Adaptive NDN, DTN and NoD Deployment in Smart-City Networks Using SDN

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*
{emamatas}(at)uom.edu.gr

2023 IEEE 20th Consumer Communications & Networking Conference (CCNC) | 978-1-6654-9734-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/CCNC51644.2023.10060803

***Abstract*—Internet of Things (IoT) evolution transformed modern cities into smart cities. As smart cities are becoming part of our everyday lives, it becomes essential to address challenging communication issues that are present, such as the intermittent connectivity, the link asymmetry and the indirect line-of-sight. In this work we propose an adaptive solution that employs interchangeably Named-Data Networking (NDN), Delay/Disruption-Tolerant Networking (DTN) and NDN-over-DTN (NoD) to compensate reliability issues and enable the seamless smart city network operation using the Software-Defined Networking (SDN) technology. The implemented system has been deployed and validated in the CityLab test-bed, a real smart city network located in Antwerp, Belgium. The evaluation results show that our adaptive protocol solution outperformed the single NoD scheme in terms of Content Retrieval Delay, Interest Satisfaction Ratio and Cache Hit Ratio.**

***Index Terms*—Information-Centric Networking, Named Data Networking, Delay-Tolerant Networking, Software-Defined Networks, Smart Cities**

I. INTRODUCTION

The advent of Internet of Things (IoT) [1] has enabled automation, monitoring and actuating activities in modern cities, leading the smart city paradigm to maturity [2]. Nowadays, this paradigm is associated with the city residents quality of life. Thus, significant attention is raised in order to preserve smart city's robustness and reliability.

Smart city networks involve dense IoT deployments and benefit from wireless mesh technology in terms of node reachability and indirect connectivity in a multi-hop way. However, communication in the urban environment is often disrupted due to: (i) intermittent connectivity (e.g., signal interference, link failures, prolonged delays, mobility); (ii) indirect line-of-sight; and (iii) bandwidth asymmetry [3].

The efficient and seamless operation of smart city networks, under reliable or unreliable connectivity conditions, calls for novel networking architectures. To this end, Named Data Networking (NDN) [4] has been already deployed in smart city environments [3] and contributed to reduced network traffic thanks to its in-network caching and content naming scheme. However, the default NDN architecture faces difficulties in intermittent connectivity environments due to the breadcrumbs routing limitation [5]. An elegant approach to combat unreliable connectivity in disruptive IoT networks is to graft the Delay/Disruption-Tolerant Networking (DTN) architecture [6] in NDN, and form the novel NDN-over-DTN (NoD) scheme [7]. However, NoD's large packet size may create redundant network overhead even in reliable network conditions, where the default NDN could operate adequately.

Thus, the dynamically changing network conditions of the urban environments call for adaptive deployment of NDN and NoD. Yet, this is a rather challenging task to be accomplished in a distributed manner because: (i) individual nodes are not aware of the global network conditions; (ii) alternating protocol stacks individually may lead to incompatible communication stacks; and (iii) self-configuration and self-adaptation can potentially introduce incompatibilities among homogeneous protocol stacks (e.g., incompatible protocol configuration).

In this work we present an SDN-based approach to enable the adaptive NDN, DTN and NoD deployment in smart city networks. This is done by: (i) collecting network monitoring results; (ii) performing centralized decision making (e.g., distinguishing between reliable and unreliable connectivity); and (iii) activating and configuring the appropriate protocol stack. The main contributions of this work include:

- The implementation of a multi-protocol SDN solution for smart cities, featuring: (i) Network monitoring; (ii) Decision Making to distinguish reliable and unreliable network conditions and to select the most reliable path and (iii) on-demand NDN, DTN and NoD protocol deployment and configuration.
- The evaluation of adaptive NDN, DTN and NoD deployment in real smart city environment [8] and their performance comparison against a non-adaptive NoD scheme deployment. To the best of our knowledge, this is the first work that combines these stacks and enables their adaptive deployment in a real-world smart city test-bed.

Our real-world experimental results: (i) validate the Controller's deployability and its ability to detect accurately network connectivity conditions (e.g., reliable and unreliable); (ii) show the superiority of the adaptive system compared to the non-adaptive NoD deployment, in terms of the Content Retrieval Delay, the Interest Satisfaction Ratio and the Cache Hit Ratio.

978-1-6654-9734-3/23/$31.00 ©2023 IEEE

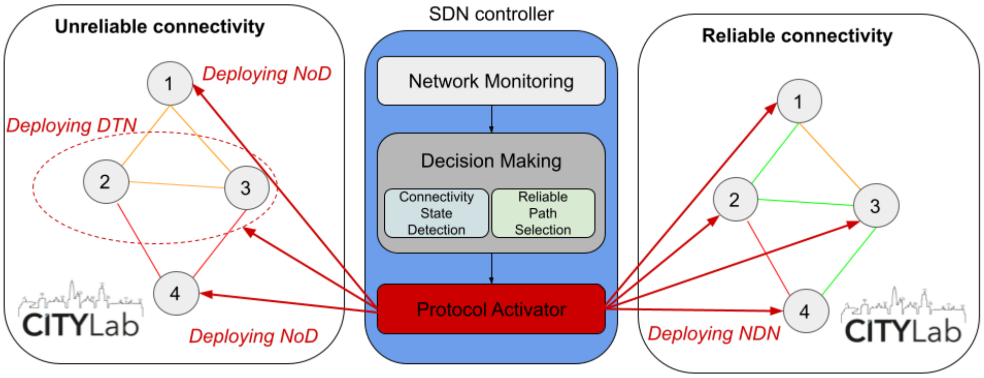


Fig. 1: Proposed System

II. RELATED WORK

The smart cities radical advances and applications in conjunction with the challenging and dynamic network conditions and the increased number of deployed IoT devices call for novel networking architectures and multi-technology solutions.

NDN, an Information-Centric Networking (ICN) architecture [9], constitutes a promising IoT networking approach suitable for rapid content retrieval as a consequence of the content naming and the in-network caching features [10]. Along these lines, NDN is considered an ideal match for smart cities [11].

In [3], authors presented an ICN deployment in the CityLab test-bed and highlighted critical intermittent connectivity issues. Their work is not focused explicitly on the communication performance under highly volatile network topologies, as in our work, thus they utilized a static routing approach.

Nevertheless, the default NDN architecture is unable to operate efficiently in volatile environments that involve mobility, link failures and unstable topologies due to the breadcrumbs routing limitation (i.e., Data packets are forced to return from the reverse path that the respective Interest packets has been sent). This limitation can severely impact the performance of the smart city applications thus alternative NDN extensions and mechanisms have been employed.

In [12] authors presented the benefits of ICN and cognitive radio network combination for content distribution in smart cities. In particular, they claim that the cognitive radio networks can address the noisy channels and signal interference issues and boost ICN's reliability in smart city networks through the anypath routing approach. Although they do not include any implementation, they present some nice ideas and future research directions about this combination.

In [13], authors present and evaluate a scheme that enhances NDN resiliency for disaster scenarios in smart metropolitan cities. This approach performs disaster detection with sensors and uses the Satisfied Interest Table (SIT) along with push-based messages to enable NDN operation in limited or lost network connectivity circumstances. In [14] authors presented a publish-subscribe NDN mechanism to support Consumer and Producer mobility and showed the superiority of their solution against the native NDN in a simulated smart city scenario. These works involve some novel ideas, which have been implemented and evaluated in simulation environments - not in real smart city networks, as is the case of our work.

A novel solution to address this limitation is introduced in the UMOBILE project [15], which inspired the realization of the NDN-over-DTN (NoD) scheme [7]. In this approach, Interest and Data packets are encapsulated into DTN bundles and are able to follow any available path. This, along with the store-carry-and-forward feature of the DTN architecture can provide reliability and robustness to smart city networks in case of unreliable connectivity. However, NoD encapsulation leads to increased packet size, which may not be always tolerant. In our work we selected to utilize the NoD approach as it: (i) enables the seamless NDN operation in volatile topologies; (ii) has been already deployed and tested in real-world experiments; (iii) enables the functional autonomy of NDN and DTN, which is essential for achieving the adaptive protocol deployment with reduced communication overhead; and (iv) includes DTN technology, which has been deployed in other smart city test-beds (such as in [16]) to support data forwarding in vehicular networks.

In our previous work [17], we enabled NDN operation over the wireless mesh networks through the SDN logic. However, dealing with intermittent connectivity and long delays in the urban environment requires the involvement of DTN and NoD, which are specifically designed for highly disruptive networks.

Unlike other related works, that are using single NDN protocol solutions, this work employs interchangeably NDN, DTN and NoD according to the network connectivity state. Furthermore, this work validates the robustness of the proposed system with experiments in a real smart city test-bed, compared to relevant works that are performing simulations.

III. PROPOSED SYSTEM

The presented system consists of the smart city network nodes and the SDN Controller, as depicted in Figure 1. The overall goal is to ensure seamless and reliable smart city communication.

A. *Smart City Network Nodes*

The Smart City Network Nodes have a key role for the communication and the reliable operation of our system, since they constitute the city's communication backbone. Towards this end, these nodes are featuring three distinct functionalities:

- *The Wireless Mesh Connectivity*: The nodes mobility and large distances, the lack of direct line-of-sight and the strong signal interference present in the city networks, are causing nodes unreachability. The wireless mesh network technology, allows nodes to extend their communication range and even to reach undirect nodes.
- *Performance Measurements*: Each individual node conducts periodically performance measurements with the adjacent network nodes, in order to collect connectivity information. The results are sent to the Controller in order to update the global network state.
- *The NDN, DTN and NoD Protocol Stacks*: A seamless network operation in challenging connectivity conditions requires the adaptable NDN, DTN and NoD deployment in the smart city network. Therefore, the nodes have the ability to run each time one of the protocol stacks according to the received control messages.

B. *Multi-protocol SDN Controller*

Global network view, dynamic network monitoring, connectivity status determination and adaptive protocol deployment are all realized by a multi-protocol SDN Controller mechanism that encompasses three distinct components:

- *Network Monitoring*: The SDN Controller collects information from each network node regarding its connectivity with the adjacent nodes. In this manner, the Controller maintains bidirectional information about each link, which is important due to link asymmetries in smart cities [3]. Additionally, the centralized collection of these information enables the Controller to capture a global network view and state.
- *Decision Making*: According to the centralized monitoring results the Controller categorizes each link as reliable or unreliable, based on a threshold value. Next, the Controller detects the global network connectivity state (i.e., *reliable state*: if there is at least one reliable path that interconnects the Consumer and the Producer, else *unreliable state* as shown in Figure 1). The latter is performed by the *Connectivity State Detection* subcomponent. Furthermore, in case of reliable state the *Reliable Path Selection* subcomponent determines the most reliable path between the Consumer and the Producer node.
- *Protocol Activator*: According to the network connectivity state, provided from the *Decision Making* module, the Controller deploys and configures the appropriate protocol stack to each network node. In case of unreliable connectivity the Controller deploys the NoD protocol stack at the edge of the Consumer and the Producer network region and DTN to the unreliable part of the network, as shown in Figure 1. In this manner, we increase the network reliability by exploiting the store-carry-and-forward approach of DTN. Otherwise, in case of reliable connectivity, the *Protocol Activator* deploys the NDN protocol at each node participating in the most reliable path and configures them accordingly.

IV. SYSTEM'S COMMUNICATION WORKFLOW

In this section we elaborate on the system's communication workflow, which can be divided in three phases: (i) Content request update; (ii) Protocol deployment and configuration; and (iii) Consumer confirmation.

In the first phase, the Consumer sends to the Controller a message, that contains the content it wishes to retrieve. Since NDN communication is Consumer-driven it is necessary to inform the Controller about the particular content prefix in order to configure the network nodes, accordingly. Afterwards, in the second phase, the Controller deploys and configures the appropriate protocol stacks to the network nodes, according to the *Network Monitoring* and the *Decision Making* results. Finally, in the third phase, the Controller triggers the Consumer to send the Interest packet. The Interest-Data exchange is concluded when the Data packet is retrieved.

A relevant example of the second phase is illustrated in Figure 1. We assume that node1 is the Consumer and node4 is the Producer. On the left side (e.g., timeframe=1) connectivity in the network is deemed unreliable by the Controller who chooses to deploy NoD at the Consumer and the Producer, and DTN at the intermediate nodes. After some time (e.g., timeframe=3) as shown on the right side, the Controller determines a reliable path in the network (e.g., node1-node2-node3-node4), deploys NDN, and configures the participating nodes accordingly.

V. EVALUATION

To assess the effectiveness of the proposed system we had to compare it with a non-adaptive NoD approach (i.e., we deployed NoD at the edge of the reliable regions and DTN at the unreliable part of the network throughout the experiment). The assessment of these approaches in a real-world smart city environment required a thorough experimental methodology and setup along with the proper performance metrics selection.

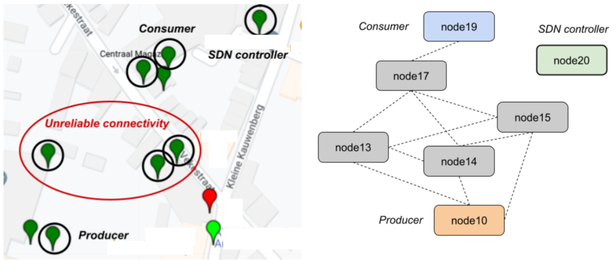
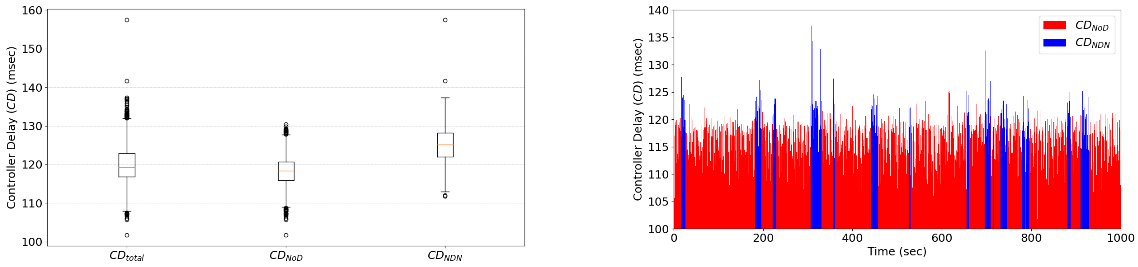


Fig. 2: Location and topology of CityLab test-bed's selected nodes



(a) Boxplots of CD_{total}, CD_{NDN} and CD_{NoD} values.

(b) CD_{NDN} and CD_{NoD} values and protocol transitions per time.

Fig. 3: Controller Performance.

A. Experiment setup

Our experiments are conducted in the CityLab testbed [8], which is located in Antwerp, Belgium. We selected the particular testbed because it offers: (i) real-world environment that reflects accurately the challenging communication conditions of smart city networks; (ii) citywide wireless mesh network experimentation infrastructure; (iii) high-end IoT nodes capable of running Linux, as well as the NDN, DTN and NoD implementations.

To this end, we selected seven nodes of the CityLab testbed, as shown in Figure 2 and created first a wireless mesh network. These nodes are featuring PCEngines apu2c4 boards equipped with a Compex WLE900VX network adapter, which contains a Qualcom Atheros QCA9880 wireless chipset. We used the *ath9k* driver and configured these nodes to run at 2.4 GHz (channel 1: 2412 MHz and width 20 MHz). We reduced the transmission power of node10 to 20 dBm and node17 to 15 dBm in order to hinder the direct connectivity of these nodes and to ensure that the Consumer and the Producer nodes communicate through the intermediate nodes.

Besides the actual delays and disruptions that are present in the CityLab test-bed, we introduced additional delay to the intermediate part of the network (e.g., node13, node14 and node15) in order to reproduce some unreliable connectivity conditions. In particular, we utilized the Linux Traffic Control (*tc*) tool and introduced additional delay ranged from 0 to 3000 msec using normal distribution (1500 msec mean value). In this manner, we created unstable network conditions in the intermediate network (e.g., alternating between reliable and unreliable state) that allowed us to validate the responsiveness of the proposed system.

The selected City-Lab nodes form a multi-hop wireless mesh network as illustrated in Figure 2. We consider an IoT Consumer node (node19) that is requesting emulated sensor measurements from an IoT Producer node (node10). The Consumer node is sending 1000 Interest packets, at a rate of 1 Interest/second and the requests follow a Zipf distribution [18] (with α=1.5), typically matching the IoT content popularity. The freshness period of each Data packet is set to 10 seconds, in order to reflect the impact of transient IoT content. In this light, we approached a real-life smart city scenario that encompasses realistic IoT traffic as well as content validity and popularity. We conducted 5 repetitions of each experiment in order to ensure the statistical validity of our results.

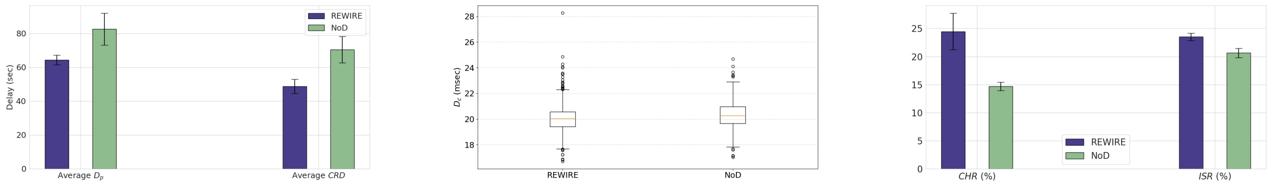
Regarding the NoD scheme experiment (i.e., the non-adaptive solution), we deployed NoD on nodes 17 and 10, DTN on nodes 13, 14, 15 and NDN on node19. The DTN deployment in the intermediate part of the network was necessary in order to enable the seamless network operation in terms of the introduced delays and disruptions. Thus, the in-network caching feature is supported by the Consumer node (node19) and the NoD gateway (node17) only.

In the adaptive protocol deployment, called the REWIRE experiment, the Controller node (node20) communicates with the smart city nodes using an ethernet interface. This setup was deemed appropriate because of the introduced delay in the network (i.e., a delayed control message might cause inappropriate protocol deployment). Furthermore, the adaptive protocol deployment is achieved using a containerized implementation of the NoD scheme, which enables the functional autonomy of the NDN and the DTN stacks and thus allows for efficient protocol activation, reconfiguration and deactivation. In particular, in case of reliable path detection, which is based on the links' packet loss monitoring results, the Controller deploys and configures the NDN to the participating nodes, otherwise uses the same deployment as in the NoD scheme experiment. At this stage, the implemented *Connectivity State Detection* component is threshold-based (i.e., based on a 50% threshold), for simplicity. In the near future, we plan to investigate monitoring employing change-point detection techniques.

B. Metrics

The purpose of our experiment is to validate the proposed system and compare it with a non-adaptive NoD approach. To do so, we used appropriate performance metrics to reflect the effectiveness and reliability of each particular approach. More precisely, our assessment includes three metrics:

- *Delay*: Our system performance is highly influenced by: (i) the *Controller Delay (CD)*, that corresponds to the delay between the Controller receiving a Consumer's request until the time the Controller triggers the Consumer



(a) Average D_P and Content Retrieval Delay (CRD) values. (b) Boxplots of Caching Delay (D_C) values. (c) Interest Satisfaction Ratio (ISR) and Cache Hit Ratio (CHR) results.

Fig. 4: Smart City Network Nodes Performance.

to send the Interest packet. This can be further analyzed as $CD_{total} = CD_{NDN} + CD_{NoD}$, where CD_{NDN} and CD_{NoD} are the Controller delays that were measured when configuring and deploying NDN or NoD to the smart city nodes, respectively and CD_{total} is the total Controller Delay; (ii) the *Content Retrieval Delay (CRD)*, that corresponds to the time elapsed between sending an Interest packet until receiving the respective Data packet. This can be analyzed as $CRD_{total} = D_P + D_C$, where D_C is the caching delay, i.e., the delay of retrieving a content from the local cache and D_P is the delay of retrieving a content form the Producer. Thus the total system's delay can be described as:

$$Delay_{total} = CD_{total} + CRD_{total} \tag{1}$$

- *Interest Satisfaction Ratio (ISR)*: is defined as $ISR = SatisfiedInterests/SentInterests$, where $SatisfiedInterests$ is the number of Interests that retrieved a Data packet while $SentInterests$ is the Consumer's total Interest transmissions.
- *Cache Hit Ratio (CHR)*: is defined as $CHR = CacheHits/SatisfiedInterests$, where $CacheHits$ is the total number of cache hits.

C. Results

1) Controller Performance: The results for the Controller performance are summarized in Figures 3a-3b. In our experiments the communication between the Controller and the smart city network nodes was performed using the ethernet interface. This was deemed necessary in order to perform on-time the protocol deployment and configuration and reflect the impact of the adaptive mechanism, accordingly. Therefore, the mean CD_{total} value that corresponds to the REWIRE experiment (e.g., includes NDN and NoD protocol switching) is 119 msec, as shown in the respective boxplot of Figure 3a.

Comparing separately CD_{NDN} with the CD_{NoD} values we can conclude that the NDN deployment and configuration requires more time than NoD (i.e., 125 vs 118 msec, mean values). This is because in the case of NDN the Controller has to configure all the participating nodes. Instead, in the NoD case, the Controller configures the NoD and DTN protocols to all the nodes once and afterwards reconfigures only the NoD gateway (e.g., to utilize the DTN face for for every new content prefix).

Comparing the Figures 3a and 4a, we notice that the Controller Delay (CD) of our system was significantly smaller than the Content Retrieval Delay (CRD) (i.e., milliseconds vs seconds). Thus, in our experiments the contribution of the Controller Delay to the total system's delay, as described in Equation 1, can be considered negligible. However, in case of a wireless control channel, an adaptive approach could introduce additional delay overhead in the network, as was also indicated in [17].

Figure 3b shows the CD_{NDN} and CD_{NoD} values as well as the protocol transitions per time unit. The threshold-based detection mechanism captures network connectivity changes with high sensitivity, in order to validate the adaptive system's performance. Hence, Figure 3b reflects the proposed system's robustness and responsiveness, which are essential requirements for the reliable smart city network operation.

2) Smart City Network Performance: The results of our experiments are illustrated in Figures 4a-4c. In Figure 4a we present the Average Delay and D_P values. We observe that the proposed system achieved significantly smaller values in both metrics compared to the NoD approach. Regarding the D_P values, REWIRE succeeded 64.352 sec (mean value), which is almost 22% less than the NoD case (82.537 sec mean value). This is justified because REWIRE utilized NDN in case of reliable network conditions and thus achieved less communication overhead, compared to the NoD approach that uses bundle encapsulation under all circumstances. Furthermore, concerning the Average Delay values, we observe an even bigger decrease of 30.8% (REWIRE: 48.696 sec and NoD: 70.406 sec), which is attributed to the better caching exploitation of the REWIRE approach. Specifically, smaller D_P values are leading to more cache hits, because the content is already stored in the node caches. In contrast, larger D_P values are causing previously cached content to expire, as the expiration time of the cached content (i.e., the freshness period) in our case is limited to 10 seconds. The latter observation can be examined in conjunction with Figure 4c that depicts the Cache Hit Ratio results.

In Figure 4b we examine the impact of each approach on the caching delay. Although the two approaches are causing comparable caching delays, the REWIRE's mean D_C appears slightly smaller (20.08 msec) than the NoD case (20.35 msec).

This is related to the NDN implementation and may be attributed to the increased Interest packet transmission in the NoD case (i.e., as the transmissions are increased the caching delay is also increased [7]).

In Figure 4c we present the ISR and CHR results of each approach. REWIRE achieved better performance with both metrics. It achieved 13.9% increased ISR compared to NoD scheme, which is associated with less Interest packet transmissions. The Controller can detect reliable network conditions and deploy NDN and thus facilitates rapid content retrieval (e.g., due to the smaller packet size and the more lightweight implementation). This causes less Interest packet transmissions, since the content is retrieved prior to the Interest lifetime expiration. Furthermore, REWIRE achieved 66.8% increased CHR compared to NoD, which is associated with smaller REWIRE's D_P values that lead to more cache hits.

VI. CONCLUSIONS AND FUTURE WORK

We presented an adaptive NDN, DTN and NoD solution based on SDN logic to facilitate reliable smart city network operations. Our real-world evaluation results showed that adaptive protocol deployment outperforms the non-adaptive NoD approach, in terms of Content Retrieval Delay, Cache Hit Ratio and Interest Satisfaction Ratio. Also, we showed that the implemented Controller detects the network changes and performs on-demand protocol deployment, accordingly.

Since the purpose of this study was to investigate the potential applicability of the adaptive protocol deployments, we aim to extend this work in several directions including:

- Additional Controller components such as: (i) caching prediction mechanism in order to facilitate the content retrieval procedure and lower the communication overhead; (ii) alternative control channel support, as in [19], including separate control and data channel; and (iii) anomaly detection mechanism [20] that can distinguish between reliable and unreliable time periods.
- Large-scale experimentation in the CityLab test-bed in order to evaluate scalability aspects of the proposed solution as well as experiments with mobile nodes.
- Investigation of the SDN control overhead and performance comparison of our solution with other multi-protocol SDN Controllers, such as [21].

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] Y. Mehmood, F. Ahmad, I. Yaqoob, A. Adnane, M. Imran, and S. Guizani, "Internet-of-things-based smart cities: Recent advances and challenges," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 16–24, 2017.

[3] 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.

[4] 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.

[5] 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.

[6] K. Fall, "A delay-tolerant network architecture for challenged internets," in *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '03, (New York, NY, USA), p. 27–34, Association for Computing Machinery, 2003.

[7] V. Demiroglou, C.-A. Sarros, and V. Tsaoussidis, "Nod: A content retrieval scheme for intermittently-connected iot networks," *Ad Hoc Networks*, vol. 130, p. 102825, 2022.

[8] 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.

[9] 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.

[10] 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.

[11] S. H. Bouk, S. H. Ahmed, D. Kim, and H. Song, "Named-data-networking-based its for smart cities," *IEEE Communications Magazine*, vol. 55, no. 1, pp. 105–111, 2017.

[12] A. Boukerche, R. W. L. Coutinho, and A. A. F. Loureiro, "Information-centric cognitive radio networks for content distribution in smart cities," *IEEE Network*, vol. 33, no. 3, pp. 146–151, 2019.

[13] O. A. Khan, M. A. Shah, I. Ud Din, B.-S. Kim, H. A. Khattak, J. J. P. C. Rodrigues, H. Farman, and B. Jan, "Leveraging named data networking for fragmented networks in smart metropolitan cities," *IEEE Access*, vol. 6, pp. 75899–75911, 2018.

[14] D. Hernandez, L. Gameiro, C. Senna, M. Luís, and S. Sargento, "Handling producer and consumer mobility in iot publish–subscribe named data networks," *IEEE Internet of Things Journal*, vol. 9, no. 2, pp. 868–884, 2022.

[15] C.-A. Sarros, S. Diamantopoulos, S. Rene, I. Psaras, A. Lertsinsrubtavee, C. Molina-Jimenez, P. Mendes, R. Sofia, A. Sathiaseelan, G. Pavlou, J. Crowcroft, and V. Tsaoussidis, "Connecting the edges: A universal, mobile-centric, and opportunistic communications architecture," *IEEE Communications Magazine*, vol. 56, no. 2, pp. 136–143, 2018.

[16] P. M. Santos, J. G. P. Rodrigues, S. B. Cruz, T. Lourenço, P. M. d'Orey, Y. Luis, C. Rocha, S. Sousa, S. Crisóstomo, C. Queirós, S. Sargento, A. Aguiar, and J. Barros, "Portolivinglab: An iot-based sensing platform for smart cities," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 523–532, 2018.

[17] S. Kalafatidis, V. Demiroglou, L. Mamatas, and V. Tsaoussidis, "Experimenting with an sdn-based ndn deployment over wireless mesh networks," in *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 1–6, 2022.

[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] T. Theodorou and L. Mamatas, "A versatile out-of-band software-defined networking solution for the internet of things," *IEEE Access*, vol. 8, pp. 103710–103733, 2020.

[20] S. Skaperas, L. Mamatas, and A. Chorti, "Real-time video content popularity detection based on mean change point analysis," *IEEE Access*, vol. 7, pp. 142246–142260, 2019.

[21] T. Theodorou, G. Violettas, P. Valsamas, S. Petridou, and L. Mamatas, "A multi-protocol software-defined networking solution for the internet of things," *IEEE Communications Magazine*, vol. 57, no. 10, pp. 42–48, 2019.