Pseudo-Dynamic Testing of Realistic Edge-Fog Cloud Ecosystems

Massimo Ficco, Christian Esposito, Yang Xiang, and Francesco Palmieri

Testing a software artifact to be deployed in the nodes composing an edge-fog ecosystem could be extremely challenging. Pure simulated environments and real testbeds could be not representative enough of realistic scenarios or unacceptably expensive. The authors explore such issues and present a pseudo-dynamic testing approach.

ABSTRACT

Currently, our society is undergoing a radical change due to the increasing pervasive use of ICT within all of its processes. Such an evolution has been triggered by the advent of the Internet of Things vision, where smart sensing devices can be integrated in the daily objects surrounding any human being. The increasing demand of dealing with the big data generated, managed, and stored by the applications built on top of such novel sensory networks has called for innovative architectures, such as edge and fog computing, which have not yet been fully standardized. Considering the large scale and complexity of these architectures, testing a software artifact to be deployed in the nodes composing an edge-fog ecosystem could be extremely challenging. Pure simulated environments and real testbeds could be not representative enough of realistic scenarios or unacceptably expensive. We explore such issues and present a pseudo-dynamic testing approach, where a portion of the experimental scenario is simulated, while the edge and fog nodes under test are emulated or executed in a real environment.

THE ERA OF EDGE-FOG COMPUTING

The success of the Internet of Things (IoT) is essentially due to the diffusion of a huge amount of very cheap embedded devices equipped with smart sensors and flexible actuators, which can easily be used to monitor and control a significant variety of elements in almost any daily life scenario. This paradigm promises to make anything part of the Internet in order to share sensory data and execute advanced control logic, opening up new horizons to innovative services and applications, which will see massive interaction among things and humans. IoT applications are currently expected to grow in scale, due to the phenomenon of federating different IoT solutions in order to realize more complex ones. This demands increasing memory and processing resources that easily exceed the possibility of current single-server architectures, as traditionally adopted by wireless sensor networks (WSNs). On the contrary, it calls for more elastic resource provisioning, by using a collection of servers and, in a later evolution, a cloud infrastructure providing enough capacity for quickly processing all the data generated by IoT objects, and making sense of them through proper analytics applications and practices. Cloud computing, thanks to its elasticity, flexibility, and scalability, offers the right infrastructure-level facilities to support the IoT runtime and storage requirements, and easily adapt to any change in the associated demand and operating scenario. However, the network traffic generated from hundreds of thousands of IoT devices toward the cloud may also become a major challenge in the presence of significant bandwidth resources, resulting in transmission and hence processing-response delays that may become crucial for many applications, such as power control, assisted driving, and health monitoring.

On the other hand, in order to extend the cloud computing paradigm to the edge of the network, where delays are critical and bandwidth is limited, the recent trend is to allow processing tasks, analytics, and knowledge generation to be handled closer to the data sources (i.e., sensors and actuators; edge computing), or between the edge and the cloud (fog computing), managing massive temporary storage and heavier analytics tasks. Both evolutions push the frontier of modern data processing applications away from centralized nodes, making the cloud as close as possible to the entities that produce and act on IoT data, in order to provide additional distributed intelligence and to deliver a faster response in analytics [1]. Edge and fog-level facilities will also participate in the management of network switching, routing, load balancing and security tasks, becoming the ingress points for the data coming from multiple heterogeneous sources and deciding if it has to be analyzed locally or conveyed through a specific path to the cloud for further processing. Such a complex ecosystem is referred to as edge-fog cloud computing. Its scalability, flexibility, and performance characteristics represent a driving force for a new breed of critical IoT services and applications that involve effective and efficient data management and analytics, such as latency-sensitive applications for smart grids, traffic management of vehicles, connected vehicles, smart cities, and services for enhancing the quality of life.

EDGE-FOG CLOUD AS A LARGE-SCALE COMPLEX SYSTEM

Applications and services tagged by the edge-fog cloud paradigm are characterized at the lower level by thousands or millions of networked objects generating and consuming data, deployed

Digital Object Identifier:
10.1109/MCOM.2017.1700328

Massimo Ficco is with the Università degli Studi della Campania Luigi Vanvitelli; Yang Xiang is with Deakin University; Christian Esposito and Francesco Palmieri are with the Università di Salerno.

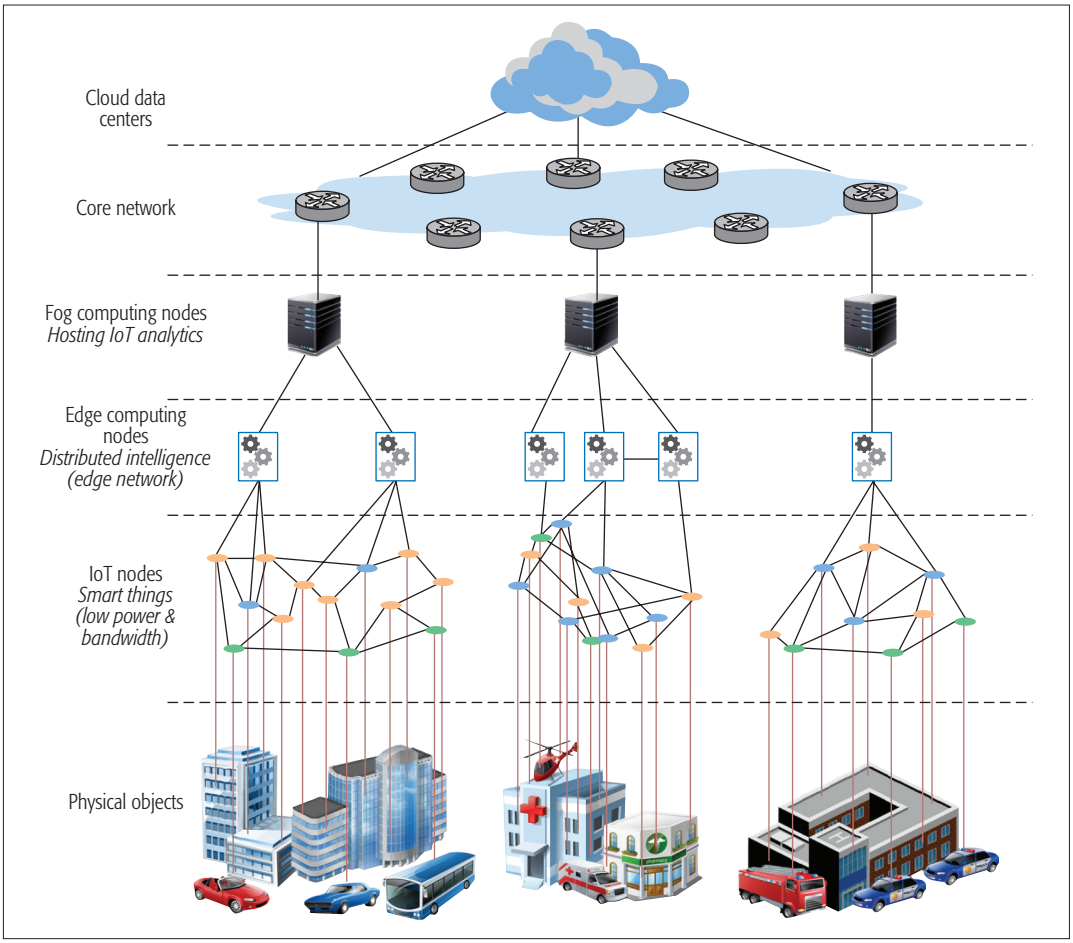

Figure 1. Edge-fog cloud computing architecture as an ecosystem of different cyber-physiscal systems.

In order to extend the services provided by the Cloud to the edge of the network, the new Fog paradigm involves application components, as well as virtualized network entities running both in the Cloud and on devices between the Cloud and the network edge.

across a large geographic area. Figure 1 presents a hierarchical distributed architecture supporting the future IoT applications, which shows the role of edge-fog cloud computing facilities. Specifically, the figure illustrates a typical setting for a smart city, which includes three interleaved and promiscuous applications: one (with orange IoT nodes) for the structural and environmental monitoring of the buildings, one (with blue IoT nodes) for patient management and healthcare, and the last one (with green IoT nodes) for traffic management. All the physical objects in the city, such as vehicles, buildings, and even humans, are equipped with proper devices, which can be abstracted as IoT nodes. For example, the orange IoT nodes can be sensors used to measure oscillations, temperature, or humidity in a room; the blue IoT nodes can be devices monitoring patient movements, vital signs, or drug consumption; and the green IoT nodes can be devices monitoring driving activities, checking traffic conditions, or alerting in case of emergencies.

Close to the IoT nodes (i.e., within a one- or two-hop distance from the IoT devices, and hence on the *network edge*), we can find a collection of loosely coupled and often human-operated nodes, such as tablets, laptops, workstations, wireless access points, and nano data centers. These edge nodes are usually provided with wireless device-to-device connectivity among them, and reliable connectivity to fog nodes. Their main task is to collect the sensor data and perform preliminary processing activities on them. In the example illustrated in Fig. 1, the edge nodes can be entitled to summarize the environmental data of the orange IoT nodes and to rapidly detect possible events (e.g., fire or structures that are about to collapse), or in the case of the green IoT nodes, an edge node can be used to detect a sudden accident on a road and to alert other reachable cars that are on the same route or planning to take it.

The fog nodes are more distant from the IoT objects and consist of more powerful machines, whose main task is supporting massive big data analytics and temporary storage activities. However, even if such nodes can provide significant computational capabilities, they are not able to cope with the increasing complexity of the envisioned applications, so most of the analytics and runtime capabilities have to still be allocated to the central cloud. Thus, in order to extend the services provided by the cloud to the edge of the network, the new fog paradigm involves application components, as well as virtualized network entities running both in the cloud and on devices between the cloud and the network edge. Specifically, they may consist of virtualized appliances or networking devices, such as smart gateways, next generation firewalls, and routers with high computing capabilities, interconnected with high speed and reliable links to the core network. Fog nodes are enabled to run cloud application logic on their native architecture, and can be managed

A real testbed, although desirable, in many cases is too expensive and does not provide a repeatable and controllable testing environment. An alternative approach consists of the emulation of a portion of the IoT solution by using a virtual network and virtual devices, typically for the sensing and embedded devices, where the real software is executed and tested.

and deployed by cloud providers. They are used to perform time-sensitive data analysis, as well as computationally intensive tasks offloaded by edge nodes, in order to reduce latency and traffic in the core network. Data that is less time-sensitive is sent to the nodes in the cloud for historical analysis of large sets of data collected from multiple fog nodes, big data analytics, and long-term storage. In the scenario shown in Fig. 1, the fog nodes can be used in order to make analysis on the traffic in the given portion of the city and to disseminate this information to all the cars placed in this portion, or to detect a critical situation for a patient and to call an ambulance. The fog nodes may be shared among the different applications running within the IoT infrastructure, so in case of a call for an ambulance, its driver can be informed of the best path to reach the patient, and all the cars along the path are instructed to yield to the ambulance.

On the other hand, the structural and dynamic complexity of this kind of information and communications technology (ICT) infrastructure makes the testing and vulnerability analysis of applications running on top of an edge-fog cloud ecosystem extremely challenging, due to the lack of a proper testing environment able to provide a realistic representation of these infrastructures. Large-scale and structural complexity derives from a wide and dense geographically distributed deployment of a huge number of IoT, edge, and fog nodes (in general not belonging to the same provider or organization), running in a wide variety of heterogeneity environments. They should be in charge of interacting and interoperating across different communications technologies, different providers, and federated domains. Their dynamic complexity can introduce behaviors that usually become evident only during on-site system operations, as well as during unpredictable fluctuations in both workloads and computational and communication resources needed to process these workloads. In addition, a proper testing environment should be not only limited to the hardware and software perspectives of an edge-fog cloud ecosystem, but it should also consider the human counterpart in order to have human-driven testing by mimicking how humans behave and interact with the cyber-physical components at the edge of the network. Identifying, understanding, and representing such complexity represent a challenge to support effective test and vulnerability analysis activities [2]. Testing and simulation environments should be provided to reproduce such complex and distributed systems locally in order to gain knowledge about their real behavior as it would be on site.

Testing Edge-Fog Cloud Ecosystems

In order to support development and test of software artifacts and services located at different levels of edge-fog cloud architecture, and assess the offered functionalities and perceived quality of service properties, several commercial and open source solutions have been proposed.

FIT IoT-LAB is a testbed equipped with thousands of wireless nodes located in six different sites across France, which can be used to test protocols and applications in a large-scale wireless IoT environment [3]. SmartSantander is a European project that offers a city-scale experimental facility for testing smart city applications [4]. The testbed comprises a large number of devices deployed in several urban locations. The logical architecture is organized into two tiers: *IoT nodes*, responsible for sensing the environmental parameters; and *gateways*, linking the IoT nodes at the network edge to a core infrastructure, providing more powerful server devices. Gateways enable interworking and integration testing between different IoT and server solutions. Cisco proposed an end-to-end fog application framework called IOx [5], providing facilities for orchestrating and managing applications on thousands of fog nodes in order to enable operations at multiple scales, as well as monitoring the application performance. Specific IOx middleware services and application programming interfaces (APIs) available on fog nodes make runtime and storage capabilities available to applications hosted on the network edge. By using device abstractions provided by Cisco IOx APIs, applications running on the fog nodes can communicate with IoT devices and applications hosted in the cloud.

However, a real testbed, although desirable, in many cases is too expensive, and does not provide a repeatable and controllable testing environment. An alternative approach consists of the emulation of a portion of the IoT solution by using a virtual network and virtual devices, typically for the sensing and embedded devices, where the real software is executed and tested. For example, the solution presented in [6] emulates an IoT environment by using the OpenStack cloud infrastructure, and in [7] a solution called MAMMotH is illustrated for the emulation of IoT nodes. Although this solution allows the implementation of experimental scenarios that are almost identical to real deployment settings, both at the hardware and software level, it can be used to emulate only a part of these complex infrastructures, and is not able to mimic and reproduce the effects of humans in the control loop, mainly referred to as human interactions with IoT devices.

Thus, simulation represents a valuable and cost-effective support for reproducing the more dense architectures that can be found at the edge of the network, as well as their iteration with the fog nodes and the cloud. It could be exploited to perform scheduling, migration, and resource management analysis at the edge-fog level by accurately modeling the involved reality and by carefully quantifying the defined performance metrics. In this direction, SimpleIoTSimulator is a commercial tool for simulating IoT-edge experimental scenarios, consisting of sensors and gateways interacting though common publish/subscribe protocols, including Constrained Application Protocol (CoAP) and MQTT [8]. Brambilla *et al.* [9] proposed a methodology for modeling and simulating large-scale IoT system deployments. It was designed to study low-level networking aspects by analyzing different mobility, network, and energy consumption models. Other concrete examples are [10, 11], which use the Omnet++ and NS3 simulators to reproduce IoT solutions and test their correctness. These simulators provide ad hoc software modules dealing with all the issues related to movement trails, network discovery, ongoing transmissions and

receptions, radio signals, battery consumption, and much more in order to build realistic simulated sensor networks. The application logic can be coded as it would be in a real deployment by means of the programming language supported by the simulator.

However, the presented simulators can be mainly used to model IoT environments. A specific framework designed to model fog environments along with IoT and cloud is iFogSim [12]. It enables performance evaluation of resource management policies applicable to fog environments with respect to their impact on latency, energy consumption, network congestion, and operational costs. It simulates edge devices, cloud data centers, and network links to measure performance metrics. This enables performance evaluation of resource management and scheduling policies across edge and cloud resources under multiple scenarios, such as real-time stream processing in a comprehensive end-to-end environment.

However, the representation and modeling capabilities of all the above simulation-based solutions and approaches are not able to fully capture the characteristics of the most sophisticated and articulated architectures, and in such scenarios a significant degree of uncertainty remains always present and impossible to represent and quantify in a reliable way, due to the heterogeneity as well as the structural and dynamic complexity of the involved systems. Moreover, considering the complexity of such systems and the huge number of involved entities at the different architectural levels, the processing of a single monolithic simulation system can lead to very high simulation cost and time. Thus, to be effective, simulations should manage these complexity aspects, and at the same time be realistic, time-optimal, and cost-effective, which, of course, are objectives that are in contrast to each other. Therefore, specific hybrid and distributed modeling strategies represent a viable alternative to design the simulation environments needed to support the evaluation of such complex systems [13]. In particular, pseudo-dynamic testing, which integrates simulation, emulation, and real components, represents the most effective solution for testing an extremely complex and large ecosystem such as the one composing the upcoming edge-fog cloud computing. This means that all or some of the above-mentioned approaches have to be jointly employed in the testing environment, where each approach will focus on a particular aspect or portion of the architecture. Moreover, such a solution is able to put humans in the loop by encompassing human activities and interactions in the testing activities of the ecosystem.

Pseudo-Dynamic Testing Approach

In order to enable testers to assemble complex, distributed, and cost-efficient edge-fog cloud experimental environments, pseudo-dynamic testing can be exploited. Such an approach leverages multiple testing methodologies and architectures, where traditional modeling practices are hybridized with experimental testing of systems of systems whose dynamic behavior can be condensed into a reduced number of degrees of freedom. In particular, it combines simulation and emulation, and also supports interaction with real systems. The emulated parts are the components under test, such as fog routers and edge devices, which can easily be assessed as prototypes that can be refined and strengthened during the testing activities. Simulation can be exploited to reproduce the behavior of the external systems, such as IoT objects used to generate the experimental workload and the testing stimulation. Unfortunately, some specific devices and phenomena cannot be simulated in a reliable way, so that real IoT devices, or specific activities associated with real human users, can be linked to the emulated fog/edge computing environment, supporting the testing and analysis of extremely realistic interactions between existing sensors and novel system architectures to be integrated into edge-layer services. Moreover, infrastructural components, such as the underlying communication network and cloud data services can also be simulated in order to abstract the testing scenario from unnecessary details and focus only on the elements that are really meaningful for a specific evaluation task, such as evaluating the effect of specific modifications to communication protocols and/or architectures in a fully controllable way. This results in cost-effective, more scalable, stable, and reliable testing frameworks, where the network architecture/layout is not constrained by economic or technological availability factors, and the networking behavior, including the link error rate, delay, and so on, is always verifiable and reproducible during the whole testing process.

Summing up, all the components involved in the experimental scenario, which are not the target of the test analysis, can be simulated, whereas external sources from which it is possible to obtain, in real time, the data streams needed to reproduce reality with a high degree of verisimilitude (e.g., meteorological services) can be associated with real systems connected to the hybrid testing scenarios. Furthermore, emulation of the edge nodes favors the interaction of humans with the systems involved in the scenario under evaluation, by providing a more realistic user experience to the whole testbed through the support of new experiments involving human in the loop without the necessity of a real-world infrastructure based on a complex communication environment. For example, the effect of human operators on edge nodes can be assessed by relying on the real contribution of human perception and decision making capabilities.

The Reference Architecture

Figure 2 illustrates the proposed solution for pseudo-dynamic testing, which encompasses the integration of simulated, emulated, and real components. Specifically, the sensory part made of IoT nodes and network interconnections among them can be simulated by means of several event-based simulators, such as TOSSIM, OMNET++, and NS3. Such tools have been extensively used in the academic literature in the context of WSNs, ad hoc wireless networks, and IoT. Despite being affected by the problem of accurately simulating wireless channels and sensors, they are a suitable solution for reproducing the behavior of a massive number of sensors characterizing an IoT deployment. Moreover, the simulators can be populated by statistics taken from measurement campaigns on

Pseudo-dynamic testing, which integrates simulation, emulation, and real components, represents the most effective solution for testing an extremely complex and large ecosystem such as the one composing the upcoming edge-fog cloud computing.

In order to exploit such a hybrid simulation approach, specific features should be offered to support interoperability and synchronization among the different simulators, as well as communication among simulated and emulated parts for correct evaluation of the scenario.

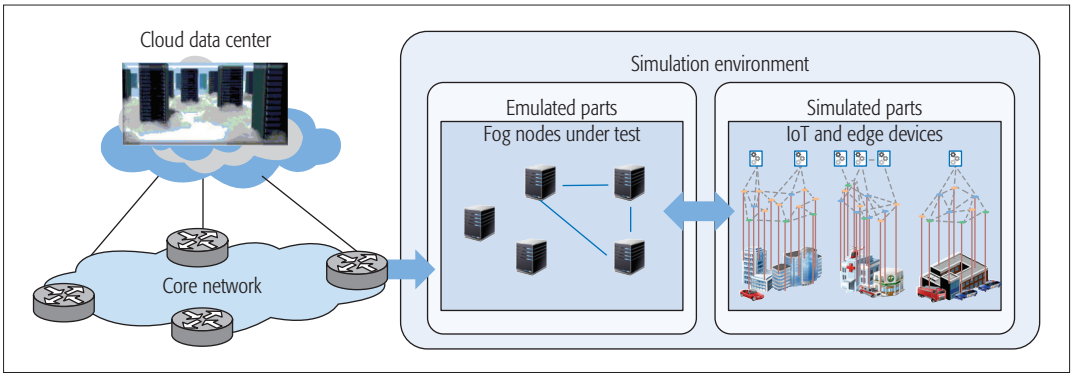

Figure 2. Simulation environment.

real sensor testbeds in order to improve the representativeness of the simulation. Each simulator is characterized by a set of key features and is specialized for selected scenarios (e.g., TOSSIM is dedicated to model sensor networks made of devices running the TinyOS operative system), while others are more general-purpose solutions (e.g., OMNET++ covers the broad types of networks from wired to wireless ones). Moreover, TOSSIM is more efficient in dealing with energy-related issues and obtains more realistic results than OMNET++. Finally, for simulating a realistic experimental scenario, specific sensor behaviors should be simulated, and human activity should also be reproduced to provide inputs to the IoT objects (e.g., by using agent-based models).

However, considering the complexity of IoT models to be simulated, the implementation of realistic large monolithic simulation systems from scratch, by using a single tool, could be excessively complex and time consuming. The simulation approach should exploit facilities for reuse that already exist: system components simulated by dedicated tools. For example, as for disaster management due to earthquakes, several simulators for disaster prediction, intensity analysis, damage estimation, and response have been implemented to provide meaningful outcomes. Therefore, it is more time- and cost-efficient to integrate multiple independent and heterogeneous simulation environments, each with its own features, languages, and operating systems, within a more complex federated simulation system, by enabling the reuse of already existing solutions for IoT simulation. By exploiting such federation dynamics and considering the elastic nature of modern federated computing environments, the resulting testbeds can reach a degree of scalability that is practically impossible in traditional architectures. Indeed, by relying on modern cloud-based runtime capabilities, we can use a virtually unlimited number of systems, located everywhere on the Internet, to host the virtual machines running the emulated or simulated entities, resulting in fully distributed testing architectures.

Moreover, according to the presented solution, the edge and fog parts of the envisioned ecosystem are the components under test. Therefore, they are reproduced using an emulation approach, by running the real software to be deployed on virtual nodes for representative machinery of the operational environment. In the same way, the networks connecting emulated nodes, the IoT objects and the external systems (i.e., core network and the cloud) can be emulated or simulated depending on the testing objectives.

On the other hand, in order to exploit such a hybrid simulation approach, specific features should be offered to support interoperability and synchronization among the different simulators, as well as communication among simulated and emulated parts for correct evaluation of the scenario. In this direction, the high-level architecture (HLA) represents the IEEE standard for distributed simulation [14]. It is an architecture developed to facilitate the reuse, interoperability, and synchronization of different simulation tools and assets, implementing a federation of interoperable simulation members called federates. HLA defines the runtime infrastructure (RTI) specification, which describes how to communicate within the federation. Each federate is represented by a simulation object model (SOM), which specifies the types of information that the federate can provide to or receive from the federation. The interactions among the federates are implemented by exploiting the publish/subscribe paradigm, and are described by the federation object model (FOM). Moreover, RTI features can be exploited in order to coordinate the synchronization among federates, which evolves according to a different emulated and simulated temporal model, as well as manages how fast the simulators advance. As Fig. 3 shows, in order to enable the interoperability among the federates, an *Ambassador* must be implemented for integrating each simulated and emulated component in the federation.

In particular, the interoperability among the simulation components with the emulated nodes is enabled by using a specific gateway (named *Emu-Ambassador*). The emulated fog and edge nodes run on virtual nodes (i.e., virtual machines or containers) with specific IP addresses assigned. When the simulation starts, each Ambassador registers the associated nodes to the federation, and a symbolic name is associated with each IP address, which is used to identify the involved component in the federation. The Emu-Ambassador operates as a bridge, which publishes and subscribes messages in the federation and interacts with the fog nodes by using the traditional socket interface.

Finally, in order to support the setup of such large-scale complex testing scenarios, massive virtualization technologies are exploited. Each scenario could involve several simulation tools

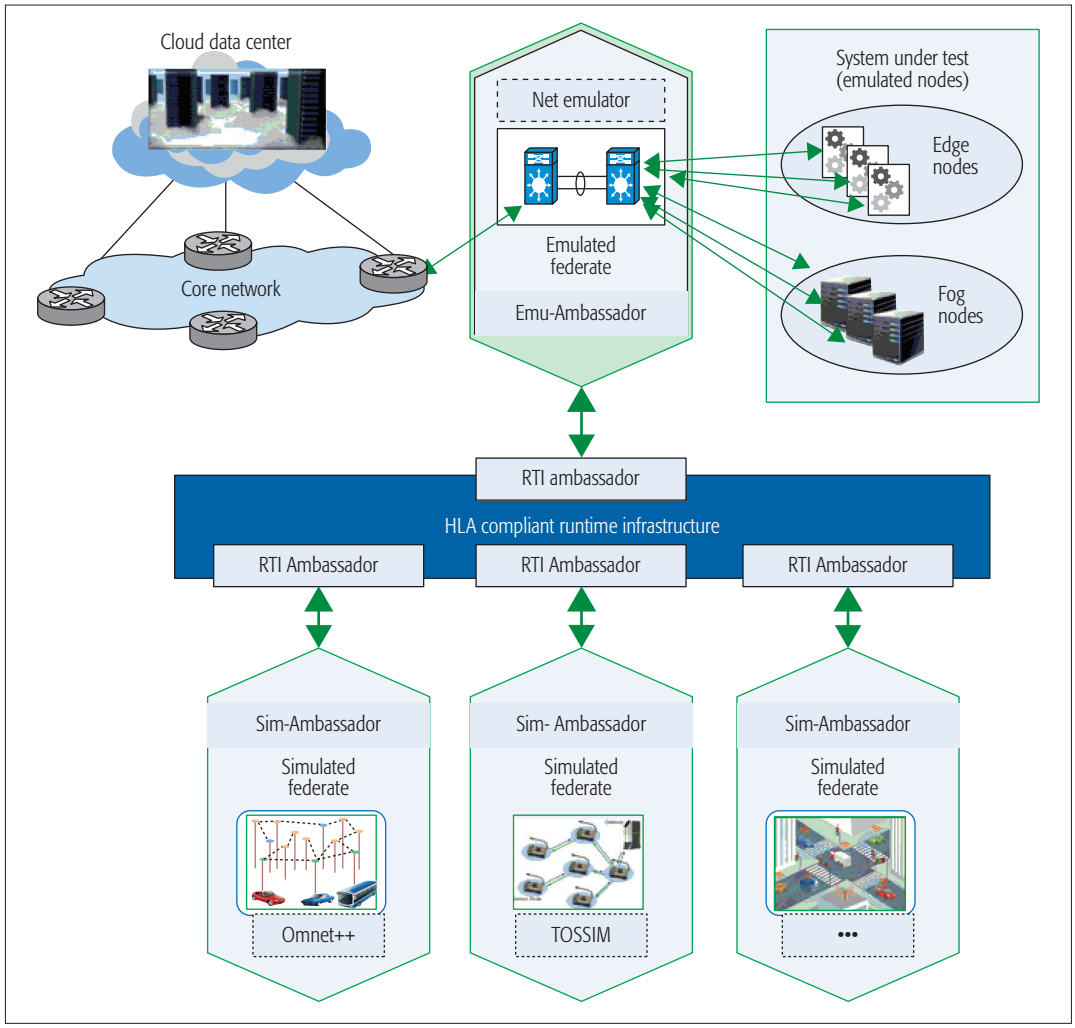

Figure 3. Hybrid simulation environment.

A distributed runtime platform, such as a cloud, can be adopted to provide the set of virtual nodes hosting the simulation tasks and the emulated nodes needed to reproduce a realistic edge-fog ecosystem. Moreover, specific mechanisms can be adopted to dynamically add or remove virtual nodes during the the experiments in order to simulate specific scaling behaviors within the edge-fog cloud ecosystem.

(which simulate dozens or hundreds of heterogeneous IoT objects), and edge and fog nodes to reproduce the experimental environment. A distributed runtime platform, such as a cloud, can be adopted to provide the set of virtual nodes hosting the simulation tasks and the emulated nodes needed to reproduce a realistic edge-fog ecosystem. Moreover, specific mechanisms can be adopted to dynamically add or remove virtual nodes during the the experiments in order to simulate specific scaling behaviors within the edge-fog cloud ecosystem.

Prototype Implementation Details

The fundamental architectural choices for a proof-of-concept implementation, entirely based on open-source components, of the proposed pseudo-dynamic testing framework, are described in the following. Specifically, in order to reduce both the number of virtual nodes to be scheduled (i.e., the required computational resources) and the boot-up time needed for the setup of the whole test scenario, lightweight Linux containers can be adopted for the simulation part, whereas for the emulated components that must be "physically" tested, KVM-based full virtualization technology is a better choice. A container is a packaged, self-contained, ready-to-deploy set of parts of an application, represented by a lightweight virtual image that can include both the middleware and business logic (binaries and libraries) needed to run it. Instead, a virtual machine (VM) is a full monolithic image, which requires guest OS images in addition to the binaries and libraries necessary for the applications. The life cycle of containers is managed by the Docker virtualization technology, supported by the OpenShift cloud platform as a service (PaaS) framework, whereas Kubernetes is used to orchestrate Docker containers on cluster nodes. Finally, OpenStack provides the needed cloud infrastructure as a service (IaaS). This solution acts as a container manager, which enables a registry for the images of the simulated components to be deployed on the virtual nodes. It can be exploited to keep track of the images executed on each node, and identify the virtual nodes on which the images are deployed and downloaded from the registry, needed for instantiating the test scenario. Moreover, the IaaS management layer is exploited in order to access and control the virtualization infrastructure, for supporting on-demand resources provisioning, running simulations on the cloud efficiently, and improving load-balancing capability. Specifically, the Chef technology has been used to simplify the configuration and deployment of the OpenShift nodes and emulated components under test on cloud resources.

In order to cope with the complexity and scale of the edge-fog cloud ecosystem, hybrid and distributed simulation practices supported by the virtualization technologies should be exploited to reproduce reality with a high degree of verisimilitude in order to set up realistic test environments.

The hybrid simulation platform proposed in Fig. 3 has been adopted to reproduce the edge-fog cloud computing scenario presented in Fig. 1, which includes about 1000 IoT nodes simulated through two different tools (i.e., NS3 and OMNET++), interoperating via HLA-RTI infrastructure, and nine edge-fog computing nodes, emulated by the Common Open Research Emulator (CORE) [15] and the OpenvSwitch network emulator. The whole solution has been hosted on the top of a cluster consisting of 8 Dell PowerEdge M610 Blade servers, each equipped with two Quad-Core Intel Xeon E5420 2.50 GHz processors, 16 GB of RAM memory, and 4 Gigabit Ethernet adapters, and running Linux CentOS 6.4. L3 Switching module (Dell M6220), which provides network connectivity to the nodes.

Conclusions

In order to cope with the complexity and scale of the edge-fog cloud ecosystem, hybrid and distributed simulation practices supported by virtualization technologies should be exploited to reproduce reality with a high degree of verisimilitude in order to set up realistic test environments.

Pseudo-dynamic simulation represents one of the most promising ways to support the key players in the edge-fog services market in dealing with the new challenges posed by IoT applications, including:

- The development of new scheduling algorithms for balancing load distribution between edge and cloud resources in order to minimize latency and maximize throughput
- The definition of effective resource management strategies for managing tenant environments, in which multiple application instances with different quality of service requirements share the same edge and fog nodes, networks, and sensing resources
- The definition of new policies for dynamic migration of processing tasks among edge, fog, and cloud nodes (based on the battery life of devices, the kind of operators, etc.) — the question is when and where to migrate what
- The introduction of authentication and authorization techniques that can work with multiple fog nodes belonging to different operators
- The simulation and analysis of the effects of advanced cyber attacks and catastrophic events that could compromise the edge-fog cloud infrastructure, as well as the definition and evaluation of possible recovery policies

References

[1] H. Gupta *et al.*, "iFogSim: A Toolkit for Modeling and Simulation of Resource Management Techniques in Internet of Things, Edge and Fog Computing Environments," June 2016, pp. 1–22; https://arxiv.org/pdf/1606.02007.pdf, accessed Feb. 2017.

[2] G. Kecskemeti *et al.*, "Modelling and Simulation Challenges in Internet of Things," *IEEE Cloud Computing*, vol. 4, no. 1, Jan.–Feb. 2017, pp. 62–69.

[3] C. Adjih *et al.*, "Fit IoT-Lab: A Large Scale Open Experimental IoT Testbed," *Proc. 2nd IEEE World Forum on Internet of Things*, 2015.

[4] L. Sanchez *et al.*, "Smartsantander: IoT Experimentation over a Smart City Testbed," *Computer Networks*, vol. 61, 2014, pp. 217–38.

[5] IOx — Cisco Framework; https://developer.cisco.com/site/iox/documents/developer-guide/?ref=overview, accessed Mar. 2, 2017).

[6] Q. Le-Trung, "Towards an IoT Network Testbed Emulated over OpenStack Cloud Infrastructure," *Proc. Int'l. Conf. Recent Advances in Signal Processing, Telecommu. & Computing*, 2017, pp. 246–51.

[7] V. Looga *et al.*," MAMMOTH: A Massive-Scale Emulation Platform for Internet of Things," *Proc. IEEE 2nd Int'l. Conf. Cloud Computing and Intelligence Systems*, 2012, pp. 1235–39.

[8] Simpleiotsimulator: The internetofthings simulator, http://www.smplsft.com/SimpleIoTSimulator.html, accessed Mar. 2, 2017.

[9] G. Brambilla *et al.*, "A Simulation Platform for Large-Scale Internet of Things Scenarios in Urban Environments," *Proc. 1st Int'l. Conf. IoT in Urban Space*, 2014, pp. 50–55.

[10] P Wehner, and D. Göhringer, "Internet of Things Simulation Using OMNeT++ and Hardware in the Loop," *Components and Services for IoT Platforms*, Sept. 2016, pp. 77–87.

[11] S. Tozlu *et al.*, "Wi-Fi Enabled Sensors for Internet of Things: A Ppractical Approach," *IEEE Commun. Mag.*, vol. 50, no. 6, June 2012, pp. 134–43.

[12] H. Gupta *et al.*, "iFogSim: A Toolkit for Modeling and Simulation of Resource Management Techniques in Internet of Things, Edge and Fog Computing Environments," *Report no. CLOUDS-TR-2016-2*,June 2016.

[13] M. Ficco *et al.*, "An HLA-Based Framework for Simulation of Large-Scale Critical Systems," *Concurrency Computation*, vol. 28, no. 2, 2016, pp. 400–19.

[14] IEEE Std. 1516-2000, "1516–2010 — IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) — Framework and Rules," Aug. 2010; http://ieeexplore.ieee.org/document/5553440/, accessed Nov. 2016.

[15] CORE — Common Open Research Emulato;: http://www.nrl.navy.mil/itd/ncs/products/core, accessed Jan. 2017.

Biographies

Massimo Ficco (massimo.ficco@unicampania.it) is an assistant professor at the Universita degli Studi della Campania Luigi Vanvitelli. He has a Ph.D. in computer engineering from the University of Napoli Parthenope, Italy. His research interests include security, cloud computing, and pervasive systems.

Cristian Esposito (christian.esposito@dia.unisa.it) is an adjunct professor at the University of Napoli Federico II and a research fellow at the University of Salerno. His research interests include information security and reliability, middleware, and distributed systems. He has a Ph.D. in computer engineering from the University of Napoli Federico II, Italy.

Yang Xiang (yang.xiang@swin.edu.au) received his Ph.D. in computer science from Deakin University, Australia. He is the Dean of Digital Research & Innovation Capability Platform, Swinburne University of Technology, Australia. His research interests include cyber security, which covers network and system security, data analytics, distributed systems, and networking.

Francesco Palmieri (fpalmieri@unisa.it) received his M.S. and Ph.D. degrees in computer science from he University of Salerno. He is an associate professor at the same university. His research interests concern networking protocols, architectures, and security. He directed the Networking Division of the University of Napoli Federico II and is a Senior Member of the Technical-Scientific Advisory Committee e of the Italian NREN.