Content Retrieval while Moving Across IP and NDN Network Architectures

Carlos Guimarães, José Quevedo, Rui Ferreira, Daniel Corujo, Rui L. Aguiar
Instituto de Telecomunicações
Universidade de Aveiro
Aveiro, Portugal
Email: {cguimaraes, quevedo, rferreira, dcorujo}@av.it.pt, ruilaa@ua.pt

***Abstract*—Research on Future Internet has gained traction in recent years, with a variety of clean-slate network architectures being proposed. The realization of such proposals may lead to a period of coexistence with the current Internet, creating a heterogeneous Future Internet. In such a vision, mobile nodes (MNs) can move across access networks supporting different network architectures, while being able to maintain the access to content during this movement.**

In order to support such scenarios, this paper proposes an inter-network architecture mobility framework that allows MNs to move across different network architectures without losing access to the contents being accessed. The usage of the proposed framework is exemplified and evaluated in a mobility scenario targeting IP and NDN network architectures in a content retrieval use case. The obtained results validate the proposed framework while highlighting the impact on the overall communication between the MN and content source.

I. Introduction

To provide better solutions to the challenges faced by the current Internet architecture (e.g., scalable content distribution and security), clean-slate network architectures started to be considered as an alternative to the current incremental evolution of the Internet [1]. Information-Centric Networking (ICN) [2] is one of the proposed paradigms for the Future Internet architecture, which aims to improve e.g. network efficiency, content dissemination and security, by shifting from a host-centric to a data-centric paradigm.

The realization of ICN-based architectures will eventually lead to a period where new network architectures might coexist in parallel with the current Internet architecture [3]. The edge of the network is a possible location for the initial roll out of native deployments of ICN-based architectures [4], by means of isolated network architectural islands interconnected with one another either through dedicated links or as an overlay over IP. Such approach creates a heterogeneous networking landscape at the access networks, paving the way for a new set of possibilities where a mobile node (MN) not only moves between access networks supporting the same network architecture, but also from IP-based to ICN-based access networks (and vice-versa). In this respect, we argue that mobility management mechanisms across different network architectures must enable the MN to preserve reachability to contents, whenever the MN changes the network architecture it is attached to.

This is where this work contributes by introducing an inter-network architecture mobility framework that enables a MN to move between access networks supporting different network architectures while being able to maintain the access to content. For that, it introduces a new network entity that acts as an anchor point and as a gateway in the communication between the MN and content source, when they are on different network architectures. The usage of the proposed framework is exemplified in two content retrieval use case scenarios featuring the mobility of a MN between IP and Named Data Networking (NDN) [5] (i.e., an ICN instantiation) architectures and further evaluated in a physical deployment. Results validated the proposed framework, while highlighting its impact on the overall communication.

The remainder of the paper is structured as follows. The related work is presented in Section II. Section III details the proposed framework, which is evaluated in Section IV. Finally, in Section V, the main conclusions are presented.

II. Related Work

To promote the deployment of ICN architectures and their coexistence with the existing networking environment, interoperability mechanisms between the IP and ICN architectures have been targeted by the research community.

One of the first works addressing the interoperability between HTTP and Content-Centric Networking (CCN) proposes a gateway that converts HTTP requests and responses into CCN Interests and Data respectively [6]. In [7], the authors propose a HTTP/NDN gateway to interconnect ICN islands to the IP by mapping HTTP protocol with NDN messages. In [8], a TCP/ICN proxy is proposed, allowing TCP traffic between TCP/IP endpoints to be carried over an ICN network. The work presented in [9] proposes an intermediary entity that allows NDN and MQTT protocols to coexist in IoT scenarios, by converting messages between protocols. The H2020 POINT project [10] targets the deployment of IP-based applications over ICN-based architectures. It proposes a gateway-based approach that allows the IP interfaces towards the user to be preserved. Additionally, it supports handlers for specific IP-based protocols such as HTTP and CoAP [11]. In [12], the Versatile ICN Deployment Framework (VICN) is proposed with the purpose of facilitating the deployment of different ICN instantiations, while enabling the interoperation

among them. Finally, Hybrid ICN solution [13] embedded ICN semantics into IPv6 packets, integrating ICN names in existing IPv6 headers and other ICN information carried as payload inside IPv6 packets.

Nonetheless, to the best of our knowledge, none of these approaches considers the mobility of a MN across different network architectures.

III. Proposed framework

The proposed framework (shown in Figure 1) introduces a new entity in the network, named Future Internet eXchange Anchor (FIXA). This new entity enables a MN to continue reaching a given content after moving to an access network of another network architecture, different from the one where the content is deployed. For that, the FIXA doubles as both anchor point and gateway for communications across different network architectures (i.e., the communication endpoints are on different network architectures). The proposed mobility mechanism complements the local mobility management mechanisms that exists on each network architecture which aims to provide reachability/connectivity inside the same network architecture (i.e., while the MN moves across access networks of the same network architecture).

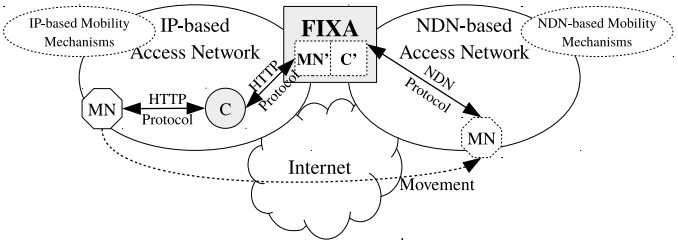


Fig. 1: Framework Overview

If the MN and the accessed content are deployed in the same network architecture, both endpoints communicate directly, not requiring the FIXA to intermediate the communication, as depicted in the example of Figure 1 where the *MN* communicates directly with the content source *C*.

When the MN moves to an access network of a different network architecture, the content source may become unreachable through the protocols previously used by the MN, because the access network does not support them. In the example of Figure 1, the *MN* cannot use the HTTP protocol since it is not natively supported in the NDN-based access network. As such, the *MN* needs to be able to reestablish the communication using NDN protocol (i.e., a protocol supported by its current access network). Still, the content source (and its access network) may not support the protocol used by the *MN*, as it is the case exemplified in Figure 1. An entity capable of communicating via both protocols is therefore required, which could adapt the protocol signaling across the different network architectures. This is the main reason for the introduction of the FIXA in the network, as an intermediary entity in the communication between endpoints deployed on different network architectures. In doing so, the content source *C*, deployed in IP, remains reachable by the *MN*, when it moves to NDN, through the FIXA, which intermediates the communication between the *MN* and the content source *C*.

A. Future Internet eXchange Anchor (FIXA)

The FIXA is the core entity of the proposed framework, responsible for providing an inter-network architecture mobility management service. It enables the MN to continue reaching content deployed in a given network architecture after moving to a different one. To achieve its purpose, the FIXA combines the capabilities of an anchor point for communication between different network architectures and of a gateway that converts messages between protocols supported by each network architecture.

Upon request by the MN, the FIXA generates an on-demand mapping that defines how the content can be accessed by the requester MN from a different network architecture than the one where the content is deployed (Figure 2). A mapping is limited to the scope of the particular MN it was generated for and it needs to comply with two main properties: (i) uniquely identify a content in the network architecture where it is deployed; and (ii) enable messages to be forwarded towards the FIXA. A mapping URI can be divided into three parts, as depicted in Figure 2: (i) the protocol to use, which is identified in the scheme part of the URI; (ii) a prefix, which is used by the MN to identify the FIXA, allowing the MN to forward messages addressing the mapping URI towards a FIXA; and (iii) a part used to map and identify the content being addressed by the MN, which is used by the FIXA to lookup for the original URI of the content on the network architecture where it is deployed.

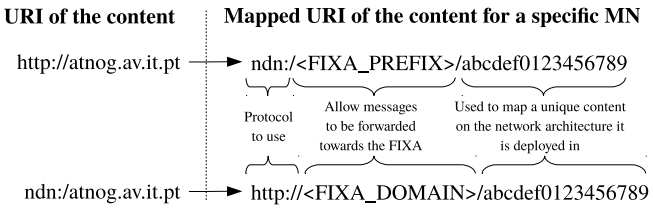


Fig. 2: Mappings examples between HTTP (over IP) and NDN

When the MN addresses the generated mapping, the FIXA anchors the communication with messages being sent towards it. Meanwhile, the FIXA intermediates the communication between the MN and the content source (i.e., acts as a gateway), being established two connections: (i) between the MN and the FIXA; and (ii) between the FIXA and the content source. In doing so, the FIXA simultaneously behaves as the destination and source of messages, converting messages between each connection. Requests received by the FIXA are converted and sent to the correspondent content source. Similarly, responses from the content source are received by the FIXA, which are then converted and sent to the MN.

FIXAs are deployed and distributed across network points that interconnect different network architectures, being addressed by well-known identifiers on each network architec-

ture. In IP, a set of IP address(es) and a hostname could be used to access the a FIXA. In NDN, a reserved (global) prefix could be used to access a FIXA. Additionally, FIXAs may be deployed as a virtual network function in a cloud/fog infrastructure, allowing it to better face the computational demands. Since the FIXA holds no state outside of the mappings it configures for MNs and the ongoing requests and responses, both vertical and horizontal scalability strategies are possible to be applied to face an eventual increase of traffic requiring conversion resulting from the mobility of MNs.

B. Mobile Node (MN)

The MN also needs to be enhanced to support the proposed inter-network architecture mobility mechanisms. It is composed by three layers responsible for the operations related with the applications, network stacks and networking hardware and by a cross-layer responsible for operations related with the inter-network architecture mobility procedures.

Figure 3 presents the enhanced internal architecture of a MN supporting both IP and NDN in a dual stack operation.

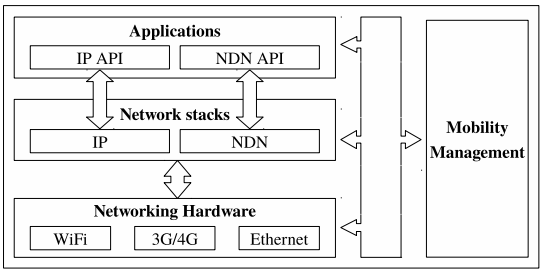


Fig. 3: Enhanced internal architecture of MN

*1) **Application Layer**:* In this work, we expect that future applications will be capable of communicating on each network architecture by implementing the logic to access content using the supported protocols. For example, [14] already proposes a web browser capable of fetching content through CCNx or HTTP protocols regarding, respectively, "ccnx:/" and "http:/" URI schemes. These novel applications can send and receive messages when connected to different network architectures, being able to fetch content independently of the network architecture the MN is connected to. Although the support of legacy applications is out of scope of this work, we envisioned that a subset of the FIXA features could be implemented as middleware in the MN itself to support inter-network architecture mobility to those applications.

*2) **Network Stacks Layer**:* To be able to communicate on each network architecture, the MN is enhanced with the support of multiple network stacks, namely IP and NDN. It can then receive, identify, process and send messages with respect to the protocols supported on each network architecture.

*3) **Networking Hardware Layer**:* For remote communications, the MN may encompass multiple network interface cards (NICs) of the same or different link technologies, either wired (e.g., Ethernet) or wireless (e.g., WiFi and 3G/4G).

*4) **Mobility Management Layer**:* This layer is responsible for managing the mobility procedures on the MN. With respect to the inter-network architecture mobility procedures, it acts as a cross layer component that interacts with the remaining layers as follows: (i) detect changes regarding the connections of the MN and the points of attachment (PoAs) (e.g., detection of link-layer attachment and deattachment events); (ii) discover the network architectures supported by the access networks that the MN is connected to (e.g., by means of information elements contained in management frames of each link technology or by the successful completion of bootstrap mechanisms related with each network architecture); (iii) notify applications about changes regarding the supported network architectures (e.g., through events between the mobility management layer and the application layer).

This layer is also responsible for handling the local mobility of the MN on each network architecture, using specific mobility management mechanisms existing therein.

C. Resource Binding Procedure

After the handover to a different network architecture, applications in the MN need to discover how to reach the content being accessed before the handover from the new network architecture with the purpose of reestablishing the retrieval of the content. For that purpose, the proposed framework defines a *Resource Binding Procedure* (Figure 4) that allows the MN to request the FIXA to provide access to the content using protocols supported in the new network architecture. The FIXA generates an on-demand mapping to the content (Figure 2) that is then delivered to the MN. This mapping allows the MN, intermediated by the FIXA, to access the content from the new network architecture.

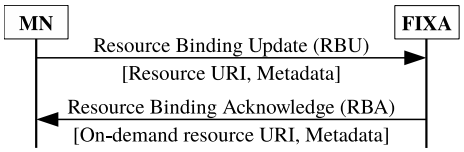


Fig. 4: *Resource Binding Procedure*

The *Resource Binding Procedure* is initiated by the MN by issuing a *Resource Binding Update* (RBU) message for each content being accessed before the handover. This message contains the original URI of the content (i.e., the known address of the content on the network architecture where it is deployed) as well as optional metadata (e.g. the preferred protocol to be used in the new network architecture).

Upon reception of the RBU message, the FIXA generates an on-demand mapping for the identified content that will enable the MN to reach it, through the FIXA, from the new network architecture. This mapping is delivered to the MN via the *Resource Binding Acknowledge* (RBA) message, along with metadata related to the protocol used to access the content.

IV. Evaluation

To verify the feasibility of the proposed framework in providing reachability to content after the handover of the

MN to a different network architecture, a proof-of-concept prototype of the proposed framework was implemented and physically deployed over an evaluation scenario targeting IP and NDN network architectures, as depicted in Figure 5. This evaluation focuses on the impact of the FIXA in the communication between the MN and the content source.

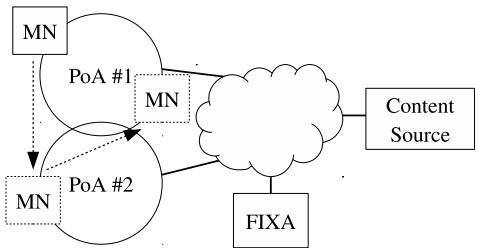


Fig. 5: Evaluation Scenario

This scenario is composed by two PoAs supporting IP or NDN architectures, a FIXA, a content source supporting only IP or NDN network architectures and a single interfaced MN supporting IP and NDN architectures.

NDN (NDN Platform v0.6.2[1]) is deployed as an overlay over the IP network. Notwithstanding, if the MN is connected to the PoA supporting NDN, the application only communicates via the NDN protocol.

The MN and the PoAs are physical machines configured with a AMD Embedded G series GX-412TC processor and with 4GB RAM running Ubuntu 14.04, while the content source and the FIXA are deployed in virtual machines with 8 core processor and 8GB RAM running Ubuntu 16.04.

The experiments were run 30 times, being presented the averaged results with a 95% confidence interval.

A. Use Case Signaling

Figure 6 presents the signaling of the evaluated use cases, for the handover of a given MN between IP and NDN. These use cases focus on retrieval of content given the relevance of web-like traffic, including file transfer using HTTP, in current usage patterns of the Internet [15]. Nevertheless, the proposed framework is capable of supporting different use case scenarios (e.g., live and on-demand video streaming) involving a different set of protocols.

1) ***From IP to NDN to IP*** *(Figure 6a):* The MN is initially connected to an IP-based access network (i.e., PoA #1) and, since the content source is deployed over the same network architecture (i.e., IP), the MN starts downloading the content from the content source via HTTP.

After moving to the PoA #2, which only supports the NDN network architecture, the MN cannot directly download the content from the content source because they are on different network architectures. The MN initiates the *Resource Binding Procedure* with the purpose of discovering how to reach the content from the NDN network architecture. The discovered mapping allows the MN to reestablish the communication with the content source, intermediated by the FIXA which converts messages from one protocol into the other (i.e., *NDN Interests* into *HTTP GET* messages and *HTTP response* messages into *NDN Data* messages). To continue the download, a *NDN Interest* message addressing the next byte segment offset is issued, which is calculated with respect to the already received content. The byte segment offset is used by the FIXA to include the *Range* header in the *HTTP GET* message so that only the specific part of the desired content is downloaded, avoiding the need to download the entire content from the content source. After receiving the *HTTP 206 Partial Content* message, the FIXA extracts the content from the received message, which is then used to generate the corresponding *NDN Data* to be sent to the MN. If an HTTP error message is received by the FIXA, it returns to the MN an *NDN Data* message containing a negative acknowledgement (NACK).

Upon moving back to the PoA #1, the MN requests the remaining content directly from the content source. In this case, since part of the content has already been received, the MN includes the *Range* header in the *HTTP GET* message in order to request only the missing content.

2) ***From NDN to IP to NDN*** *(Figure 6b):* The MN is initially connected to PoA #1 which supports the NDN network architecture and, since the content source is also deployed over the NDN network architecture, the MN requests the content from the content source via the NDN protocol.

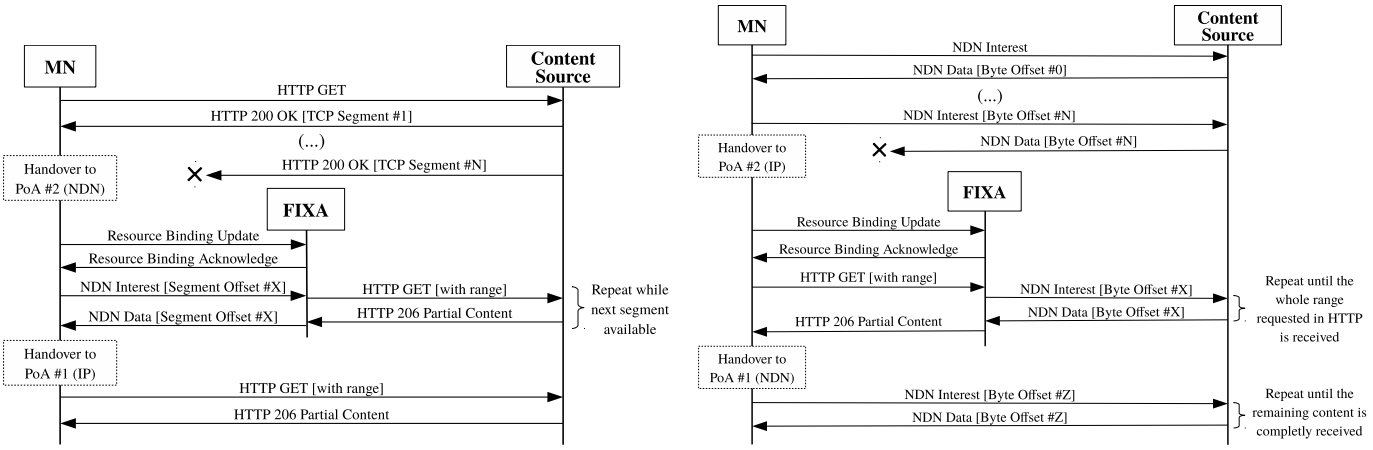
After moving to an access network supporting only IP (i.e., PoA #2), the MN is not able to directly get the content from the content source, due to being on different network architectures. As such, it triggers the *Resource Binding Procedure* to discover how to reach the content from the IP network architecture. The MN can then request the remaining content through the FIXA, which converts messages between network architectures. More specifically, it converts *HTTP GET* messages into *NDN Interests* and *NDN Data* messages into *HTTP response* messages. Based on the *Range* header of the *HTTP GET* messages, the FIXA issues the *NDN Interest* message for the corresponding byte segment offset. Whenever the requested content is received by the FIXA, it generates a *HTTP 206 Partial Content* message to be sent to the MN. If a *NDN Data* message containing a NACK is received by the FIXA, an HTTP response message with an error code is sent.

When the MN handovers back to PoA #1, it requests the remaining content via the NDN network architecture, issuing *NDN Interest* messages for the next byte offset segment.

B. Validation

To validate the mobility mechanism of the proposed framework, both use cases depicted above were implemented and deployed over the evaluation scenario. While downloading the content, the MN had to move between both IP and NDN network architectures and, consequently, the content was downloaded via both HTTP (over IP) and NDN protocols. After finishing the download, the SHA256 hash of the downloaded content was generated and compared with the SHA256 hash of the correspondent content in content source, verifying

[1]NDN Platform - http://named-data.net



(a) From IP to NDN to IP

(b) From NDN to IP to NDN

Fig. 6: Signaling of handover between IP and NDN use cases

that both hashes match, which indicates that the content was correctly downloaded by the application in the MN.

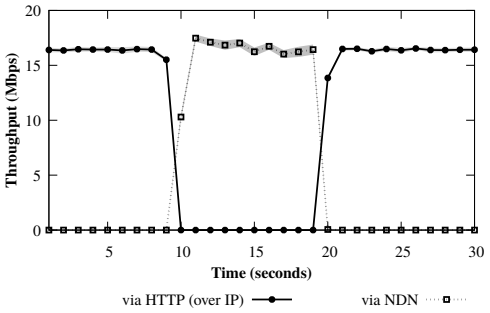
C. Performance Evaluation

In our evaluation, two parameters affected the throughput in NDN: (i) the Interest window size; and (ii) the round trip time (RTT) between the MN and the content source. For example, by using an Interest window of size 1, a new *NDN Interest* message is only issued after the previous *NDN Interest* message is satisfied (i.e., after receiving the corresponding *NDN Data* message). Consequently, the time between *NDN Interest* messages is affected by the RTT between the MN and the content source. For this experiment, NDN consumer was configured to use a Interest window of size 5 to allow simultaneous requests of different parts of the content and the *SignatureSha256WithEcdsa* algorithm was used to reduce the delay in signing the *NDN Data* messages and, consequently, the RTT between the MN and the content source.

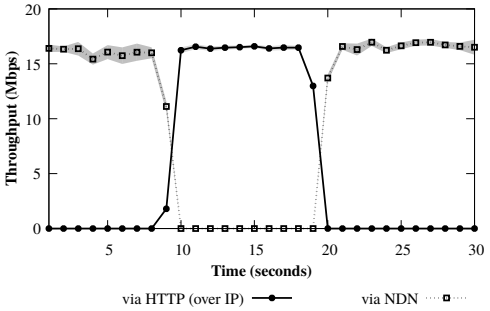
Figure 7 presents the throughput measured in the MN regarding the download of a given content while switching across IP and NDN network architectures. In both use case scenarios, after the MN moves to a different network architecture, the application in the MN was able to resume the download by using the protocols supported on each network architecture. When the MN and the content source are on different network architectures (meaning that the communication needs to be intermediated by the FIXA), a similar performance in terms of throughput was achieved when compared with the case where the MN and the content source are on the same network architecture and no intervention of the FIXA is required.

D. Resource Binding Procedure

After the handover procedure, the MN discovers the mapping to continue reaching the content from its current network architecture through the *Resource Binding Procedure*. This procedure took about 2.83 $\pm$ 0.36 ms when requested over NDN and 1.10 $\pm$ 0.19 ms when requested over HTTP over



(a) IP-NDN-IP use case



(b) NDN-IP-NDN use case

Fig. 7: Throughput at the MN

the IP. Among the reasons for this procedure being more time consuming in NDN are the additional time required to register each generated mapping with the connected NDN forwarder and the need to sign the *NDN Data* message. Still, this delay is only introduced once in the communication and it impacts the time to restore the download of the content after moving to a different network architecture.

If prior knowledge about the supported network architectures by each PoA is acquired before the handover procedure

itself, a preemptive discovery of the mappings could be performed, allowing applications to immediately reestablish the access to contents after the handover procedure.

E. FIXA delay

Figure 8 depicts the delay introduced by the FIXA while intermediating the communication between the MN and the content source.

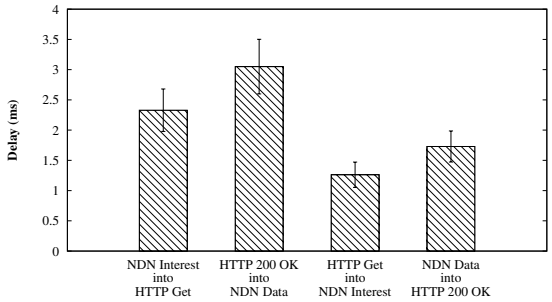


Fig. 8: Delay introduced by the FIXA

This delay is impacted not only by the operation of the FIXA of acting as a source and destination of messages on each network architecture, but also by the time required to convert messages from one protocol into another. In the evaluated use case scenarios and depending on the message conversion, the FIXA introduced a delay between 1.2 ms and 3.1 ms, increasing the RTT in the communication between the MN and the content source. This delay may have a significant impact while dealing with delay-sensitive applications which requires content to be retrieved with very low latency requirements, in contrast to other type of applications where the delay is not critical. Notwithstanding, the introduced delay in the communication is one of the tradeoffs of the proposed framework to enable content to continue being reached by the MN when it moves to a different network architecture, which otherwise was not possible.

V. Conclusion

This paper proposes an inter-network architecture mobility framework that enables a MN to move across IP and NDN network architectures, while maintaining the access to contents. For that, it introduces a new network entity, named Future Internet eXchange Anchor (FIXA), which acts as an anchor point and a gateway, as well as it proposes a procedure for the MN to discover how to address the content after the handover. The proposed framework is exemplified and evaluated in content retrieval use cases while moving across NDN-based and IP-based access networks, with results validating its mobility mechanisms and highlighting its impact on the communication between the MN and the content source.

As future work, we expect to expand the set of studied use case scenarios, as ICNs matures by defining upper-layer protocols or intrinsic mechanisms to support different applications commonly taken for granted in the current Internet.

Acknowledgment

This work is supported by the European Regional Development Fund (FEDER), through the Regional Operational Programme of Lisbon (POR LISBOA 2020) and the Competitiveness and Internationalization Operational Programme (COMPETE 2020) of the Portugal 2020 framework [Project 5G with Nr. 024539 (POCI-01-0247-FEDER-024539)].

References

[1] A. Feldmann, "Internet Clean-slate Design: What and Why?" *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 3, pp. 59–64, Jul. 2007.

[2] G. Xylomenos, C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, and G. C. Polyzos, "A Survey of Information-Centric Networking Research," *IEEE Communications Surveys Tutorials*, vol. 16, no. 2, pp. 1024–1049, Second 2014.

[3] D. Fisher, "A Look Behind the Future Internet Architectures Efforts," *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 3, pp. 45–49, Jul. 2014.

[4] A. Rahman, D. Trossen, D. Kutscher, and R. Ravindran, "Deployment Considerations for Information-Centric Networking (ICN)," Internet Engineering Task Force, Internet-Draft draft-irtf-icnrg-deployment-guidelines-03, Jun. 2018, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-irtf-icnrg-deployment-guidelines-03

[5] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, k. claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang, "Named data networking," *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 3, pp. 66–73, Jul. 2014.

[6] S. Wang, J. Bi, J. Wu, X. Yang, and L. Fan, "On Adapting HTTP Protocol to Content Centric Networking," in *Proceedings of the 7th International Conference on Future Internet Technologies*, ser. CFI '12. New York, NY, USA: ACM, 2012, pp. 1–6.

[7] X. Marchal, M. E. Aoun, B. Mathieu, T. Cholez, G. Doyen, W. Mallouli, and O. Festor, "Leveraging nfv for the deployment of ndn: Application to http traffic transport," in *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, April 2018, pp. 1–5.

[8] I. Moiseenko and D. Oran, "TCP/ICN: Carrying TCP over Content Centric and Named Data Networks," in *Proceedings of the 3rd ACM Conference on Information-Centric Networking*, ser. ACM-ICN '16. New York, NY, USA: ACM, 2016, pp. 112–121.

[9] J. Quevedo, R. Ferreira, C. Guimarães, R. L. Aguiar, and D. Corujo, "Internet of things discovery in interoperable information centric and ip networks," *Internet Technology Letters*, vol. 1, no. 1, 2018.

[10] D. Trossen, M. J. Reed, J. Riihijrvi, M. Georgiades, N. Fotiou, and G. Xylomenos, "IP over ICN - The better IP?" in *Networks and Communications (EuCNC), 2015 European Conference on*, June 2015, pp. 413–417.

[11] N. Fotiou, H. Islam, D. Lagutin, T. Hakala, and G. C. Polyzos, "CoAP over ICN," in *2016 8th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, Nov 2016, pp. 1–4.

[12] J. Ren, K. Lu, S. Wang, X. Wang, S. Xu, L. Li, and S. Liu, "VICN: a versatile deployment framework for information-centric networks," *IEEE Network*, vol. 28, no. 3, pp. 26–34, May 2014.

[13] L. Muscariello, G. Carofiglio, J. Auge, and M. Papalini, "Hybrid Information-Centric Networking," Internet Engineering Task Force, Internet-Draft draft-muscariello-intarea-hicn-00, Jun. 2018, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-muscariello-intarea-hicn-00

[14] X. Qiao, G. Nan, Y. Peng, L. Guo, J. Chen, Y. Sun, and J. Chen, "NDNBrowser: An extended web browser for named data networking," *Journal of Network and Computer Applications*, vol. 50, pp. 134 – 147, 2015.

[15] Cisco. (2017, Sep.) Cisco Visual Networking Index: Forecast and Methodology, 2016-2021. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html