# IoT Data Collection Over Dynamic Networks: A Performance Comparison of NDN, DTN and NoD

Vassilis Demiroglou†, Michail Martinis†, Dimitra Florou†, Vassilis Tsaoussidis†
† *Department of Electrical and Computer Engineering, Democritus University of Thrace, Greece*
Email: {*vdemirog, vtsaousi*}@*ee.duth.gr* {*michailmartinis, dimitraflor*}@*gmail.com*

*paper accepted to IEEE WFIoT 2023

***Abstract*—Internet of Things (IoT) deployments experience dynamic wireless connectivity, due to nodes mobility and poor coverage. Therefore, data collection in such dynamic networks is challenging and can't be efficiently supported by traditional networks. Named Data Networking (NDN), Delay/Disruption Tolerant Networking (DTN) architectures and the combined NDN-over-DTN (NoD) scheme are endorsed by the research community to address this issue, since they provide performance and reliability benefits to dynamic IoT networks. In this work we present a comprehensive performance comparison of these solution, in the context of dynamic IoT networks. The emulation results showed that NDN is the most prominent solution for stationary networks with limited packet loss, DTN shows adequate reliability in all cases, while the combined NoD scheme prevails for highly-disruptive stationary and mobile networks.**

***Index Terms*—Internet of Things (IoT), dynamic networks, Named Data Networking (NDN), Delay-Tolerant Networking (DTN), performance comparison**

## I. Introduction

Internet of Things (IoT) [1] consists of devices that sense, process and communicate massive amount of data among themselves and with the Internet. IoT has been widely adopted in diverse application domains, such as environmental monitoring, smart cities and smart transportation. Given the exponential growth of deployed IoT devices, these IoT applications call for efficient data collection methods in order to cope with their latency and reliability requirements.

Such environments typically include wireless IoT devices and experience dynamic network conditions. Dynamic IoT networks can be formed both by stationary and mobile nodes [2]; stationary wireless networks may experience disconnected periods as a results of poor coverage or highly asymmetric links [3] and mobile wireless networks due to frequent connection changes [4]. IoT data collection in such challenging network conditions can't be efficiently supported by traditional networks, thus alternative communication paradigms are investigated.

Delay-/Disruption Tolerant Networking (DTN) [5] and Named Data Networking (NDN) [6] are two prominent architectures for supporting dynamic IoT communications. Specifically, DTN is a store-carry-and-forward architecture, designed for partially-connected environments, such as opportunistic and infrastructure-free networks [7]. NDN is an Information-Centric Networking (ICN) [8] architecture that improves the performance of IoT networks, thanks to its data naming scheme and in-network caching support. Furthermore, the NDN-over-DTN (NoD) scheme [9], is recently introduced as a promising IoT solution, that combines the benefits of NDN and DTN and facilitates content retrieval from communication-disruptive deployments [3], [10].

Although, many works are employing these architectures for supporting dynamic IoT networks [4], [7], [9], their distinct benefits and limitations have not been comparatively examined in this context. In this work, we fill this gap by individually assessing the performance, reliability and network overhead of each architecture over dynamic stationary and mobile IoT networks. The emulation results showed that NDN is the most prominent solution for stationary networks with limited packet loss, DTN shows adequate reliability in all cases, while the combined NoD scheme prevails for highly-disruptive stationary and mobile networks.

The remainder of this paper is structured as follows. In Sections II and III we provide brief background information of NDN, DTN and NoD and discuss related works. In Section IV we summarize the key properties of each approach for dynamic IoT networks. The experiment setup and evaluation results are presented in Section V. In Section VI we summarize the main conclusions and discuss future work.

## II. Background

In this section we provide background information about NDN, DTN and NoD operation.

### A. NDN

NDN [6] operation is based on the exchange of Interest and Data packets. These packets contain names, instead of addresses, in order to be identified [11]. Specifically, an NDN consumer sends one Interest packet that contains the name of the requested data (i.e., *ndn/temperature*), to the network. The Interest packet is forwarded in a hop-by-hop manner across the network until it reaches a node that has the requested data (i.e., data can be retrieved either from the original producer or any intermediate node that has previously retrieved this Data packet). When the Interest packet reaches this node, the respective Data packet is forwarded back to the consumer node, through the reverse path of the Interest.

To achieve this packet exchange, each NDN node maintains three data structures: a Content Store (CS) to cache incoming data locally, a Pending Interest Table (PIT) to store the forwarded Interest packets that have not been satisfied and a Forwarding Information Base (FIB) to keep information about where to forward Interests.

### *B. DTN*

DTN [5] architecture has been designed to allow communication in scenarios that end-to-end path is not always available. Specifically, it provides an overlay approach, called Bundle Protocol (BP), that allows heterogeneous underline networks to operate over delayed or intermittent connectivity. This is achieved by using convergence layer adapters (e.g., TCP, UDP) with the underline networks.

DTN enables nodes to store-carry-and-forward bundles until reaching the receiver node. Bundle destinations are identified using endpoint identifiers (EIDs), which follow a URI-based addressing scheme.

### *C. NoD*

NoD is a combined NDN-DTN scheme [9], [10] that supports the NDN-over-DTN stack [12]. Specifically, its design is inspired by the breadcrumbs routing limitation of NDN (i.e., each Data packet follows the reverse path of the respective Interest packet) that impedes its operation in volatile networks [4]. By encapsulating NDN Interest and Data packets to bundles, NDN operation is seamlessly supported in dynamic network scenarios, while preserving NDN and DTN compatibility.

Since NoD includes two distinct protocols, it features a more complex implementation and is designed as a gateway-oriented solution. In particular, NoD gateways being deployed at the edge of remote NDN networks, enable remotely-deployed NDN nodes to be interconnected through DTN data mules.

## III. RELATED WORK

NDN and DTN architectures have been extensively studied in the context of dynamic networks and IoT. In [11] and [13] authors quantify the performance of each approach and demonstrate their data retrieval benefits, considering similar performance metrics with our work. In the following, we present relevant studies that compare the performance of these architectures with other prominent solutions.

Many works focused on the performance comparison of DTN with TCP/IP solutions. Specifically, [14] focused on the evaluation of DTN in various disruptive networks, including stationary and mobile networks, however, is not an IoT-oriented work. In [7], authors investigated the employment of DTN for IoT data collection and showed its reliability benefits compared to TCP/IP. Beyond its novelty, this study focuses on the performance improvements of DTN-based data collection and does not emphasize on the protocols comparison aspect.

Another group of works compare the performance of NDN with host-centric approaches. In [15] authors introduced a performance comparison of NDN and epidemic routing. In particular, they employed the Distributed Dataset Synchronization in Disruptive Networks (DDSN) protocol as a transport mechanism of the default NDN. The results showed that NDN/DDSN retrieves content rapidly and efficiently than epidemic routing. However, DDSN involves broadcast Interest transmissions, which may not be the optimal solution in IoT environments, due to the increased network overhead. In [16] authors compared NDN with CoAP and MQTT, which are the most dominant host-centric IoT protocols. The results showed that the IP-based approaches are showing better performance over single-hop topologies, while NDN outperformed them over multi-hop topologies. Unlike [16], we emphasize on the performance assessment of native NDN in highly dynamic networks.

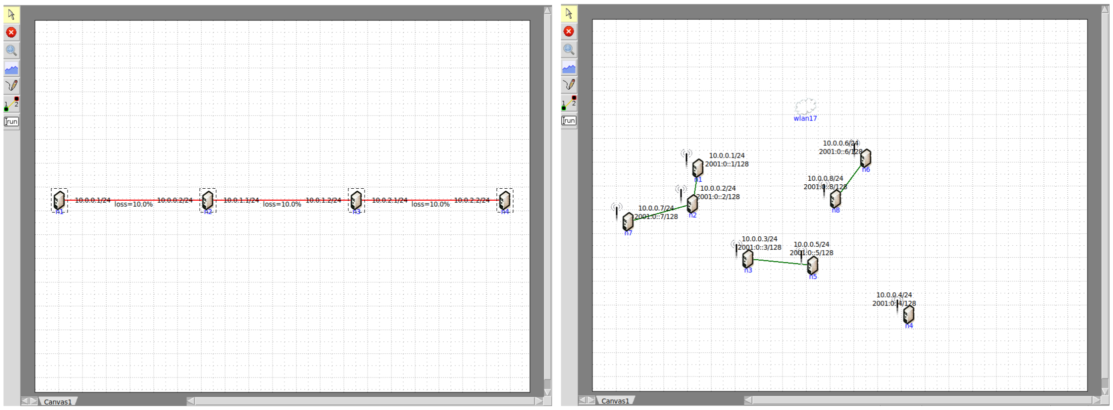
In our previous works [9] [10], we introduced NoD scheme and showed that it outperforms the DTN approach in data muling scenarios. Nonetheless those works: (i) have not compared NDN with NoD and DTN; (ii) include real-world experiments with limited number of nodes; and (iii) do not consider dynamic stationary networks. Furthermore, in [3] and [17] we showed that the adaptive NDN, DTN and NoD employment in stationary smart city networks outperforms NDN and NoD. However the performance benefits and limitations of NDN, DTN and NoD have not been quantified thoroughly in any of those works.

In this work, we fill these gaps by individually assessing NDN, DTN and NoD in dynamic stationary and mobile IoT networks. Specifically, we conduct emulation-based experiments that allow us to compare the performance, reliability and network overhead of each approach and unravel their distinct capabilities for varying number of nodes and diverse network conditions.

## IV. NDN AND DTN FEATURES FOR DYNAMIC IOT NETWORKS

Given the volatile network conditions and the IoT constraints, dynamic IoT networks require protocols able to handle intermittent connections in a communication-efficient way [2]. To emphasize on the suitability of NDN and DTN protocols in dynamic IoT networks, we analyze and compare their key related features bellow.

1) *Hop-by-hop Transfer*: The absence of end-to-end path in communication-disruptive environments calls for hop-by-hop approaches. Both architectures operate in a per-hop basis, however follow different routing schemes. Native NDN imposes the breadcrumbs routing limitation [4] that may degrade its overall performance in highly dynamic environments. In contrast, DTN supports one-way communications and its routing approaches bring versatility and robustness in volatile networks [18].
2) *In-network Storage*: Network-based storage capabilities are essential to enable IoT nodes to transmit data several times until reaching the next hop. Both architectures support built-in storage mechanisms that allow packets/bundles to be stored. NDN nodes can carry Interest and Data packets, to the PIT and CS, respectively, while



(a) Stationary network. (b) Mobile network.

Fig. 1: Considered dynamic networks emulated using CORE.

DTN nodes can permanently store bundles in their local persistent storage.

3) *Lifetime Parameters*: NDN and DTN lifetime parameters influences storage usage, retransmissions and communication efficiency [19]. After lifetime expiration nodes drop the carried Interest packet or bundle (some protocol strategies may perform retransmissions instead). Additionally, NDN Data packets have a freshness period, that corresponds to the content validity period (i.e., reflects for how long this packet can be considered fresh and remain stored at NDN nodes caches). These parameters should be configured according to network conditions and requirements, as well as to IoT data validity [19] (e.g., transient or persistent content).
4) *Naming Scheme*: NDN and DTN architectures offer distinct naming features [20] that bring application flexibility to IoT environments [21]. DTN uses EIDs to name endpoints and applications (e.g., *dtn://node2/pressure*). In this manner, DTN clients could determine the receiving node and its corresponding application. Contrary, NDN names content instead of endpoints and allows for improved data dissemination, compared to host-centric approaches.

Although each approach could be extended in order to enable non-supported features, in this work we focus on their default operation. Therefore, we consider NoD scheme as a relevant approach as it preserves the default NDN and DTN operations. The impact of grafting additional functionality to each individual protocol (e.g., employing cache in DTN) is beyond the scope of this work.

## V. Evaluation

To compare the performance of NDN, DTN and NoD over dynamic IoT networks, we consider two typical IoT scenarios: a stationary and a mobile network. In both scenarios we assume an IoT consumer that requests sensor measurements from an IoT producer. Specifically, the consumer node is sending 1000 requests (e.g., bundles or Interest packets) with a rate of 1 request/second. The content requests follow a zipf distribution (with $\alpha$=1.5), typically matching the IoT content popularity [19]. The producer node generates 150 different sensor measurements. In the following sub-sections we discuss the experiment set-up, the considered metrics and the obtained results.

### A. Experiment Setup

We conducted our experiments using the Common Open Research Emulator (CORE) [22] (version 7.5.2) that allows for the reproduction of the considered scenarios, as illustrated in Figure 1. We utilized a machine featuring an Intel Core i5-10210U CPU and 8GB of RAM and running *Ubuntu 20.04.4 LTS* to run our experiments.

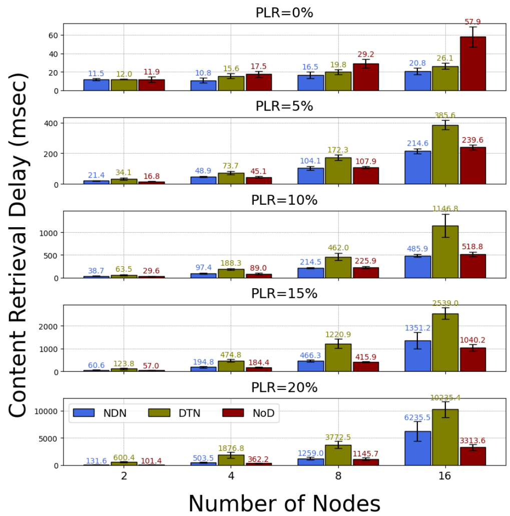
TABLE I: Average packet/bundle size of each protocol, as measured in our experiments

| | **Request size (Bytes)** | **Response size (Bytes)** |
|---|---|---|
| **NDN** | 28 | 352 |
| **DTN** | 72 | 75 |
| **NoD** | 83 | 409 |

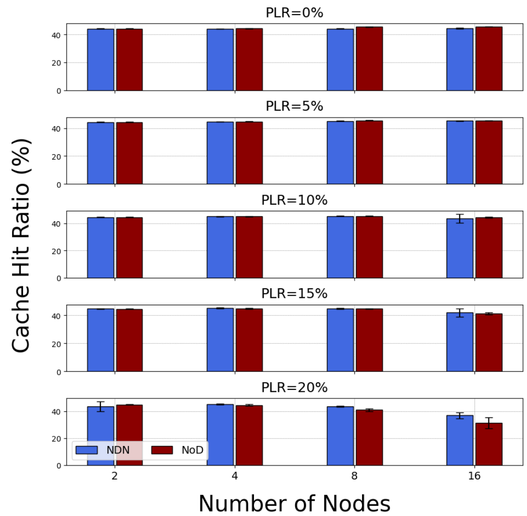
In both setups we set bandwidth to 54 Mbps to approach realistic wireless link performance and varied the number of nodes in the range of $[2, 4, 8, 16]$. We conducted 10 iterations of each experiment to guarantee statistically valid results.

We utilized the NDN Forwarding Daemon (NFD), IBR-DTN [23] and their combined NoD implementation [9], for running NDN, DTN and NoD, respectively. In all NoD experiments, we employed NoD scheme at the consumer and producer and DTN to all the intermediates nodes, as in [9]. In Table I, we summarize the average size of packet/bundle requests and responses of each approach, as measured in our experiments.

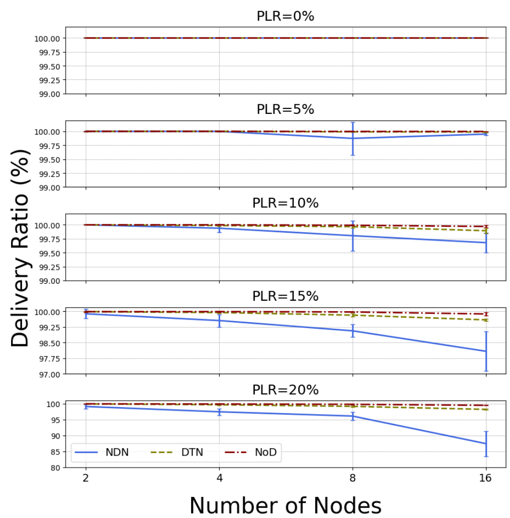
In the stationary network we designed a chain topology, as shown in Figure 1a, and varied the packet loss ratio (PLR) in the range of $[0, 5, 10, 15, 20]$ %. To obtain comparable results, we set the lifetime and freshness period to 10 seconds and



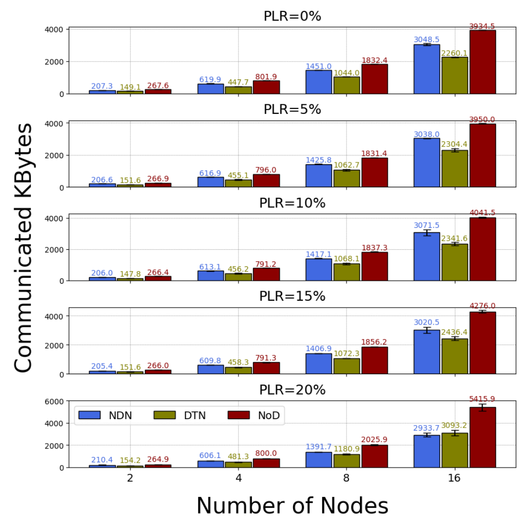

(a) Content Retrieval Delay (CRD).




(b) Cache Hit Ratio (CHR).




(c) Delivery Ratio (DR).




(d) Communicated KBytes.

Fig. 2: Stationary network results.

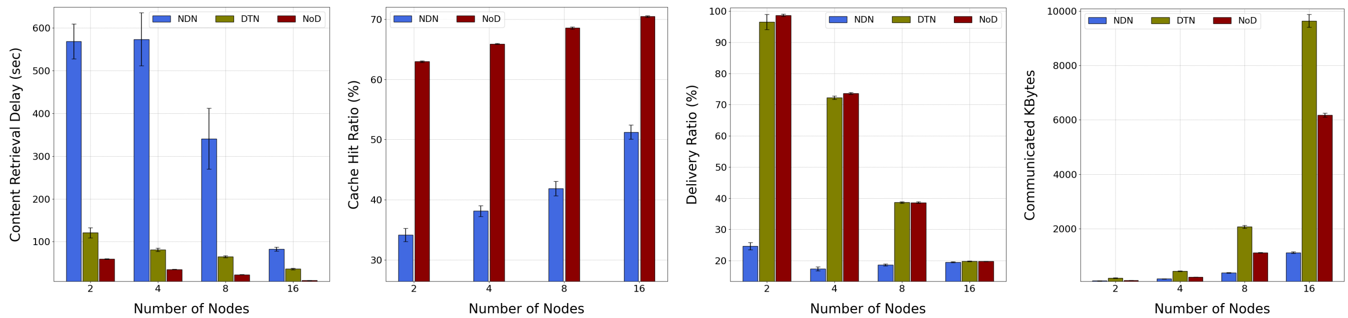
used the TCP face/convergence layer to all protocols. In this experiment we disabled the Interest aggregation feature of NDN, to approach equivalent request-response behaviour with DTN.

In the mobile network scenario, nodes are moving in an 650 m x 650 m area and communicate through a wireless interface (with a maximum communication range of 140 m). Specifically, nodes are moving based on the Random Walk model and change their direction and speed every 120 seconds. The speed ranges from 0.5 to 20 m/s. We generated this mobility model using BonnMotion (version 3.0.1) [24]. In this scenario all three protocols are configured to use UDP face/convergence layer, in order to reflect their performance without a reliable underlying transport mechanism. To exploit all possible paths and achieve comparable results, we configured NDN to flood Interest packets and used DTN flooding routing (in DTN and NoD experiments). Interest aggregation is enabled in this experiment, in order to minimize the communicated bytes of NDN and NoD. To avoid frequent retransmissions we set the lifetime and freshness period to 60 and 120 sec, respectively.

### B. Metrics

The following metrics are considered to evaluate the performance of each approach:

- **Content Retrieval Delay (CRD)**: Corresponds to the time elapsed from the transmission of a content request until the reception of the respective content by the IoT consumer.
- **Delivery Ratio (DR)**: We calculated this metric as the ratio between the total received and the total sent packets/bundles in the network.
- **Cache Hit Ratio (CHR)**: Corresponds to the ratio between the cache hits and the total number of retrieved




(a) Content Retrieval Delay (CRD). (b) Cache Hit Ratio (CHR). (c) Delivery Ratio (DR). (d) Communicated KBytes.

Fig. 3: Mobile network results.

content. This metric concerns only NDN and NoD, since DTN does not support in-network caching. To obtain comparable results CHR is measured at the consumer node, which is the only node that supports in-network caching in our NoD experiments.

- **Communicated KBytes**: Corresponds to the total generated Kbytes produced by each protocol.

### C. Results

*1) Stationary Network:* The results of the stationary network are summarized in Figure 2. Note that the illustrated error bars represent the standard deviation. Figure 2a depicts the CRD results. We observe that, in the majority of experiments, DTN accomplished increased CRD compared to NDN and NoD, which is mostly attributed to DTN's absence of in-network caching support.

In ideal conditions (i.e., 0% loss), NDN achieved lower CRD values than NoD due to the increased packet size of the latter. Under the loss rates of 5% and 10%, NoD showed reduced CRD for 2 and 4 nodes, but NDN outperformed NoD for increased number of nodes. Concerning the 15% and 20% loss, we observe that NoD outperforms NDN, thanks to the better disruption handling of the DTN layer. Note that in this experiment NDN and NoD achieved similar CHR values, as shown in Figure 2b.

In Figure 2c, we illustrate the Delivery Ratio of each approach. All protocols accomplished high DR values (i.e., roughly 100%), in cases of 0% and 5% loss. In the rest cases, we observe that: (i) NoD and DTN maintained increased reliability (i.e., no less than 99.5% and 98.2%, respectively); and (ii) NDN's DR is significantly reduced as the number of nodes increases. The latter becomes more apparent in the 20% PLR case, where 16 NDN nodes showed 11.6% decreased DR, compared to the single-hop case.

In Figure 2d, we present the Communicated KBytes of each approach. Despite the in-network caching support, NDN and NoD generated more Kbytes than DTN. This is caused by the increased Bytes per exchange, as shown in Table I. Specifically, the minimum communicated Kbytes for retrieving a content (assuming that only one request/response exchange required) are approximately 380, 177 and 492 for NDN, DTN and NoD, respectively. Note that NoD has the largest packet size, since its request and response correspond to a bundle that encapsulates an Interest and a Data packet, respectively.

*2) Mobile Network:* Figure 3 summarizes the results of the mobile network. In Figure 3a we present the CRD results of each approach for varying numbers of nodes. In this scenario NDN showed significantly larger CRD, compared to DTN and NoD. This is primarily caused by NDN's breadcrumbs routing limitation [4], as explained in Section II. Another contributing factor may be the absence of content discovery methods, that we did not employ in our experiments to avoid increased network overhead (e.g., due to Interest broadcasting). The impact of content discovery in mobile IoT environments is beyond the scope of this work, but could be investigated in a future study. However, we observe that the impact of these factors becomes significantly smaller as the number of nodes increases. DTN showed smaller CRD than NDN, despite the lack of in-network caching. NoD accomplished the lowest CRD values, since it combines the in-network caching support of NDN with the reliability benefits of DTN.

Regarding CHR, NoD outperformed NDN, as shown in Figure 3b. This is justified because NoD retrieved content rapidly, as shown in Figure 3a, and thus increased the probability of finding fresh content in the local consumer cache (i.e., the content is not expired). Note that CHR influences significantly all the other performance metrics, since a cache hit contributes to lower CRD, less NDN transmissions and communicated KBytes.

In Figure 3c, we observe that NDN showed significantly reduced DR compared to DTN and NoD in the cases of 2, 4 and 8 nodes. This is caused because the network is sparse and volatile, rendering native NDN unable to operate reliably. However, in the 16 nodes case, where the network is more dense, NDN showed similar DR values with DTN and NoD. This demonstrates the benefits of content-centric approaches for larger deployments, as mentioned in [16]. Also, NoD and DTN achieved similar DR values, reflecting their reliability advantages in such dynamic conditions.

In Figure 3d we observe that NDN communicated significantly less Kbytes than all protocols and NoD achieved less communicated Kbytes than DTN. This is a result of in-network caching that allowed the retrieval of an increased amount of content from consumer's local cache, as shown in Figure 3b. Furthermore, the reduced communicated Kbytes of NDN and NoD is enhanced by the Interest aggregation feature. We conclude that the overall NDN and DTN performance in mobile environments could be significantly improved by their combined NoD approach.

## VI. Conclusions

NDN and DTN are the most prominent network architectures to handle intermittent and dynamic IoT connectivity. In this work we compared the performance of NDN, DTN and NoD over dynamic stationary and mobile IoT networks. Our experimental results revealed that native NDN outperformed DTN and NoD in stationary networks with limited packet loss. DTN demonstrates adequate reliability in both stationary and mobile networks, while NoD prevails under highly-disruptive stationary and mobile networks.

As a future work, we plan to extend this comparative study by performing real-world experiments and investigating the impact of NDN and DTN protocols on constrained IoT devices. Another future direction of this work concerns the investigation of mixed stationary and mobile IoT networks, as well as other challenging communication environments (e.g., space-air-ground IoT networks).

## Acknowledgment

## 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] W. Gao, J. H. Nguyen, W. Yu, C. Lu, D. T. Ku, and W. G. Hatcher, "Toward emulation-based performance assessment of constrained application protocol in dynamic networks," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1597–1610, 2017.

[3] V. Demiroglou, L. Mamatas, and V. Tsaoussidis, "Adaptive ndn, dtn and nod deployment in smart-city networks using sdn," in *2023 IEEE 20th Consumer Communications & Networking Conference (CCNC)*, pp. 1092–1097, 2023.

[4] 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.

[5] 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.

[6] 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.

[7] H. Liang, W. Gao, J. H. Nguyen, M. F. Orpilla, and W. Yu, "Internet of things data collection using unmanned aerial vehicles in infrastructure free environments," *IEEE Access*, vol. 8, pp. 3932–3944, 2020.

[8] 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.

[9] 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.

[10] 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.

[11] C. Nasis, C.-A. Sarros, and V. Tsaoussidis, "The impact of chunk size on named data networking performance," in *2020 3rd International Conference on Hot Information-Centric Networking (HotICN)*, pp. 108–113, 2020.

[12] 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 Commun. Mag.*, vol. 56, no. 2, pp. 136–143, 2018.

[13] Y. Zguira, H. Rivano, and A. Meddeb, "Internet of bikes: A dtn protocol with data aggregation for urban data collection," *Sensors*, vol. 18, no. 9, 2018.

[14] W. Gao, H. Liang, J. Nguyen, F. Liang, W. Yu, C. Lu, and M. Orpilla, *Emulation-Based Performance Evaluation of the Delay Tolerant Networking (DTN) in Dynamic Network Topologies*, pp. 23–41. Cham: Springer International Publishing, 2020.

[15] T. Li, Z. Kong, and L. Zhang, "Supporting delay tolerant networking: A comparative study of epidemic routing and ndn," in *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 1–6, 2020.

[16] C. Gündoğan, P. Kietzmann, M. Lenders, H. Petersen, T. C. Schmidt, and M. Wählisch, "Ndn, coap, and mqtt: A comparative measurement study in the iot," in *Proceedings of the 5th ACM Conference on Information-Centric Networking*, ICN '18, (New York, NY, USA), p. 159–171, Association for Computing Machinery, 2018.

[17] L. Mamatas, V. Demiroglou, S. Kalafatidis, S. Skaperas, and V. Tsaoussidis, "Protocol-adaptive strategies for wireless mesh smart city networks," *IEEE Network*, vol. 37, no. 2, pp. 136–143, 2023.

[18] E. A. Abdellaoui Alaoui, S. C. KOumetio Tekouabou, A. Gallais, and S. Agoujil, "Dtn routing hierarchical topology for the internet of things," *Procedia Computer Science*, vol. 170, pp. 490–497, 2020. The 11th International Conference on Ambient Systems, Networks and Technologies (ANT) / The 3rd International Conference on Emerging Data and Industry 4.0 (EDI40) / Affiliated Workshops.

[19] M. Amadeo, G. Ruggeri, C. Campolo, A. Molinaro, and G. Mangiullo, "Caching popular and fresh iot contents at the edge via named data networking," in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 610–615, 2020.

[20] K. Fall, "Comparing information-centric and delay-tolerant networking," in *Proceedings of the 2012 IEEE 37th Conference on Local Computer Networks (LCN 2012)*, LCN '12, (USA), p. xxxiii, IEEE Computer Society, 2012.

[21] E. Baccelli, C. Mehlis, O. Hahm, T. C. Schmidt, and M. Wählisch, "Information centric networking in the iot: Experiments with ndn in the wild," in *Proceedings of the 1st ACM Conference on Information-Centric Networking*, ACM-ICN '14, (New York, NY, USA), p. 77–86, Association for Computing Machinery, 2014.

[22] "Common Open Research Emulator." https://www.nrl.navy.mil/Our-Work/Areas-of-Research/Information-Technology/NCS/CORE/. Accessed: 2023-07-07.

[23] M. Doering, S. Lahde, J. Morgenroth, and L. Wolf, "Ibr-dtn: an efficient implementation for embedded systems," in *Proceedings of the Third Workshop on Challenged Networks, CHANTS 2008, San Francisco, California, USA, September 15, 2008*, pp. 117–120, Sept. 2008.

[24] N. Aschenbruck, R. Ernst, E. Gerhards-Padilla, and M. Schwamborn, "Bonnmotion: A mobility scenario generation and analysis tool," in *Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques*, SIMUTools '10, (Brussels, BEL), ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2010.