A Named Data Networking Flexible Framework for Management Communications

Daniel Corujo[a], Iván Vidal[b], Jaime Garcia-Reinoso[b], Rui L. Aguiar[a]

[a] Universidade de Aveiro, Instituto de Telecomunicações, Campus Universitário de Santiago, 3810-193, Aveiro, Portugal

[b] Universidad Carlos III de Madrid, Av. Universidad, 30, 28911, Leganés (Madrid), Spain

Abstract - The ongoing changes in the way we use the Internet are motivating the definition of new information-distributing designs for an interworking layer. Recently, information-centric networking (ICN) concepts have defined mechanisms focusing on *what* information to get rather than *where* it is located. However, the still unfledged architectures instantiating such concepts have disregarded vital operations, such as management aspects aiming to optimize content retrieval by the User Equipment (UE). In this paper, a flexible management framework is proposed, which enhances existing Named Data Networking (NDN) architectures in allowing both the network and the UE to employ management mechanisms over the NDN fabric. We illustrate our management framework with an implementation over an open-source NDN software, considering the specific case study of interface management. We quantitatively access the performance gains achieved through the usage of this framework in such scenarios, when compared to un-managed NDN mechanisms.

Index Terms – Information-Centric Networking (ICN), Named-Data Networking (NDN), Network Management

1. Introduction

In recent years, rapid developments in how we access an ever-increasing amount of new online content have changed current communication patterns. Traditional communication between a pair of networked machines has evolved into a scenario where new services generate unprecedented amounts of content (e.g., video traffic and cloud access), and at the same time allow multi-link mobile devices to access that information via different connectivity opportunities. The existing host-centric architecture has been patched to encompass content-oriented mechanisms such as Content Delivery Networks (CDN), peer-to-peer overlays and HTTP proxies, deployed over the existing infrastructure. Supporting these enhancements to access content, network operators employ a series of management procedures governing a complex heterogeneous environment, through the cumulative usage of mechanisms such as Quality of Service (QoS), network policies, load balancing, failback actions, over-the-air updates, amongst others. However, the sheer amount of content being accessed has highlighted functional limitations in terms of flexibility, performance and cost, motivating the development of more adequate network paradigms for an Internet of the Future.

The focus on content distribution has led to the concept of Information-Centric Networking (ICN) allowing content to be addressed by name and not by location or end-point addresses. This principle has motivated different approaches, such as Data-

Oriented Network Architecture (DONA) [1], Publish/Subscribe Internet Routing Paradigm (PSIRP) [2], Network of Information (NetInf) [3] and Named Data Networking (NDN) [4]. A survey on ICN approaches can be found in [5].

These proposals leverage the evolution required by today's Internet to support this content-centric vision, focusing on aspects such as hierarchical content naming, information-centric communication and content-based security, which are core features of NDN [4]. However, there has been very little consideration on management aspects for ICN architectures, which are fundamental for supporting network operations, such as, e.g., those aiming to increase user experience when accessing content with different requirements. We argue that the provision of intrinsic management functionalities is important for the successful future deployment of ICNs, incrementing existing and new Future Internet scenarios, with support for QoS, network policies management, amongst other operations.

In this paper, we focus on the NDN architecture to discuss management enhancements potentially addressable as well by other ICN architectures. To achieve that, we define an integrated and flexible framework that:

1) Considers a generic approach allowing management procedures to collect input or triggers from various sources, and use that information to optimize network operations. These aim to assist or control network attachment procedures of terminals, towards network resource optimization and user experience satisfaction;

2) Defines interfacing points with the NDN fabric while proposing improved mechanisms allowing the integration of management procedures.

The remainder of this document is organized as follows. Section 2 introduces the base mechanisms of NDN and provides insight into current evolution. Section 3 presents our NDN flexible management framework, followed by Section 4 where we focus on a specific use case (i.e. management of the terminal's connection point for content retrieval) and we evaluate the performance that can be achieved with our framework, in this specific use case, from an experimental perspective. Lastly, Section 6 presents conclusions and future work.

2. Background and Related Work

2.1. NDN Basic Operation

NDN relies on two packet types defined in [4]: *Interest* and *Data*. *Interests* are used by a consumer to ask for content. They contain a Content Name based on a hierarchical structure: its components are separated by '/', (similar to a URL, such as */domain/content/Videos/videoA.mpg/version/segment*), with the prefix providing global and organizational routing information, and a suffix showing versioning and segmentation details. When an *Interest* packet reaches any node with data matching the Content Name, it consumes the *Interest* and responds with a matching *Data* packet, carrying back the content.

When an NDN-node receives an *Interest* packet, a set of functional structures is consulted: the Content Store (CS), the Pending *Interest* Table (PIT) and the Forwarding Information Base (FIB).

The CS is checked to see if the requested data is already available. If the name requested no entry there, the PIT is checked. This structure keeps track of *Interests* forwarded to content sources that were not yet consumed. If an *Interest* has no match in either the CS or PIT, a new PIT entry is created and the packet is forwarded towards one or more interfaces that might lead to the respective content sources. For this, the FIB associates Content Name prefixes towards potential holders of the content, with some routing protocol defining the forwarding state in the FIB, (e.g., routes to applications or physical interfaces), or through a registration in a local NDN store.

An interesting detail from [4] and [6], is the ability for FIB entries to address multiple interfaces[1]. Each NDN-node contains a *Strategy Layer* that can use several options to forward an *Interest* packet. The forwarding strategy can vary from sending an *Interest* sequentially on each face until a *Data* is received, to more elaborated designs that evaluate which interfaces provide better performance in retrieving specific content. A specific mechanism for best face determination is described in [4], where the Strategy Layer runs experiments in which an *Interest* is sent (e.g., every second or when a packet is lost) through all available faces, towards a given prefix. If a face provides lower end-to-end delay than the previous *best face*, it becomes the new one until the next test. Hereinafter, we refer to this process as probing. Another possibility suggested in [4] and [6], is to define a program within each FIB entry that defines forwarding choices and behaviour. This program could be configured with a set of instructions, such as *sendToAll* and *sendToBest*, determining the forwarding of *Interests* under a predefined and static strategy. However, this behaviour is not thoroughly explored in [4] and [6], which provides a default strategy of sending *Interests* to all broadcast capable faces and, if no answer is received, all other faces are tried out in sequence.

2.2. NDN Evolution

There have been several proposals enhancing the base architecture of NDN not only to better tackle the problems it proposes to address, but also to improve its behaviour when handling current or future traffic patterns.

The work in [7] evaluates the performance of NDN when mapped to a Voice over IP (VoIP) application, transporting SIP and RTP data in a real-time conversation.

Extensions supporting NDN-node mobility, without having to undergo a full *Interest* re-routing toward the content source, have been proposed [8]. This work considers a proxy acting as an anchor point for *Interest* and *Data* packets. When nodes connect to an NDN-domain, they associate with that proxy (using it to reach content on their behalf) and when they move, they report movement changes to it, updating the *Data* packets being forwarded to the new location. However, this update does not provide any preferences, policies or any other information which could be used for assisting and managing this mobility procedure, thus allowing only un-optimized handovers.

The authors of [9] analyse how *Interest* route selection is affected by policies, exploring content name granularity. However, it focuses on the economic incentives for routing only, not considering aspects such as different content requirements,

[1] In fact, under NDN, the term *interface* is replaced by *face* (which we will use from now on) because packets are not only forwarded over hardware interfaces but also directly between application processes.

content/application-driven policies or dynamic route selection based on different interface technologies.

2.3. Management Requirements in NDN

A thorough analysis of deployment motivations for NDN as a future internetworking architecture is presented in [10], highlighting the need for network-grade solutions, such as manageability aspects that allow the network to control the content reception by the user (i.e., optimal link selection, Quality of Service, policing, etc.). In fact, considering the outline of the NDN project [6], management procedures are only considered for Storage and Usable Trust. These, along with the previously mentioned Strategy Layer, do not consider network input or intervention in their management processes. As such, they are limited to operation based on static rules, or relying on information collected locally by the node. Its management, as it is, is completely local. As a result, there is no coordinated effort between NDN-nodes and the network as a whole, with which to provision or optimize not only NDN operating aspects, but also the performance of *Data* packet reception in terms of policies. The usage of policies, analogous to the IP world, allows the network to control and suggest to NDN-nodes the most preferable access network, by providing discovery information (e.g., which WLAN networks are available within a 3G cell) or by indicating a set of rules considering preferred access points at different times of the day, or routing traffic flows depending on the content[2].

3. The NDN Flexible Management Framework

3.1 General Architecture

This section describes the management framework proposed in this paper for NDN. We argue that the interaction between information existing in the network and information within the point-of-view of the terminal could be used to impact NDN operations, optimizing both network procedures and user experience. This framework was designed as a comprehensive and flexible solution, capable of supporting the functionalities that may be required by different management procedures in a future content-oriented network, such as QoS provision and face management.

Figure 1 shows a general overview of the different functional entities comprising our framework. It considers the deployment of a Manager Entity (**ME**) in the network, able to interact with a set of Management Agents (**MA**) located in different devices, such as network and user equipment. For simplicity, in the remaining of this paper we focus on MAs deployed in User Equipment (UE). As a result of the interaction between MEs and MAs, the network and any UE can exchange information to appropriately coordinate procedures, taking advantage of the different information available in the network and locally at the UE. By deploying the different management entities as application processes, the framework can be easily decoupled from the underlying network architecture, enabling easy interoperation with any Information-Centric network approach.

---

[2] The example policy characteristics highlighted here are based on the Discovery and Inter-Routing policies as defined by the 3GPP in the 3GPP TS 24.312 technical specification (Release 10).

Considering this traditional network management approach, the challenge is centred on how the MA and ME entities interface with the NDN fabric. Figure 1a presents the core components of the NDN architecture at the UE, also coupling an MA. This functional entity can be deployed as a single application process that interfaces with internal NDN structures, such as the PIT and the FIB. Thus, the MA is able to access and update these structures as a result of any management procedures. The UE can run end-user applications, such as web browsing, telephony or Internet TV, which are able to exchange *Interest* and *Data* packets. Our framework also regards management (e.g., event generation and sending/receiving commands to optimize a NDN-related process) as content exchanged through *Interest* and *Data* packets. MAs residing inside NDN-enabled UEs behave as producers and consumers of content related to management operations, and any management information provided by the ME is viewed as an NDN name (e.g., under the name prefix */domain/management/ME/*). Moreover, the MA can also interface with local applications to obtain information about the content to be retrieved. This provides relevant information that may be used to guide the decisions of the management processes. As an example, a local application can provide the MA with information about QoS requirements for a specific video content. This information can be provided to the ME, which can coordinate an appropriate resource delivery in the network. Finally, the MA is capable of interfacing with the lower layers of the UE, obtaining link information that can also be relevant to the management procedures (e.g., identifying available wireless access networks in the vicinity of the UE).

Shown in Figure 1b, the ME is a functional entity located in the operator network that interacts with MAs to handle management procedures. The ME is triggered by different mechanisms, existing elsewhere in the network, to report information that can be relevant to aid management. The ME processes this information, which can then be used to coordinate any necessary management operations. For example, the ME can be notified when the traffic load of a given access network exceeds predefined thresholds. Information about traffic load can then be used by the ME to coordinate an appropriate selection of faces in the UE for content retrieval. This logical entity can be implemented in a single centralized network node, or distributed over nodes residing in different parts of the network, for the sake of scalability and redundancy. The flexibility of the proposed NDN management mechanism can be reutilized in different ways to support distributed coordination, allowing the exchange of state amongst different MEs belonging to different domains, or by using other distributed network mechanisms to act as triggers for NDN management operations. Such interfacing can go beyond triggering and can take many forms (e.g., SNMP), but is out of scope of this paper.

The flexibility of this model allows its application to a wide range of management scenarios. As an example, in QoS provisioning, the MA could be used to dynamically provide user and application requirements to the ME in the network. This information can be used to initiate the necessary admission control and resource reservation procedures in the network, using NDN interfaces located in the required entities. On the other hand, in the case of face management, a change in access network traffic conditions can trigger a management procedure in the ME. Here, the ME can provide new traffic policies to the MA of the UE, describing other usable access networks. As a result, internal NDN structures such as the FIB can be updated at the UE to enforce network management decisions. All management traffic will look like regular NDN traffic to the NDN-nodes, identified by specific names.

3.2 Support Procedures

The framework presented in Section 3.1 requires the exchange of management data between the MA and an ME. This exchange must fulfil the following properties:

- **Security.** MA and ME must be able to authenticate and determine the trust that can be established on management data. Also, a UE can use a broadcast interface towards a Point of Attachment (PoA), and it is necessary to protect the confidentiality, integrity and authenticity of the content exchanged, as it may contain sensible information that must not be vulnerable to unauthorized eavesdropping, modification or creation.

- **Asynchronous exchange**. Other than just pulling content via an *Interest*/*Data* packet exchange, both the MA and the ME must be able to push unsolicited management content to one another.

- **Reliability** (optional). NDN transport can operate on top of unreliable data delivery services. Nevertheless, in some use cases, content exchanges between the MA and the ME must proceed reliably (e.g., to send the ME a set of QoS requirements that should be satisfied to retrieve a specific video content).

3.2.1 Bootstrapping for Reliable and Secure Management Content Exchange

Inspired by the procedures used to setup a secure Voice-over-NDN conversation [7], we have defined a bootstrapping procedure between the MA and the ME, illustrated in Figure 2.

The first step involves the MA discovering an appropriate ME, by broadcasting an *Interest* with the name */domain/management/mgmt-case/ME* to its local network. The name component *mgmt-case* refers to the management capacities that the MA requires from the ME (e.g. an ME to handle QoS provisioning can be discovered by issuing an *Interest* to */domain/management/qos/ME*). As a result, the UE obtains a short hand identifier for the ME (i.e. the *ME-publisher-id*) and a key locator, which indicates the name that can be used to retrieve the public key of the ME. Assuming that the public key of the ME is authorized by another key trusted by the MA, (e.g. a public key corresponding to */domain*), the MA can identify the ME as an acceptable signer for management data. The MA selects an encryption algorithm, out of those indicated by the ME in the *Data* packet, and generates a session key, *Ks*. Then, it registers its desire to serve *Interests* matching a given NDN name (e.g. */domain/management/mgmt-case/MA-publisher-id*), where *MA-publisher-id* is a global and unique identifier for the MA, such as the cryptographic digest of its public key). Finally, the MA sends a new *Interest* to retrieve management *Data* from the ME that includes, as NDN name components, the shorthand identifier of the MA (*MA-publisher-id*) and some additional information encrypted with the public key of the ME, such as the encryption algorithm (*security-mechanism*) and *Ks* chosen to guarantee confidentially of the content exchanged between the MA and the ME. Note that different security infrastructures could be used in this approach, suitably adapted. With this, the MA is prepared to securely receive content from the ME.

Nevertheless, before the ME starts exchanging management content with the MA, it generates a challenge (i.e. a *nonce*) and expresses the *Interest* in obtaining the response to this challenge from the MA. The MA responds to the *Interest* with a *Data* packet containing the answer to the challenge. Consequently, the ME can retrieve the public key of the MA and identify it as an acceptable signer for management content. In

addition, the ME verifies the signature of the *Data* packet and checks the validity of the answer to the challenge. Therefore, this exchange allows the ME to verify that the encryption algorithm and the session key are valid for the MA. At this point, the ME is prepared to receive management data from the MA.

Once these initialization procedures conclude, MA and ME can exchange information to coordinate the execution of any management activities.

3.2.2 Asynchronous Exchange of Management Data

After bootstrapping, the framework allows the MA to securely pull management content from the ME and vice versa. Case 1 in Figure 2 shows the scenario where the MA retrieves a specific management information item *content-name* from the ME (the procedure of having the ME pulling content from the MA would proceed in a similar way).

In addition, pushing unsolicited content (e.g., commands to nodes, or informational events to the network) between the MA and the ME is also supported. As suggested in [6], we support pushing content between applications by implementing a double *Interest*/*Data* exchange. Case 2 of Figure 2 shows the necessary procedures allowing the MA at the UE to push management data towards the ME. The procedure to push content from the ME to the MA follows a similar approach.

This procedure starts with the MA sending an *Interest* to the ME soliciting it to receive management content with a local sequence number. Sequencing is necessary to enable the recovery of new content instead of cached content. If the ME is interested in retrieving content from the MA, it answers back with a *Data* packet, indicating its willingness to accept management content. Then, the ME sends an *Interest* to retrieve the management data with the sequence number given by the MA. The MA responds with a *Data* packet containing the information it wanted to push to the ME. The information contained in the *Data* packet is encrypted with the session key established during bootstrapping.

Finally, if reliability is desired, MAs and MEs must retransmit *Interest* packets not satisfied in a reasonable period of time (either to pull or push management content). This mechanism is suggested in [4] and [6], and improves the reliability of the asynchronous data exchange.

4 Evaluation

In this section we evaluate the feasibility of our framework, addressing the particular use case of face management (i.e., configuring and selecting an appropriate face to retrieve a given content). In this context, we argue that our approach can provide a better-performing alternative to the mechanisms presented in [4] and [6], such as probing at the NDN strategy layer.

To evaluate the benefits of our framework in this use case, we considered the simple validation scenario depicted in Figure 3. The main objective is to show that an ME residing in the network, with the ability to know the topology and the network

conditions surrounding PoAs or a UE[3], is able to assist the latter in network discovery and selection procedures, according to the operator policies. This would enable the network operator to perform a global and more appropriate management of the available access resources, achieving an adequate distribution of the load among different access networks, which is globally beneficial for the users, while incurring a negligible overhead.

The figure features a UE with two network interfaces, providing it with physical connectivity to PoA_A and PoA_B respectively. Additionally, PoA_C is available to the UE at its current location from one of its network interfaces, but is not initially attached to it. The scenario includes a Content Server that receives *Interests* matching a given prefix requested by the UE. An ME is also deployed, allowing the network to assist the UE in face management procedures. The different entities are interconnected by means of an NDN network. Under this validation scenario, we can vary the traffic load at the different PoAs and evaluate the feasibility of our framework, and the performance that can be achieved compared to probing mechanisms locally executed by the UE strategy layer.

The test-bed was deployed in different virtual machines connected to the same virtual network. We introduced bandwidth constraints, by limiting the capacity of every link to the UE to 1 Mbps (bidirectional) using the *tc* (Traffic Control) tool. The test-bed uses the CCNx software[4] and our framework software. Three Java applications were implemented, using the CCNx Java API: an NDN UE (featuring an MA), a Content Server and an ME.

The NDN UE generates periodic *Interests* matching a given prefix and computes the RTT of each *Interest*/*Data* exchange. This application can be launched in a basic NDN mode or in a framework-managed mode. The NDN Content Server receives *Interests* from the UE and replies back with random content. The basic NDN mode represents an extension of the procedure defined in [4], where the UE occasionally sends a regular *Interest* through all the available faces associated with a prefix and measures the RTT for each *Interest*/*Data* exchange. The face that obtains the lowest value of RTT becomes the current face and is used for subsequent *Interests* associated with the prefix, until the next probing takes place or an *Interest* times out. In addition, our implementation supports sending multiple *Interests* through each face in order to compute the best Interface for *Data* retrieval. In the framework-managed mode, the MA changes the UE's current face when it receives management information from the ME reporting a better face for content retrieval, or an alternative PoA that could be used to improve the performance. The mechanism is independent of the format of the management information, depending solely on the MA and ME implementation, and how they interface with the UE access interfaces and network information, respectively. For testing purposes, we assume that the ME has an up-to-date access control indication of the UE location, its supported access interfaces and current network conditions on the PoAs at its vicinity.

---

[3] For example, schemes using IEEE 802.21 provide a Media Independent Information Server indicating the topology for a network, and allow the usage of events indicating changes in wireless link conditions.

[4] http://www.ccnx.org/

Each experiment comprises a 160 seconds time interval, divided in four periods of 40 seconds. We vary the background traffic traversing each PoA in each period, according to the configuration presented in Table 1. Background traffic is generated using *iperf*[5] according to a Poisson distribution. The average rate for low, medium and high traffic loads was calculated to obtain significant differences in the RTTs measured at NDN UE, taking into account the Poisson distribution of the background traffic and the capacity of the links connecting to the UE.

Three different trials were defined to compare the performance that can be achieved governing face selection by means of probing mechanisms and our management framework: (1) basic NDN with 1 probe per face, (2) basic NDN with 5 probes per face, and (3) NDN under a framework-managed mode. In trials (1) and (2) the probing period was set to be one every 200 packets, as suggested in [4].

4.1 Results Discussion

Figure 4a shows the results for the UE running in basic NDN mode and sending a single probe *Interest* per available face. It shows the instantaneous RTT for each *Interest* sent from the UE, the average RTT computed from the instantaneous values in the last 5 seconds and the current face used by the UE to send *Interests*. As can be observed, under similar traffic load conditions (e.g. time period 0-40 seconds), probing with a single *Interest* per face may lead to the selection of any available face, (as any PoA can provide a better RTT in an isolated probe), which may imply instability due to oscillations in face selection. On the other hand, even under different average traffic conditions the decision is error-prone, (e.g., choosing a PoA with medium or high load), as can be observed from the wrong face selections made during the experiment (e.g. choosing face 0 in periods 80-120 and 120-160 seconds). Increasing the number of *Interests* used in a single probing process may improve the performance, but with the cost of increasing the overhead and thus decreasing efficiency. To illustrate this, we executed a set of experiments increasing the number of *Interests* per probing process to 5. Due to space constraints, instead of including new figures related with probing, we present a brief summary of all the experimental results that we obtained under this specific test-bed in Table 2.

Figure 4b shows the results for face management using our framework. As can be observed, face selection remains stable in the UE until performance significantly decreases at the beginning of the period 80-120 seconds. This is a consequence of the increment in the traffic load traversing PoA_A (face 0). When this happens, the ME sends an informational message to the UE's MA, which starts retrieving *Data* from PoA_B (face 1). In the period 120-160 seconds, traffic load traversing this PoA increases to medium. In this case, the ME can instruct the MA to detach the UE's face 0 from PoA_A and to attach to PoA_C, (otherwise invisible to the UE), improving performance.

As shown, the face management service implemented using our proposed framework improves the performance when compared to probing, also achieving a reduced overhead. This is due to a more stable face selection, assisted by operator policies, not subject to transitory variations of traffic load traversing the PoAs, and to the fact that the UE can take advantage of information provided by the network about

[5] http://iperf.sourceforge.net

new candidate PoAs for attachment. As a particular example of this, by using a new PoA in the interval 120-160 seconds, which is available in the vicinity of the UE, the management framework can achieve a reduction of 14.78% in the average RTT with respect to basic NDN with 1 probe per face. In addition, with the management framework, face oscillations are avoided.

Unlike probing, our framework does not require all interfaces to be always active. As shown in Figure 4b, since only face 0 is used for data retrieval in the time interval 0-80 seconds, face 1 could be deactivated and re-activated when strictly necessary, (i.e. after 80 seconds), when performance achieved through face 0 significantly decreases. Activating and deactivating network interfaces can be especially useful for resource saving (e.g. battery-operated handheld devices). Note that this cannot be done with probing, where all the faces must be active to decide which one is the best for content retrieval. Furthermore, the ME can decide which of the UEs attached to a saturated PoA should be moved to a different PoA, in order to improve the network usage.

5 Conclusions

In this paper, we presented a flexible and comprehensive management framework for NDN. This framework introduces a Manager Entity (ME) in the network, which can interact with a set of Management Agents (MAs) located in the User Equipment (UE) to coordinate management procedures. The proposed solution does not require significant changes to the NDN architecture, as the mechanisms defined to securely pull/push management content between MEs and MAs use the regular NDN transport through specific management naming. Our approach provides a reliable, secure and asynchronous management structure. Furthermore, the framework can be easily decoupled from the underlying network architecture, enabling an easy interoperation with any Information-Centric network approach.

To evaluate the feasibility of our framework, we have covered the particular use case of face management in the UE, and we have evaluated its benefits with respect to probing at the NDN strategy layer, by means of a validation scenario based on the CCNx software.

Our future work will focus on enhancing the control interfacing capabilities of the management content exchanged between the ME and the MA, as well as studying the applicability of our framework to other areas where management plays an essential role, such as QoS provisioning, policy management, remote administration of NDN-nodes and inter-domain support.

Acknowledgments

This article has been partially supported by the Portuguese Foundation for Science and Technology (FCT) grant agreement SFRH / BD / 61629 / 2009 and by the Madrid Community through the MEDIANET project (S-2009/TIC-1468).

References

[1] T. Koponen, M. Chawla, B. Chun, A. Ermolinskiy, K. Kim, S. Shenker, and I. Stoica. “A Data-Oriented (and beyond) Network Architecture,” *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 4. ACM, 2007, pp.

181–192.
[2] N. Fotiou, D. Trossen, and G. Polyzos, “Illustrating a Publish-Subscribe Internet Architecture,” *Telecommunication Systems*, pp. 1–13, 2010.
[3] K. Pentikousis, “Distributed Information Object Resolution,” *in Proceedings of the 8th International Conference on Networks.* IEEE, 2009, pp. 360– 366.
[4] V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs, and R. Braynard, “Networking Named Content,” in *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies*. ACM, 2009, pp. 1–12.
[5] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, B. Ohlman, "A survey of information-centric networking," IEEE Communications Magazine, vol. 50, no. 7, pp. 26-36, 2012.
[6] 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,” PARC, Tech. report ndn-0001, Tech. Rep., 2010. Available online (May 2012): http://www.named-data.org/ndn-proj.pdf
[7] V. Jacobson, D. Smetters, N. Briggs, M. Plass, P. Stewart, J. Thornton, and R. Braynard, “VoCCN: Voice-over Content-Centric Networks,” in *Proceedings of the 2009 Workshop on Re-architecting the Internet*. ACM, 2009, pp. 1–6.
[8] J. Lee and D. Kim, “Proxy-assisted Content Sharing Using Content Centric Networking (CCN) for Resource-limited Mobile Consumer Devices,” *IEEE Transactions on Consumer Electronics*, vol. 57, no. 2, pp. 477–483, 2011.
[9] S. DiBenedetto, C. Papadopoulos, and D. Massey, “Routing Policies in Named Data Networking,” in *Proceedings of the ACM SIGCOMM Workshop on Information-Centric Networking*. ACM, 2011, pp. 38–43.
[10] D. Trossen, M. Sarela, and K. Sollins, “Arguments for an Information-Centric Internetworking Architecture,” *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 2, pp. 26–33, 2010.

DANIEL CORUJO (dcorujo@av.it.pt) received his computer and telematics engineering degree in 2006 and his MsC in 2007 from the Electronics, Telecommunication and Informatics Department, University of Aveiro, Portugal, where he is currently pursuing his PhD in Communication Models for the Future Mobile Internet. He has previously worked in telecommunication management software at Nokia Siemens Networks and as a IMS deployment executive for the research branch of Portugal Telecom. He is currently working as a researcher for the Advanced Telecommunications and Networks Group at Instituto de Telecomunicações, Universidade de Aveiro, Portugal, where he works in several EU research projects and is pursuing research areas in mobility mechanisms for heterogeneous networks, Future Internet Architectures and the Internet of Things.

IVÁN VIDAL (ividal@it.uc3m.es) received the Telecommunication Engineering degree in 2001 from the University of Vigo, and the Ph.D. in Telematics Engineering in 2008 from the University Carlos III of Madrid. He is currently working as visiting professor at Universidad Carlos III de Madrid. His research interests include Multimedia networking, IP Multimedia Subsystem (IMS), P2P networks and Information-Centric Networking (ICN). Currently, he is working in the Spanish MEDIANET project related with video streaming distribution over peer-to-peer networks in the future Internet.

JAIME GARCIA-REINOSO (jgr@it.uc3m.es) received the Telecommunications Engineering degree in 2000 from the University of Vigo, Spain and the Ph.D. in Telecommunications in 2003 from the University Carlos III of Madrid, Spain. He is currently an associate professor at Univ. Carlos III of Madrid having joined in 2002 and he has published over 35 papers in the field of broadband computer networks in magazines and congresses. He has been involved in several international and national projects related with protocol design, user localization, broadband access, peer-to-peer overlays, Next Generation Networks and signaling protocols like the EU IST MUSE, the EU NoE CONTENT, the Spanish BioGridNet and currently he is working in the Spanish MEDIANET project related with video streaming distribution over peer-to-peer networks in the future Internet.

RUI L. AGUIAR (ruilaa@ua.pt) received a Ph.D. degree in electrical engineering in 2001 from the University of Aveiro. He is currently an Associate Professor with “Agregação” at the University of Aveiro and an adjunct professor at the INI, Carnegie Mellon University. He is leading a research team at the Institute of Telecommunications, Aveiro, on next-generation network architectures and protocols. His current research interests are centered on the implementation of advanced wireless networks, systems, and circuits, with special emphasis on QoS and mobility aspects. He has more than 300 published papers in those areas. He has served as technical and general chair of several conferences, such as ICNS ’05, ICT ’06, and ISCC ’07, and is associate editor of several journals. He is a Senior member of IEEE and a member of ACM.

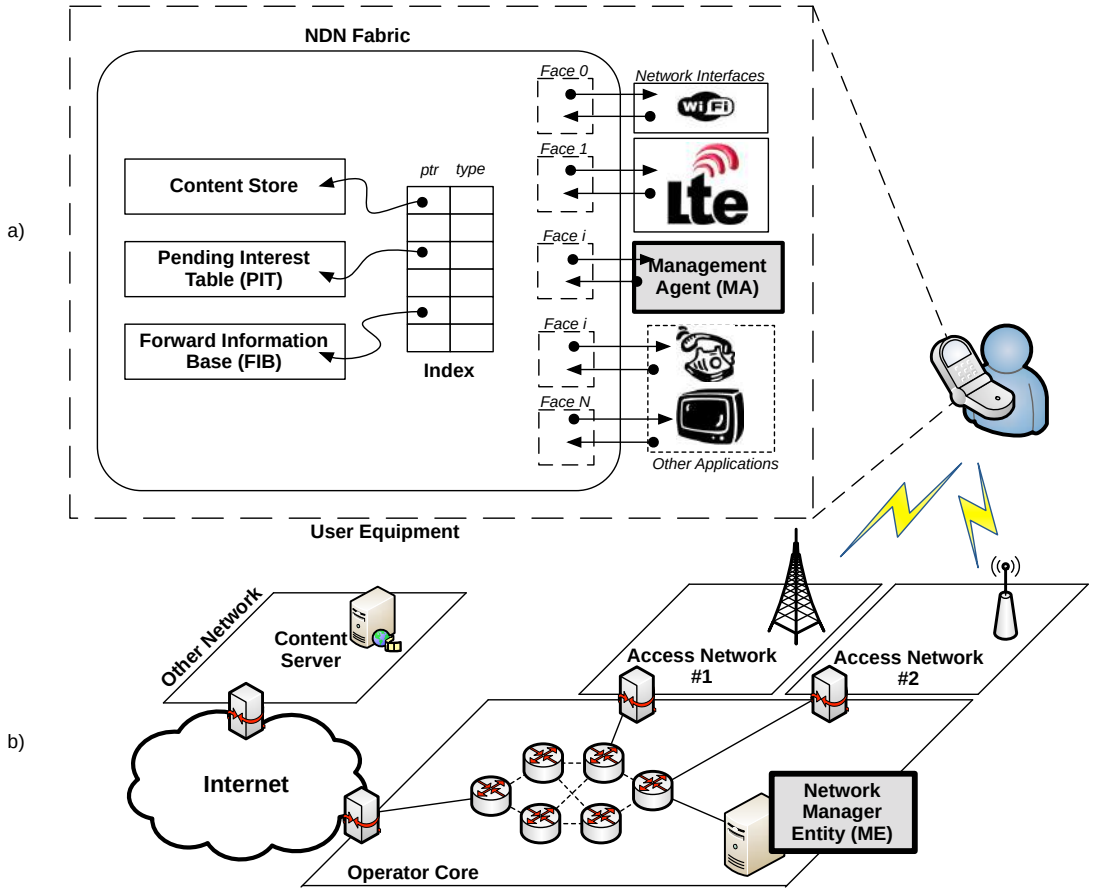

Figure 1 – The NDN Flexible Management Framework: a) Manager Agent interfacing with the NDN fabric; b) Deployment of the management framework in an operator network featuring a Manager Entity

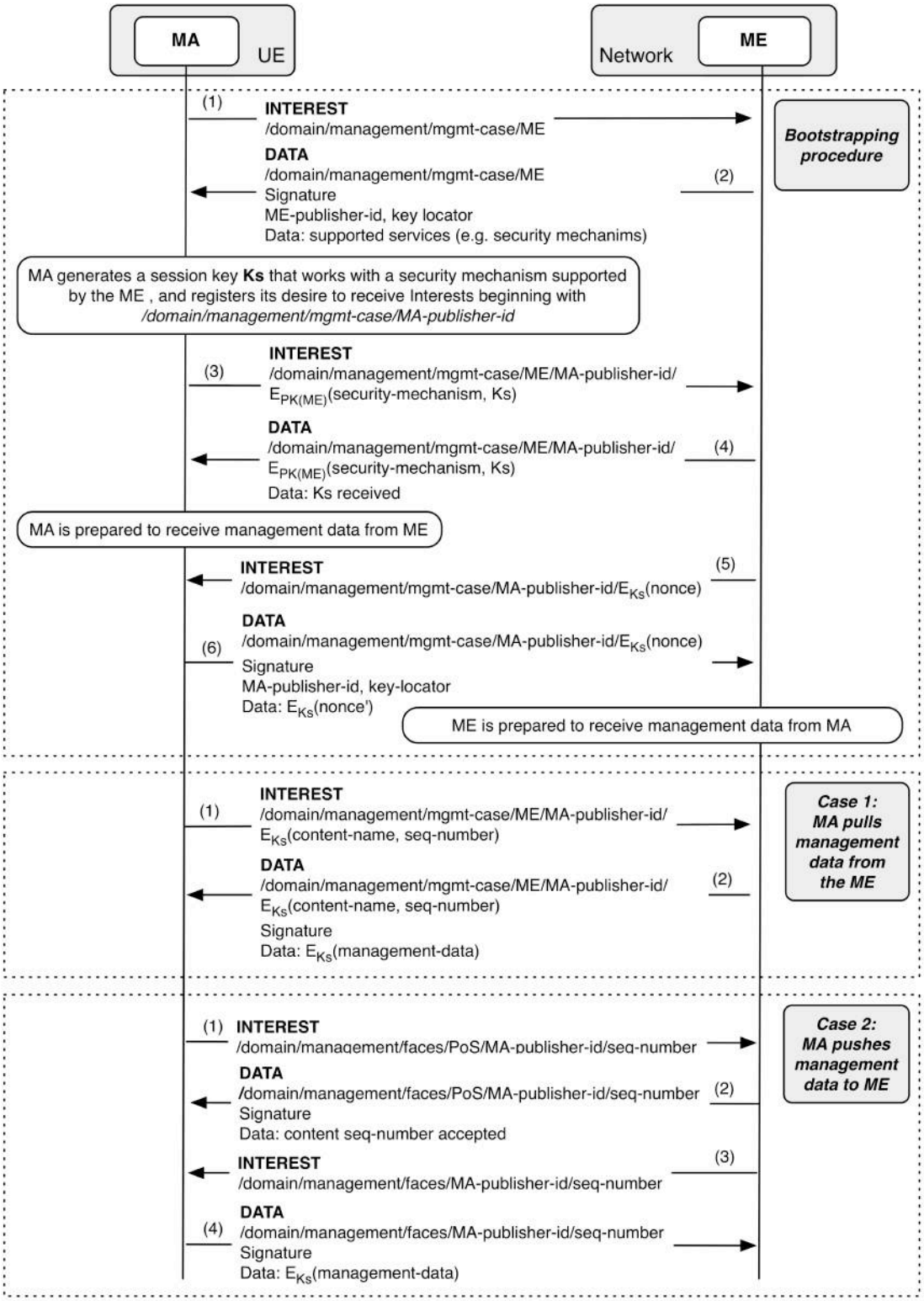

Figure 2: Bootstrapping and management data exchange

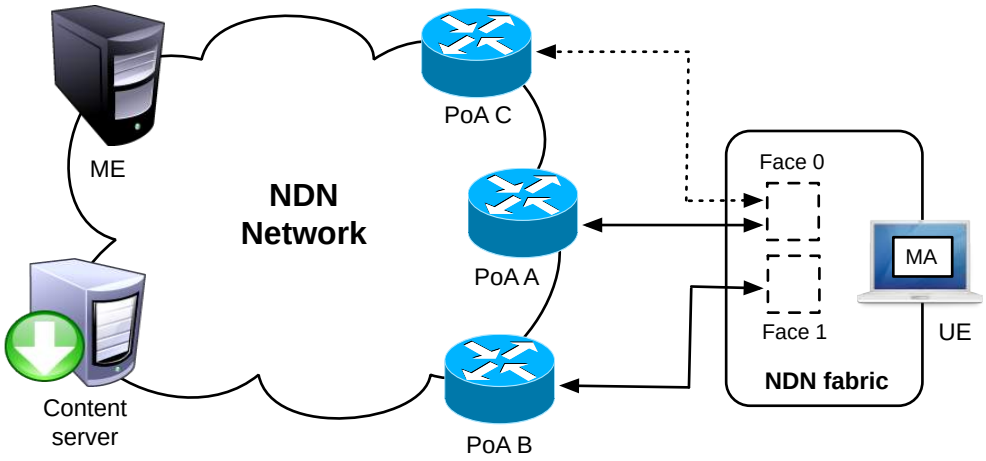

Figure 3: Validation scenario

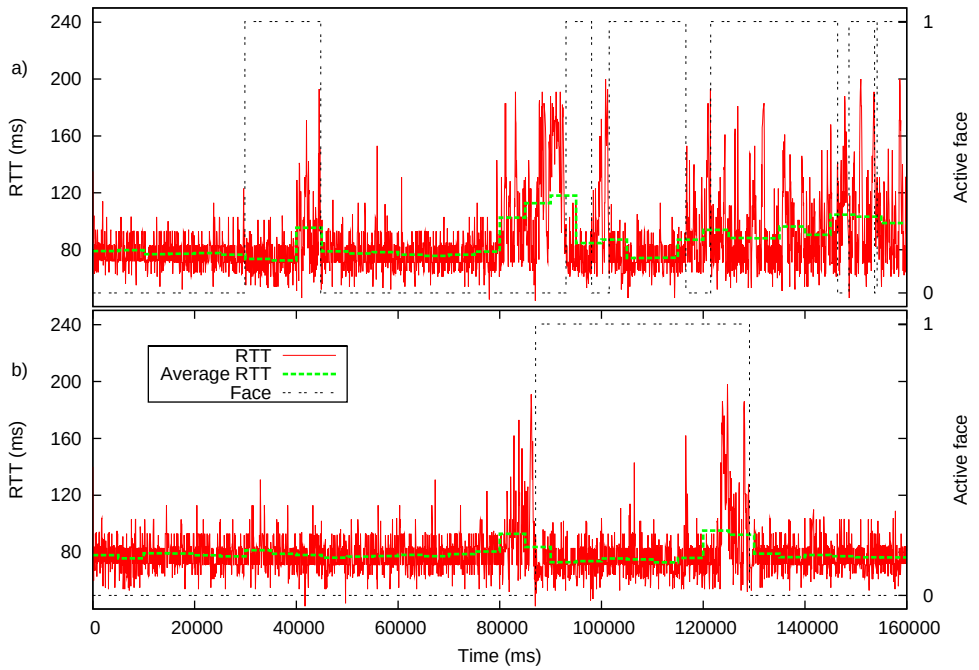

Figure 4: Results of the evaluation: a) RTT in basic NDN with probing, 1 probe per face; b) RTT in NDN under a framework-managed mode

Period	Load in the Point of Access		
	PoA_A	**PoA_B**	**PoA_C**
First (1-40)	Low	Low	Low
Second (41-80)	Low	Medium	Low
Third (81-120)	Medium	Low	Low
Fourth (121-160)	High	Medium	Low

Table 1. Load of the PoAs for the different time periods of the tests

	Basic NDN with 1 probe per face	Basic NDN with 5 probes per face	Framework-managed NDN
Average RTT (ms)	86.6140	84.685	78.9840
CI (ms)	(84.6350, 88. 5930)	(83.8573, 85.5127)	(78.5232, 79.4448)
Overhead (%)	1.2908	5.7677	0.1121
Losses (%)	1.8102	1.5478	0.3728
Handovers/s	0.0594	0.06	0.0125

Table 2. Summary of experimental results.