An Orchestrated NDN Virtual Infrastructure Transporting Web Traffic: Design, Implementation, and First Experiments with Real End Users

Guillaume Doyen, Thibault Cholez, Wissam Mallouli, Bertrand Mathieu, Hoang-Long Mai, Xavier Marchal, Daishi Kondo,
Messaoud Aouadj, Alain Ploix, Edgardo Montes-de-Oca, and Olivier Festor

After more than one decade of research efforts, ICN technologies seem mature enough to move from a design and implementation phase to early deployment trials. However, if one wants an ICN stack to be operated in a production context, some major locks, related to the lack of adequate deployment infrastructure, the migration of relevant services, and the capability to accurately monitor an overall ICN domain, must be addressed.

ABSTRACT

After more than one decade of research efforts, ICN technologies seem mature enough to move from a design and implementation phase to early deployment trials. However, if one wants an ICN stack to be operated in a production context, some major blocks, related to the lack of adequate deployment infrastructure, the migration of relevant services, and the capability to accurately monitor an overall ICN domain, must be addressed. In this article we present a feedback experience of the deployment of a Named Data Networking island. The latter considers HTTP over NDN as a primary service deployed through dedicated gateways and NFV as a substrate for deploying and orchestrating ICN components. From the performance assessment of individual components up to the opening of the testbed to end users in a university campus, we propose an analysis that can guide further research efforts in the ICN area.

INTRODUCTION

After more than one decade of research and development, the information-centric networking (ICN) [1] paradigm has now reached a level of maturity that makes it a promising candidate to replace or complete standard IP stacks. Among the numerous insights that attest to this maturity, one can observe that:

1. Several ICN architectures proposed to date have stable and fully operational implementations (e.g., CCNX, NFD, Pursuit's Blackadder, Netinf).
2. Some worldwide testbeds allow the deployment of real services for research and experimentation purposes (e.g., the Named Data Network, NDN, testbed[1] or the ICN testbed federation).[2]
3. The ICN Research Group (ICNRG), a dedicated working group of the Internet Research Task Force (IRTF), actively works on the standardization of several key points of this paradigm such as deployment considerations [2].

Consequently, ICN is now moving from a pure research stage toward early trials of deployments, which exhibit novel challenges this paradigm has to face. The first one is related to deployment consideration, requiring novel infrastructure means able to host all ICN network functions. The second relies on the identification and migration of relevant services that can benefit from deployment over an ICN. Finally, novel monitoring and security facilities have to be designed to enable stakeholders to deploy ICN domains in a safe and manageable way.

Focusing on the NDN solution [3], the most federated ICN architecture to date, and after having summarized our contributions about security in [4], we present in this article our achievements in:

1. Service migration with the presentation of a HTTP/NDN gateway able to carry web traffic [5]
2. Infrastructure means that leverages network functions virtualization (NFV) to propose content-oriented orchestration [6]

Original contributions depicted in this article especially focus on NDN caching of web content and first results of experiments performed with real users.

The article is organized as follows. We present the related work on ICN deployment and service migration. We present two major contributions, which are an HTTP/NDN gateway enabling web traffic to cross NDN islands and content-oriented management and orchestration (MANO), which enables the automated deployment and dynamic enforcement of NDN policies. We depict the different evaluation results from unitary tests of the components and a measurement campaign involving real end users. Finally, we draw some lessons from this first deployment experience that can help guide any further contributions in this area.

RELATED WORK

SDN AND NFV FOR ICN DEPLOYMENT

It is commonly admitted that NDN will not replace

This work is partially co-funded by (1) the French National Research Agency (ANR), DOCTOR project, < ANR-14-CE28-0001>, started in 01/12/2014 and supported by the French Systematic cluster, and (2) the CRCA Excellence grant < A2101-03>, and the CRCA-FEDER CyberSec Platform <201304601>.

[1] https://named-data.net/ndn-testbed/, accessed on 04/02/19

[2] http://www.icn2020.org/2018/01/15/testbed-federation/, accessed on 04/02/19

Digital Object Identifier:
10.1109/MCOM.2019.1800730

Guillaume Doyen and Alain Ploix are with Troyes University of Technology; Hoang-Long Mai, Wissam Mallouli, and Edgardo Montes-De-Oca are with Montimage Research Labs; Thibault Cholez, Xavier Marchal, Daishi Kondo, and Olivier Festor are with LORIA-CNRS; Bertrand Mathieu is with Orange Labs.

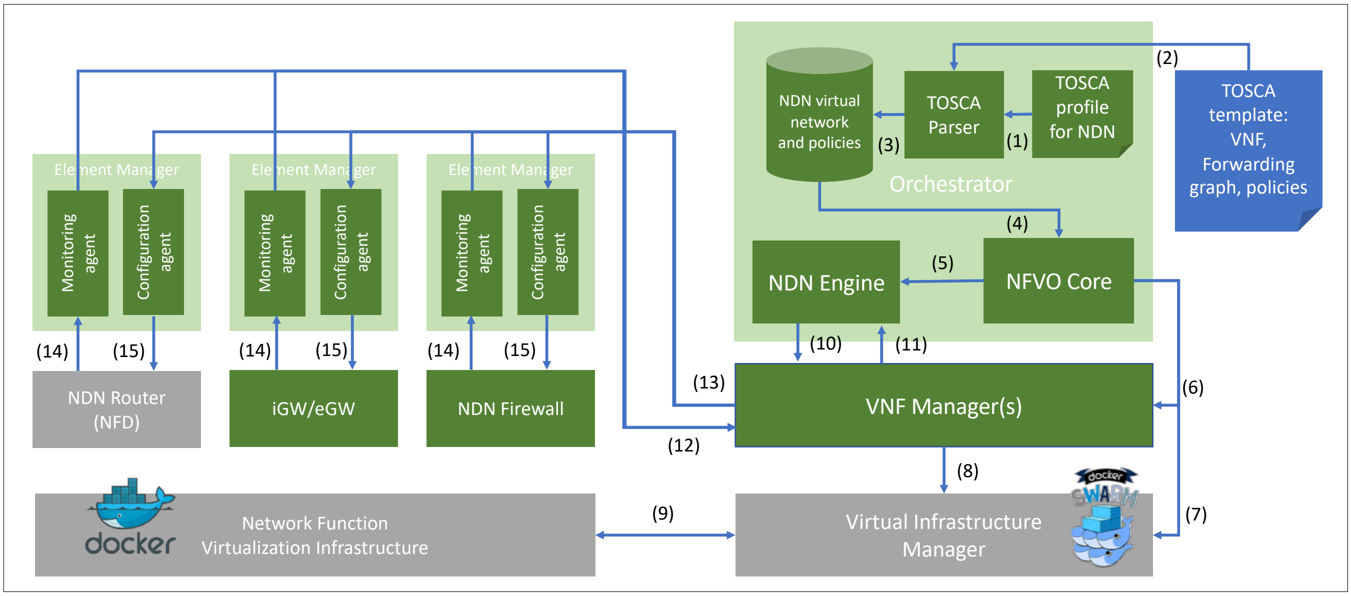

Figure 1. Virtual NDN Network and MANO Architecture. Green boxes stand for novel NFV components and grey ones stand for existing ones. The TOSCA virtual NDN network specification is depicted as a blue box.

IP in a one-shot phase, but that the deployment will rather be progressive, and software defined networking (SDN) and NFV are two key technologies that allow it.

Many research efforts argue for the cohabitation of IP and ICN on a common layer 2 network by leveraging the network programmability offered by SDN. In [7], the authors propose to use a dedicated UDP or TCP port to identify ICN protocol and to extend the SDN controller with an ICN module. Salsano *et al.* [8] propose a framework for deploying ICN functionalities over SDN using the IP option header as a name field for ICN. Meanwhile, the authors of [9] propose and implement an ICN module in the SDN controller to process the forwarding path computation for NDN flows separate from IP flows. Nguyen *et al.* [10] implement an intermediate layer between a CCN node and an OpenFlow switch called *Wrapper*, and the combination of the three elements acts as an ICN router.

NFV instead argues for the separation of IP and ICN protocols by leveraging the isolation property of virtualization. It also enables the deployment of ICN without requiring any change in the current network infrastructure, thus acting as a key enabler. Sardara *et al.* [11] follow this direction by proposing vICN (virtualized ICN). The authors provide a flexible unified framework for ICN, which includes several functions such as monitoring. The H2020 FLAME project aims to fully integrate ICN into an overall media function platform using the concepts of service function chaining (SFC). The fifth generation (5G) system architecture defined by the Third Generation Partnership Project (3GPP) leverages NFV and SDN technologies to provide the flexibility to deploy ICN as a slice. Finally, in [12], the authors benefit from NFV to provide contextualized edge services relying on ICN protocol stacks.

To date, several solutions aimed at providing practical deployment solutions of ICN with SDN and NFV have been proposed. However, to the best of our knowledge, our architecture is the first approach that pushes the content-oriented paradigm of ICN up to high-level orchestration templates while keeping interoperability with existing standards for network virtualization such as the European Telecommunications Standards Institute (ETSI) MANO reference architecture, TOSCA as a high-level specification language, and Docker as a virtualization substrate.

SERVICE MIGRATION

The prime protocol to deliver content nowadays is HTTP, and it is natural that a few initiatives already tried to make the bridge between HTTP and the NDN world. A first approach was proposed by Wang *et al.* [13] that translates HTTP requests to CCN *Interests* packets, but it exhibits weaknesses such as the unability to communicate with a native CCN producer/consumer and the misuse of the "metadata" field in the CCN *Interest* packet to carry a HTTP request's header. The authors of [14] explain how CDN could benefit from ICN and identify CCN/HTTP translation and CCN/IP tunneling as key technologies of their architecture, but they only provide a high-level description of the gateway. Another more generic approach, also relying on a gateway, was proposed by Moiseenko *et al.* [15]. They succeeded in carrying TCP over NDN, and consequently all the upper protocols that use TCP, which include HTTP. While their work is a significant step toward the adoption and deployment of NDN, their generic solution also misses an important incentive because it does not use one of the main features of NDN, the cache, when carrying web contents because it is not possible to have efficient content-level caching when only TCP-level information is considered.

To conclude, to the best of our knowledge, our work is the only one to address the most relevant service migration case while preserving all the features of both the HTTP protocol and the NDN one. It is also the only one to have been extensively evaluated in a realistic deployment situation with end users accessing real web sites.

To overcome the current limitations of service migrations toward NDN, we designed an HTTP/NDN mapping protocol and architecture, whose implementation takes the form of a gateway that can be used to seamlessly transport HTTP traffic over an NDN island.

A VIRTUALIZED NDN INFRASTRUCTURE FOR WEB TRAFFIC

LEVERAGING NFV AS A SUBSTRATE FOR NDN DEPLOYMENT

The overall architecture of our content-oriented MANO is illustrated in Fig. 1. It strictly follows the ETSI reference architecture specification.[3] It leverages Docker as the core technology for the NFV infrastructure (NFVI) and VXLAN as an encapsulation strategy for the NDN data plane traffic, thus making the NFVI agnostic to the carried traffic nature. Consequently, the virtual infrastructure manager (VIM), does not need to be extended to support NDN traffic too, and we have selected Docker Swarm as a ready-to-use technology (arrow 9). As such, the methodology we followed has consisted in solely extending or redesigning the MANO components that need NDN awareness without diverting them from their initial purpose. These are the virtual network function manager (VNFM) and the NFV orchestrator (NFVO).

To provide network administrators with a high-level and intuitive way to specify virtualized NDN services, we have designed a novel TOSCA profile for NDN that also integrates novel policies to dynamically react against NDN-specific performance and security incidents. The specifications for *virtual deployment units*, *virtual links*, and *connection points* have been kept unchanged since they only relate to the infrastructure layer, while those for VNF, *forwarding path*, and *policies* have been extended to encompass NDN features. For instance, the VNF specification includes configuration parameters that represent the set of NDN prefixes to be announced as well as the status of a signature verification module in an NDN router, while the *forwarding path* specification captures the list of VNFs that a particular set of NDN packets will follow defined on either the forwarder or the NDN prefix. Finally, our NDN TOSCA extension enables the specification policies modeled with Event-Condition-Action (ECA) rules that apply dynamically during service runtime for dynamic reconfiguration operations (e.g., enforcement of signature verification to be applied on *Data* packets) and scale-out.

We have designed and implemented a dedicated NFVO that includes two main blocks: a TOSCA parser and an Orchestration engine. Given our TOSCA profile for NDN (arrow 1), the TOSCA parser reads TOSCA templates (arrow 2) and creates an in-memory graph of TOSCA nodes and their relationships (arrow 3). The graph is then passed to the NFVO Core component (arrow 4), which delegates any NDN-specific operation to the NDN engine (arrow 5), which extracts the forwarding information defined in the forwarding path and translates it into forwarding information base (FIB) configuration for each VNF, which includes the NDN prefix, the address of the next hop, and the port number of the NDN process (e.g., NFD, NDN firewall, or signature verification module). In the case of a deployment automation, the NFVO core orders the VIM to deploy a management network (arrow 7) to ensure the communication between VNFs and the VNFM, followed by the deployment of the VNFM as a container connecting to this network (arrow 6).

The latter, in charge of the life cycle management of VNFs, acts as a central point between them and the orchestrator. As such, it forwards all specific NDN configurations (the NDN prefix-based routing information, the NDN prefix to be blocked in the NDN firewall, etc.) from the NDN engine (arrow 10) to the element managers (EMs, arrow 13), which are the unified management components of VNFs that abstract their specificities. The VNFM also receives notifications from EMs that can issue notifications when an NDN configuration is fully applied or an NDN attack is detected (arrow 12) by a monitoring agent which embeds security applications that allow it to detect NDN attacks. In the case of a routing VNF, an EM uses the NFD management protocol to configure (arrow 15) and monitor (arrow 14) NFD. For scaleout operations, the VNFM also orders the VIM (arrow 8) to spawn new containers that will be configured through previously exposed configuration paths.

MIGRATING HTTP THROUGH DEDICATED GATEWAYS

To overcome the current limitations of service migrations toward NDN, we designed an HTTP/NDN mapping protocol and architecture, whose implementation takes the form of a gateway that can be used to seamlessly transport HTTP traffic over an NDN island [5].

An NDN island using our HTTP over NDN architecture must instantiate two kinds of gateways:

1. An ingress gateway (iGW), that converts HTTP user requests into NDN messages and returning NDN messages into HTTP responses
2. An egress gateway (eGW), the counterpart of the first one, that converts NDN messages into HTTP requests toward web sites and converts HTTP responses into NDN messages

Several eGWs can coexist in the island, each receiving the requests for a given name prefix based on the defined NDN routes. Because HTTPS is by nature in opposition to ICN (no interest to cache end-to-end encrypted content), it is not supported by the gateway. iGW is simply seen by IP users as a web proxy, but their traffic is partly transported by NDN.

The gateways follow a naming pattern based on a naming proposition to convert URL to ICN names.[4] Since an NDN Interest packet cannot carry data while an HTTP request's header does, an iGW (or a native NDN web-client) must exchange different messages in three steps, defined in Table 1, to retrieve the web content. First, the iGW sends an *Interest* whose name components contain, as illustrated in Table 1a:

- The requested domain split by sub-domains and in reverse order (e.g., /com/google/www)
- The path of the content on the web server
- The route toward the sender as a single name component
- A hash of the HTTP request's header (a SHA1 of the header and up to 1024 bytes of the request body) to perfectly identify HTTP requests and their corresponding Data packet

This Interest packet is sent in the NDN net-

---

[3] https://www.etsi.org/technologies-clusters/technologies/nfv, accessed 4 Feb. 2019.

[4] http://www.icn-names.net/, accessed 4 Feb. 2019.

work to ask someone to handle the request (Fig. 2, steps 1–2). Consequently, eGW (or a native NDN web server) knows upon reception that an HTTP request must be satisfied, but also the network name to contact the client to get the request details. Please note that the SHA1 is used to be sure we respect the parameters of the HTTP request to match users' properties. However, carefully choosing a subset of fields to be considered to compute the hash can vastly improve NDN caching while giving consistent results to users.

Indeed, our preliminary experiments showed that users cannot benefit from the cached responses initiated by another web browser. This is mainly due to the relative uniqueness of HTTP requests sent by a browser. As such, HTTP responses' packets can only come from the NDN cache if users ask for the same content in the very same way regarding all options in HTTP requests' headers. By default, a user reloading a page gets 75 percent of packets from the NDN cache, while a subsequent user with the same browser only gets 38 percent, and another subsequent user with a different browser 0 percent. This can endanger the ability of HTTP traffic to take advantage of the NDN island. A solution is to ignore or replace some common but "useless" fields of the header when computing the hash in order to maximize the reusability of concurrent requests for a given content. Our tests show that the cachability is thus vastly improved without affecting the accuracy of delivered content by modifying *accept-encoding* and ignoring the *user-agent*, *accept-language*, *accept*, and *cookie* fields for static contents. With this optimization that preserves all significant HTTP features, a second user with the same browser now achieves 78 percent of cache hit, and a subsequent user with a different browser 61 percent.

In the second step, the eGW extracts information from the first Interest sent by the iGW, more precisely the two last components, the sender route and the hash, in order to retrieve the full HTTP request (Fig. 2, steps 3–4, and Table 1b). The sender route is coded as a single binary field name component to be extracted easily. This preliminary exchange including the sender route is mandatory because Interest packets cannot transport Data in NDN, so the HTTP request must be retrieved specifically. Once the full HTTP request is received by the eGW, it can now ask the HTTP server in the IP domain for the actual web content. After the reception of the HTTP response from a regular HTTP server, the eGW prepares the NDN Data packets. Following the NDN principle, it is up to the NDN client to send Interest packets to retrieve each chunk of the HTTP response (Fig. 2, steps 5–6, and Table 1c). Please notice that the six steps of the mapping protocol are actually done in two round-trip times (RTTs) because some steps are done in parallel (3–4 and 5–6). In the case of a native NDN web server, the mapping protocol is the same, but content is directly sent by the server without issuing traffic on the IP network.

a.	/http/reverse_splitted_domain_name/path/sender_route/sha1
b.	/sender_route/sha1(/segment)
c.	/http/reverse_splitted_domain_name/path/sha1(/version/segment)

Table 1. Naming pattern of the mapping protocol.

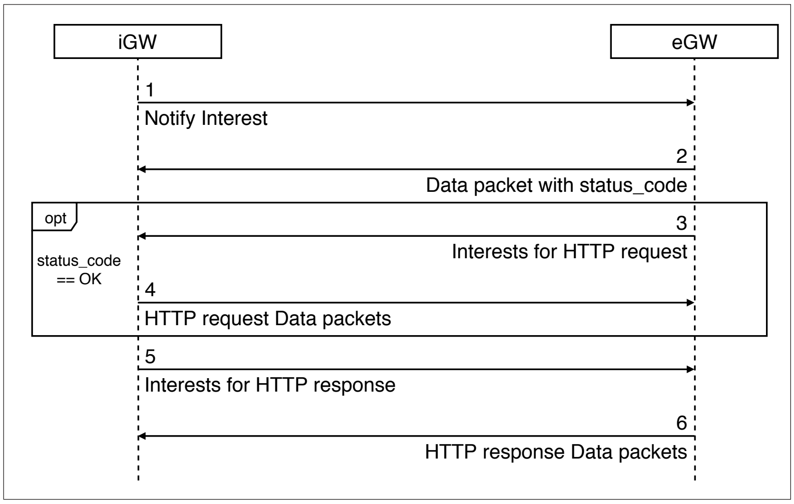

Figure 2. Sequence diagram of the communication protocol between gateways.

EVALUATION

We followed a specific experimental approach to evaluate each of our contributions: unitary tests for the HTTP/NDN gateway, test scenarios with emulated users for the content-oriented orchestration infrastructure, and finally *in vivo* tests with real end users.

EVALUATION OF SERVICE GATEWAYS

We evaluated our gateway both in terms of performance and reliability and in terms of interest and efficiency for end users. For the performance tests, we used one scrapper we internally developed. For the interest and quality for end-users, we used Webview,[5] a tool developed by Orange, allowing us to perform automation of tests for the web browsing service and measuring quality metrics, such as the page load time (PLT), as defined by the W3C consortium.

For the performance and accuracy of our gateway, we evaluated the success of retrieving the requested objects. In Fig. 3 we plot the frequency distribution of the page content, given by the number of HTTP objects retrieved from the set of all objects that the infrastructure successfully requests and retrieves. The result (plot in semi-log scale) shows that most of the top 1000 HTTP websites can be retrieved entirely, those with bad results being mainly remote websites (Chinese, Korean and Russian).

We evaluated the additional delay resulting from the usage of the gateways. This experiment shows that the additional delay resulting from the usage of the gateways is nearly constant and in our case equal to 29+/–3 ms. We can get a constant time because the eGW does not wait for the completion of the HTTP responses and starts to generate Data packets as soon as possible.

EVALUATION OF CONTENT-ORIENTED ORCHESTRATION COMPONENTS

The first experiments we have performed to

---

[5] https://webview.orange.com/, accessed on 04/02/19.

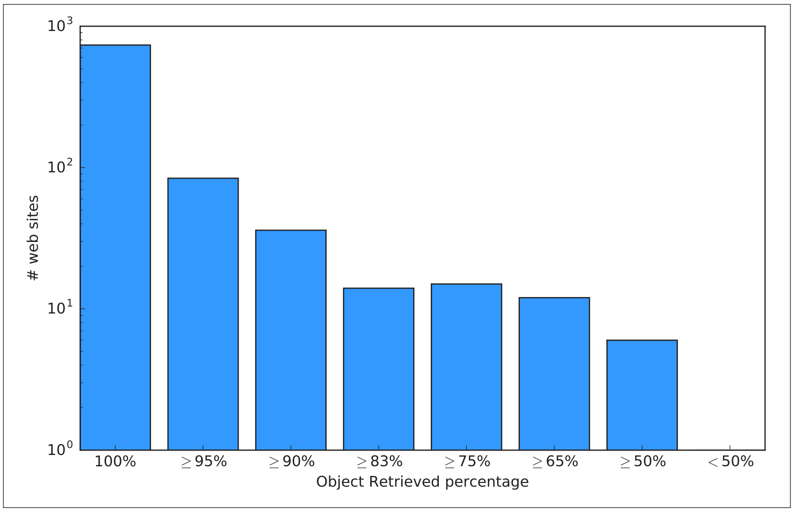

Figure 3. Distribution of web sites based on percentage retrieval.

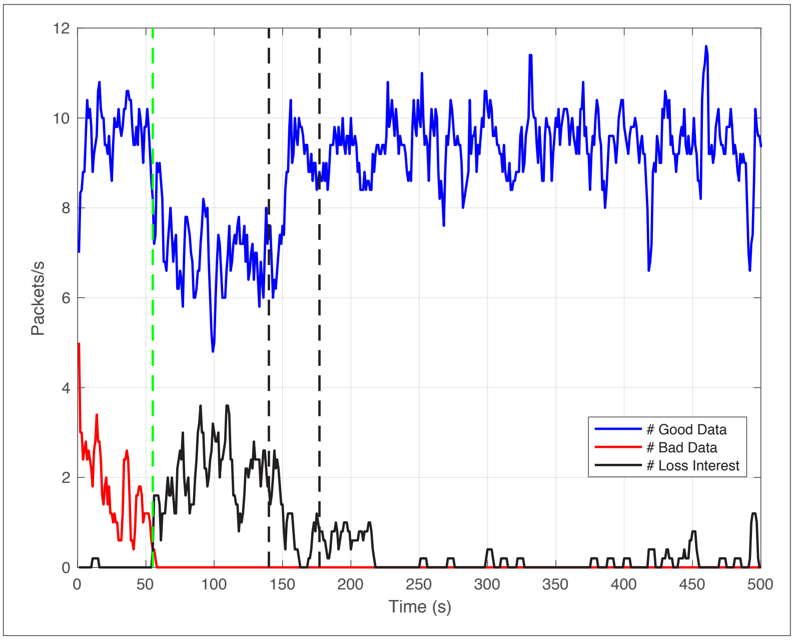

Figure 4. Time series of the dynamic orchestration of CPA mitigation; (red, blue, black) lines stand for (good, bad, lost) data packets, respectively.

assess the performance of our content-oriented MANO framework have consisted of evaluating the capability of our solution to automatically deploy an NDN virtual topology. Different topologies exhibiting various numbers of VNFs and different connectivity degrees have been considered (e.g., star, ring, and mesh). The collected results have shown that:

1. Our content-oriented MANO is able to spawn and configure VNFs in a reasonable time (from 10 to 30 s on average per VNF).
2. The NDN configuration enforcement is reasonably time consuming as compared to standard container spawning.
3. The total deployment time grows linearly according to the number of VNFs.

The bottleneck we identified for the deployment automation resides in the connection degree of VNFs, which induces orchestration operations of both infrastructure layer endpoints as well as the convergence of forwarding routes in the NDN virtual network, thus leading to an potential exponential growth of the deployment time in the case of a full-mesh topology.

Then, in order to evaluate the dynamic part of our orchestration framework, we considered the case of a content poisoning attack (CPA), which is one of the current threats in NDN. Our solution could be extended to other security threats (e.g., interest flooding attack, cache probing, route poisoning), but such a study is left for future work. CPA detection is achieved through a dedicated detector presented in [4], and from an orchestration perspective, we consider three mitigation policies that bring a virtual NDN network under attack back to normal. First, on a CPA detection by a security probe, our orchestrator dynamically enforces the signature verification module, which checks the integrity of each Data packet, of NDN routers located at the edge of the NDN virtual network. Then, for each corrupted prefix, the orchestrator dynamically reconfigures the black list of the upstream virtual NDN firewall to prevent such traffic from entering the network. Finally, to deal with any potential overload of NDN routers enforcing the signature verification, which may induce a too high computing cost, the orchestrator enforces a scale-out policy that dynamically spawns and configures replicas of NDN routers.

Figure 4 exhibits time series of *good data* (blue lines), *bad data* (red lines), and *lost data* (black lines), expressed in terms of number of packets per second, received by a good user under a CPA. The green dotted line shows the time at which the network detects the attack and triggers the firewall and signature verification policies to mitigate the attack.

We can observe that, after this moment, the amount of bad data abruptly drops to zero. However, the amount of good data also decreases slightly, and the number of lost data increases remarkably. The reason lies in the overload the routers performing the signature verification suffer, which prevents them from reliably achieving their basic forwarding operations. Then the two black dotted lines show the moment when the scale-out configuration operations start and terminate. After that time, the network returns to a normal state: good data gets back to its initial rate, and lost data drops to a negligible value. We can also observe that during the period of scale-out, there is no downtime of the network. Overall, these results show the effectiveness of the content-oriented orchestration in terms of security and performance.

REAL USER EXPERIMENTS

To assess the performance of our overall deployment solution for NDN, we conducted a real-scale experiment involving real users. During five days, the HTTP traffic issued by a dozen volunteer students from the Troyes University of Technology campus was routed to a basic version of our testbed, integrating our gateways and a single NDN router but without dynamic reconfiguration.

In total, 38 reports of users' perception of the HTTP service operated over our ICN island were provided. They were questioned on:

1. Any additional delay they perceived as compared to their everyday life navigation

Proposal	% of answers
Delay to retrieve web content	
Really worse than usual	5.26%
Slightly worse than usual	18.42%
No difference	55.26%
Slightly better than usual	10.53%
Much better than usual	10.53%
Missing elements in the retrieved content	
Many elements are missing	5.26%
Some elements are missing	36.84%
All items are fully loaded	57.89%
Adoption for real-life usage	
Yes	89.47%
No	10.53%

Table 2. Quality of experience perception of real users of our HTTP/NDN island.

2. The incomplete retrieval of web contents in the pages they accessed
3. Their will to adopt this technology in real life

The collected results are synthesized in Table 2. They are globally promising since for all questions, the results are satisfying. On the question related to their perceived additional delay, more than half of the users did not notice any change, while two almost identical subsets notice a degradation (23.68 percent) or improvement (21.06 percent), leading us to conclude that the island does not significantly alter the perceived quality of experience, while it acts as an additional domain to cross before reaching the Internet. Similarly, to the question about the capability of our solution to retrieve any web content, more than half of the users were able to retrieve all web objects they accessed, and only a very small portion noticed an important lack in the content retrieval (5.26 percent) due to a bug occurring with non-occidental encoding. Finally, to the last question related to the ICN adoption in the future, a large part of these testers indicated that they are in favor of adopting this technology given their quality of experience during this experiment.

These results lead us to conclude that an ICN island is a promising candidate for a telco to provide web content to its users, but the diversity and richness of web content must be well caught by ICN components and especially gateways to enable fully reliable content retrieval.

LESSONS LEARNED AND FUTURE RESEARCH DIRECTIONS

In this article, we present two contributions aimed at pushing forward the deployment of ICN, and NDN in particular. The first one is a content-oriented orchestrator for NDN networks which brings the content paradigm up to service specification. By extending the TOSCA standard profile, the latter allows us to dynamically manage the life cycle of an NDN network deployed in a virtualized infrastructure and to dynamically react to events or anomalies by reconfiguring a virtualized NDN network. As a use case of service that can benefit from a migration to NDN, we designed a protocol mapping from HTTP to NDN and implemented an operational gateway that allows end users to use current web browsers to reach public web servers via an NDN network. Our solution is compliant with current standards of network virtualization, transparent for end users, adaptive to allow the communication between the NDN and IP worlds, efficient, and reliable. The components have been exhaustively evaluated in the context of a security scenario, emulated web traffic, and also a real deployment campaign where real end users from a university campus, via our NDN testbed, accessed public websites using their own browsers. In this context, we also discovered that caching is not as efficient as we could expect with real end users, due to the variety of accessed web sites, the various web browser configurations, and the personalization of content. As a guideline for future work, novel NDN caching features moving away from generic usage should be refined for the particular web browsing service to improve its general performance and consequently the quality of experience of end users.

The future direction of work in our research area is as follows. Our short-term perspective consists of analyzing the set of metrics we have collected during the experiments with real end users. Our purpose is to provide recommendations and models for subsequent deployments of NDN. Then, to promote our gateway, we will join the federated NDN testbed, and hope to cause an increase of its inner traffic once the web is made available and with the iGW constituting a cheap entry point for users. Finally, the genericity of our solution, which is currently restricted to NDN, is an open question. One can envisage an adaptation of our work for similar ICN solutions such as CCN, which would ease the mapping with HTTP because CCN Interest packets can directly transport the HTTP request, but the design and implementation of content-oriented MANO components is still dependent on the actual ICN technology they manage and would require some effort.

One can envisage an adaptation of our work for similar ICN solutions such as CCN, which would ease the mapping with HTTP because CCN Interest packets can directly transport the HTTP request, but the design and implementation of content-oriented MANO components is still dependent on the actual ICN technology they manage and would require some effort.

REFERENCES

[1] B. Ahlgren *et al.*, "A Survey of Information-Centric Networking," *IEEE Commun. Mag.*, vol. 50, no. 7, July 2012, pp. 26–36.

[2] A. Rahman *et al.*, "Deployment Considerations for Information-Centric Networking," ICNRG draft; https://tools.ietf.org/html/draft-irtf-icnrg-deployment-guidelines-04, 2018.

[3] L. Zhang *et al.*, "Named Data Networking," *ACM SIGCOMM Computer Commun. Rev.*, vol. 44, no. 3, 2014, pp. 66–73.

[4] T. N. Nguyen *et al.*, "A Security Monitoring Plane for Named Data Networking Deployment," *IEEE Commun. Mag.*, vol. 56, no. 11, Nov. 2018, pp. 88–94.

[5] X. Marchal *et al.*, "Leveraging NFV for the Deployment of NDN: Application to HTTP Traffic Transport," *2018 Proc. IEEE/IFIP NOMS*, 2018, pp. 1–5.

[6] H. L. Mai *et al.*, "Toward Content-Oriented Orchestration: SDN and NFV as Enabling Technologies for NDN," to appear, *2019 Proc. IEEE/IFIP IM*, 2019, pp. 1–5.

[7] M. Vahlenkamp *et al.*, "Enabling Information Centric Networking in IP Networks Using SDN," *2013 Proc. IEEE SDN-4FNS*, 2013, pp. 1–6.

[8] S. Salsano *et al.*, "Information Centric Networking Over SDN and Open-Flow: Architectural Aspects and Experiments on the OFELIA Testbed," *Computer Networks*, vol. 57, no. 16, 2013, pp. 3207–21.

[9] N. L. M. van Adrichem *et al.*, "Ndnflow: Software-Defined Named Data Networking," *2015 Proc. IEEE NetSoft*, 2015, pp. 1–5.

[10] X. N. Nguyen *et al.*, "Efficient Caching in Content-Centric

Networks Using Openflow," *2013 Proc. IEEE INFOCOM*, 2013, pp. 1–2.
[11] M. Sardara *et al.*, "Virtualized ICN (vICN): Towards A Unified Network Virtualization Framework for ICN Experimentation," *2017 Proc. ACM ICN*, 2017, pp. 109–115.
[12] P. TalebiFard *et al.*, "An Information Centric Networking Approach Towards Contextualized Edge Service," *2015 Proc. IEEE CCNC*, 2015, pp. 250–55.
[13] S. Wang *et al.*, "On Adapting HTTP Protocol to Content Centric Networking," *2012 Proc. ACM CFI*, 2012, pp. 1–6.
[14] G. White *et al.*, "Content Delivery With Content Centric Networking," Cable Television Labs., Tech. Rep., 2016, pp. 1–26.
[15] I. Moiseenko *et al.*, "TCP/ICN: Carrying TCP over Content Centric and Named Data Networks," *2016 Proc. ACM ICN*, 2016, pp. 112–21.

BIOGRAPHIES

GUILLAUME DOYEN (guillaume.doyen@utt.fr) has been an associate professor at Troyes University of Technology (UTT) since 2006. His current research focuses on the design of autonomous management solutions for content distribution and virtualized infrastructures.

THIBAULT CHOLEZ is an associate professor at Université de Lorraine, where he works in the Inria RESIST research team on data network monitoring and analytics.

WISSAM MALLOULI is currently a research and development project manager at Montimage, France. His topics of interest cover formal methods for monitoring of functional, performance, and security aspects of networks and applications.

BERTRAND MATHIEU joined Orange Labs in 1994. Since 1999, he has worked on distributed computing, programmable networks, overlay networks, QoS and QoE, and information-centric networking.

HOANG-LONG MAI is a Ph.D. student in an Industrial Convention of Formation by Research (CIFRE) contract between Montimage, UTT, and INRIA Lorraine. His Ph.D. topic focuses on the autonomous monitoring and control of virtualized network functions.

XAVIER MARCHAL is a Ph.D. student at LORIA/INRIA. He works on ways to efficiently deploy new networks like NDN thanks to new paradigms like NFV and SDN, but also thinks about ways for these networks to cohabit.

DAISHI KONDO received his Ph.D. degree in computer science from the University of Lorraine in 2018. His research interests include information-centric networking, network security, privacy, and peer-to-peer networking.

MESSAOUD AOUADJ is a postdoctoral researcher at UTT. His research interests focus on network virtualization and software-defined networking.

ALAIN PLOIX has been an associate professor at UTT since 2004. He was the head of the Network and Telecommunication Department from 2010 to 2018.

EDGARDO MONTES DE OCA founded Montimage, a research-oriented SME, in 2004. His main interests include monitoring the security and performance of 4G/5G networks.

OLIVIER FESTOR is a professor in computer science at the University of Lorraine and director of TELECOM Nancy, the Graduate Engineering School in Computer Science.