCN/NDN host-to-name communications are sufficient. However, in our use case, each sender may have different roles, persona and responsibilities (as an individual, as an authority). When that person wishes to send some information, e.g., related to a disaster, the initial communication could be viewed as coming from an authority (identified by a name) to a designated set of recipients (identified by another name). Unlike CCN/NDN [1], the return message could be addressed to the authority (i.e., a name) and could follow a different path for communication than the original path in the reverse direction. The responses would be delivered to the original name that transmitted the initial message, even if the named entity moves from its original location, or the original named entity is mapped to a different logical entity with a related location (e.g. because the original one is not reachable anymore), going beyond simple source mobility. Moreover, the response could be delivered to more than one entity associated with that name (in a sense, it is the reverse of a traditional multicast, having the information flow from receivers to all senders). The NRS would take care of such dynamic mapping, which can be also a function of network conditions, reducing the states maintained in the network (e.g., in the PIT in CCN/NDN enabled routers, by resorting to name resolution both in the forward and in the reverse path) and performing additional services, such as sender/receiver name-based authorization, and facilitation of communication between a group of senders and receivers where each set is identified by a single name. Finally, if connectivity to NRS entity(ies) is impeded by the disaster, resolution services could be entrusted to suitable local servers or even to other user devices or simply mapping every request to broadcast (think of people trapped under a collapsed building).

II. Related Work

A. *CCN/NDN based approaches*

CCN/NDN[2] is the most popular ICN approach; essentially it is a Query-Response based model that uses states stored in a Pending Interest Table (PIT) in order to perform reverse path forwarding towards the host(s) that requested the content. COPSS [2] enhances NDN by providing push capability.

1) Migration path: The transition to a worldwide CCN raises serious migration issues. With Internames, evolution would be more gradual and mostly concentrated at network edges and hosts. The NRS could be initially the current DNS, gradually evolving to a more powerful and dynamic and localized version of itself.

2) Routing scalability: A full worldwide CCN implies very large routing tables and high frequency of updates. Internames envisages the presence of multiple network realms including

[2] In this work, we use the acronyms CCN and NDN interchangeably since they conceptually follow the same approach

pure CCN realms, pure IP realms or hybrids between the two. This allows various levels of separation that could facilitate efficient management of routing tables. Moreover, routers in the network need not have the same capability and can be grouped based on their functionality and capability into different sub-NRSs. Furthermore, Internames relies on name-to-name communication also for the return path, and therefore does not need additional state in the router.

3) Security: In CCN, the network delivers data upon request only, eliminating several Denial of Service attacks. However, it is still prone to Interest packet flooding as PITs could be flooded with fake entries.

B. Pursuit

The PURSUIT [4] project proposed a publish-subscribe architecture based on names. Network entities such as Rendezvous and Topology Management entities play a similar role than Internames' NRS (and RRS see later), i.e. resolve names to locations and support routing. LIPSIN Bloom filters are used for source routing multicasting to distribute publications from a publisher to subscribers.

C. NetInf

The NetInf [11] project proposed the use of a Name resolution service mapping a name to a locator, if required. Moreover, in NetInf, the reverse path can be different from the forward path. Internames borrows such principles and enhances them by designing the resolution service to return the network domain(s) where the destination name can be found, the protocol that can be used to reach the next realm and the next hop gateway towards the destination network realm.

D. Internames: main characteristics

Internames complements existing ICN solutions with a focus on name-to-name, rather than host-to-name communication, gradual migration from current solutions and co-existence with other network domains. Its scalability properties are similar to those of Pursuit (NRS' complexity being similar to that of the Rendezvous and Topology Management entities). As regards privacy and security concerns, related to trusting a centralized authority, we argue that an open/standard NRS would be better than current proprietary search engines, which play today a similarly worrying role. Internames retains most of the advantages promised by ICN. It can perform content-based routing; it can provide off-path and in-path caching, not only in ICN routers, but also in IP routers using mechanisms similar to that proposed in [16]. It can facilitate mobility and offer all the advantages of content-based operations, including content-oriented security, content-oriented access control, content-oriented QoS differentiation and pricing. Since the NRS is aware of the transferred content, Internames allows for better control of information and of related revenue flows, as information on requested data could be made available under suitable agreements to interested parties. Name to name communications and bidirectional, two-way links between content could become a reality as envisioned in works such as [17]: In a network with two-way links each node knows what other nodes are linked to it and preserve context.

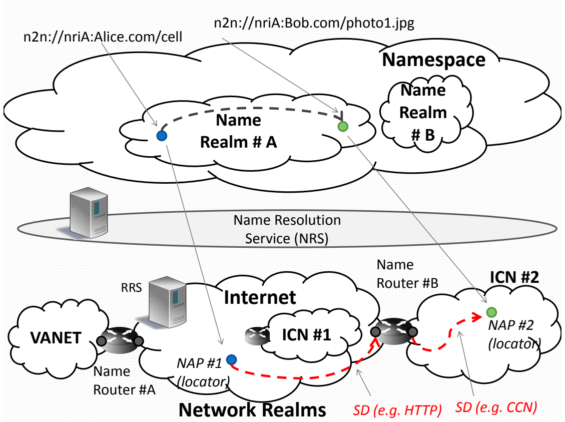


Fig. 2: Architecture Components

It is well known that large parts of this picture have already been proposed, several times, by several researchers (e.g. [2], [4], [5], [8]–[11], [19]). The aim of this work is to advocate the need for a name-to-name approach, highlight its benefits, provide insights into a possible overarching architecture and identify key components and challenges that need to be solved in order to realize this vision. In order to do so, we borrow from past work in the area.

III. Internames: A Name to Name Principle

Internames enables communications among (mobile) named-entities. It is designed to use names not only within an ICN network realm but for communicating over any network. Using names as the primary means for applications to access entities and a name-based routing and forwarding plane (e.g., name-based routers), Internames aims to do for content and services what IP did with IP addresses and routers: create a glue to interconnect networks, potentially of different technologies. In this vision, names provide access to content and service access points distributed on networks of any type, including public/private IPv4/v6 networks; public/private overlay/clean-slate ICN and IoT networks; Data Centers and Cloud; ad-hoc/mesh/cellular networks; DTNs; etc.. Different name spaces could be used to allow the architectural framework to adapt to the different needs of different contexts, where the interpretation of the names could vary based on the context. Names could be associated with additional meta-data information, e.g., description of the content, rights to use it; expiry date for the content, capabilities of devices, etc. The NRS is the entity providing high-level, network-wide interoperability and built-in capacity for evolution, as it can provide dynamic mapping of named entities to their current locations.

A. Internames: Architecture and functional components

The basic building blocks of Internames are: i) a name-based Application Programming Interface; ii) a separation of identifiers (names) and locators (addresses); iii) a Name Resolution Service (NRS) that dynamically maps names to locators, as a function of time/location/context/service; iv) a built-in capacity of evolution, allowing a smooth migration from current location-oriented networks with the ability to include as part of the network current specific architectures and technologies; v) compatibility with current ICN architectures.

Internames exposes to applications a name-oriented API (see Figure 1) that enables get, publish and subscribe to content identified by name. It also enables push of data towards the communication interface (port) of an application (i.e., a service access point) that is identified by a name. It also provides the capability to update the location of a given name, i.e., the binding between the name and its current location(s). The architecture uses a namespace, where names are associated with entities (Figure 2). Entities may be digital data (content) or service access points through which an application can send or receive data (e.g., a TCP/UDP port). We refer to an entity associated with a name as a named-entity (NE). For instance, to retrieve a content item by its name or for example to push a talkspurt of speech to a mobile phone application, the API provides access to a named-entity.

1) Name-realms: The namespace is formed by name-realms that may be disjoint containers of names, managed by different administrations. Name realms use name schema that may differ. A generic name (e.g., n2n://nriA:Alice.com/cell) is a URI composed by a name-realm identifier (e.g. nriA) followed by an identifier that uses the local naming scheme (e.g. Alice.com/cell). A name-realm may be a set of names using the CCNx naming scheme, or may be the set of current DNS names, or flat identifiers, as suggested in [19].

2) Network-realms: A named-entity is dynamically or statically bound to one or more Network Attachment Points (NAPs), i.e., addressable network ports or interfaces. These may be possibly available in different network-realms. We refer to such a NAP as a locator of a NE. A network-realm is an autonomous network using a local networking stack (e.g., IP, Ethernet, ICN, etc.), and whose routing scope is bounded to that network domain only. For instance, a network-realm may be the public Internet, an ICN, a Content Delivery Network, or a Data Centre/Cloud. Network Realms may be nested. A nested network-realm uses the networking services of the underlying realm to interconnect its nodes. Practically a nested network-realm is an overlay network. For instance, it may be an autonomous CCN within the public Internet network-realm. Like an autonomous system in IP, a network-realm has a unique network-realm identifier used for inter-realm routing.

3) Object Resolution Service (ORS): This is a searchable database that contains all names of the namespace together with related metadata. Users searching for content or a service access point would query the ORS (as with current search engines) and obtain a name or list of names. The ORS is not strictly necessary, and may be replaced by private/commercial search engines and could also be implemented in a distributed manner. Its main role would be to provide trusted association of names to descriptions, keywords and metadata.

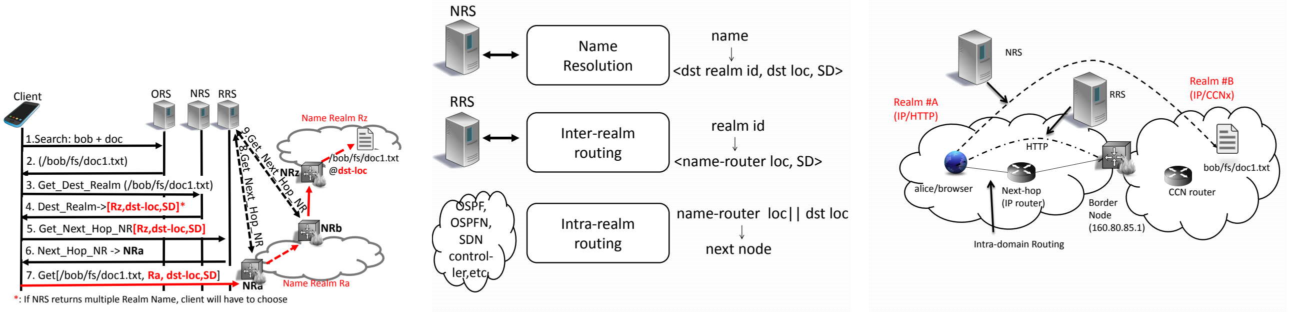
4) Name Resolution Service (NRS): The Name Resolution Service (NRS) resolves the mapping between the name-realms and network-realms. The NRS provides the information to reach a name, including the identifier of the hosting network-realm, its locator and also a service descriptor (SD), which describes the available name-oriented protocols that can be used to interact with the named-entity. By the term name-oriented protocol we refer to a communication facility used to get, push, subscribe or publish data by name, e.g., HTTP (note that HTTP 2.0 will support a PUSH primitive), CCN, SIP, etc. We believe that future mobile (e.g., 5G) networks should be characterized by the ability to handle worldwide mobility of any named-entity (and not only e.g., mobile phone numbers). Thus, the NRS has to handle trillions of names that may change location. It should manage fast location updates and support high query loads. The NRS can be a distributed system, like the DNS, and may potentially be seen as the future DNS. The resolution could be a function of time, context, location and requested service, or in general, accommodate various policies. For instance, in a disaster situation in which a section of the network is isolated, names would be resolved to different locators with respect to normal conditions. This adaptation of resolution would be performed transparently to users, who can continue using their applications as usual, as far as possible (e.g., redirecting communications to first responders).

5) Name-router: Network-realms are interconnected to each other through dedicated nodes called name-routers (Figure 2). To support the flexible integration of the architecture with current protocols, a name-router can act as a protocol proxy between the name-oriented protocols operating in each of the interconnected network-realms. For instance, as shown in Figure 2, a name-router could receive an HTTP request coming from a NAP hosted in the public Internet and translate it into a CCN request, when the destination named-entity is contained in an ICN network-realm based on CCN technology. This proxy-based modus operandi is clearly different from IP or CCN routers, which instead forward only their own protocol data units. We recognize that this proxy approach may give rise to concerns about the complexity involved in carrying out protocol translation at line rate. However, with current computing power we believe it might be an easier path than achieving worldwide consensus on a new networking protocol.

6) Routing Resolution Service (RRS): In a transit network-realm, a Routing Resolution Service (RRS) assists in the inter-realm routing. It provides the locator and the service descriptor that should be used to contact the next name-router on the path towards the destination network-realm that contains the named-entity. For instance, in Fig. 2, the RRS of the Internet realm provides the IP address of the name router #B and specifies HTTP as the name-oriented protocol to relay the users request. In a ICN/CCN realm the RRS provides a name addressing a CCN application on a name-router that performs proxy operations. For scalability reasons, the inter-realm routing is carried out on the basis of the destination network-realm identifier (resolved by the NRS), rather than on the basis of the name of the named-entity. As with BGP, the RRSs of the different realms have peering relationships through which they exchange network-realm based routing information.

7) Name-to-Name end to end approach: In Internames, the forward path and the reverse path need not necessarily be the same. Since a bidirectional flow could traverse different network realms, we treat the flow in the forward and backward direction as two flows that are resolved based on the source/destination name. In order to be backward compatible with NDN and/or utilize the additional feature provided by routers in an NDN realm where the network provider chooses to enable PIT, Internames can use the same path for the forward and reverse direction with the correct combination of gateway to the NDN realm and service descriptor.

8) Putting it all together: Figure 3a illustrates the overall message flow. The interaction of a user with the infrastructure



(a) Example Message Flow

(b) Routing Functionality

(c) Routing Example

Fig. 3: **Internames Routing**

starts with a query to the ORS; the user sends a list of key words to the ORS, which in turn finds one or more objects that satisfy this query, each of them associated with a named-entity, which are returned to the user. The user chooses one of them and asks the Name Resolution System (NRS) to resolve the current location of the named-entity in terms of its network-realm (realm id), its locator within the realm (dst-loc) and the protocol means to reach it within the realm (service descriptor: SD). The locator of the named-entity complies with the network technology used in that network-realm, e.g., may be an IP address, an Ethernet addresses or the name itself in the case of CCN realms. Then, to transfer a message towards the destination realm through one or more transit network-realms, the RRS provides the locator of the next hop name-router towards the destination realm and the related name-oriented protocol to be used to reach it. In Figure 3a the name-routers contact the logical RRS multiple times to obtain the next hop name-router. Note that step-8 could also be a request to a local-RRS within the domain Ra. To route the packets of the name-oriented protocol in a network-domain, an intra-domain routing protocol can be used, like OSPF in an IP-based network-realm or OSPFN in a CCN-based network realm (see Figure 3b). When the final network-realm is reached, the locator (dst-loc) and the SD provided by the NRS are used, without involving the RRS. Figure 3c shows as an example the case in which the browser of Alice (a named-entity with name alice/browser) gets the bob/fs/doc1.txt content. Within each realm there are different transport sessions and (in case) protocols (e.g. TCP for Internet realm or receiver driven flow control for CCN), similarly to what occurs in a proxy cascade.

Alices browser is located in an IP network-realm, while the network-realm of Bob's document is a CCN one. Alices Internames functionality queries the NRS, which returns the current location of the content, i.e., the triple < dst realm id = realm #B, dst-loc = bob/fs/doc1.txt, SD=CCNx >. Then Alice queries its RRS for realm #B and the RRS returns the tuple < name-router loc = 160.80.85.1, SD = HTTP >. Now Alice uses an HTTP GET of bob/fs/doc1.txt to the proxy located at 160.80.85.1. The IP intra-domain routing will setup the route towards such a name router. The HTTP/CCN proxy functionality located on the name router handles the HTTP session and repeats the same process done by Alice but with a final CCN Get of bob/fs/doc1.txt. Moreover, in this specific case the RRS lookup phase is skipped since we have reached the destination realm and we can use the final locator and SD provided by the NRS, i.e. dst-loc = bob/fs/doc1.txt and SD=CCN. In other words, the RRS functionality is used until we reach the final realm; inside the final realm local routing uses the information provided by the NRS. It is noteworthy that in case of a single IP realm, this approach is similar to what is used in the current Internet , with the role of NRS similar to DNS.

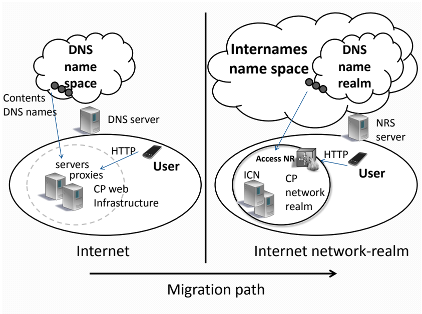


Fig. 4: Migration (left: before migration; right: after migration)

IV. Migration Path and Example

Internames inherently supports a smooth migration from current networks. Indeed, name and network realms can be progressively deployed. The NRS technology can be made compatible with DNS and the DNS namespace can be considered as the first name-realm of the Internames, as much as the current Internet can be considered as the first network-realm. Production IP-based services are transparently merged in the Internames architecture. Operators do not need to necessarily switch from IP to ICN to support name-based communication. In what follows we discuss the possible steps of network actors (content providers, network carriers, and users) to migrate IP/DNS services to Internames-based services.

a) Content-provider: Consider a content provider (CP) whose business is storing and distributing content by using its own distribution network, such as current CDN service providers. Today, a CP delivers content to users by using an infrastructure comprising web servers, web proxies and DNS servers (Figure 4-left). Content items have names that are in the DNS namespace. Users access content located on servers and/or proxies using the HTTP protocol. As shown in Figure 4-right, a first step of the migration consists in replacing the authoritative DNS server of the content provider with an NRS server. This does not perturb production services, since the NRS would be backward compatible with DNS. A second step is to deploy a private network-realm, connected to the Internet through a name-router (Access NR). The connectivity of such network-realm could be an overlay IP network; i.e., the network-realm would be nested in the Internet network-realm. The networking technology used within the network-realm (e.g., ICN) may be optimized for the distribution of the specific content (e.g., videos) of the provider. The final step consists in updating the NRS with proper Service Descriptors, which use the IP address of the name-router as next-hop and selects HTTP as the name-oriented protocol of choice. Now content that was distributed in the Internet network-realm can be migrated into the new network-realm.

b) ISP: Consider an Internet Service Provider (ISP) that transports data from/to user premises or customer networks. For instance, companies providing fixed and wireless Internet access, or Internet Tier-1 carriers. ISPs may deploy new network-realms (e.g. nested in their actual IP infrastructure) either to build name-based transit networks, interconnecting newly deployed name-realms of content-providers, or to proxy name-based services that are available on the Web (Internet Realm). The transit service may use the same technology as the interconnected realms, thus avoiding CP access routers having to perform protocol bridging functions. Proxy services may be useful to reduce inter-domain traffic, with an ISP realm implementing an ICN or HTTP proxy hierarchy.

c) Users: End-users can continue to use IP and DNS based services, as the NRS would be DNS compatible; access routers of newly deployed network-realms would support the proxying of current name-based protocols such as HTTP. In this case, end-user devices can belong to the Internet realm. In an evolutionary scenario end-user devices can have interfaces connected (directly or via tunnels) to the newly deployed network realms.

V. Summary

Moving to an ubiquitous ICN would mean a radical change in the way the Internet works today. On the other hand, if we give up implementing ICN functionality within the network, such as routing-by-name and forwarding-by-name, then ICN as a concept would collapse either into CDN internetworking or into an application-layer ICN, exploiting HTTP, or just evolving from HTTP, without network-layer support. We propose a third way for ICN by confining ICN operation in sections/portions of the networks, leaving the core untouched (at least initially). We believe that ICN, augmented with Internames, could be of interest in access networks, and in ad hoc environments without infrastructure support, such as isolated sections of the networks and IoT scenarios. In these environments, scalability and deployment issues are less important than in the Internet at large. In addition, as noted in recent papers (e.g., [3]), in-network caching and performance gains brought about by ICN in the core network are not as compelling as initially foreseen by ICN advocates. Instead, the benefits of ICN described above in specific realms and access sections are very attractive and indeed attainable. Internames would also empower advanced mobility functionality for every kind of network. The challenge is in designing and implementing the NRS to be scalable and to have high performance. To this end, we expect to learn from past efforts in evolving the DNS and recent proposals (e.g. [19]). Another issue is the alternative between name-routers working as protocol proxies between interconnected network-realms or with a new protocol operating over the network layers of the interconnected network-realms, with a related corresponding new protocol data unit. In any case, the issue of complexity of this architecture is to be evaluated taking into account the availability of cloud networking functionality that would have been hardly predictable only few years ago.

Acknowledgment

This research was funded by the joint EU FP7/NICT GreenICN Project, Contract No. 608518 and NICT No. 167. We would also like to thank the anonymous reviewers for their insightful comments.

References

[1] V. Jacobson, D. K. Smetters *et al.*, "Networking Named Content," in *CoNEXT*, 2009.

[2] J. Chen, M. Arumaithurai *et al.*, "COPSS: An Efficient Content Oriented Pub/Sub System," in *ANCS*, 2011.

[3] S. K. Fayazbakhsh, Y. Lin *et al.*, "Less Pain, Most of the Gain: Incrementally Deployable ICN," in *ACM SIGCOMM*, 2013.

[4] D. Trossen and G. Parisis, "Designing and realizing an information-centric internet," *Communications Magazine, IEEE*, vol. 50, no. 7, pp. 60–67, July 2012.

[5] D. Han, A. Anand *et al.*, "Xia: Efficient support for evolvable internet-working," in *USENIX NSDI*, 2012.

[6] GreenICN: Architecture and Applications of Green Information Centric Networking, a joint EU-Japan research project. www.greenicn.org

[7] Saltzer, J. H., D. P. Reed, and D. D. Clark "End-to-End Arguments in System Design". International Conference on Distributed Computing Systems. Paris. April, 1981. IEEE Computer Society, pp. 509-512

[8] T. Koponen et al. : Architecting for Innovation, ACM SIGCOMM Computer Communication Review, Volume 41 Issue 3, July 2011

[9] R. Moskowitz, P. Nikander: Host Identity Protocol (HIP) Architecture, RFC 4423, May 2006

[10] D. Farinacci, V. Fuller, D. Meyer, D. Lewis: The Locator/ID Separation Protocol (LISP), RFC 6830, Jan. 2013

[11] C. Dannewitza, D. Kutscher, B. Ohlmanc, S. Farrelld, B. Ahlgrene, H. Karla: Network of Information (NetInf) An information-centric networking architecture, Computer Communications, Elsevier, Volume 36, Issue 7, 1 April 2013, Pages 721735

[12] Ahlgren, B. ; Dannewitz, C. ; Imbrenda, C. ; Kutscher, D. ; Ohlman, B..A survey of Information-Centric Networking, IEEE Communications Magazine, Volume:50, Issue: 7, 2012

[13] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J.Rexford, S. Shenker, J. Turner, OpenFlow: enabling innovation in campus networks ACM CCR, Vol. 38, Issue 2, pp. 69-74, 2008

[14] Network Functions Virtualisation Introductory White Paper, portal.etsi.org/NFV/NFV_White_Paper.pdf, OpenFlow/SDN World Congress (Darmstadt, Oct. 2012

[15] GreenICN: Architecture and Applications of Green Information Centric Networking, a joint EU-Japan research project. www.greenicn.org

[16] A. Detti, N. Blefari Melazzi, S. Salsano, M. Pomposini: CONET: A Content Centric Inter-Networking Architecture, ACM SIGCOMM ICN Workshop 2011, August 19, 2011, Toronto, Canada

[17] Jaron Lanier, Who Owns the Future, 2013, London: Allen Lane.

[18] C. Perkins, "IP Mobility Support for IPv4", IETF Request for Comments (RFC 3220), 2002.

[19] D. Raychaudhuri, K. Kiran Nagaraja, A. Venkataramani, MobilityFirst: A Robust and Trustworthy Mobility-Centric Architecture for the Future Internet, ACM SIGMOBILE MCCR, Volume 16 Issue 3, July 2012