Experiments with SDN-based Adaptable Non-IP Protocol Stacks in Smart-City Environments

Sarantis Kalafatidis*‡, Vassilis Demiroglou *†, Sotiris Skaperas*‡, Georgios Tsoulouhas*†, Polychronis Valsamas*‡, Lefteris Mamatas*‡, Vassilis Tsaoussidis*†,

* *Athena Research and Innovation Center, Greece*

† *Department of Electrical and Computer Engineering, Democritus University of Thrace, Greece*

{vdemirog, gtsoulou, vtsaousi}(at)ee.duth.gr

‡ *Department of Applied Informatics, University of Macedonia, Greece*

{kalafatidis, sotskap, xvalsama, emamatas}(at)uom.edu.gr

***Abstract*—We propose an SDN-based Non-IP multi-protocol platform (REWIRE) that mixes and matches, on-demand, multiple Non-IP protocol strategies with real rapidly-detected network conditions and IoT data communication patterns. The REWIRE solution supports centralized monitoring of Wireless Mesh Networks (WMN) and manages alternative Non-IP protocols stacks (i.e., NDN, DTN & NoD) in an adaptable manner based on change-point analysis & clustering mechanisms. To this end, our proposal grafts flexibility and adaptability capabilities to the WMN providing the communication backbone in real Smart-City environments. Our platform was implemented in real WMNs environments over Fed4FIRE+ test-beds considering an IoT scenario using traffic patterns based on real sensor measurements.**

***Index Terms*—Software-Defined Networks, Wireless Mesh Networks, Named Data Networking, Delay-Tolerant Networking.**

I. Introduction

The Smart-City environment may consist of numerous heterogeneous IoT devices generating traffic that is not always smooth and manageable. These aspects have an impact on the IoT applications' performance, which are typically associated with critical requirements, such as low-delay and high-throughput. Furthermore, Smart-Cities usually encompass Wireless Mesh Networks (WMNs) in order to extend the communication range at large distances, however, the WMNs may face unreliable connectivity issues, such as link failures, no line-of-sight, and signal interference.

The Named Data Networking (NDN) [1] is ideal solution to handle efficient data management in Smart-City environments thanks to its data-oriented nature and network caching. Furthermore, Delay-Tolerant Networks (DTN) [2] are suitable when connectivity is intermittent. The NDN-over-DTN (NoD) [3], [4] scheme is a novel IoT oriented solution that combines these architectures and demonstrated significant performance benefits in data communication over intermittently connected IoT devices. However, such multi-protocol strategy calls for a platform that rapidly detects challenging network conditions and deploys, on-demand, the most appropriate protocol strategy to each node.

In this context, we propose REWIRE, a novel SDN-based management platform that applies, on-demand, protocol-adaptive strategies to mitigate the above challenging communication issues, characterizing real Smart-City deployments. Here, we detail a demonstration of our initial experiments based on the REWIRE prototype, which includes the following key features:

- Wireless Mesh Network (WNM) deployments over Fed4FIRE+ Smart-City test-beds [5], [6].
- Autonomous deployment of containerized Non-IP protocol stacks.
- Change-point analysis & clustering mechanisms for unreliable link detection.
- Realistic IoT application traffic patterns based on collected data [7] from SmartSantander test-bed [8].

II. The REWIRE platform

Figure 1 gives a high-level overview of the REWIRE prototype and its experimentation facilities. Its main functionalities are the following: (i) it manages alternative Non-IP protocols stacks (i.e., NDN [1], DTN [2] and NoD [3] [4]) in an adaptable manner; (ii) it incorporates multiple decision-making capabilities (e.g., path selection and detection mechanisms); (iii) it provides centralized monitoring of the wireless mesh network; (iv) it accommodates producers with realistic characteristics (i.e., generating real IoT measurements) and forecasting capabilities (e.g., identifying the freshness period of IoT measurements); (v) it supports both wired and wireless interfaces for the control channel; and (vi) it emulates various network connectivity conditions.

As it is shown in Fig. 1, the platform consists of: (i) the REWIRE Experimentation Manager, (ii) the SDN Controller, and (iii) the Network Nodes communicating through a wireless mesh plane. The major sub-components of the SDN Controller are the *Decision-making*, the *Protocol Control Engine* and the *Monitoring Facility*. The *Decision-making* component is responsible for: (i) selecting on-demand the appropriate path; and (ii) detecting the connectivity issues between wireless

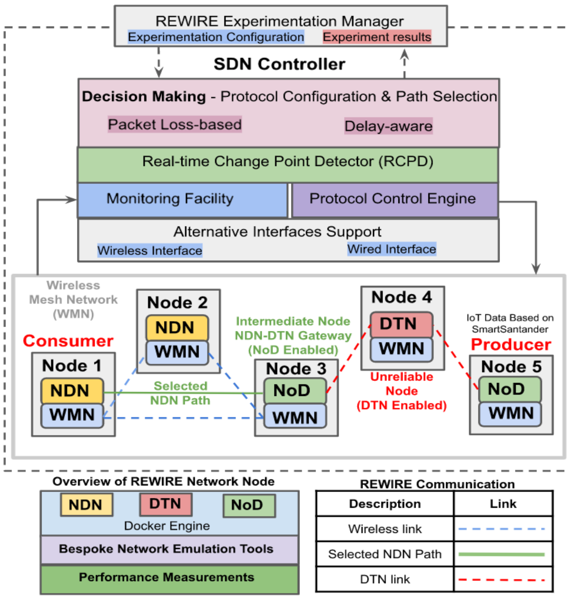

Fig. 1. Overview of REWIRE prototype

nodes, considering the overall network status. A brief description of each of the main REWIRE components follows:

Experimentation Manager (EM): This component is located on top of the REWIRE platform and is an interface between the user and the experimentation environment, i.e., enabling the adjustment of the REWIRE subsystems, as well as carrying out the aggregation of the experimental results. Fig. 2 illustrates a screenshot of the EM, during a running experiment, showing: (i) the jFed GUI tool which presents the wireless network topology (top-left); (ii) the output of the NDN-Producer (bottom-left) and the NDN-Consumer (bottom-right); and (iii) the SDN-NDN Controller (top-right).

Path Selection: Depending on the network conditions and the application requirements, the proposed path selection mechanism features both delay-aware (e.g., determining a low-latency path for facilitating a rapid IoT data retrieval process) and packet loss-based path selection (e.g., determining the most reliable path).

Real-time Change Point Detector: According to the targeted/intended detection sensitivity, our system accommodates two distinct approaches for link-quality detection including: (i) a classification/CP-based method [9] dedicated to identify unreliable time-periods over the links, and (ii) a threshold-based mechanism, suitable to detect unreliable time-instances (peaks) over the communication links.

Protocol Control Engine: It deploys and configures on-demand the NDN, DTN and NoD protocols depending on the outcome of the Decision-making mechanism.

Alternative Interface support: the Controller accommodates multiple interfaces and can use the appropriate interface according to: (i) the application domain (e.g., infrastructure-free vs infrastructure-based deployments); (ii) the application requirements (e.g., ultra-low latency vs delay-tolerant) and the Controller's deployment location (e.g., participating in the

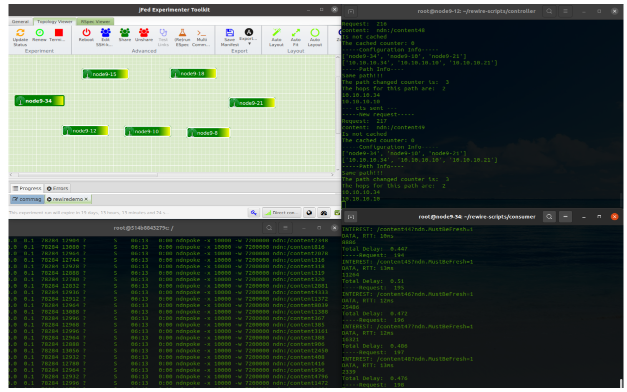

Fig. 2. REWIRE Experimentation Manger

WMN vs infrastructure deployment).

Monitoring Facility: aggregates the collected performance measurements from the network nodes and enables the global network view.

An overview of the Network Nodes' functionalities follows next. The *Wireless Mesh Network* enables the indirect communication between non-adjacent nodes in a multi-hop manner. Each individual node conducts performance measurements with the adjacent network nodes (e.g., of packet loss), which is communicated to the Controller, where the latter updates the global network view it maintains.

The *Container Engine* enables on-demand Future Internet Architecture (e.g., NDN, DTN and NoD) deployment and configuration. This is a key-enabling feature since it (i) accommodates distinct protocols in isolated virtualized environments; (ii) allows the rapid protocol deployment and configuration; and (iii) supports vertical and horizontal protocol deployment, according to the network requirements.

The *Bespoke Network Emulation Tool* facilitates the reproduction of unreliable connectivity and is based on the Linux Traffic Control (tc) utility that can introduce: (i) additional delays (e.g., prolonged delays, delays following the normal distribution); (ii) random packet loss; and (iii) data rate limitations. Thus, our ecosystem supports the reproduction of various connectivity scenarios that facilitate the testing and experimentation of REWIRE.

In this manner, our system grafts network flexibility and adaptability to allow for scalable and reliable Smart-City operations.

III. Demo Description

Our paper highlights the REWIRE platform's novel features through implementing our SDN controller in real WMNs environments over Fed4FIRE+ test-beds (w-iLab.1 & CityLab [5]), as well as considering an IoT scenario using real sensor measurements from SmartSantander test-bed. Especially, we demonstrate: (i) the flexibility and adaptability capabilities of our solution by optimizing the the NDN operation in unstable wireless conditions of WMN; (ii) the impact of adopting multiple Non-IP protocol strategies to handle unreliable connectivity conditions across Smart-City environments,

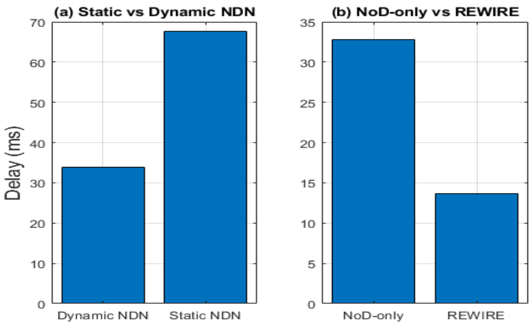

Fig. 3. REWIRE experimental results

while identifying these communication issues based on a novel statistical-based mechanism.

The demo presents two scenarios implemented in separate test-beds, i.e., aiming to evaluate our approach in indoor and outdoor wireless connectivity conditions. In both cases, we consider a WMN managed by the BATMAN protocol [10]. The wireless nodes host our containerized Non-IP protocol stacks that are centrally managed by the REWIRE SDN controller. Also in both cases, the NDN network includes an NDN Producer that generates real sensor measurements collected from the SmartSantander test-bed and an NDN producer that fetches the generated IoT contents.

The first scenario highlights the integration between the Layer 2 wireless mesh and the NDN planes. In particular, the REWIRE platform supports centralized monitoring of the wireless nodes in terms of wireless mesh routing protocol information and configures the NDN paths, accordingly. In paper [11], we present and describe in detail an initial implementation of the corresponding functionality.

In this scenario, we use the w-iLab.1 test-bed, which provides us with the ability to adapt the transmission power of the wireless interfaces, i.e., to enable both multi-hop communication and network mobility emulation. The REWIRE platform detects the network topology changes (via the BATMAN protocol) and configures the NDN paths (i.e., the NDN Forwarding Information Base, FIB), so they are aligned with the routing decisions of BATMAN.

Indicatively in Fig. 3(a), we present the results of a corresponding experiment in terms of NDN delay, where the NDN consumer fetches 1500 generated IoT data from the NDN producer, in two cases: (1) the NDN path remains static throughout the experiment; and (2) the network is managed by the REWIRE platform. Fig. 3 shows a significant improvement of NDN RTT, in the case our solution is utilized.

Scenario 2 demonstrates the REWIRE's unreliable link detection mechanisms and its autonomous deployment of containerized Non-IP protocol stacks. Specifically, the SDN Controller receives Packet Loss Ratio or Delay measurements for each link, which are processed by the Real-time Change Point Detector algorithm. Whenever the algorithm detects changes in the quality of communication, the SDN Controller configures the nodes with the most appropriate selection of protocol stack.

Specifically, the Controller deploys NDN for the nodes with reliable connectivity, and NoD for the nodes with challenging connectivity. In this manner, we utilize the more lightweight NDN implementation, compared to NoD, and facilitate the rapid content retrieval, when the path is reliable. In this experiment, we utilize the CityLab test-bed, which provides us with real outdoor wireless conditions.

Fig. 3(b) presents the performance of a static NoD deployment, over a single link with periodic emulated challenging connectivity, compared to a REWIRE-based experiment, which detects the quality of communication and applies NoD, for the time period with challenging connectivity, and NDN for the period with reliable communication. These results illustrate that REWIRE contributes to reduced Delay values, compared to the static NoD case.

IV. Conclusions

We demonstrated the capabilities of REWIRE experimentation platform, highlighting that: (i) our SDN-based approach effectively supports the NDN operation in unstable topologies with frequent dynamic changes, such as the WMNs; and (ii) the adaptive Non-IP protocol deployment based on a real-time link-quality detector mitigates unreliable connectivity conditions in Smart-City networks.

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] L. Zhang *et al.*, "Named data networking", *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, pp. 66–73, 2014
[2] Kevin Fall, "A delay-tolerant network architecture for challenged internets", *SIGCOMM '03*, 2003
[3] 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.
[4] V. Demiroglou, C. Sarros, and V. Tsaoussidis, "NoD: A content retrieval scheme for intermittently-connected IoT networks", *Ad Hoc Networks*, 130, 102825, 2022.
[5] "FED4FIRE+ - Federated Testbeds," FED4FIRE+, Oct. 29, 2014. https://www.fed4fire.eu/testbeds/ (accessed Jun. 03, 2022).
[6] 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," *INFOCOM WKSHPS*, pp. 529–534, 2018.
[7] V. Tsaoussidis *et al.*, "Experimental Results of the REWIRE FED4FIRE+ Open Call (OC) 9 Project." *Zenodo*, Nov. 29, 2022. doi: 10.5281/zenodo.6395256.
[8] L. Sanchez *et al.*, "Smartsantander: The meeting point between future internet research and experimentation and the smart cities." 2011 Future Network Mobile Summit. IEEE, 2011.
[9] S. Skaperas, L. Mamatas, and A. Chorti, "Real-time algorithms for the detection of changes in the variance of video content popularity", *IEEE Access*, 8, 30445-30457, 2020.
[10] "BATMAN Concept - open-mesh - open mesh." https://www.open-mesh.org/projects/open-mesh/wiki/BATMANConcept.
[11] S. Kalafatidis, V. Demiroglou, L. Mamatas, and V. Tsaoussidis, "Experimenting with an SDN-Based NDN Deployment over Wireless Mesh Networks", *IEEE INFOCOM WKSHPS CNERT 2022*, 2022.