

A global review of routing mechanisms in the named data network

Yassine Sembati^{1,*}, Najib Naja^{2,*}, Abedallah Jamali²

¹*Institut national des postes et télécommunications,
RABAT Morocco*

²*National School of Applied Sciences of Berrechid (ENSA)
y.sembati@gmail.com naja@inpt.ac.ma jamali.uh1@gmail.com*

ABSTRACT

Named Data Networking (NDN) is the future architecture for the internet, as it should have a good design routing to handle massive node connections with different behaviors. Routing is an important part of NDN since it configures the topology as well as policies, and manages long-term changes to the forwarding table. However, routing protocols are burdened with many challenges that reduce the connection performance. This paper investigates and critically evaluates the classification of NDN routing plane researches. Also, this paper classifies all types of routing protocols that exist in the literature. To our best knowledge, this is the first review that identifies the major challenges, such as routing overhead, in all NDN routing protocol types. This review will serve as guidelines for our future work in NDN routing research.

Keywords: *Named Data network, Routing ,Softwar Defined network, routing overhead*

1. Introduction

Named Data Networking (NDN), [1] [2] is one of the most popular ICN proposals, and it is one of five projects established by the US National Science Foundation as part of its Future Internet Architecture Program. In contrast to IP networks, where the semantics of network services are delivering packets to a specific destination address, the semantics of NDN services are retrieving data identified by a specific name.

The NDN has two types of messages: interest and data. The interest message is used by the content requester to send content requests. The response to a content request is a Data Message sent to the content requester. The content request is forwarded to a content replicator.

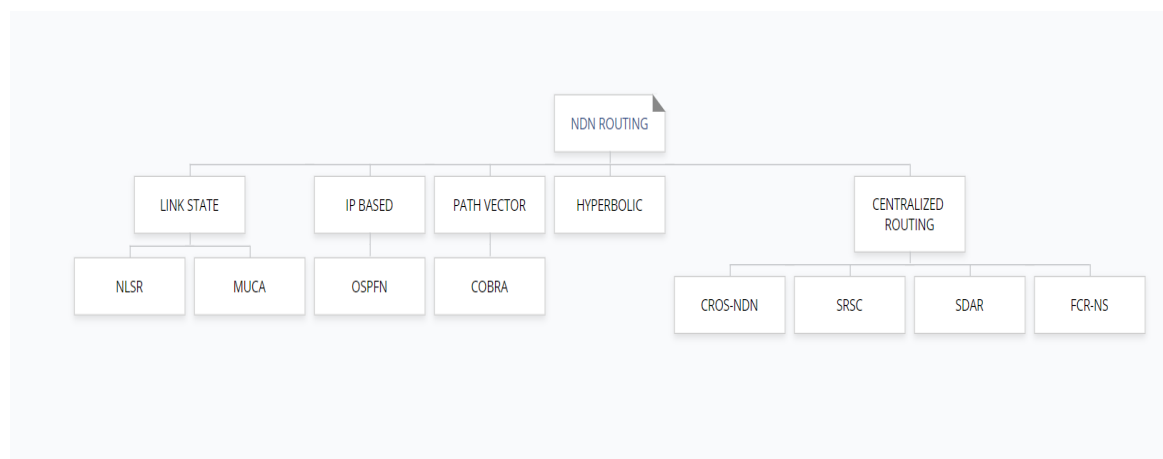
The basic idea behind this concept is that a requester can send requests of interest for content without prior

knowledge of the location of the content. In NDN, content is directly addressable, and routing is done using content identifiers. NDN node uses three tables: Forwarding Information Base (FIB), Pending Interest Table (PIT) and Content Store (CS) to forward interest [3], by default the main strategy of forwarding in NDN is flooding consist of transferring the interest in all faces of the router that engender a high signalization overhead the main role of routing protocols is to fill the table of FIB nodes to avoid flooding caused by the Shema of forwarding nodes that transmit interests on all possible sides. An NDN network uses routing protocols to spread the reachability of data names, similar to how an IP network uses routing protocols to spread the reachability of IP addresses. However, there are several significant differences in routing between IP and NDN networks. as well as to update the forwarding table. NDN Routing Protocol Coordinates with NDN Forwarding Plan for Interface Ranking and Polling In NDN, , the only

difference between routing and forwarding is, while routing decides on the availability of routes, forwarding makes decisions about the preference and use of routes based on their performance/status. The routing information and user-requested content are stored in the router, maximizing the storage capacity of the router. If the link fails, the data can be obtained from the buffer of the intermediate node after recovery, without routing to the original data producer, to reduce delay and bandwidth consumption and improve content delivery efficiency.

This paper is organized as follows. Section 2 presents an overview of ndn routing concepts. Section 3 provides a review of routing protocols mechanism and classification. Section 4 explores the evaluation tools using to investigate the ndn routing protocols. Finally, in section 5 conclusion and future perspectives.

Figure 1. Classification of NDN routing mechanisms



plane, which solely assists the forwarding plan by populating the FIB.

Multipath routing is built into NDN. As a result, each NDN FIB entry contains a prioritized list of numerous interfaces, as well as information on their routing and forwarding planes

3. Classification of routing protocol

3.1 IP based Routing

This protocol of routing is part of NDN routing, which is based on the direct expansion of IP-based routing.

3.1.1 OSPFN

OSPFN is a content-based routing protocol[4] designed for the NDN architecture and is an extension of the OSPF protocol dedicated to the TCP/IP architecture.

2. NDN Routing

Routing schemes that apply to IP-based networks can, in general, be applied to NDN networks as well. In contrast to IP networks, the subject of routing in NDN networks are interests and data packets rather than single network packets.

NDN routing serves as a logical organization layer for NDN interests and packet flows. When NDN is used in its native environment, the NDN routing has a direct impact on network behavior. When NDN is used as an overlay network on top of TCP/IP, the routing of NDN packets is independent of the routing of IP packets used for data transport in lower network layers.

Because the NDN network contains an intelligent and adaptive data plane, the routing plane architecture for the NDN network can differ from that of today's routing

The Opaque Link State Advertisement [5] is a new form of Opaque Link State Advertisement that OSPFN defines (OLSA) which is similar to two layer routing protocol[13]. It supports name prefix advertising from multiple sites (for the same or different prefixes), In addition, since OSPF only provides a single best access to each destination, a multipath configured feature has been added to allow users to specify links to use when the best route fails to bring back data. Although OSPFN does not support full dynamic multipath bordered routing capability, the multipath configured feature has already helped to better understand the forwarding behavior of the current ccnd-station implementation.

OSPFN is currently deployed at the ten participating institutions in the NDN testbed.

3.2 Link State routing

3.2.1 NLSR

The Named-data Link State Routing Protocol (NLSR) was proposed by Hoque, et al. [6] to work on an NDN network and eliminate the need for an IP mechanism, NLSR is a link state routing protocol that runs on top of NDN where each Data_paket used for routing updates are signed by the originating router for authentication support. It also distributes routing updates using the packets Interest and Data. It can route message exchanges using any of NDN's underlying communication technologies (Ethernet, IP tunnels, TCP/UDP tunnels).

NLSR uses a hierarchically structured names scheme to identify various components: routers, routing processes, routing data, and keys because the relationship among them is inherently hierarchical. Furthermore to distribute outing information, NLSR employs Link State Advertisements (LSAs), It uses two kinds of LSAs, adjacency LSAs and Prefix LSAs, adjacency LSAs is used by NLSR to build a network topology while prefix LSAs used to distributing name prefix reachability.

All NLSR routers will be faster in terms of route convergence, however they may have a problem with routing overhead.

3.2.2 MUCA

Since NLSR running the Dijkstra algorithm in a router for each of its active interfaces produces a list of ranked next-hops, which has a substantial processing overhead, especially for routers with high connectivity. To address this issue C. Ghasemi et al [7] propose MUCA a new intra-domain name-based routing protocol to provide simple and scalable support for Multipath forwarding and in-network Caching. As a link-state routing protocol, MUCA employs a distance-vector mechanism to retrieve routing tables from neighboring routers rather than computing them. The prioritized list of possible next-hops can be quickly figured out from these retrieved routing tables, saving CPU cycles for the local router. In addition to support explicit caching and efficient multipath support, MUCA uses a simpler mechanism than NLSR to propagate incremental routing updates. The routing update propagation problem is treated by NLSR as a data synchronization problem, and ChronoSync [8] is used to synchronize the link state database (LSDB) between neighboring routers. MUCA simply informs the neighbor router that a routing update is available and expects the neighbors to retrieve it. As a result, it effectively reduces routing overheads.

Routing Proposal	Main papers	Routing Shema	Mecanisme	Evaluation tools
OSPFN	L. Wang et al [4]	IP-Based	-OSPF enhancement -OLSA -Using repo/sync	NDN TESTBED
NLSR	Hoque, et al [6]	Name-based	-OSPFN enhancement -OLSA -LSA Broadcast -Use cheronoSyc	NDNTESTBED/ MINI-NDN
MUCA	C. Ghasemi et al [7]	Name-based	multipath forwarding - learns multiple alternative paths from neighboring routers	NDNSIM
COBRA	M. Tortelli et al [9]	Name-based	-Use Bloom filter -creates routes based on paths used previously	NDNSIM
Hyperbolic routing	V. Lehman et al [10] A. Aboud et al [11]	Location-based	-utilizes greedy -Shortest distance -Low memory	MINI-NDN
SDAR	Y. Liu et al [16]	Controller-based	-SDN/NDN combinaison -Efficient communication controller /router	NDNSIM
SRSC	E. Aubry et al [17]	Controller-based	-High, caching performance -Centralized Control plane	DOCKER TESTBED/ NDNx
CROS-NDN	Torres, et al [18]	Controller-based	-supports content mobility -Provides fast content recovery -improves the routing efficiency	NDNSIM
FCR-NS	Kalghoum et al [19]	Controller-based	-cache policy based on popularity of data - use bloom filter	NDNSIM

Table. 1. Classification of ndn routing mechanisme

3.3 Path vector

3.3.1 COBRA

Tortelli et al. proposed COBRA [9] (COntent-driven Bloom filter based Routing Algorithm), a bloom-filter based intra-domain routing algorithm for NDN. COBRA builds routes based on previously used paths for content retrieval and keeps routing information current without requiring substantial transmission between nodes. COBRA does not add any new data structures or packet processing operations that could increase the computational overhead at an NDN node, thus is fully distributed and does not require any update message exchanges between nodes to maintain the consistency of the collected information; in fact, it is locally guaranteed by the COBRA SBF combination. Furthermore, COBRA does not require a fallback IP-based routing system or one that is responsible for announcing routes to permanent content copies.

COBRA was not implemented in a large topology (use GEANT topology.) to evaluate the impact of increasing nodes on the routing overhead, and it's not compared with other routing protocols (NLSR).

3.4 Hyperbolic Routing

Hyperbolic routing is a geometric routing technique that sends packets over a network effectively using hyperbolic coordinates. It was investigated in [9]. Assuming that a mechanism for retrieving the coordinates of a name prefix exists (e.g., through NDNS), each Interest can carry the name prefix's coordinates, and routers can use greedy forwarding to forward the Interest, selecting the next hop(s) based on the distances between the router's neighbors and the Name prefix. There is no need to maintain a routing table or FIB, and there are no dynamic routing updates, hence this technique is very scalable.

V. Lehman et al proposes HR as an NDN routing mechanism for the following reasons: (i) real-time determination and calculation of the best path, as well as resource, memory, and routing overhead friendliness; and (ii) expected to be stable within topology changes, i.e., backtracking capability. (iii) HR is more reliable than pure geographic routing [11] for distributed network traffic. Despite the fact that the NDN namespace

In [10] in order to confirm that hyperbolic routing is a viable candidate for NDN inter-domain routing, its performance in a large topology was evaluated and compared to link-state routing. The following criteria were used for comparison: packet loss ratio, RTT, message overhead, and failure response time of data retrieval. Experiments are carried out with various forwarding strategies. According to the findings, hyperbolic routing must have a much lower forwarding overhead than the smart flooding strategy in order to be a viable solution for NDN.

The results of the performance evaluation show that the forwarding plane HR overhead in such networks is significantly lower than the control overhead of NLSR while achieving comparable packet delay and loss rates. Furthermore, as the network size grows, the per-node message overhead of HR remains nearly constant and close to zero, as opposed to the usual polynomial growth experienced by traditional links-state networks.

There are several technical issues that need to be addressed before including mechanisms to compute and distribute hyperbolic coordinates in a decentralized manner while properly accounting for link delays [10].

3.5 Centralised Routing

This part explores centralized routing which is part of NDN routing. In recent years, a new paradigm, centralized network, has gained traction. The old paradigm in which network control is distributed to a

centralized control on each node, known as Software Defined Network (SDN) with OpenFlow protocol. [14]–[15]

SDN is a network architecture design that focuses on centralized control plane functions and the separation of user and control planes.

The SDN function improves scalability by separating the data plane and control plane. Furthermore, centralized control plane functions make complex mechanisms like smart forwarding and caching easier to implement without adding to routing overhead.

3.5.1 SDAR

Taking advantage of integration ndn-sdn Y. Liu et al design and develop SDAR [16], a Software Defined Intra-Domain Routing Control Platform in NDN, that can manage network-wide routers and various dynamics effectively, by leveraging the unique feature-adaptive forwarding in NDN and the centralized management and control in Software Defined Networking (SDN).

Through SDAR, Y. Liu et al show how SDN concepts can be applied to the NDN context to manage and control networks in a feasible and efficient manner.

SDAR is the first implementation that combined SDN and NDN concepts to implement adaptive forwarding via intra-domain multi-path routing algorithms.

Instead of all participating routers, designing and implementing a centralized routing system on a centralized controller in NDN can considerably minimize the complexity and overhead in network management and routing setups.

3.5.2 SRCS

SDN-based routing scheme for CCN developed by SRSC [17] This scheme employs both data messages and a content-centric network (CCN) interest to establish a communication channel between the controller and the nodes. As a result, when this scheme is used and no input is assigned to the node in its FIB, the node passes the interest to the controller.

As a result, the latter must represent the network's overall knowledge (paths, nodes, contents stored in the caches and topology). It can respond with a piggy-backed forwarding rule embedded in the data message sent by the controller via the reverse path to the node. As a result, by employing SRSC, no network flooding will occur in order to obtain the content.

Furthermore, the controller informs the nodes about the routing of the relevant message to the nearest requested content. Its workload varies according to the size of the network and the number of nodes. As a result,

a large-scale network must be divided into domains that are managed by a controller.

SRSC, which has been integrated into NDNx and deployed on the virtual Docker testbed. Furthermore SRSC has a low overhead (18%) yet still performs well in terms of caching (47 percent of Cache Hit), comparing to the default forwarding strategy in ndn.

As perspective E. Aubry et al suggest extend the comparison to englobe other routing protocols (NLSR), and calibrate the SRSC parameter to reduce the control message.

3.5.3 CROS-NDN

The CroS-NDN [18] approach isolates the control plane from the data plane. It consolidates the former onto a dedicated node (controller). This consolidation allows network nodes to store and request network data without being distributed throughout the network. The network information is dynamically stored on a controller in a controller-based routing scheme for NDN nodes.. These nodes also request new routes to the controller based on consumer interests at locally unknown name prefixes.

To minimize additional routing signaling, CroS-NDN limits the interest of the flooded network. In effect, the various nodes in the network simply flood it to look for the controller. Then, the latter finds the flooding that occurs exactly on the nodes for the controller's interest delay. The main functions of each CroS-NDN node are to monitor its one-hop neighbors and to store all variations in the controller.

3.5.6 FCR-NS

FCR-NS is an SDN-based routing protocol [19] that employs a new cache replacement policy that calculates the popularity of local data in switches, as well as the bloom filter structure, to accelerate forwarding. It is completely independent of current Internet architecture and only uses NDN messages. and because it is based on SDN, it provides complete separation of the control and data planes.

The proposal's main goal was to improve the performance of the current NDN network in order to facilitate its deployment in the real Internet network.

FCR-NS is a new NDN architecture that has the following features:

The data plane is isolated from the control plane in this NDN architecture based on SDN.

The usage of a collection of bloom filters tables allows for a very rapid forwarding procedure.

A highly effective cache replacement approach based on the switches' real-time estimate of data popularity.

SDN controller manages a very quick intra-zone routing procedure.

Based on the use of bloom filter tables in SDN controllers, a very quick inter-zone routing process between zones.

The SDN architecture allows for a considerable reduction in interest overhead. Indeed, in SDN architecture, the controller had a global picture of the data distribution in the network, which reduced the overhead of messages and the time spent searching for data.

When compared FCR-NS to other solutions such as NDN, it may require more memory space.

4. EVALUATION PLATFORMS

Typically, modeling and simulation are used to investigate any problem in the field of NDN networking.

An emulator or testbed can be used to conduct the investigation. While simulators such as ndnSIM can be used as simulation tools

4.1 NDNSIM

We use a dedicated NDN network evaluation and simulation environment to evaluate the approaches presented in the preceding sections. There are currently two major simulation environments available. ndnSIM 2.0 [20] is a simulation for low-level NDN network activities.

which is based on ns-3 [21], a discrete-event network simulator for Internet systems NdnSIM implements all basic NDN operations and allows for the simulation of NDN networks with NFDs, network components, and generators for various data request distributions.

NdnSIM is written in C++ and simulates all NFD components such as PIT, FIB, and Faces as modular components. To accomplish this, ndnSIM includes C++-APIs that are compatible with the native C++ NDN development libraries ndn-cxx (NDN C++ library with eXperimental eXtensions) [22]. Furthermore pyNDN is available as C++-Python-Bindings for ndn-cxx [23] for using Python applications in ndnSIM. C++ and Python applications can be used as native applications in simulated network environments by using the ndnSIM simulation environment. To accomplish this, the application makes use of compatible simulator APIs rather than the regular NFD APIs that handle communication.

4.2 MINI-NDN

Mini-NDN is a network emulation tool that is both lightweight and scalable [24]. It is used on the NDN platform for testing, experimenting, and research. Mini-

NDN combines the NDN libraries NFD, NLSR, and the NDN project's tools to provide a single system for simulating an NDN network, It is a free and open software tool distributed under the GPL 3.0 license.

Although it has primarily used Mini-NDN to test and assess NLSR applications, researchers can program any experiment or test and evaluate any NDN application within the Mini-NDN framework.

4.3 NDN Testbed

The NDN team at Washington University runs and oversees the global NDN testbed [25], which included 33 nodes in 15 countries as of April 2017. Integration testing and deployment are also handled by the Washington University team. They create and maintain monitoring tools for the NDN Testbed, including a Cacti-based node resource (CPU, memory, etc.) and a Cacti-based node resource.

Monitoring facility, as well as a real-time testbed usage mapping tool based on NDN. The usage data is displayed on a web page at <http://ndnmap.arl.wustl.edu>. The deployment, testing, debugging, and evaluation of hyperbolic routing have all been key testbed advancements, revisions, and deployments in the last year.

CONCLUSION AND FUTURE PERSPECTIVES

The research community has been paying close attention to named data networking. In particular, the routing strategy is one major property that affects NDN efficiency. This work provides a brief review of research works in the area of NDN routing. We have broadly classified the NDN routing mechanisms. We finally show the evaluation tools used to run the different experiences of routing protocols.

In future work, we plan to implement different routing protocols in a controller using the software define network paradigm to investigate routing scalability and overhead signalization, and other QoS metrics, routing efficiently.

REFERENCES

- [1] V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs, and R. Braynard, "Networking Named Content," in Proc. of CoNEXT, 2009.
- [2] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, K. Claffy, P. Crowley, C. Papadopoulos, L. Wang, and [1] B. Zhang, "Named Data Networking," ACM Computer Communication Review, July 2014.
- [3] C. Yi, A. Afanasyev, I. Moiseenko, L. Wang, B. Zhang, L. Zhang, "A case for stateful forwarding plane,"
- [4] L. Wang and A. K. M. M. Hoque, "OSPFN: An OSPF Based Routing Protocol for Named Data Networking," pp. 1–15, 2012.
- [5] L. Berger, "RFC 5250 - The OSPF Opaque LSA Option," 2008.
- [6] Hoque, A.K.M.M., Amin, S.O., Alyyan, A., Zhang, B., Zhang, L., Wang, L.: NLSR: named-data link state routing protocol. In: Proceedings of the 3rd ACM SIGCOMM Workshop on Information-Centric Networking, pp. 1–6 (2013). <https://doi.org/10.1145/2491224.2491231>
- [7] C. Ghasemi, H. Yousefi, K. G. Shin, and B. Zhang. MUCA: New Routing for Named Data Networking. In 2018 IFIP Networking Conference (IFIP Networking) and Workshops, pages 289–297, May 2018.
- [8] Z. Zhu and A. Afanasyev, "Let's ChronoSync: Decentralized dataset state synchronization in named data networking," in IEEE ICNP'13, 2013, pp. 1–10.
- [9] M. Tortelli, L. A. Grieco, G. Boggia, and K. Pentikousis. Cobra: Lean intra-domain routing in
- [10] V. Lehman, A. Gawande, R. Aldecoa, D. Krioukov, and L. Wang, "An Experimental Investigation of Hyperbolic Routing with a Smart Forwarding Plane in NDN," 2016.
- [11] A. Aboud and H. Touati, "Geographic interest forwarding in NDN-based wireless sensor networks," in Proceedings of IEEE/ACS International Conference on Computer Systems and Applications, AICCSA, 2017,
- [12] V. Jacobson, J. Burke, L. Zhang, B. Zhang, K. Claffy, C. Papadopoulos, T. Abdelzaher, L. Wang, J. A. Halderman, and P. Crowley, "Named data networking next phase (ndn-np) project may 2014-april 2015 annual report,"
- [13] Huichen Dai, Jianyuan Lu, Yi Wang and Bin Liu, "A two-layer intra-domain routing scheme for named data networking," 2012 IEEE Global Communications Conference (GLOBECOM), 2012, pp. 2815-2820, doi: 10.1109/GLOCOM.2012.6503543.
- [14] N. McKeown et al., "OpenFlow: Enabling Innovation in Campus Networks," ACM SIGCOMM Comput. Commun. Rev., vol. 38, no. 2, p. 69, 2008.

- “Opendaylight web.” [Online]. Available: <https://www.opendaylight.org/>
- [15] Y. Liu and H. Wadekar, "SDAR: Software Defined Intra-Domain Routing in Named Data Networks," 2016 IEEE 15th International Symposium on Network Computing and Applications (NCA), 2016, pp. 158-161, doi: 10.1109/NCA.2016.7778611.
- [16] E. Aubry, T. Silverston and I. Chrismen, "Implementation and Evaluation of a Controller-Based Forwarding Scheme for NDN," *2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA)*, 2017, pp. 144-151, doi: 10.1109/AINA.2017.83.
- [17] Torres, J.V., Duarte, C.: CROS-NDN: controller-based routing strategy for named data networking. In: Universidade Federal do Rio de Janeiro-GTA/COPPE/UFRJ, pp. 1–6 (2014)
- [18] Kalghoum, A.; Saidane, L.A. FCR-NS: A novel caching and forwarding strategy for Named Data Networking based on Software Defined Networking. *Clust. Comput.* 2019, 22, 981–994.
- [19] Spyridon Mastorakis, Alexander Afanasyev, Ilya Moiseenko, and Lixia Zhang. 2016. ndnSIM 2: An updated NDN simulator for NS-3. Technical Report NDN-0028, Revision 2. NDN
- [20] G. F. Riley and T. R. Henderson, “The ns-3 Network Simulator”, in *Modeling and Tools for Network Simulation*, K. Wehrle, M. Gunes, and J. Gross, Eds., Berlin, Heidelberg: Springer, 2010, pp. 15–34.
- [21] A. Afanasyev, Y. Yingdi, J. Shi, D. Pesavento, E. Newberry, J. Pereira, S. Mastorakis, W. Shang, M. Sweatt, V. Lehman, J. Thompson, T. Jiewam, J. Quevedo, I. Moiseenko, S. Chen, X. Jiang, and M. Juszkievicz, *Ndn-cxx: NDN C++ library with eXperimental eXtensions*, Jul. 2016.
- [22] D. Kulinski, A. Afanasyev, W. Shang, and Y. Yingdi, *PyNDN - NDN bindings for Python*, May 2016. [Online]. Available: <https://github.com/cawka/PyNDN> (visited on 07/04/2016).
- [23] “Mini-NDN GitHub,” <https://github.com/named-data/mini-ndn>.
- [24] Ze'ev Lailari, Hila Ben Abraham, Ben Aronberg, Jackie Hudepohl, Haowei Yuan, John D. DeHart, Jyoti Parwatar, and Patrick Crowley. Experiments with the Emulated NDN Testbed in ONL. In *Proceedings of the 2nd ACM Conference on Information-Centric Networking*. ACM, 2015.