Protocol-Adaptive Strategies for Wireless Mesh Smart City Networks

Lefteris Mamatas, Vassilis Demiroglou, Sarantis Kalafatidis, Sotiris Skaperas, and Vassilis Tsaoussidis

***Abstract*—Wireless mesh networks, especially those typically utilized in smart city deployments for their low-cost and adaptable topologies, are characterized by challenging requirements for communication performance, reliability, as well as adaptability to dynamic network conditions. In this context, Named Data Networking (NDN) introduces a novel packet naming scheme and in-network caching for efficient data retrieval. Although NDN reliability can be damaged by prolonged delays and intermittent connectivity, this impact can be largely canceled by incorporating the Delay-Tolerant-Networking (DTN) paradigm. Hence, we argue that the challenging, dynamic network conditions of smart cities can be handled accordingly by a protocol-adaptive solution that deploys and configures on-demand the most appropriate protocol strategy per node. Software-Defined Networking (SDN) provides the missing features of intelligent centralized control and programmability. Here, we propose REWIRE: an SDN-based protocol-adaptive solution for smart city networking with low-delay communication and reliable interactions. We employ SDN control features, containerized non-IP protocol stacks, clustering and change point (CCP) mechanisms. We also conduct a preliminary investigation of our solution based on real experimentation over two novel smart city testbeds.**[1]

***Index Terms*—Wireless Mesh Smart City Networks, Software-Defined Networks (SDN), Named Data Networking (NDN), Delay-Tolerant Networks (DTN).**

I. Introduction

Cities are becoming smarter along with the technological evolution of Internet of Things (IoT), wireless sensor networks, network virtualization and automation. This evolution shapes applications such as environmental monitoring services, intelligent transportation and smart lighting systems that improve citizens' life quality and reduce city expenses. However, smart city ecosystems face scalability challenges due to the massive number of nodes deployed in large areas but also heterogeneity given that they integrate a wide-range of device types with different characteristics, each.

The devices generate and communicate big amounts of data that are reposited at some central point and drive context-sensitive intelligent mechanisms, e.g., an open-data repository for multiple uses. According to [1] data communication performance is crucial for smart city applications; Along the same line, according to [2] connectivity plays major role in the smooth operation and co-existence of city-wide deployments. In this work, we mainly focus on improvements to achieve reliable communication with adequate performance for the implementation of sophisticated smart city services.

[1]accepted for publication in IEEE Network, special issue on "Smart Community Network and Systems", Nov/Dec 2022

Wireless mesh network (WMN) technology brings particular advantages in smart cities, such as: (i) unstructured topology, i.e., blanket network connectivity with self-sustaining and self-configuring paths; (ii) flexibility and independent placement installation, i.e., not restricted to a specific technology or communication protocol; and (iii) extended communication range with scalable connections (rapid discovery of newly added/removed devices). Consequently, WMN is an attractive choice for implementing the communication backbone for smart city applications that require low deployment/maintenance cost and self-organized communication. For example, Stratford [2] is a smart city implemented with WMN that targets energy conservation and economic growth.

However, WMNs may suffer from unreliable and low-quality communications in open-air and crowded areas due to signal blocking/interference and mobility; unstable topologies and connectivity failures impact the operation and performance of IoT applications [1]. Hence, smart city networks require smart protocol strategies that adapt to dynamic changes.

Named Data Networking (NDN) is an Information Centric Networking (ICN) architecture, ideal to handle efficient IoT data communication in smart city environments thanks to its data-oriented nature and network caching [4]. Beyond NDN, Delay/Disruption Tolerant Networking (DTN) architecture [10] can enhance network's reliability given its store-carry-and-forward approach, when connectivity is intermittent; such conditions may appear in a challenging, communication-wise, smart city environment. Nevertheless, the DTN architecture alone cannot provide the content-centric benefits of NDN (e.g., in-network caching, packet-naming scheme), while native NDN is unable to deal with volatile network topologies, because of its breadcrumbs routing mechanism [11]. That said, the combined NDN-over-DTN (NoD) scheme [8] is bringing significant performance and reliability benefits in data communication over intermittently connected IoT devices.

On the other hand, a single protocol alone is not able to address the diversity of connectivity conditions and application requirements of smart cities' networks. Software-Defined Networking (SDN) becomes an important emerging technology for the appropriate selection, deployment and configuration of communication protocols. A centralized SDN platform, that enables global monitoring and network configuration, allows for protocol-adaptive strategies to the wireless mesh smart city networks. For example to employ decision-making mechanisms based on the global network view and improve the reliability of network operation through dynamic protocol configuration in groups of nodes. Our previous work on a relevant platform [9], considering a resource-constrained IoT

Solution	Protocol Solution(s)	Addressed Communication Challenges	Considered Environment	SDN-based Wireless Mesh	Protocol-adaptive Strategies	Experimentation Facility
DOCTOR [3]	ICN	Deployment & Security issues	Virtualized NDN environment	No	No	Dedicated wireless testbed
PiGeon [4]	ICN	Reliability, Service Placement	Smart city backbone network	No	No	CityLab testbed
ICN SDN-WMN [5]	ICN	Performance	Wireless Mesh Network	Yes	No	Dedicated wireless testbed
GreenICN [6]	ICN	Scalability, Energy-efficiency	Disaster scenario, Video delivery	No	No	ndnSIM simulator
U-MOBILE [7]	NDN-over-DTN	Reliability	Disaster Scenario	No	No	None
NoD scheme [8]	NDN, DTN & NoD	Reliability	Disruptive IoT network	No	No	Dedicated wireless testbed
MINOS [9]	RPL & CORAL SDN	Adaptation in network conditions	Mobile IoT network	No	Yes	Cooja emulator
REWIRE	WMP, NDN, DTN & NoD	Performance, Reliability	Smart city backbone network	Yes	Yes	CityLab and w-iLab testbeds

TABLE I: A comparison of REWIRE with state-of-the-art protocol strategies.

environment, revealed that the SDN architecture may deploy and configure multiple protocol strategies on demand, and also mitigate network heterogeneity by decoupling data from the control plane by introducing abstracted southbound interfaces.

In paper [12], we conducted an initial evaluation of the SDN-NDN integration over a WMN, which we further extend here with respect to the challenging intermittent connectivity in wireless mesh smart city networks. In our evaluation a subset of nodes characterized temporarily by connectivity disruptions due to occasional communication issues, such as interference. More importantly, we introduce and investigate experimentally SDN-based hybrid protocol strategies that mitigate dynamic network conditions. In particular, we evaluate the efficiency of interchanging NDN, NoD or DTN protocols, when disruptions evolve and eventually dominate. Efficiency here requires not only a corresponding adequate protocol solution for each particular condition challenge, but also an ability to realize the change. Hence, our experimental tools include methodologies for detecting significant network changes that call for corresponding adaptive strategies.

The main contribution of this work is the integration of those tools into a single experimental platform, called REWIRE. In summary, REWIRE exhibits the following novel properties: (i) applies NDN when topology changes are at a low or a moderate level; (ii) deploys NoD or DTN for links with severe communication issues, such as those with intermittent communication or prolonged delays; and (iii) rapidly detects disruptions in links' reliability based on a novel clustering and change point (CCP) mechanism. Last but not least, we document real proof-of-concept experiments over the novel w-iLab.t and CityLab Fed4FIRE+ testbeds and quantify the performance impact of (i) - (iii), in terms of communication delay and packet loss rate (PLR) under diverse network conditions. Since in large-scale experiments complex dynamics emerge in connectivity maintenance, this aspect deserves an independent study, e.g., to consider a hierarchy or distributed SDN controllers and multiple WMN islands connected with wireless optical links.

In turn, we highlight that deploying the most appropriate strategy dynamically (e.g., among NDN, NoD or DTN) allowed us to improve the communication performance and reliability of wireless mesh based smart city environments. We also show that the proposed link reliability detection mechanism is indeed adequate to detect significant changes and rapidly trigger the corresponding protocol adaptations.

The remainder of this paper is organized as follows. The next section contrasts our proposal against the related works. In sections III and IV we detail and evaluate the proposed system, respectively. Finally, in section V we conclude the paper and discuss important open issues.

II. Related Work

Smart city deployments are often architected by dispersed IoT networks and service entities communicating with each other over wireless mesh network backbones. However, they face *performance* and *reliability* issues due to interference and weak signal propagation of city environments, such as high delays and intermittent connectivity. Our proposal targets performance enhancements by mitigating the impact of dynamic and unreliable network conditions in wireless mesh smart city networks.

Here, we discuss the main proposed solutions sharing similar design characteristics with REWIRE. In Table I, we enlist such proposals along with their supported protocol technologies, targeted communication challenges and considered network environments. Next, we contrast the main contributions of REWIRE against representative related works, also supporting our choices to utilize ICN and the recently introduced NoD protocol in a smart-city context, as well as our multi-protocol approach to tackle the challenging connectivity requirements of wireless mesh smart city networks.

In particular, a number of solutions adopt the ICN technology to improve the performance of IoT data communication based on its novel naming scheme and in-network caching features. For example, the DOCTOR project [3] targets network performance optimization and security in Wide Area Networks (WANs) and investigates the co-existence and interoperability between IP and NDN by employing containerized protocol stacks. PiGeon [4] demonstrates the performance advantages of ICN in CityLab, the real outdoor smart city testbed we also utilize. Furthermore, it exploits ICN principles for containerized service placement in WMNs. The authors of [5] implement efficient content delivery over WMNs based on ICN caching controlled by an SDN controller. The GreenICN project [6] investigates scalability and energy-efficiency aspects of ICNs, also demonstrating that a disaster scenario

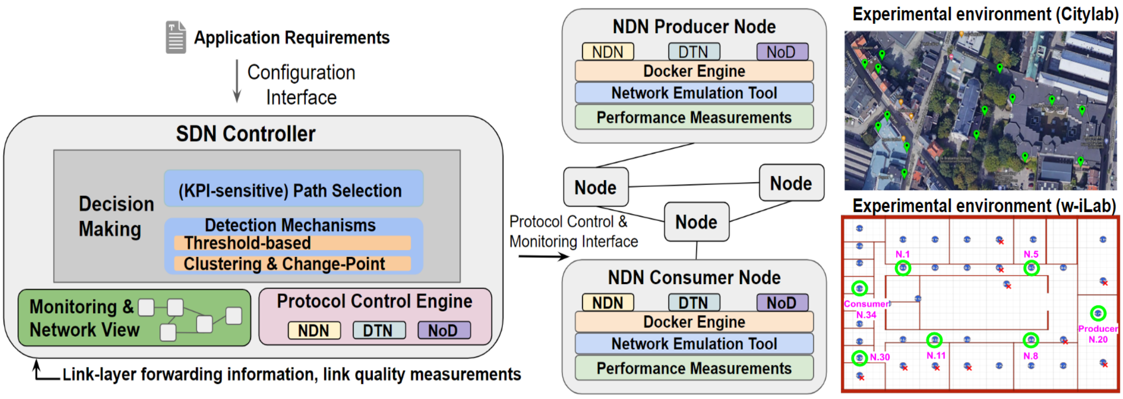

Fig. 1: The proposed REWIRE platform along with the utilized testbeds.

can lead to network fragmentation. In this context, GreenICN highlights that ICN may face reliability issues in challenging network conditions that call for dynamic topology adaptations.

Our proposal [12], which is an earlier and quite primitive version of REWIRE, highlights that an SDN approach can adapt ICN to the dynamic conditions of wireless mesh networks. An SDN controller monitors the WMN for dynamic changes in the wireless connectivity and adapts the forwarding information base (FIB) of ICN accordingly. However, in our experience with real smart city experimentation (e.g., as shown later in Fig. 3), particular nodes may face prolonged isolation. This part of the network can employ the DTN architecture, which is particularly designed for extraordinary conditions (e.g., opportunistic networks), but should be used with caution, given its extra communication overhead. Consequently, a hybrid ICN-DTN deployment is a potential solution.

Along these lines, the co-existence of NDN and DTN technologies has been proposed in a number of works. The U-MOBILE project [7] tackled the communication limitations of NDN in disaster environments (i.e., to handle intermittent connectivity) based on an NDN-over-DTN stack. That design also inspired the novel scheme for disruptive IoT networks, named NoD [8]. A real-world implementation and evaluation of NoD strengthens the claims for its suitability in disruptive IoT networks, including smart city networks. It appears that NoD: (i) improves the performance of both NDN and DTN stacks; (ii) extends NDN operation in intermittent connectivity situations; and (iii) enables their harmonious coexistence preserving the design advantages of both stacks. Although NoD scheme supports independent NDN and DTN operations, it does not include adaptable protocol synergies, by design.

MINOS [9] adopts SDN-based protocol-adaptive strategies by deploying and configuring on-demand two IoT network protocols (e.g., RPL and CORAL-SDN). However, it focuses on the performance and mobility support of constrained IoT devices without considering the limitation of data communication over a wireless mesh network backbone.

In contrast to the above works, REWIRE provides SDN-based control of multiple protocol stacks (i.e., wireless mesh, NDN, DTN, and NoD), handling both performance and reliability issues in smart city WMNs. For example, the co-existence and interoperability of different protocol stacks allow our solution to handle jointly data communication performance over WMN (i.e., through NDN), dynamic topology changes (i.e., through SDN-based wireless mesh) and communication disruptions (i.e., through DTN or NoD). Furthermore, we introduce a CCP mechanism that detects and classifies the different network conditions, guiding the appropriate selection of protocol stack in each node. The latter is deployed on-demand based on containerization technology.

Furthermore, our literature review identified a research gap in the experimentation of relevant solutions over real wireless-mesh-based smart city environments facing performance and reliability issues. As shown in the last column of Table I, commonly, relevant works are validated in simulators only (e.g., [6] in ndnSim), emulators (e.g., [9] in Cooja), or dedicated lab-based wireless testbeds (as in [3], [8] and [5]). In contrast, our approach is evaluated in two state-of-the-art smart city testbeds facing interference from co-existing users: (i) the CityLab testbed that enabled testing our reliability detection mechanism in real outdoor conditions, like in PiGeon [4]; and (ii) the w.iLab testbed that facilitates WMN and IoT experimentation in a heavily-used indoor environment. A description of the proposed system follows.

III. Proposed System

A. REWIRE Architecture and Main Facilities

We detail the REWIRE architecture and its associated subsystems being deployed over the w-iLab and CityLab testbeds, as illustrated in Fig. 1. REWIRE brings the following unique features: (i) protocol-adaptive strategies for wireless mesh smart city networks based on the deployment and management of alternative non-IP protocol stacks (i.e., NDN, DTN and NoD), seamlessly integrated over the WMN; (ii) a complete control loop based on centralized monitoring, decision-making capabilities enabled by a novel CCP mechanism, as well as unified control abstractions for the supported protocol stacks; and (iii) emulation facilities enabling REWIRE experimentation, including the reproduction of real IoT application communication patterns and challenging connectivity conditions.

The REWIRE system consists of an *SDN controller* and smart city *nodes*. The *SDN controller* supports three main subsystems: the *decision-making*, the *monitoring and network view*, and the *protocol control engine*, discussed briefly below.

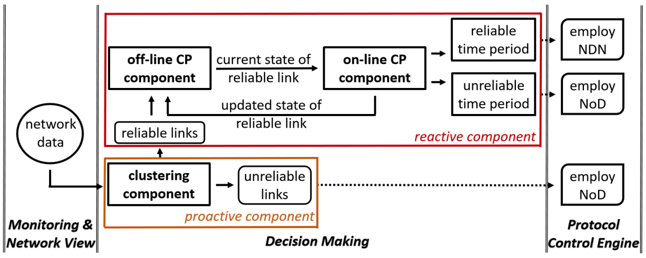

Fig. 2: Flow diagram of the overall detection mechanism.

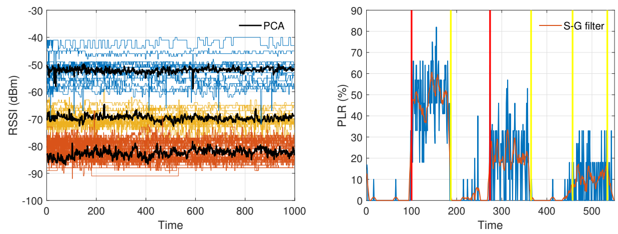

Fig. 3: a) Links' clustering of CityLab RSSI measurements using the proactive component. b) Output of the reactive component on a specific link. Red (yellow) vertical line denotes a CP leading to an unreliable (reliable) time period.

The *decision-making* subsystem incorporates the following functionalities: The *path selection*, which is responsible for dynamic path establishment and operates in two modes: simple and extended. The *simple mode* exploits link-layer data from all nodes and synchronizes the forwarding updates with the Forwarding Information Base (FIB) of the NDN stack (i.e., leaves the routing decisions to the wireless mesh protocol). The *extended mode* defines paths according to configurable key performance indicators (KPIs), e.g., delay or packet loss measurements, and the Dijkstra's algorithm.

The *detection mechanisms* facility supports multiple algorithms for unreliable connectivity detection, including threshold based (i.e., being suitable to detect time-instances/peaks of unreliable communication), as well as CCP based (i.e., to identify groups of unreliable links and time-periods with challenging communication concerning particular links). Such decisions drive on-demand adaptations of the protocol operation, ranging from protocol configurations to the protocol stack deployment itself. The appropriate selection of path establishment criteria and detection mechanism of choice depends on the *application requirements* that are being communicated to the *controller* through its *configuration interface*. At this point of investigation, we use static configuration files, but more sophisticated application-based adaptability is foreseen, incorporating artificial intelligence (AI) or machine-learning (ML) techniques.

The *protocol control engine* deploys and configures on-demand the NDN, DTN and NoD protocol stacks based on the outcome of the *decision making* feature and on appropriate control abstractions of all available protocols. NoD is used for nodes connected with unreliable links within the NDN path, while DTN for unreliable off-path intermediate forwarding nodes. The actual deployment is implemented using Docker containers that support all these stacks in a common code-base, which allows for rapid switching between the protocols.

Furthermore, the *monitoring and network view* collects and aggregates link-layer protocol data and link quality measurements from the network nodes, enabling the global network view of the REWIRE system. The above protocol control and monitoring facilities communicate with the *nodes* through the *protocol control and monitoring interface*. The REWIRE platform triggers on-demand deployment and configuration of each containerized protocol stack through the *control interface* and collects periodically information about the network state from the network nodes through the *monitoring interface*. At this stage, the controller configures the network nodes over custom TCP socket communication. However, as a next step, we plan to introduce a relevant SDN control protocol as well as REWIRE features in one of the existing open-source controllers (e.g., ONOS or OpenDaylight).

Regarding the communication among the SDN controller and the network nodes, we assume a separate control channel approach, like in [9], [13], consisting of (i) a reliable channel (i.e., low-data rate and long-range broadcast) for the control plane carrying the data from *protocol control* and *monitoring interfaces*; and (ii) a high-data-rate and short-range broadcast (e.g., 802.11) for the data plane. This approach allows efficient control of protocol stacks, in cases of intermittent or disruptive communication of the data plane. At this point of investigation, we have a limited implementation of separate control channel features, as discussed in Section IV.

All *nodes* are encompassing several functionalities, including: (i) the support of *wireless mesh networking* capabilities, enabling multi-hop communication between non-adjacent nodes; (ii) scripts periodically collecting *performance measurements* of each node with its neighbors, e.g., PLR or received signal strength indicator (RSSI), which are being transmitted to the *controller*; (iii) the *container engine* utilizing Docker to host the containerized protocol stacks; (iv) the *network emulation tool* facilitating the reproduction of unreliable connectivity conditions (i.e., additional delays that are either fixed or based on a probability distribution, packet loss probabilities or data rate limits); and (v) NDN consumer or producer features.

B. Detection mechanisms

We pay particular attention to the detection mechanisms of REWIRE system, due to their importance for the protocol-adaptive operation. We consider two approaches, a *threshold-based* and a novel CCP mechanism. The *threshold-based* applies configurable thresholds that detect significant changes in particular KPIs, as defined by the *application requirements*.

The design of CCP algorithm relies on the following observations, guided by our initial experimental results: (i) links that are generally unreliable may use NoD or DTN in cases they belong to the NDN path or are solely forwarding nodes, respectively; (ii) links that are generally reliable may use NDN; and (iii) reliable links with periods of disruptions, e.g., due to temporary interference, should be able to dynamicaly switch between NDN and NoD (or DTN). Consequently, we target here low-complexity structures to match potential resource constraints in IoT environments. Accordingly, we consider unsupervised ML techniques, since we prefer small-sized training samples rather than huge chunks of data or labeling overhead.

The CCP approach, as illustrated in Fig. 2, consists of two major components: (i) the proactive detector, which categorizes the links' quality based on historical data expressed in configurable KPIs; and (ii) the reactive detector, that progressively estimates in real-time, potential shifts on the links' quality level. We note that the reactive component is disabled for unreliable links to reduce communication and processing cost, since we assume that these links remain unreliable for a significant period of time. We further detail these major components below.

The proactive detector combines similarity based techniques, i.e., the dynamic time warping (DTW), with clustering, i.e., *k-medoids*. First, the DTW measure estimates the cross-dependence between the links, since common clustering techniques, e.g., *k-means*, assume independent observations. Then, the *k-medoids* algorithm is applied to cluster the links, where the number of clusters is determined by the gap statistic. Finally, each cluster is characterized in terms of reliability by principal component analysis (PCA).

The reactive component can be viewed as a change point (CP) detector, targeting to identify drifts from the "normal" behavior, i.e., the current reliability status. The proposed detector monitors connectivity metrics, e.g., PLR, by applying a hybrid scheme based on the offline-online CP algorithms introduced in [14]. First, the Savitzky-Golay (S-G) filter smooths the observed time-series. Second, the off-line CP detector, combined with an augmented binary segmentation algorithm, is applied to determine the current reliability status and extract the training sample, by segmenting the historical data in stationary intervals. Third, the sequential on-line detector is implemented, comparing the real-time observations with those of the training sample. Finally, if a CP is detected then its magnitude is estimated to justify and characterize a potential shift in reliability status.

Fig. 3 depicts the outputs of both proactive and reactive components, over real data collected from the CityLab smart city testbed. In Fig. 3(a), the proactive component categorizes efficiently the links in three clusters, as illustrated by the coloring, also assuming that the links' cluster with RSSI > -60 dBm groups the reliable links. Then, in Fig. 3(b), we depict the results of the reactive component on an arbitrary reliable link, where we manually increase (and decrease) the delay for specific time periods based on the normal distribution. Our approach detects efficiently the CPs in the mean of the PLR (i.e., as marked in Fig. 3 (b) with vertical lines). For instance, in $t = 100$ a CP is detected in the PLR labeling the link as unreliable, while, in $t = 200$, the link is considered again reliable. Note that in $t = 460$ and $t = 580$ the detected CPs provide a marginal magnitude of mean shift and the algorithm should not update the link connectivity status. Concluding, our CCP schema is able to distinguish between reliable and unreliable links with on-line link quality detection.

IV. Proof-of-concept Evaluation

Our experimental results demonstrate the capabilities of REWIRE to tackle the challenging performance and reliability requirements of wireless mesh smart city networks. We consider two experimental scenarios: (1) an assessment of SDN-based NDN-WMN integration, demonstrating the performance advantages of REWIRE over dynamic small-scale topologies; and (2) an evaluation of hybrid NDN / DTN strategies, validating the reliable communication of our solution in challenging communication conditions.

We generated simple, yet dynamic connectivity conditions in the two scenarios, as follows: In the mobility node case of scenario 1, we periodically adjust the wireless NIC transmission power between $5 - 15$ dBm, i.e., for 2.5 mins, each 2.5 mins. To implement the intermittent node case, we interrupt the wireless connection for 30 seconds, every 2.5 mins. In both cases, the overall experiment's duration is 10 mins. In scenario 2, additional delays are periodically introduced (i.e., for 5 mins, each 5 mins) based on random numbers that follow the normal distribution (i.e., in a $[800, 1200]$ msec interval) and using the Linux traffic control (tc) tool. The duration of the particular experiment is 30 mins.

The two scenarios utilize the w-iLab.t Fed4FIRE+ IoT testbed, depicted in Fig. 1, that supports Intel NUC devices with an Intel i5-4250U CPU, 8GB of memory and both wired (ethernet) and wireless 802.11abgn interfaces (AR9462 wireless network adapter), i.e., allowing the implementation of an ideal separate control channel. Also, the wireless connectivity in all experiments is handled by the Better Approach to Mobile Ad-hoc Networking (B.A.T.M.A.N.) WMN protocol.

The experiments utilize containerized implementations of NDN and DTN i.e., the named-data forwarding daemon (NFD) and IBR-DTN bundle protocol implementation, respectively. We note that NoD scheme uses both implementations. We also assume an IoT application where IoT consumers request sensor data from IoT producers communicating emulated sensor measurements. To confirm the statistical accuracy of our results, we conducted at least 10 repetitions per experiment and visualized the average values of delay, i.e., the NDN interest-data exchange round-trip time that reflects the end-to-end delay and PLR, i.e., the NDN packet loss ratio of each protocol solution.

Moreover, we consider two different cases for the SDN control channel: (i) scenario 1 adopts in-band control, which is a worst-case scenario in this respect (as in [12]), i.e., control traffic faces the same communication issues with data traffic; and (ii) scenario 2 uses an ideal wired separate control channel, focusing on the impact of REWIRE strategies over the data plane. Although assessing the impact of alternative control channel implementations and protocols is a complex matter outside the scope of this paper, we attempt an initial observation on the control channel dynamics.

At this point, we assess REWIRE's gains in terms of performance and reliability over a small-scale network with challenging connectivity. However, it may also be utilized in

TABLE II: Comparing NDN with REWIRE in dynamic topology conditions of w-iLab.t testbed.

	Case 1: Mobile Node				Case 2: Intermittent Node			
	Delay (ms)		PLR (%)		Delay (ms)		PLR (%)	
Metric	**NDN**	**REWIRE**	**NDN**	**REWIRE**	**NDN**	**REWIRE**	**NDN**	**REWIRE**
mean	67.7	33.8	4.7	3.3	36.8	33	8.6	3.6
sd	25.8	5.9	2.6	1	2.4	8.5	2.3	1.3

a large-scale network, targeting a limited number of nodes with unstable connectivity, e.g., in a crowded city center. We have released relevant source code and open-data in GitHub (https://github.com/athenarc/rewire).

A. Scenario 1: SDN-based NDN Communication

This scenario aims to investigate and quantify potential performance advantages that REWIRE brings in NDN deployments over wireless mesh smart city networks facing dynamic topology changes. In practical terms, the REWIRE controller monitors the forwarding information from the wireless mesh protocol of each node and synchronizes all routing updates with the forwarding information base (FIB) of the NDN solution, i.e., utilizing the *monitoring and network view* and *path selection / protocol control engine* components, respectively. We evaluate REWIRE performance against a typical NDN deployment that is not aware of the underlying link-layer protocol. A relevant scenario with static nodes can be found in our earlier conference paper [12].

Our small-scale experimental scenario consists of seven (7) nodes from the w-iLab.t testbed, an adequate but also manageable amount that ensures multihop communication within the given deployment area. We also reduced the nodes' wireless transmission power to 3 dBm, for the same reason. We consider two cases. In the first case, we assume an emulated mobile node at the center of the network, which moves periodically and causes routing changes without connectivity disruptions. In the second case, we increase the topology dynamicity by implementing connectivity disruptions at a central node with a fixed movement period and duration.

In Table II, we depict the mean along with the corresponding standard deviation (sd) values of the delay and PLR measurements for NDN and REWIRE. For both mobile and intermittent node cases, REWIRE outperforms the NDN solution. Specifically, in the mobile node case, the results demonstrate a superior performance of REWIRE, i.e., achieving a 50% delay and a 29% PLR reduction over the typical NDN, respectively. In the intermittent node case, REWIRE provides significant gains in terms of PLR, which is attributed to the REWIRE's platform capability to re-adjust the NDN path after the communication disruptions. Also, NDN and REWIRE depict comparable performance in terms of delay, i.e., lower mean and higher sd. However, REWIRE serves packets with longer delays, due to the lower PLR, which contributes to its higher sd value. Finally, more severe connectivity disruptions call for the deployment of either NoD or DTN. A relevant experimental evaluation follows next.

B. Scenario 2: Hybrid NDN / NoD Strategies

Here, we investigate the performance impact of the REWIRE protocol-adaptive strategy over a single w.iLab wireless link when intermittent connectivity occasionally prevails. More precisely, the *monitoring and network view* component collects and forwards (each 3 sec) the link's PLR. The *decision making* component applies the CCP mechanism to detect potential shifts between reliable, i.e., PLR<15%, and unreliable time-periods. Finally, the *protocol control engine* component activates the appropriate protocol, i.e, NDN or NoD, according to the *decision making* outputs. REWIRE's performance is compared with NoD and NDN (as single protocol solutions).

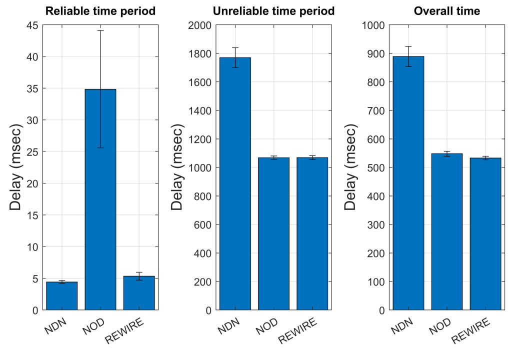

Fig. 4: Comparison of NDN, NoD and REWIRE over the w.iLab testbed, during the reliable and unreliable time-periods of the experiment, as well as for its total duration.

The outcomes are summarized in Fig. 4, illustrating the measured delay of NDN, NoD and REWIRE in both reliable and unreliable conditions, i.e., depicting the relevant measurements for the reliable and the unreliable parts of the experiment, as well as its overall duration. As it is shown, NDN outperforms NoD in the case of seamless connectivity, since it is more lightweight than the NoD (e.g., utilizes a smaller packet size). However, NoD achieves lower delay values compared to NDN when communication is disrupted, due to the reliability advantages of store-carry-and-forward paradigm implemented by the bundle protocol (i.e., DTN). The latter affirms that NoD is indeed suitable to handle unreliable connectivity, while NDN is superior in reliable network conditions.

Furthermore, Fig. 4 highlights the improved delay performance of REWIRE compared to both NDN and NoD. In particular, REWIRE reduces delay by 39% and 84%, compared to NDN for unreliable and to NoD for reliable connectivity, respectively. Regarding the overall experiment, REWIRE achieved a respective 40% and 2.7% delay reduction in contrast to NDN and NoD. We note that high delay values contribute much more to the overall delay, compared to low delay ones, i.e., impacting on a lower delay reduction in the overall experiment.

In the case of PLR, the NDN, NoD and REWIRE approaches performed almost 0%, 0.5% and 0.6%, respectively. NoD's increased PLR is attributed to the more complex implementation, while REWIRE faces a slightly larger value, due to both the protocol switching (around 0.3%) and NoD's implementation limitations (almost 0.3%). These PLR values are considered negligible given the unreliable network conditions of this scenario. Comparing the two scenarios, we observe an increased PLR in scenario 1, partially caused from control message failures, since both data and control planes use the same communication channel. Consequently, our results indicate that a selective NDN and NoD deployment is a reasonable strategy to address both performance and reliability issues of smart city WMN environments.

V. Conclusions and Open Issues

We detailed and evaluated REWIRE, a protocol-adaptive system that improves the network performance in wireless mesh smart city networks facing challenging communication conditions. Our experiments provided the following insights: (i) NDN is indeed an appropriate choice for smart city nodes with connectivity of low or moderate quality, as long as the NDN paths are synchronized with the corresponding topology changes of the wireless mesh protocol; (ii) NoD maintains the connectivity of the NDN path, even with temporary disruptions or long delays; (iii) protocol switching strategies can improve communication performance compared to single protocol approaches, such as NoD or NDN; and (iv) persistent monitoring of particular metrics, e.g., delay or PLR, can guide the appropriate protocol deployment or adaptations based on lightweight CCP procedures. Our future work plans include:

- experimentation over large-scale networks, studying the new dynamics and issues that scalability brings. We plan to introduce a distributed or hierarchical SDN controller architecture.
- extensive evaluations on the accuracy/detection time and resource-efficiency of the proposed CCP mechanisms.
- validation of REWIRE in other contexts, including over terrestrial-to-space communication establishments, i.e., a DTN-enabled satellite constellation [15].
- improvements/extensions of controller features, including: (i) a new decision making component based on AI/ML; (ii) additional protocol aspects and configurations (e.g., NDN caching); and (iii) in its control channel facility.
- integration/experimentation with other WMN protocols.

Acknowledgment

This work received funding from EU's H2020 research and innovation programme through the 9th open call scheme of the Fed4FIRE+ project (grant agr. num. 732638). It is also co-funded by Greece and the European Union (European Social Fund-ESF) through the Operational Programme "Human Resources Development, Education and Lifelong Learning" in the context of action "Enhancing Human Resources Research Potential by undertaking a Doctoral Research", sub-action 2: "IKY Scholarship Programme for PhD candidates in Greek Universities".

References

[1] W. Tu, "Data-driven QoS and QoE management in smart cities: A tutorial study," *IEEE Commun. Mag.*, vol. 56, no. 12, pp. 126–133, 2018.

[2] I. Yaqoob, I. A. T. Hashem, Y. Mehmood, A. Gani, S. Mokhtar, and S. Guizani, "Enabling communication technologies for smart cities," *IEEE Commun. Mag.*, vol. 55, no. 1, pp. 112–120, 2017.

[3] G. Doyen, T. Cholez, W. Mallouli, B. Mathieu, *et al.*, "An Orchestrated NDN Virtual Infrastructure Transporting Web Traffic: Design, Implementation, and First Experiments with Real End Users," *IEEE Commun. Mag.*, vol. 57, no. 6, pp. 33–39, 2019.

[4] 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 *Proc. IEEE Int. Conf. Fog Comput. (ICFC)*, pp. 50–55, IEEE, 2020.

[5] W.-S. Kim, C. Sang-Hwa, and M. Jae-Won, "Improved content management for information-centric networking in SDN-based wireless mesh network," *Computer Networks*, vol. 92, pp. 316–329, 2015.

[6] O. Ascigil, V. Sourlas, I. Psaras, and G. Pavlou, "Opportunistic off-path content discovery in information-centric networks," in *Proc. IEEE Int. Symp. Local Metrop. Area Netw. (LANMAN)*, pp. 1–7, 2016.

[7] C.-A. Sarros, S. Diamantopoulos, S. Rene, I. Psaras, *et al.*, "Connecting the edges: A universal, mobile-centric, and opportunistic communications architecture," *IEEE Commun. Mag.*, vol. 56, no. 2, pp. 136–143, 2018.

[8] 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.

[9] T. Theodorou, G. Violettas, P. Valsamas, S. Petridou, and L. Mamatas, "A Multi-Protocol Software-Defined Networking Solution for the Internet of Things," *IEEE Commun. Mag.*, vol. 57, no. 10, pp. 42–48, 2019.

[10] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, *et al.*, "Delay-tolerant networking architecture," 2007.

[11] C.-A. Sarros, V. Demiroglou, and V. Tsaoussidis, "Intermittently-connected IoT devices: Experiments with an NDN-DTN architecture," in *Proc. IEEE 18th Annual Consumer Commun. Netw. Conf. (CCNC)*, pp. 1–9, 2021.

[12] S. Kalafatidis, V. Demiroglou, L. Mamatas, and V. Tsaoussidis, "Experimenting with an SDN-Based NDN Deployment over Wireless Mesh Networks," *IEEE INFOCOM WKSHPS CNERT*, vol. 2022, 2022.

[13] T. Theodorou and L. Mamatas, "SD-MIoT: A Software-Defined Networking Solution for Mobile Internet of Things," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4604–4617, 2021.

[14] 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.

[15] S. Burleigh, V. G. Cerf, J. Crowcroft, and V. Tsaoussidis, "Space for internet and internet for space," *Ad Hoc Networks*, vol. 23, pp. 80–86, 2014.

LEFTERIS MAMATAS is an Associate Professor at the Department of Applied Informatics, University of Macedonia, Greece.

VASSILIS DEMIROGLOU received his Diploma from Electrical and Computer Engineering Department of Democritus University of Thrace, Greece. Currently, he is a Ph.D. student at the same department and a research associate at Athena Research and Innovation Center.

SARANTIS KALAFATIDIS pursues his Ph.D. at the University of Macedonia, Greece, in the areas of SDN and Cloud Computing. He holds an MSc Degree in Applied Informatics from the same University.

SOTIRIS SKAPERAS received the B.Sc. and the M.Sc. degree in Mathematics, Aristotle University, Greece, and the Ph.D degree in 2021, from the Department of Applied Informatics, University of Macedonia, Greece.

VASSILIS TSAOUSSIDIS is a professor at the Democritus University of Thrace, Greece and adjunct professor at Athena RC, holding degrees in applied mathematics (Aristotle University) and computer science (Ph.D. in computer science, Humboldt University, Berlin).