FLEXNET: Flexible Networks for IoT based services

Johnny Choque
Communications Department
University of Cantabria
Santander, Spain
jchoque@tlmat.unican.es

Ramón Agüero
Communications Department
University of Cantabria
Santander, Spain
ramon@tlmat.unican.es

Zbigniew Kopertowski
Orange Labs Poland
Orange Polska
Warsaw, Poland
Zbigniew.Kopertowski@orange.com

Kim Khoa Nguyen
Synchromedia
Universite du Quebec
Montreal, Canada
kim-khoa.nguyen@civimetrix.com

Arturo Medela
Network & Systems Department
TST Sistemas S.A.
Santander, Spain
amedela@tst-sistemas.es

Esteban Municio
IDLab research group
University of Antwerp - IMEC
Antwerp, Belgium
esteban.municio@uantwerpen.be

Johann M. Marquez-Barja
IDLab research group
University of Antwerp - IMEC
Antwerp, Belgium
johann.marquez-barja@uantwerpen.be

Jaroslaw Domaszewicz
Inst. of Telecommunications
Warsaw University of Technology
Warsaw, Poland
domaszew@tele.pw.edu.pl

Andrzej Bak
Inst. of Telecommunications
Warsaw University of Technology
Warsaw, Poland
bak@tele.pw.edu.pl

Jeong Hyop Lee
Research Institute
Data Alliance Inc.
Seoul, Korea
jhlee@data-alliance.com

Seongsu Noh
Research Institute
Data Alliance Inc.
Seoul, Korea
sbison@data-alliance.com

Luis Muñoz
Communications Department
University of Cantabria
Santander, Spain
luis@tlmat.unican.es

Abstract*—Internet of Things is becoming one of the main triggers in designing and deploying new services aiming at fulfilling the wide demand imposed by end-users. Usually, concrete solutions addressing the optimization of the wireless segment are found in the literature. However, it is much less frequent to find end-to-end solutions to be easily adopted by the corresponding stakeholders. It is in this context that FLEXNET brings an integrated solution, relying on cutting-edge technologies, dealing with a wide set of technical requirements imposed by the different applications and services.*

***Keywords*—** *Flexible networks, Internet of Things, services, software defined network, wireless segment.*

I. INTRODUCTION

Internet of Things (IoT) driven application and services are imposing a plethora of requirements to both network core and edge segments. Hence, it is not just a matter of optimizing specific local resources either in the wired or wireless domains but a more holistic approach is needed enabling the synergies among the different domains. Furthermore, although most of the theoretical foundations, protocols and interfaces are well established, it is not usual to find holistic solutions or platforms, which can be easily adopted for customized usage.

Based on this, the present paper shows an integrated architecture dealing with both wired and wireless infrastructures, which aims at optimizing end-to-end performance by providing the required components. The proposal is underpinned around Software Define Network (SDN) [1] concept tightly correlated with concepts such edge computing techniques or dynamic resource allocation fitting Service Level Agreements (SLA).

The paper has been organized according to the following sections. Section II addresses the use cases and requirements which provide the fundamentals for platform design. Section III provides a short review of the state-of-the-art related to the employed technologies. Section IV discusses the main hints related to the platform design as well as the adopted validation approach. Finally, Section V provides main conclusions and potential further work.

II. FLEXNET ARCHITECTURE: USE CASES AND GOALS

FLEXNET (Flexible Network) proposes a set of canonical use cases in the security domain, which will enable to derive the corresponding requirements in the different network segments. Security domain provides the perfect framework for identifying some of the main constraints in existing network infrastructures. Hence, three scenarios will be conceived aiming at stressing the network in its different subsystems and setting up the basis for flexible network solutions. The idea of reconfiguring the network according to event detection activating different concurrent routes from one source to multiple destinations guaranteeing some specific quality of service parameters plays a capital role.

A. Video surveillance

The initial use case relies on an SDN based architecture to improve video streaming in a city area were video surveillance is required. Hence, the platform should support a set of video cameras implementing a surveillance security system in a city spot where caravans park. In essence, as long as the presence of an intruder is not detected, low-quality video streaming video is transmitted; in another case, a higher quality streaming video is transmitted in order to visualize the potential intruder with higher resolution.

This means that by default the video is transmitted in the scenario under low-quality consideration. Thus, routing within the emulated FLEXNET network is done over the low-bandwidth links. At the other end of the communication

channel a web application receives and displays the streaming video. When presence sensors or any other IoT device detects a potential intruder, they trigger a request for setting up high quality video streaming from the cameras.

The web application sends the corresponding OpenFlow [2] command to the SDN controller, which in turn reconfigures the paths to support the session with higher bandwidth requirements. The employment of a programmable OpenFlow-enabled SDN architecture facilitates a fine-grained control over the traffic flows. Fig. 1 depicts all these issues.

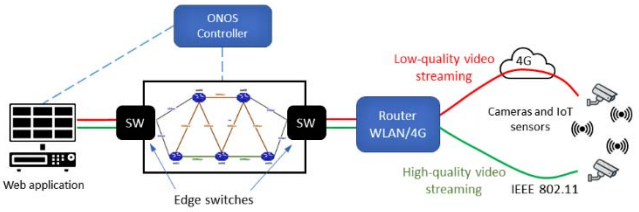

Fig. 1. SDN-based architecture to improve video streaming

B. Missing People Tracking

In 2016, the total number of reported missing people was 38,281 (approximately 105 persons/day) in South Korea. It incurs enormous social and economic loss, which can be reduced by applying IoT technologies based on LoRa (Long Range) and its next generation LoRa Gateways that enable a GPS-free geolocation service.

In an ordinary situation, the approximate location of a registered user can be monitored by their guardians using a Web app. In order to do this, a device periodically transmits a packet, and the approximate location is determined by the TDOA (Time Difference of Arrival) method on the server. The built-in GPS module on the device is in a sleep mode to save a battery life. When the application receives a report (or an alert) on a missing person, the application must transmit a message to the device to activate the GPS mode of the device so that the device can transmit the exact location data using GPS to the server. The server should transmit Assisted GPS (A-GPS) information to the device so that the GPS module of the device can measure the location more quickly, and the device should be able to measure the current location as soon as possible by loading the corresponding A-GPS data into the built-in GPS module (Fig. 2).

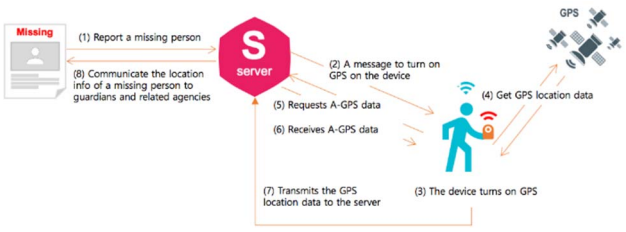

Fig. 2. Service Scenario for Missing People Tracking

A LoRa Gateway has multiple physical Network Interfaces and configured OVS (Open Virtual Switches) with them. When it transmits LoRa packets to a Network Server, the OVS adopts one of multiple physical NICs (Network Interface Controller) installed on a LoRa Gateway. The physical NIC is determined by the SDN Controller. A routing path from a LoRa Gateway to a Network Server and a routing path from a Network Server to an Application Server are all decided by the SDN Controller (Fig. 3). One of the physical network servers is allocated among cluster of multiple network server instances by the ONOS (Open Networking Operating System) controller [3], which produces a virtual address. The combination of multiple networks and network servers can enlarge the network capacity to accommodate a large-scale missing people tracking service.

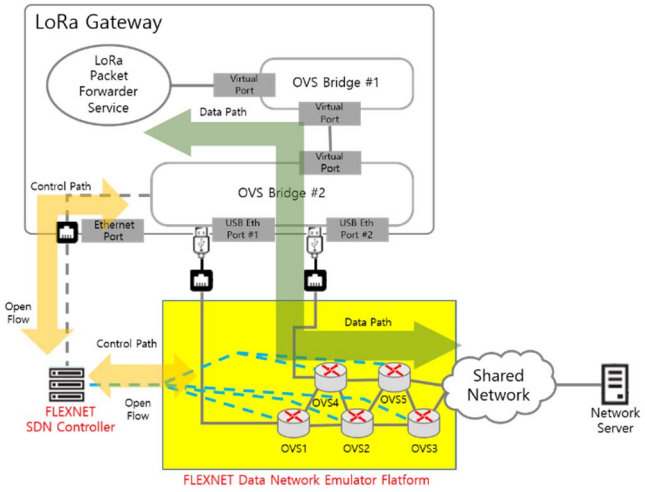

Fig. 3. The FLEXNET SDN Controller managing the physical network

C. Flexible Wireless Gateway

Finally, FLEXNET considers a third use case that aims to also virtualize the actual wireless segments. This done through a general purpose multi-RAT (Radio Access Technology) Flexible Gateway that transport data from sensors and other traffic sources to the backhaul and core networks through a virtualized wireless MAC layer. This VMAC (Virtual MAC) consist of a server and a number of clients, and would allow performing non-trivial operations such as seamless handover, packet duplication or load balancing to transparently transport the data in a technology agnostic manner. The Flexible Gateway has being conceived to support three applications of interest (see Fig. 4):

- *Maritime safeguard*. Currently, many transport vessels sail or anchor in offshore wind-farms and coastal areas where there is often Wi-Fi (Wireless Fidelity) or LTE (Long Term Evolution) coverage. By placing such Flexible Gateways in the vessels, seamless connectivity for industrial or leisure data could be provided with different levels of QoS (Quality of Service) according to the different wireless technologies available.

- *Public safety*. Emergency management and firefighting services could be enhanced with drone-assisted tasks, such as danger assessment in fires or collapsing buildings. Since reliability is a must when piloting drones and transmitting data streams, a Flexible Gateway mounted in a drone enables maximum usage of the available wireless technologies (e.g., Wi-Fi or LTE) to offer the best QoS policy fulfilment.

- *Industrial Safety*. In port and industrial areas, many spaces, such as the interior of fuel deposits or grain silos are wirelessly isolated because of they behave as a Faraday cage. In order to detect accidents or anomalies in the interior, the inside UWB (Ultra Wide Band) sensors could be connected through the Flexible Gateway to reliably forward sensor data through

different technologies such as (LTE-M, LoraWAN or Wi-Fi) to the outer internet.

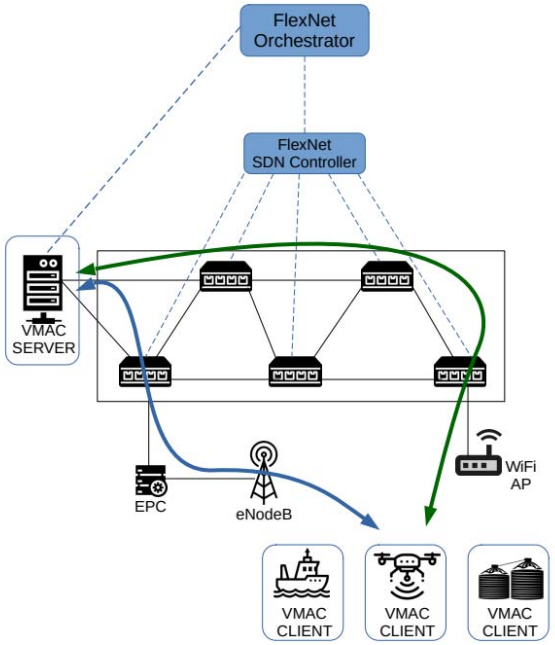

Fig. 4. The Flexible Gateway (VMAC Client) offering multi-RAT connectivity to different domains

III. STATE OF THE ART REVIEW

While network softwarization in the wired domain has been already used for more than a decade [4][5] the concept of Wireless SDN (and specially SDN-on-IoT) [6][7] has not really lifted off yet, due to the unreliable nature of wireless links, devices' limited resources and the excessive control overhead. However many works have engineers solutions to tackle these problems [8][9][10]. It is also worth to refer to other existing reference architectures such the ones adopted in SELFNET [11], SOFTFIRE [12] and V-SDN [13].

For this type of solutions to be widely adopted in the industry, they need to be fully integrated with the current programmable networking fabric [14][15][16] and support a representative number of different use cases. This is one the goals of FLEXNET.

Additionally, FLEXNET enables the integration of the wireless network segments with traditional, heterogeneous SDN/NFV (Network Function Virtualization) domains. In order to do this, FLEXNET uses the concept of a Virtual MAC for the different wireless technologies, such as Wi-Fi, 4G or LTE-M. We leverage the Orchestra tool [17] to provide the global orchestrator with inter-technology network management in the wireless segments. This includes e.g., inter-technology seamless handover, load balancing (interface bonding) or packet duplication. Additionally we can perform scheduling and configure PHY parameters under orchestrator's commands. These functions and their location in the OSI (Open System Interconnection) model are depicted in Fig. 5.

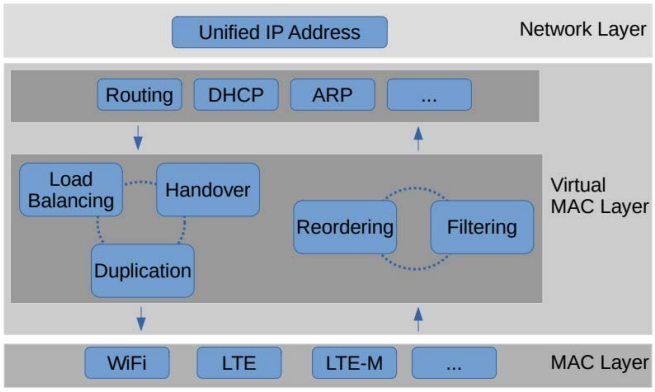

Fig. 5. The Flexible Gateway (VMAC Client) offering multi-RAT connectivity to different domains

While some of this features are already present in MTCP [18], LTE-LWA [19] or IEEE 1905.1 [20], Orchestra provides with a combination of network-wide and packet-level control, some of the required key features for full programmability in wireless networks (see Table I).

TABLE I. ORCHESTRA WITHIN THE STATE-OF-THE-ART

	IEEE 1905.1	3GPP	MPTCP	Orchestra
Domain	LAN	LAN-Radio	Any	Any
Technology	Eth, Wi-Fi	Wi-Fi, LTE	All	All
Coordination	Global	Local	Peer to Peer	Global
Control Level	Flow-based	Flow-based	Packet-based	Packet-based
Transport	Any	Any	TCP	Any
Vertical HO	No	Yes	Yes	Yes

A. *Software Defined Network platforms*

The backbone network for the FLEXNET platform can be composed by different public and private wired networks based on different technologies in the physical, data and transport layer. It is envisioned to use an SDN solution with their capability of programmable flexible control of network resources and dynamic on demand configuration according to application requirements.

In this approach, we can distinguish three layers: application, control and infrastructure (as sketched in Fig. 6). The main role in control plane is held by the Network Controller. Its precise role is to control the forwarding rules in the network devices using the so-called Openflow protocol. Network devices are thus called Openflow switches. Openflow is a vendor independent standard protocol that allows controlling the forwarding behaviour without knowledge about the vendor device.

Such approach allows for flexible creation of new services and applications that are installed over the network controller while no changes in the network devices are needed. Developers need to write their own network policies and services using high level programming language and API's (Application Programming Interfaces) of the controller. The network controller role is to translate such high-level programs into forwarding devices.

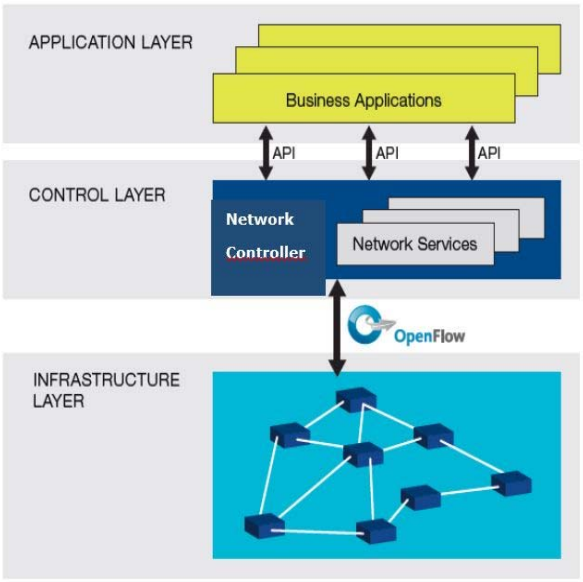

Fig. 6. General SDN architecture

For the SDN management solution there are many open source projects developing SDN controllers such as:

- ONOS
- Open Daylight
- Project Floodlight
- Beacon
- NOX/POX
- Open vSwitch
- Ryu Controller (supported by NTT Labs)
- Faucet

For FLEXNET platform the latest ONOS release was selected to use, this is the one called Toucan 2.3.0.

B. *Wireless IoT Domain*

The IoT is a growing area where industry, R&D (both public & private), alliances and standardization organizations are very active. Different achievements have already been done, such as the standardization of IoT LPWAN (Low power Wide Area Network) access technologies: NB-IoT (Narrow Band IoT), eMTC, EC-GSM, or LoRa. Others are ongoing or still to be designed, such as 5G. Other areas of definition and development include the creation of Evolved Packet Cores for IoT. Indeed, with the sheer ubiquity and volume of IoT devices, contention for finite control plane network resources will be extreme, impacting other human-critical services. As a contrast to human communications, collaboration and content delivery applications, different IoT applications have different traffic profiles and also do not necessarily follow human diurnal cycles. IoT applications also have highly diverse requirements in terms of network connectivity, reliability, security, latency, data rate, mobility and battery life. These requirements must also be met at extremely lost cost per bit, due to the lower value per bit (for example, to communicate an on/off state for a "thing") than for a typical (human) cellular connection. On the standardization side, we have Device Management and Data Collection, with OMA (Open Mobile Alliance) as the main reference in this area, and with its standardization of protocols such as Lightweight M2M or FUMO (Firmware Update Management Object).

FLEXNET will be based on all available standards and technologies in the area of IoT mentioned above, taking into account their requirements in so-called IoT platform. This platform provides the functionality for Device management, Data collection, and Application enablement. Its main role is to manage the complete thing lifecycle: installation, activation & configuration, diagnostic & maintenance, software/firmware control & upgrade, configuration update, fault management and any other lifecycle device management. It is also in charge of the Data Collection, making sure the data transport is performed securely. Both the Device Management and the Data Collection are essential to assure the trust of the data provided by the thing. Unmanaged data sources and/or data collection over untrusted network, provide full loss of data value, since the chain of trust is broken. Added to this Device Management and Data Collection, the IoT Platform also provides Application Enablement capabilities, so IoT value creators can do their applications easily.

The combination of all these technologies, in an end to end approach, as well as its alignment with the industry and R&D activities towards the 5G is the hearth of the present proposal. FLEXNET aims to perform probe of concepts and functionalities of the network programmability and elasticity in defined use cases with the objective to evaluate their limitations and challenges future 5G will have to face in a more widespread, heterogeneous scenario, actively contributing to the 5G design, implementation and widespread use.

IV. FLEXNET PLATFORM AND ITS VALIDATION

A. *Platform description*

The reference architecture which will drive aiming at consolidating platform design is shown in Fig. 7. Considering that architecture as the FLEXNET solution, the open source based approach was chosen to meet the main requirements of the system related to dynamic network resource allocation according to demands from IoT applications. The FLEXNET solution is designed to be flexible for IoT applications demands and have generic architecture of the system allowing for providing resources on demand independently from IoT vertical type. In the project designed architecture will be demonstrated for the three selected IoT use cases but solution is not limited to this chosen applications. The main platform elements are related to network control function allowing for network resource allocation (network orchestration). Many solutions in this area are currently elaborated and provided in research projects and first initial commercial products. Project partners are focused on open source solutions, especially to study their functionalities, reliability and possible deployment in the future offered services. There were considered different existing open source projects like ONAP (Open Network Automation Platform), and more focused on virtualised network controllers like OpenDayLight, Ryu or ONOS. The last one was chosen to use for FLEXNET platform implementation as most stable, and with capabilities required for platform implementation. Moreover, project partners like Orange Poland and Civimetrix actively work in their laboratories with their extensions and application for real deployments.

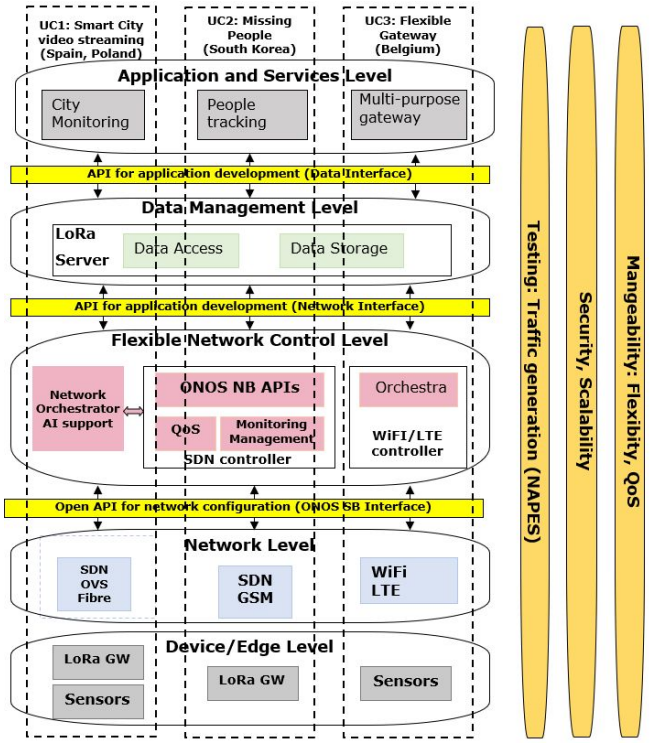

Fig. 7. FLEXNET designed architecture [21]

Based on open source solution, partners are looking for providing the cost effective system, open for developers, secure, scalable and reliable to evaluate it for future deployments. In the project, ONOS controller for SDN network is used for implementing network orchestrator functions. Additionally, AI (Artificial Intelligence) mechanism to support network orchestrator in network resources allocation capabilities is under development. Moreover, to validate designed FLEXNET platform the IoT traffic generator called NAPES (Networked Application Emulation System) is under development. It allows for IoT application emulation close to real one with implemented protocols and different traffic generation patterns. Using NAPES, IoT platform validation with different types of IoT use cases, with large amount of traffic and large number of connections is getting easier, cost efficient and not required big amount of real IoT equipment. It also allows for traffic load scenarios generation for AI functionalities developing and testing.

Based on the specified architecture the initial platform prototype has been set-up (see Fig. 8).

For demonstration purposes different IoT use cases will be deployed and for each of them network slice will be setting up dynamically. Different use cases (IoT applications) will demand network slices with different QoS parameters. For starting point in the initial version of the prototype we will use as network slice parameter the capacity link. In each use case the different level of integration with platform and network control level is required: (1) simply connection where sensors and IoT gateways not support Open Flow protocol, (2) SDN switch integrated in IoT gateway or IoT devices (e.g. LoRa Gateway), (3) at level of orchestration (existing orchestrator and FLEXNET orchestrator).

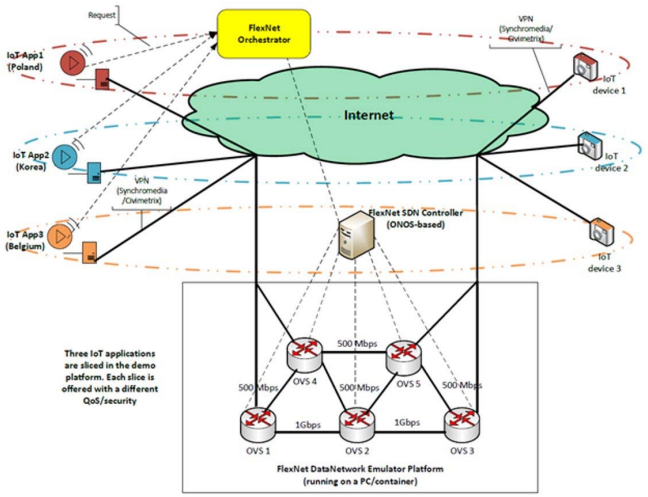

Fig. 8. FLEXNET implementation approach

B. *FLEXNET Validation with NAPES*

Using the real-life use case applications outlined above, the ultimate validation of the FLEXNET solution will be done. However, the process may face some challenges. The environmental conditions desirable for FLEXNET testing (e.g., ones that cause a rapid change of required network throughput) may not be under experimenters' control.

Accordingly, the FLEXNET project includes the development of a software framework, called NAPES, that makes it possible to *emulate* actual IoT applications to stress test the FLEXNET network. In NAPES, an emulating application, just like a real (emulated) one, is a collection of communicating components, running on IoT nodes and in the cloud. However, these components do not access real sensors or actuators, nor do they implement the actual application logic; instead, they just communicate among themselves in the way that closely resembles traffic produced by the real application. NAPES is meant to allow rapid FLEXNET testing, without an extensive investment in the development and deployment of real IoT applications. The "development" of an emulating application should be at least an order of magnitude less labor-intensive than that of a real one.

The design priorities for NAPES are (a) the flexibility in defining the structure of the emulating application (as a graph of communicating components), (b) the flexibility in defining the application's logic (via cooperating state machines), (c) the flexibility of making the workload traffic dependent on the states of the state machines, and (d) the flexibility in defining individual traffic flows which add up to the overall workload.

The emulating application consists of reusable, communicating components "running" on NAPES nodes (more precisely, interpreted by the so-called NAPES runtime). A connector links a port of one component with a port of another. The workload traffic consists of flows occurring on connectors. A component's internal logic is modelled as a finite state machine. To coordinate their work, the components exchange application events. Further, the components receive environmental events, which represent sensed phenomena occurring in the environment where the application is "deployed". The events drive the state machines. The flows generated by a component depend on the state of its state machine; this way the emulating application can generate very different network traffic at different times.

The elements of the NAPES framework are as follows: (a) the runtime, (b) the node register, (c) the emulation register, (d) the manager, (e) the event player, (f) the event relay, and, (g) the log register.

The *runtime* is a program installed on every (physical) node that may participate in an emulation session. The runtime transforms an "ordinary" node into a NAPES node, ready to host a component of an emulating application. The runtime can be implemented on different platforms, from resource-constrained IoT nodes to powerful cloud servers.

The *node register* holds a list of all NAPES nodes. The *emulation register* holds emulating applications. The *manager* inspects the node register and the emulation register to prepare an emulation session. It produces the component-to-node mapping, which assign, for each component of the emulating application, the node where it should run. Also, it translates a user-produced "code" of the emulating application into runtime component representations. The manager then uploads the runtime component representations to the nodes and instructs both the runtimes and the event player to start the emulation. After the emulation, the manager collects logs from the runtimes.

During the emulation, the *event player* sends environmental events to the components, according to the user-produced environmental event script. The *event relay* delivers environmental and application events to the components. The event relay adopts an MQTT-like, topic-based publish/subscribe paradigm. Finally, the *log register* is the place where all the logs produced during an emulation session are collected for analysis.

V. CONCLUSIONS AND FUTURE WORK

IoT infrastructure is offering a plethora of new services and applications filling heterogeneous end-user demands. Although there is an intensive activity around such technologies, and in particular in the edge domain, it is not so frequent to have access to end-to-end architectures, which optimize service performance in the different intermediate components. This is the reason that makes FLEXNET architecture an appealing option for network architects, service designers or even end-users. It provides a practical approach integrating SDN technologies, edge computing techniques and IoT applications and services aiming at fitting the corresponding service level agreements. Furthermore, from the validation perspective both a set of assessing techniques have been conceived for making the process systematic. Additionally, NAPES tool has demonstrated itself as a really convenient asset which enables to speed up the assessment phase.

ACKNOWLEDGMENT

This work was supported by FLEXNET Project: "Flexible IoT Networks for Value Creators" (Celtic 2016/3), in the Eureka Celtic-Next Cluster.

REFERENCES

[1] Madhusanka Liyanage; Andrei Gurtov; Mika Ylianttila, "Software Defined Networking Concepts," in Software Defined Mobile Networks (SDMN): Beyond LTE Network Architecture, Wiley, 2015, pp.21-44, doi: 10.1002/9781118900253.ch3.

[2] S. J. Vaughan-Nichols, "OpenFlow: The Next Generation of the Network?" in Computer, vol. 44, no. 8, pp. 13-15, Aug. 2011, doi: 10.1109/MC.2011.250.

[3] Open Network Operating System (ONOS). https://www.opennetworking.org/onos/

[4] Greenberg, Albert, et al. "A clean slate 4D approach to network control and management." ACM SIGCOMM Computer Communication Review35.5 (2005): 41-54.

[5] Gude, Natasha, et al. "NOX: towards an operating system for networks."ACM SIGCOMM Computer Communication Review 38.3 (2008): 105-110.

[6] Bera, Samaresh, Sudip Misra, and Athanasios V. Vasilakos. "Software-defined networking for internet of things: A survey."IEEE Internet of Things Journal4.6 (2017): 1994-2008.

[7] Thubert, Pascal, Maria Rita Palattella, and Thomas Engel. "6TiSCH centralized scheduling: When SDN meet IoT." 2015 IEEE conference on standards for communications and networking (CSCN). IEEE, 2015.

[8] De Oliveira, Bruno Trevizan, Lucas Batista Gabriel, and Cintia Borges Margi. "TinySDN: Enabling multiple controllers for software-defined wireless sensor networks." IEEE Latin America Transactions 13.11 (2015): 3690-3696.

[9] Galluccio, Laura, et al. "SDN-WISE: Design, prototyping and experimentation of a stateful SDN solution for WIreless SEnsor networks." 2015 IEEE Conference on Computer Communications (INFOCOM). IEEE, 2015.

[10] Municio, Esteban, et al. "Whisper: Programmable and flexible control on industrial IoT networks." Sensors18.11 (2018): 4048.

[11] SELFNET, Framework for Self-Organized Network Management in virtualized and Software Defined Networks. https://selfnet-5g.eu/

[12] SoftFIRE, Software Defined Networks and Network Function Virtualization Testbed within FIRE+. https://www.softfire.eu/

[13] V-SDN, Video streaming services with Software-Defined Networking. https://cordis.europa.eu/project/id/753685

[14] Phemius, Kevin, Mathieu Bouet, and Jérémie Leguay. "Disco: Distributed multi-domain SDNcontrollers." 2014 IEEE Network Operations and Management Symposium (NOMS). IEEE, 2014.

[15] Municio, Esteban, Steven Latre, and Johann Marquez-Barja. "Extending Network Programmability to the Things Overlay using Distributed Industrial IoT Protocols." IEEE Transactions on Industrial Informatics (2020).

[16] De la Oliva, Antonio, et al. "5G-TRANSFORMER: Slicing and orchestrating transport networks for industry verticals." IEEE Communications Magazine 56.8 (2018): 78-84.

[17] De Schepper, Tom, et al. "ORCHESTRA: enabling inter-technology network management in heterogeneous wireless networks." IEEE Transactions on Network and Service Management15.4 (2018): 1733-1746.

[18] Barré, Sébastien, Christoph Paasch, and Olivier Bonaventure. "Multipath TCP: from theory to practice." International Conference on Research in Networking. Springer, Berlin, Heidelberg, 2011.

[19] Nuggehalli, Pavan. "LTE-WLAN aggregation [industry perspectives]." IEEE Wireless Communications 23.4 (2016): 4-6.

[20] IEEE 1905.1: http://grouper.ieee.org/groups/1905/1/

[21] FLEXNET D2.2, Platform design document, March 2020