# Experimenting with an SDN-Based NDN Deployment over Wireless Mesh Networks

Sarantis Kalafatidis*‡, Vassilis Demiroglou*†, Lefteris Mamatas*‡, Vassilis Tsaoussidis*†
* *Athena Research and Innovation Center, Greece*
† *Department of Electrical and Computer Engineering, Democritus University of Thrace, Greece*
{*vdemirog, vtsaousi*}*(at)ee.duth.gr*
‡ *Department of Applied Informatics, University of Macedonia, Greece*
{*kalafatidis, emamatas*}*(at)uom.edu.gr*

***Abstract*—Internet of Things (IoT) evolution calls for stringent communication demands, including low delay and reliability. At the same time, wireless mesh technology is used to extend the communication range of IoT deployments, in a multi-hop manner. However, Wireless Mesh Networks (WMNs) are facing link failures due to unstable topologies, resulting in unsatisfied IoT requirements. Named-Data Networking (NDN) can enhance WMNs to meet such IoT requirements, thanks to the content naming scheme and in-network caching, but necessitates adaptability to the challenging conditions of WMNs.**

**In this work, we argue that Software-Defined Networking (SDN) is an ideal solution to fill this gap and introduce an integrated SDN-NDN deployment over WMNs involving: (i) global view of the network in real-time; (ii) centralized decision making; and (iii) dynamic NDN adaptation to network changes. The proposed system is deployed and evaluated over the w-iLab.1 Fed4FIRE+ test-bed. The proof-of-concept results validate that the centralized control of SDN effectively supports the NDN operation in unstable topologies with frequent dynamic changes, such as the WMNs.**

***Index Terms*—Software-Defined Networks, Information-Centric Networking, Named Data Networking, Wireless Mesh Networks, Internet of Things**

## I. Introduction

The emerging and ubiquitous Internet of Things (IoT) technology brings a profound quality impact to our every-day life with its diverse applications, including e-health monitoring, environmental quality notification and Smart-City applications [1]. Such IoT applications are typically associated with critical performance requirements, such as low-delay, reduced communication overhead and high resilience. Thus, the efficient functionality of these applications is inherently associated with the IoT network performance.

Several IoT networks are encompassing wireless mesh technology, in order to increase their communication range in a multi-hop manner. Indeed, Wireless Mesh Networks (WMNs) fulfill the multi-hop communication needs of distant or infrastructure-free IoT deployments (e.g., Smart-City, Environmental Monitoring), however are facing unstable topologies, connection failures and low-quality communication (e.g., due to signal interference, mobility). These arising issues are impacting the performance of IoT applications, as they can not meet the aforementioned requirements.

In this context, Named-Data Networking (NDN) [2], an Information-Centric Networking (ICN) [3] architecture, has been proposed as a promising approach to match the IoT application requirements [4]. In particular, NDN architecture: (i) facilitates content retrieval from the network, as NDN packets contain data names instead of IP addresses; and (ii) contributes to reduced communication overhead thanks to the in-network caching [5]. However, deploying NDN in such dense wireless mesh networks requires to employ additional mechanisms that rapidly detect network changes and appropriately adjust the NDN nodes.

Software-Defined Networking (SDN) is ideal to provide the missing features of intelligent centralized control and programmability of the NDN networks. In summary, we consider the following features offered by the integration of SDN with NDN over WMNs: (i) adaptation to dynamic changes (e.g., when a new NDN node participates in the network); (ii) flexibility in the network (e.g., using alternative paths depending on the network state); (iii) reliability/fault tolerance by the rapid detection of network failures; and (iv) content-aware decision making based on caching information.

In this work, we present our SDN-based solution to facilitate NDN adaptability in unstable wireless mesh networking conditions. In particular, our approach exploits the advantages of SDN-NDN integration enabling efficient NDN Interest-Data exchange (in terms of network delay). Our solution encompasses: (i) a global view of network topology in real-time; (ii) centralized decision making (including content-based decision making and best-path selection) and (iii) dynamic NDN adaptation to network changes.

Our proof-of-concept evaluation involves experimentation over a real WMN, investigating both the control overhead and the data messages exchange performance (e.g., NDN packets). The evaluation results affirm the deployability of the proposed system, illustrate our system's delay performance and validate the experimental methodology. Our contributions include:

- a novel SDN-NDN integration system over mesh networks.

TABLE I
RELATED WORK COMPARISON

| Feature / Works | NDN | SDN | SDN-NDN Mechanisms | Wireless Mesh | Real Experimentation |
|---|---|---|---|---|---|
| [6] | ✓ | X | X | ✓ | ✓ |
| [7] | ✓ | ✓ | SDN-based routing scheme | X | X |
| [8] | ✓ | ✓ | Multipath Forwarding | X | ✓ |
| [9] | ✓ | ✓ | FIB modifications | X | X |
| Present Work | ✓ | ✓ | NDN routing | ✓ | ✓ |

- An integrated network control, best path selection and content-aware mechanism that facilitates the coexistence of SDN, NDN and wireless mesh technologies.
- A prototype experimental design to facilitate multi-hop mesh networking experimentation, based on the w-iLab.1 [10] Fed4FIRE+ test-bed.
- Proof-of-concept experimental results of our solution.

The rest of the paper is outlined as follows. In Section II we discuss relevant approaches that deal with similar problem and solutions. In Section III we detail our proposed system and its main features. In Sections IV and V we provide our experimentation analysis and discussions of our future works, respectively.

## II. RELATED WORK

The unstable topologies of wireless mesh networks benefit from the ICN approach, thanks to the in-network caching and the reduced communication overhead, as shown in [6]. Particularly, this work examines the advances of Information-Centric Networking (ICN) protocol by conducting experiments in the CityLab test-bed [11], to demonstrate that ICN is capable of operating in outdoor environments. However, in this work static routing has been used which is not the optimal choice in dynamic networking conditions. Motivated by this fact, we employed an SDN logic to adapt NDN routing in unstable mesh networks, in a robust and flexible manner.

Many works have been trying to resolve NDN routing and forwarding limitations with SDN [12]. In [7], authors designed and evaluated an SDN-based Routing Scheme for CCN/NDN (SRSC) which fully exploits the NDN principles and thus the controller and the nodes communicate using NDN messages (exchanging control and information messages). Particularly, the controller makes the routing decisions and the NDN nodes act as forwarding devices only, as in our case. In SRSC the Controller only informs an NDN node with the entire path to the content and afterwards the NDN nodes communicate with each other (hop-by-hop), in order to create the specific path. In contrast, we selected the controller to communicate independently with each on-path node and configure the NDN network, because of the unstable mesh topology.

Authors in [9] introduced an integrated SDN-NDN framework and modified NDN's FIB design in order to address FIB overflow. In particular, FIB overflow is affected by: (i) the large number of different contents; and (ii) the long-lived FIB entries. In our work, we maintain the default NDN's FIB design and address those issues by creating short-lived FIB entries in the NDN network.

In [8], the authors proposed an SDN-enabled controller for multipath forwarding in NDN. The SDN controller analyzes the global view of the NDN network and makes appropriate forwarding decisions, according to the router states, the available forwarding paths and the cached contents. The particular centralized solution improves the performance of NDN compared to a distributed multipath forwarding strategy, which relies on *a priori* forwarding information and is inappropriate for networks with dynamic topologies, as in our case. Such solutions have been evaluated over a real-world WAN network, so frequent path changes and unstable topologies were not investigated in depth.

Most of the aforementioned works are not validated in real-world test-beds. Moreover, there is no prior work that addresses the challenging communication issues of deploying NDN in real-world mesh networks, using the SDN approach, as illustrated in Table I.

## III. PROPOSED SYSTEM

The WMNs may face instabilities (e.g., due to high interference of multiple transmissions or data loss in noisy areas), leading to frequent topology changes, even in network environments with static wireless nodes. Inherently, the NDN does not have mechanisms that can support its efficient operation over such environments, so real-time network monitoring for reliable route detection is required. Our strategy is to determine the best NDN path in terms of reliability and network performance (i.e., delay) for each destination in a WMN, according to the monitoring data. To achieve this, we exploit the information of a mesh network routing protocol to configure NDN paths through centralized control.

According to these lines, we present our proposed SDN-based system and its corresponding mechanisms, aiming to the flexible and adaptive NDN operation over the WMNs. Its main functionalities are: (i) real-time centralized monitoring of the wireless mesh network; (ii) dynamic best-routing decision making; and (iii) NDN configuration according to the selected routes. The system consists of two functional entities: (i) *the SDN-NDN Controller* which is the centralized control point of the network; and (ii) *the Network Nodes* consisting of the

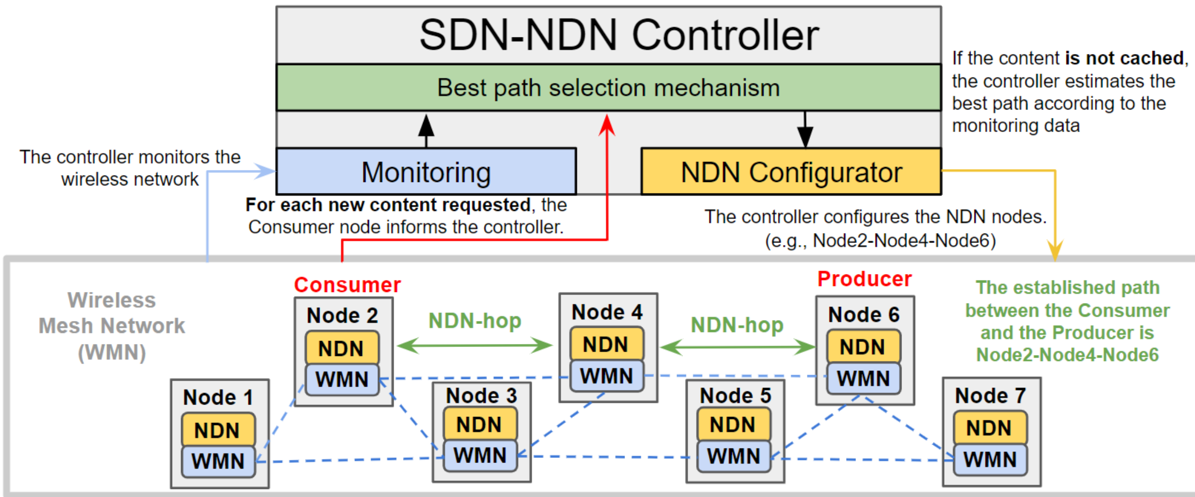


Fig. 1. Overview of the Proposed System

wireless mesh plane and the overlay NDN plane. Their detailed description follows.

### *A. SDN-NDN Controller*

The Controller is the key component that enables the integration of the NDN and SDN as it performs centralized monitoring of the wireless mesh network in order to configure the NDN network. Its main objectives are: (i) information collection about the network state and rapid detection of network changes, enabling the global view of entire network in real-time; (ii) to determine the best route per each Interest-Data packet exchange; and (iii) to establish NDN route (including per-hop NDN face[1] and FIB entry creation). The overview of the Controller's operation is presented in Fig. 1.

The monitoring data collection of WMNs is performed through the WMNs routing protocol, i.e., the blue blobs of Fig. 1. That is, the centralized information collection from the distributed wireless nodes, regarding the discovery of the neighbors of each node and the routes that occur between the hops of the network. The routing costs among the links are based on the quality metrics of B.A.T.M.A.N. [13] routing protocol (Layer 2) which is utilized in our implementation. The Controller is also located in the WMN network and communicates with the Network Nodes over IP.

The Controller manages the content requests through the reactive operation, following these four actions: (i) receives Consumer's updates for each new Interest packet; (ii) defines the best wireless routing path among the Consumer and the Producer, according to the collected monitoring data; (iii) establishes the selected NDN route; and (iv) triggers the Consumer to send the particular Interest packet. As the Data packet will travel through the reverse path, the Controller node stores the path that has cached the particular Data packet and estimates the caching remaining time. This can be accomplished by maintaining information about the particular Data packet (e.g., Freshness Period) as well as the corresponding caching information (e.g., utilized cache size, remaining entries, caching policy). In this light, our Controller associates the content prefix with the corresponding previously returned path and Freshness Period in order to determine whether the content is cached.

### *B. Network Nodes*

The infrastructure of our proposed system consists of interconnected wireless nodes that support NDN communication. The NDN and WMN functionalities are independent and are integrated with the Controller (i.e., the Controller monitors the WMN in order to configure the NDN). In this section, we describe the *NDN and WMN planes* in order to illustrate the system's network nodes operation.

*1) NDN Plane:* In a nutshell, NDN is a future Internet architecture that follows the ICN principles and accomplishes named content retrieval by employing two types of packets (e.g., the Interest and the Data packets). In NDN, Consumers send an Interest packet in the network in order to fetch the corresponding Data packet that contains the requested content. Although a Data packet is originally generated from a Producer node, it may be retrieved from intermediate nodes' caches, as NDN supports in-network caching.

Each NDN node uses three components: the Content Store (CS), the Pending Interest Table (PIT) and the Forwarding Information Base (FIB). When a new Interest packet is received, the NDN node first checks if the requested content exist in the CS (i.e., it is cached) and in that case it responds directly with the Data packet. Alternatively, if the prefix matches a specific PIT entry (which contains the already sent Interest prefixes associated with the respective faces), then the incoming face is added to the particular entry (meaning that when the Data is fetched it will be forwarded also to that face). Otherwise, a new PIT entry is created and the Interest is forwarded to the next hop according to the FIB information [5].

[1]NDN faces can be either physical (e.g., Ethernet) or logical interfaces (e.g.,TCP or UDP channel)

In our NDN deployment, face creation and prefix registration are triggered from the controller node. Since NDN communication is consumer-driven, our system performance heavily relies on the Consumer node behavior. Thus, this plane targets to fetch the data efficiently with the minimum communication delay, by exploiting the NDN features.

Here, we give an example of the NDN and SDN interaction of our system, as illustrated in Fig. 1. The NDN Consumer (node2) intends to fetch the content with the prefix *sensor/temperature*. Thus, it sends a request to the Controller in order to inform about the specific content. If the content is not cached in the network, the Controller finds the best path to the NDN Producer (e.g., Node2-Node4-Node6) and establishes the NDN routes. Finally, the Controller triggers the Consumer to transmit the Interest packet through the created path.

*2) Wireless Mesh Network Plane:* The wireless mesh network is responsible for: (i) the dynamic topology detection (i.e., rapid detection of nodes entering the network, path failures and low-quality communication); (ii) the collection of necessary network information for its smooth and efficient operation; and (iii) the Controller updating about the network state. To enable these capabilities, we use a WMN routing protocol that allows the dynamic monitoring of the WMN and the knowledge/discovery of the best routes at each node (e.g., it determines for each destination in the WMN one single-hop neighbor, which can be utilized as the best gateway). These pieces of information are decentralized, so are passed to the Controller to enable the global view of the network.

The protocol we employed to meet the above requirements is the B.A.T.M.A.N. routing protocol. It belongs to the Data Link layer and its operation is based on the logic of distance vector protocols, (i.e., it does not know the whole topology but only the neighbors and the ”best” next single-hops) and responds efficiently to dynamic topology changes and unreliable conditions of wireless mesh networks. The information collected by the B.A.T.M.A.N. at each node are: (i) the neighboring nodes; and (ii) the next hop on the best path from the current node to the destination. The information is passed to the Controller thus allowing it to know progressively the entire topology and therefore all the estimated best paths.

## IV. Real Experimentation

In this section, we demonstrate the functionality of the proposed system, performing experiments to investigate the delay performance of NDN over real WMNs, concluding to proof-of-concept results. The experimental setup includes the implementation of WMN connectivity with B.A.T.M.A.N. routing protocol and NDN-deployment by using our ready-to-use docker containers. At this point of investigation, we target demonstrating the flexibility and adaptability capabilities of our system, which will render NDN deployments over dynamic wireless mesh topologies feasible. Due to its heavy implementation requirements, we consider its performance optimization as a potential subject for future work. A detailed description of the experimental methodology follows.

### A. Experimentation Environment and Methodology

We utilized the w-iLab.1 Fed4FIRE+ [10] test-bed and reserved nodes from the ninth floor of the respective office lab. We build a multi-hop network topology consisting of seven wireless network nodes (Intel NUC devices) as illustrated in Fig. 2, including one Consumer (Node9-13) and one Producer node (Node9-21). The nodes are interconnected with the B.A.T.M.A.N. [13], forming a WMN, Fig. 3. The NDN implementation is based on the containerized Named data Forwarding Daemon (NFD) [14] and has been deployed on all network nodes. We developed the NFD in Docker containers [15] because it is lightweight, easily reconfigurable and facilitates the NDN deployment on any hardware.

The Controller participates in the same wireless topology, i.e., Node9-3 of Fig. 3, and performs centralized control of entire network. The monitoring data of each node: (i) include information about the B.A.T.M.A.N neighbors and originators enabling the global view of the network; and (ii) are sent to the Controller through the A.L.F.R.E.D. tool [16]. The Controller configures the NDN nodes by transferring messages to the NFD container over TCP. These messages include information about the next-hop and the particular content prefix and thus enable NDN face creation and prefix registration.

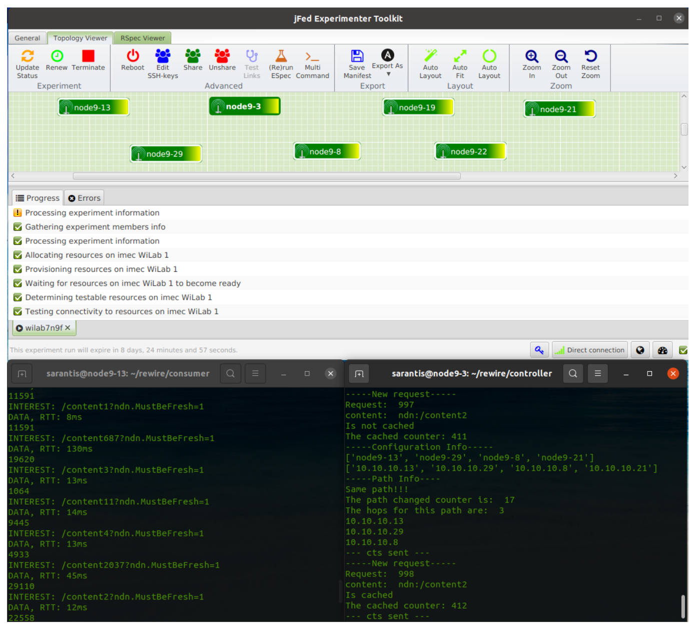


Fig. 2. Experiment demonstration using the jFed [17] GUI tool

We envisioned an IoT scenario, where an IoT Consumer requests sensor measurements from an IoT Producer, that generates emulated sensor measurements. These devices are not located nearby, thus their connectivity was accomplished through the WMN. We considered 150 different IoT contents which are generated from the Producer node. These produced Data packets have a limited size (350 Bytes) and are representing raw sensor measurements (e.g., temperature, humidity etc). The Consumer node is sending 1000 Interest packets one-by-one to fetch the generated IoT contents from the Producer node. These requests follow the Zipf distribution [18] (with $\alpha$=1.5). We set the Freshness Period to 10 seconds, which is the time that the cached content is valid. The Consumer and Producer applications are based on the *ndnpeek* and *ndnpoke* tools, respectively. Fig. 2 illustrates a screenshot of a running experiment, showing the output of the Consumer (left terminal, Node9-13) and the SDN-NDN Controller (right terminal, Node9-3).

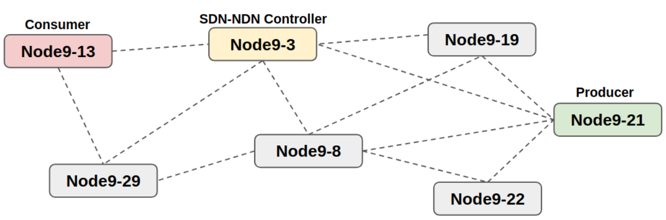


Fig. 3. w-iLab.1 Wireless Mesh Network Topology

The nodes are equipped with an AR9462 wireless network adapter that we used to construct our wifi mesh network. We used the *ath9k* driver and configured these devices to run at 2.4 GHz with 20 MHz channel width in mesh mode. Although the selected devices were deployed in close proximity, we adjusted the transmission power of the wireless nodes to 3 dBm and managed to shape a multi-hop scenario with at least 3 hops. The available links are illustrated in Fig. 3.

### B. Metrics

The evaluation of the system includes three metrics: (i) the performance of the NDN network in terms of communication delay (i.e., the Interest-Data exchange procedure between the Consumer and the Producer); (ii) the path establishment delay, i.e., the elapsed time at which the Controller makes the best path decision and configures the NDN network; and (iii) the performance of SDN-NDN dynamic decision making over the WMN. The following is a detailed description of the metrics used.

- $RTT_{NDN}$**:** this metric is obtained from the NDN Consumer node and represents the Round Trip Time (RTT) between the Interest packet transmission and the Data packet reception. The Consumer can fetch the requested content either from the Producer node or from the on-path nodes, due to the in-network caching. We can further analyze this particular metric to:

$$RTT_{NDN} = RTT_{NDN_c} + RTT_{NDN_p} \quad (1)$$

  Where $RTT_{NDN_c}$ corresponds to the RTT values in cache-hit cases , while $RTT_{NDN_p}$ corresponds to the measured RTT in case of producer-fetched content.
- *Total Delay:* This corresponds to the elapsed time between the Consumer's request to the Controller until the Data packet reception, including the path establishment delay.
- *Best Path Changes (BPC)*: represents the total number of best path changes in the whole duration of the experiment.

### C. Proof-of-concept Results

The experimental results are summarized in Figures 4-5. We conduct 10 repetitions of the experiment and illustrate the corresponding average values, i.e., for statistical validity purposes.

In Figure 4, we present the $RTT_{NDN}$ and *Total Delay* results that were measured during our experimental procedure. These delay metrics mainly impact our system's performance and reflect the latency capabilities of each functional entity. Derived from the average $RTT_{NDN}$ results, we can pose that our system enables the seamless operation of NDN in volatile network topologies, without compromising NDN performance. $RTT_{NDN_c}$ values (roughly 4.48 msec) were significantly smaller than the $RTT_{NDN_p}$ (roughly 91.45 msec) which is explained because in our case the cache hits occurred locally in the Consumer node, without requiring any network transmission. In contrast, fetching Data from the remote Producer in our experiment required Interest and Data packet transmissions over the 3-hop topology. Our future work plans include experimenting with the caching capabilities of our system in large-scale topologies with many consumers and producers.

The average *Total Delay* (roughly 1.3 seconds) was significantly larger than the average $RTT_{NDN}$ (milliseconds order of magnitude). Obviously, this is justified by the additional network control overhead introduced by the Controller, as detailed in Section III-A. In this context, we identify the relevant trade-off: our system enables the efficient NDN communication in mesh networking environment, but introduces additional control overhead. However, there is space for further optimization, e.g., employ a proactive path selection strategy, rather than reactive.

Figure 5 shows the average best-path's hops and the total *Best Path Changes (BPC)* that occured per round. In every round, the average number of hops has been preserved to 3 hops, confirming the validity of the implemented experimental methodology. Additionally the *Best Path Changes* deviation (i.e., ranging from 7 to 19) shows the wireless links volatility that were present in our test-bed setup. At the same time, it affirms our system's capability to capture frequent network changes and establish the appropriate paths.

Furthermore, Figures 4-5 jointly demonstrate the adaptability of our system to volatile network topologies. In particular, we can conclude that our proposed system can mitigate potential path changes and render NDN capable of outperforming a static NDN network, in a similar volatile topology, i.e., the latter cannot handle connection disruptions.

## V. Conclusions and Future Work

In this work, we presented an integrated SDN-NDN approach to facilitate seamless operation of NDN in a WMN environment. Our proposed system is deployed in the w-iLab.1 Fed4FIRE+ test-bed, enabling the experimentation over a real wireless mesh environment. Our proof-of-concept results demonstrate the deployability of our solution and validate that the centralized control of SDN effectively supports the

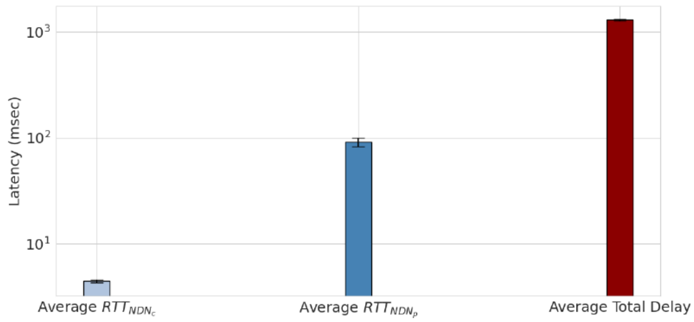


Fig. 4. Average $RTT_{NDN_c}$, $RTT_{NDN_p}$ and *Total Delay*.

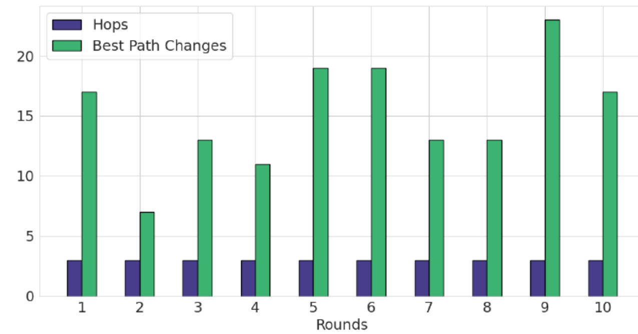


Fig. 5. Number of Hops and *Best Path Changes (BPC)* pers round.

NDN operation in unstable topologies with frequent dynamic changes, such as the WMNs.

We plan to carry out further investigations and extend the capabilities of our approach. These include:

- Extensive experimentation analysis and performance evaluation of our approach in more complex experimental scenarios (e.g., large-scale wireless mesh deployments, multiple Consumers and Producers).
- Extension of the Controller capabilities improving its decision making, taking into account additional NDN-related information (e.g., caching information of the intermediate network nodes, communication with the Producer).
- Evaluation of our approach over challenging network environments (e.g., signal interference of crowded areas, outdoor wireless conditions, long-distance), such as Smart-City networks. We plan to conduct experiments in the City-Lab test-bed [11] which provides real Smart-City conditions.
- Adopting a multi-protocol solution, i.e., also supporting on-demand Delay/Disruption Tolerant Networking (DTN), in order to provide further reliability support in highly-disruptive networking conditions (e.g., mobility, prolonged delays, frequent disruptions, link failures). In this context, we can also enable the seamless NDN operation in such challenging networking conditions with the NDN-over-DTN stack [19].

## Acknowledgment

This work has received funding from the EU's Horizon 2020 research and innovation programme through the 9th open call scheme of the Fed4FIRE+ project (grant agr.num. 732638).

## References


[1] J. A. Stankovic, "Research directions for the internet of things," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 3–9, 2014.

[2] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, K. Claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang, "Named data networking," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, pp. 66–73, 2014.

[3] 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, 2014.

[4] A. Aboodi, T.-C. Wan, and G.-C. Sodhy, "Survey on the incorporation of ndn/ccn in iot," *IEEE Access*, vol. 7, pp. 71827–71858, 2019.

[5] C. Yi, A. Afanasyev, I. Moiseenko, L. Wang, B. Zhang, and L. Zhang, "A case for stateful forwarding plane," *Computer Communications*, vol. 36, no. 7, pp. 779–791, 2013.

[6] M. Selimi, L. Navarro, B. Braem, F. Freitag, and A. Lertsinsrubtavee, "Towards information-centric edge platform for mesh networks: The case of citylab testbed," in *2020 IEEE International Conference on Fog Computing (ICFC)*, pp. 50–55, 2020.

[7] E. Aubry, T. Silverston, and I. Chrismen, "Implementation and evaluation of a controller-based forwarding scheme for ndn," in *2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA)*, pp. 144–151, 2017.

[8] M. Alhowaidi, D. Nadig, B. Ramamurthy, B. Bockelman, and D. Swanson, "Multipath forwarding strategies and sdn control for named data networking," in *2018 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, pp. 1–6, 2018.

[9] M. Amadeo, C. Campolo, G. Ruggeri, A. Molinaro, and A. Iera, "Understanding name-based forwarding rules in software-defined named data networking," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, pp. 1–6, 2020.

[10] "Wireless Testlab and OfficeLab — imec iLab.t documentation." https://doc.ilabt.imec.be/ilabt/wilab/. Accessed: 2022-01-15.

[11] J. Struye, B. Braem, S. Latré, and J. Marquez-Barja, "The citylab testbed — large-scale multi-technology wireless experimentation in a city environment: Neural network-based interference prediction in a smart city," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 529–534, 2018.

[12] Q.-Y. Zhang, X.-W. Wang, M. Huang, K.-Q. Li, and S. K. Das, "Software defined networking meets information centric networking: A survey," *IEEE Access*, vol. 6, pp. 39547–39563, 2018.

[13] "BATMAN Concept - open-mesh - open mesh." https://www.open-mesh.org/projects/open-mesh/wiki/BATMANConcept. Accessed: 2022-01-15.

[14] "NFD Overview — Named Data Networking Forwarding Daemon (NFD) 0.7.1 documentation." https://named-data.net/doc/NFD/current/overview.html. Accessed: 2022-01-15.

[15] "Empowering App Development for Developers — Docker." https://www.docker.com/. Accessed: 2022-01-15.

[16] "Wiki - alfred - Open Mesh." https://www.open-mesh.org/projects/alfred/wiki. Accessed: 2022-01-15.

[17] "jFed." https://jfed.ilabt.imec.be/. Accessed: 2022-01-15.

[18] S. K. Fayazbakhsh, Y. Lin, A. Tootoonchian, A. Ghodsi, T. Koponen, B. Maggs, K. Ng, V. Sekar, and S. Shenker, "Less pain, most of the gain: Incrementally deployable icn," in *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM*, SIGCOMM '13, (New York, NY, USA), p. 147–158, Association for Computing Machinery, 2013.

[19] C. A. Sarros, V. Demiroglou, and V. Tsaoussidis, "Intermittently-connected iot devices: Experiments with an ndn-dtn architecture," in *2021 IEEE 18th Annual Consumer Communications Networking Conference (CCNC)*, pp. 1–9, 2021.