


RESEARCH

Open Access

An agile and effective network function virtualization infrastructure for the Internet of Things



Diogo Menezes Ferrazani Mattos^{1,2*} , Pedro Braconnot Velloso² and Otto Carlos Muniz Bandeira Duarte²

Abstract

The processing and power-consumption constraints of the Internet of Things devices hinder them to offer more complex network services than the simple data transmission in smart city scenarios. The lack of complex services, such as security and quality of service, can even foster disasters in urban centers. In this paper, we propose the integration of complex network services from the IoT devices till a cloud environment through an agile and effective network function virtualization infrastructure of isolated IoT domains. Therefore, our proposal develops a simple gateway access node that virtualizes the domains to which the devices connect. A prototype for services of security and quality of service has been implemented and its evaluation shows that virtualization of the access node does not impact the performance of virtual network functions. The results also show that the proposal provides security for IoT devices, identifying malicious traffic with 99.8% accuracy, avoiding denial of essential services, and ensuring the quality of service.

Keywords: Internet of Things, Smart city, Network function virtualization, NFV, Service function chain, SFC

1 Introduction

In an increasingly interconnected world, it is estimated that 54% of the population lives in urban centers and this number will reach 66% by 2050 [1]. To respond to challenges arising from the accelerated growth of cities and to leverage the opportunities generated by continuous urbanization, government policies must converge efforts to focus on the sustainability of urban areas. An alternative to assure sustainability is to deploy the concept of smart cities, which consists of providing smart public services, smart energy and water distribution, smart health services and city governance based on data collection and decision-making platforms [2, 3]. In this context, smart city platforms are essential for sustainable urbanization and smart city governance.

Developing platforms for smart cities implies collecting data and acting using various connected devices in the urban context [4], according to the concept of the

Internet of Things (IoT) [2]. Thus, most IoT applications for smart cities consider a large number of low cost and low power consumption devices spread all over the city. As a consequence, the connection between IoT nodes depends on communication devices that are subject to severe processing and power consumption constraints. These constraints hamper offering more complex network services, such as security and quality of service, without which most applications of smart cities have their performance impaired or become even unfeasible. Although providing complex network services for IoT applications is crucial, it is not a simple task. Several works address this issue by using cloud or edge computing [5]. However, both approaches overload the access gateway, which might affect significantly the performance of IoT applications. Therefore, the main goal of this paper is to provide complex network services for IoT applications without introducing significant overhead to the access gateway. We achieve our goal by outsourcing the network functions to a virtualized infrastructure.

In this paper, we propose a network function virtualization infrastructure for the Internet of Things. The main idea is to define virtual IoT domains, in which

*Correspondence: menezes@midia.com.uff.br; diogo@gta.ufrj.br

¹Laboratório MídiaCom - TET/UFF, Universidade Federal Fluminense, Niterói, Brazil

²Grupo de Telemática e Automação - PEE/COPPE/UFRJ, Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brazil

each domain is a vertical slice of the transport network, running in the data collection until the data consumer delivery processes. Thus, in our proposal, we employ a *gateway* that multiplexes network access through the creation of virtual access points. The *gateway* connects the IoT devices of the wireless network to a Network Function Virtualization Infrastructure (NFVI). Hence, each IoT domain might deploy different network services implemented by Virtual Network Functions (VNF). The main advantage is to spare IoT devices and the *gateway* from all the packet processing overhead.

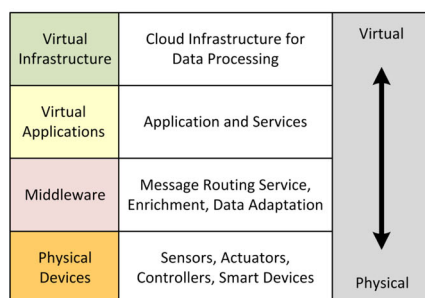
Unlike previous works that focus on data collection and processing platforms [2–4], or focus on theoretical models of IoT access-network management [6–8], our proposal is the virtualization of the access node for the IoT domains, combined with an agile and effective network function virtualization infrastructure for the deployment of the functions. The proposed infrastructure can assume packet-handling functions previously performed by IoT devices or by the *gateway*. Finally, we also propose traffic classification and reaction against attacks as virtual functions tailored to the context of the Internet of Things. We implement a prototype of our proposal and the performance evaluation shows that the delay of the communication between the access *gateway* node and the infrastructure is not significant, and that virtualization of the access point does not impact the performance of the virtual network functions. Results also show that the traffic classification VNF identifies the threat traffic in each IoT domain with 99.8% accuracy, and another VNF provides quality of service for IoT devices and avoids the denial of essential services. We also verify that the traffic classification is effective and that the infrastructure can isolate denial of service attacks in the IoT domain without affecting critical services.

The paper is organized as follows. The network service of the Internet of Things is characterized in Section 2. In

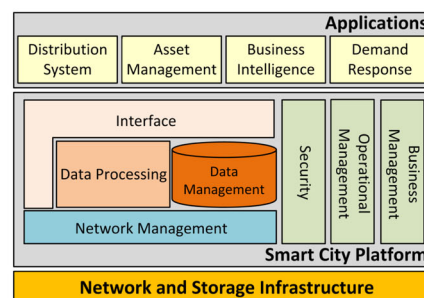
Section 3, we propose our network function virtualization infrastructure for the Internet of Things. Section 4 details the proposal prototype implementation. The evaluation of the proposal and the experimental results of the developed prototype are described and discussed in Section 5. The Section 6 discusses the related work. The Section 7 concludes the paper.

2 The network service on Internet of Things

The fundamental enabling technologies for the Internet of Things evoke four main layers: (i) physical identification and sensing devices; (ii) middleware platform; (iii) virtual applications and service composition; and (iv) virtual cloud infrastructure for device abstraction [9]. The reference architecture for the Internet of Things, Fig. 1a, highlights the transition between the physical devices and the software abstractions of IoT applications. In the lowest layer, sensors, identification tags and communication infrastructure are grouped in the category of the physical realization of the IoT, named physical devices. The middleware mediates the interaction between the IoT applications and the real world devices, acting as a request-adaptation software for each type of device [3]. The basic idea of middleware is to abstract devices into micro-services and, thus, to provide more complex services through the composition of simple micro-services. The adoption of the principles of service-oriented architecture (SOA) allows the decomposition of complex and monolithic systems into applications, which consist of more straightforward and well-defined micro-services [9, 10]. Applications, in turn, use the services provided and managed by the middleware to collect data and to act on devices in different contexts. Virtual applications run in a virtual infrastructure, usually hosted on cloud computing and data processing environments. In this context, cloud computing is an enabling technology for the Internet of Things, since it allows the timely processing of data



a Reference architecture for the Internet of Things.



b Reference architecture for the Smart Cities.

Fig. 1 Reference architecture for a smart city deployment. **a** Scheme of the relationship between physical devices and virtual applications to enable Internet of Things applications. **b** Smart city example architecture, where communication between sensors and applications interface to network access, processing and storage. The smart cities platform acts as a middleware

captured by IoT devices. Moreover, cloud computing-related technologies enable the IoT infrastructure to scale allocated resources, in real time, according to service requirements [11]. Therefore, IoT applications are the interface between the user and the IoT sensors and actuators. The IoT applications deployed as a composition of micro-services provided by the middleware, which interacts with the applications and the low-level devices. Nevertheless, in this standpoint, the IoT architecture conceals the network service specificities required for each type of application in a single infrastructure. Thus, real-time applications, such as smart industrial applications, rely on the same transport service as applications with less strict latency requirements, but sensitive to data privacy, such as house sensing devices.

Network service stands for the visible function provided by the network layer of the TCP/IP stack. Typically, the network service is characterized by the packet forwarding between two network edge nodes through the network core. Nevertheless, nodes in the core of the network perform new complex functions, such as filtering, quality of service provision, and policy enforcement that enhances the network service and changes the behavior of the network. However, IoT architectures focus on providing basic network services considering the resource restrictions of IoT devices. Thus, for the access network of IoT devices, a number of network technologies has been developed to meet different application requirements. For instance, Bluetooth Low Energy (BLE) and Zigbee for short-range communication, different versions of IEEE 802.11 for medium-range access networks, and Lora, SigFox, NB-IoT for wide-range communications. In the network and application layers, light-weight protocols for low power devices are used, such as 6LowPAN [12] and CoAP, a light version of HTTP. Generally, IoT gateways deploy at least two protocol stacks, an IoT-enabled stack, such as BLE or 6LowPan, to communicate with IoT devices, and TCP/IP stack to communicate with other devices and to forward traffic to the Internet. The translation between protocols is an example of network function for IoT network service.

In the context of IoT applications, smart cities are a good examples of the importance of providing more complex network services. In the smart cities concept, IoT devices are responsible for monitoring and sending information to data centers. The information is used to manage and govern cities more efficiently [3]. Figure 1b sums up a reference infrastructure for smart cities. In the context of smart cities, the network service has a fundamental role in the transport of the data from IoT devices to the platform for data processing and storage. This platform, shown in Fig. 1b, is deployed with cloud computing and provides interfaces between the collected data and the city applications for management and governance. Besides, real-time management and control applications, such as market

tools, operation management, and security management, must operate on sensed data with minimal delay and, thus, they have direct access to data. In this way, the transport network service for smart cities must be aware of the consumers of the data and provide differentiated services. It is also worth mentioning that the transport network infrastructure is also responsible for the adaptation of data. All these requirements on the network services imply a more complex network service which conflicts with the resource restrictions of the IoT devices. Therefore, the network service must be outsourced from the devices to a more robust and scalable network infrastructure.

Despite the success of the IoT paradigm, most of the traditional network mechanisms is not suitable for providing complex network services to IoT applications due to resource restrictions of devices and due to the number of connected devices [13]. Authentication, for instance, relies on mechanisms based on cryptography or on computational challenges and, thus, a resource-constrained device hampers, or even prevent, deploying such security service. Moreover, integrity and confidentiality also rely on computing-extensive mechanisms, such as cryptography and hash calculations. Therefore, the network service of IoT must be aware of devices' limitations and must take over the authentication, confidentiality, and integrity of the data communication [4]. Besides, the network service is also responsible for the access control of IoT devices. Access control is an important requirement, because the Distributed Denial of Service attack (DDoS) is one of the main threats to IoT, as a huge number of devices can generate a high rate of requests when orchestrated for an attack. Thus, the network service ought to isolate each IoT silo into disjoint network domains and to provide quality of service for each domain according to the criticality of services that the domain provides. It is also important to notice that while visibility is an unusual requirement in traditional network services, for the IoT network service real-time visibility of devices is a key requirement. The visibility refers to keeping device data always available and monitored in real time, which implies low latency and high goodput of IoT application as requirements. Visibility is a challenge in IoT because devices usually turn off to save battery and work with low throughput rates [14].

The key idea of our proposal is to complement the current Internet of Things platforms, developing a network service adapted to IoT applications through an access *gateway* node for IoT devices connected to a cloud of network function virtualization. Traffic from IoT devices is classified and chained into application-specific service chains. As a result, both devices and access *gateway* have computational resources released, since network packet handling, routing, policy enforcement, and queuing are carried out by the cloud. It is important for the network service to be able to abstract the access technology used

by devices and to be capable of translating IoT-dedicated application protocols, such as CoAP, to general-purpose application protocols, such as HTTP.

3 The proposed infrastructure for IoT

The network requirements of the devices in the Internet of the Things scenario are different from conventional networks due to the vertical integration of applications, called silos, and the magnitude of the amount of communication between devices (Machine to Machine - M2M) [15]. Each IoT silo is a slice of the IoT infrastructure that comprises since the devices till the applications that consumes the device data and actuate on them. Security, mobility, scalability, policy compliance and quality of service are essential requirements that are hampered in IoT due to devices' resource restrictions and the number of connected devices. In this sense, the goal of our infrastructure is to provide network services based on the function chaining [16, 17], capable of performing virtual network functions (VNF), which are outsourced by IoT devices. To this end, our proposal introduces the idea of a special access gateway to connect IoT devices to the network virtualization environment, shown in Fig. 2. As opposed to the idea of edge computing, in which the gateway performs several tasks to provide multiple network services, our special gateway performs exclusively the procedure of forwarding frames that arrive at its virtual network interfaces. Therefore, all packet processing, such as protocol adaptation, policy, and quality of service enforcement, or content analysis, is performed by the network virtualization infrastructure through virtual network functions. Sparing the gateway from all the complex tasks brings two main advantages: (i) the gateway is fully dedicated to the role of providing access to IoT devices, which allows it to serve a larger number of devices; and (ii) the delay added by the gateway is reduced significantly.

3.1 The access gateway node

The main idea of the gateway is to create virtual interfaces that act as different virtual access points for different domains of connected devices. The gateway runs on top of some hardware that has the physical resources to deploy virtual network interfaces, being a wireless IEEE 802.11, IEEE 802.15.4, an LTE base station, or other technologies. Virtualization implies isolation between each IoT domain, i.e., the devices that connect to each virtual network provided by the access gateway do not realize that share the infrastructure with devices from other virtual networks. The traffic from each IoT domain is isolated from the others from the access gateway to the final edge which is the data consumption node. In this way, IoT devices connect to the isolated virtual networks whose access is provided by the gateway. Therefore, the role of the gateway is to provide the physical realization of the access point, which is in line with the proposed 5G network and C-RAN (Cloud Radio Access Network) architectures [18]. After the device association, all frames are forwarded to the network virtualization infrastructure through a GRE tunnel (Generic Routing Encapsulation)¹. It is worth mentioning that the connection between the gateway and the network virtualization infrastructure may pass through the Internet and, therefore, it is subject to delay and packet losses.

An important advantage of configuring virtual access points is to allow a single gateway to provide connectivity service to different IoT domains. Hence, this approach ensures isolation at the physical layer since each virtual access point may have its own authentication credentials and access. With distinct access credentials, the participant of an IoT domain is not able to eavesdrop other virtual domains, since the encryption at the physical layer is different for each virtual access point.

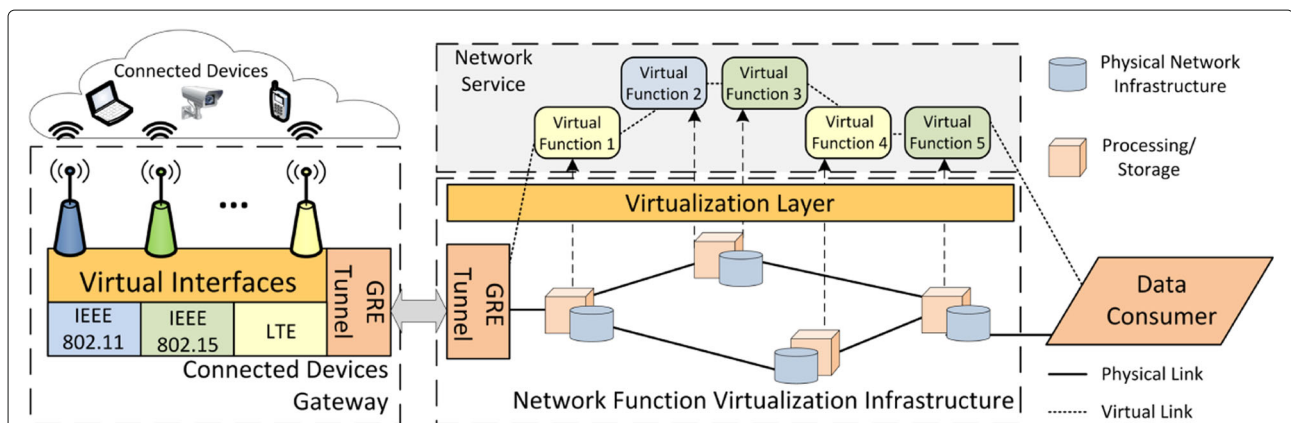


Fig. 2 The network function virtualization infrastructure for Internet of Things. The gateway for connected devices abstracts the access method through virtual interfaces tunneled to the Network Function Virtualization Infrastructure (NFVI). The gateway forwards the frames directly to the NFVI. Data processing at the network layer and above is performed by the network functions running over the NFVI

3.2 The network function virtualization infrastructure

Network Function Virtualization is characterized by the adoption of cloud computing technologies in the domain of transport networks, allowing the virtualization of network functions implemented as software [19]. Figure 2 shows the network function virtualization infrastructure based on the reference architecture of Network Function Virtualization Management and Orchestration (NFV-MANO) [20]. In the NFVI, the physical resources of the network are abstracted into virtual resources, and in virtual environments, software-based virtual network functions are chained to provide a complex network service. Therefore, the infrastructure might provide network services with different functionalities for each IoT silo, depending on the chaining of different VNFs. As a packet arrives in the NFVI, the packet is directed to one of the service function chains (SFC) in the infrastructure [21]. A service function chain consists of sequencing the virtual network functions that the packet must follow through. It is worth mentioning that the correct sequence of functions, as well as the appropriate choice of functions in the chain, define the features provided by the network service for packets crossing the NFVI. To direct the packets to the correct service function chain, the NFVI employs a classifier shortly after the packet entry point. The classifier tags the package with the correct SFC label responsible for handling the packet.

3.3 The virtual network functions

Virtual network functions consist of virtual environments that perform packet-handling functions. In a traditional Internet of Things architecture, these functions are executed by IoT devices or by the access *gateway* node between IoT devices and the Internet. In our proposed infrastructure, both IoT devices and the *gateway* do not need to perform the network functions, since these are outsourced to the NFVI in the form of virtual network functions.

3.4 The data consumer

The data consumer in our proposed infrastructure is any agent that accesses the IoT devices for retrieving sensed data or for actuating. In the case of an IP surveillance camera, for instance, the data consumer may be a web portal that accesses the images captured by the camera and forwarded and handled by the NFVI. In this case, NFVI enriches the transport network service by caching the data stream coming from the camera and also adapting the data flow of the camera to the standard supported by the end users or the web portal. In the case where IoT devices comprise a network of sensors and actuators, the data consumer might be an IoT middleware software that directly connects to the NFVI, without the need

to manage and control the access of the sensors to the network.

4 The proposal prototype implementation

In order to evaluate the performance of our infrastructure, we have deployed a prototype that implements all the components of the infrastructure, as detailed in the following sections.

4.1 The access gateway

As the main goal is to keep the gateway as simple as possible, thus our gateway is equipped with an IEEE 802.11n card and the virtualization of wireless network interfaces is accomplished by creating a virtual access points on top of the gateway node, using the `hostapd`² in line with the utilities `iw-utils`³.

The access *gateway* node for connecting IoT devices with the OPNFV environment is deployed on a computer equipped with an Intel Core i7-2600 processor, 16 GB RAM, an Intel gigabit network interface card for accessing the NFVI, and a low-cost wireless network interface card, IEEE 802.11n, Ralink RT2870 USB, for creating the virtual access points.

4.2 The NFV infrastructure

We use the Open Platform for Network Function Virtualization (OPNFV)⁴ as the network virtualization infrastructure. OPNFV is the reference platform of the NFV architecture standardized by the European Telecommunications Standards Institute (ETSI). The management of the virtualization layer is performed by OpenStack⁵. We use the Open vSwitch to classify, to tag the ingress flows and to forward them to the correct function chain in the NFVI.

The service function chain tagging can be performed with the Network Service Header (NSH) [22, 23]. However, the implementations available for the NSH protocol are still initial and the performance achieved is not satisfactory [16]. Therefore, a feasible alternative to establishing a coherent and performing service path is the usage of well-established tunneling protocols, such as GRE and VXLAN, to create the service function chains. In this paper, we use the GRE protocol to implement the chaining of service functions.

The OPNFV installed environment is composed of four nodes equipped with Intel Core i7-4770, 3.40 GHz, 32 GB RAM, and Intel *gigabit* network interface card. The environment configuration is organized with a node acting as an OpenStack cloud management controller and an OpenDaylight software-defined network controller⁶; the three other nodes are dedicated to processing and memory virtualization, through Linux KVM⁷, and to distributed disk storage through Ceph⁸. All software versions used in the environment

are the same as the OPNFV Danube 3.0 reference distribution⁹.

4.3 Virtual network functions

In this work, we implement the virtual network functions as virtual machines running Linux Ubuntu 16.04 with support for the software switch Open vSwitch¹⁰. In order to evaluate the effectiveness of our infrastructure to offer real network services, we have defined and developed two virtual network functions to provide security and QoS services to IoT applications. The first VNF classifies the traffic of each IoT domain between legitimate or malicious, using machine learning algorithms. The second enforces quality of service policies on traffic by forwarding flows by queues with previously allocated resources. Flow forwarding through correct queues is performed by rules on a software switch instantiated as a virtual network function.

5 Experimental evaluation

We evaluate the performance of our IoT infrastructure in regard of the overhead of the access gateway and the effectiveness of the proposed virtual network function. All the results in this section are reported as mean values within a 95% confidence interval.

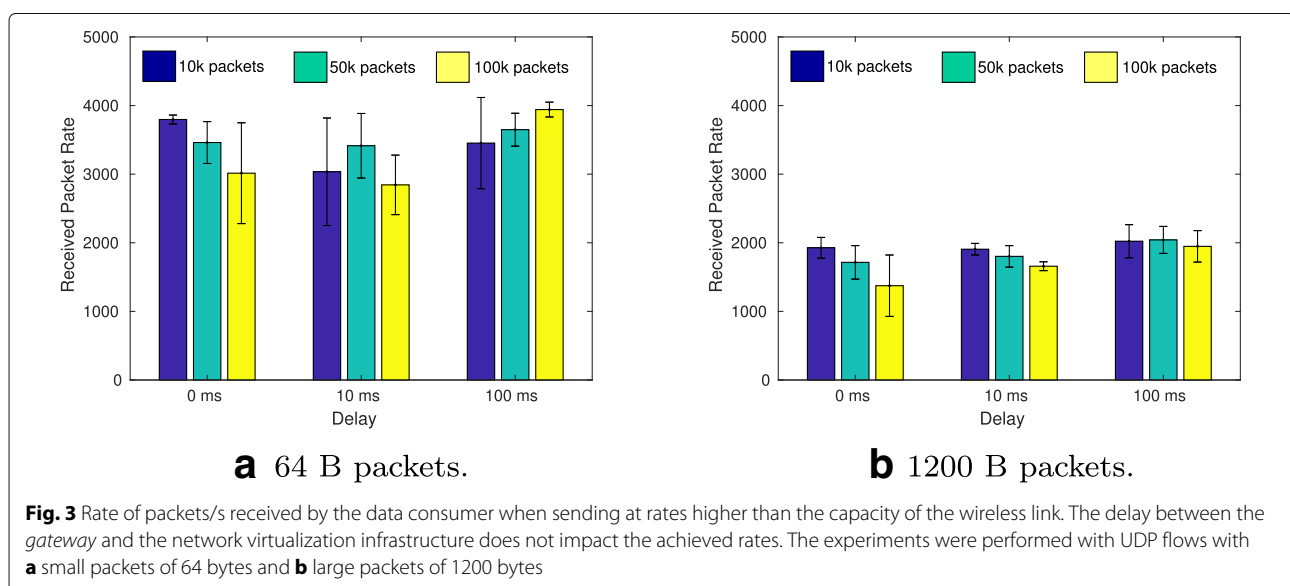
We have defined two main scenarios to evaluate our infrastructure. In the first one, we aim at assessing the performance and the overhead introduced by virtualizing multiple wireless interfaces at the same access point, which provide domain isolation to different IoT applications. In this scenario, to emulate the traffic of several IoT devices, we use two portable computers equipped with Intel Core i5-2410M processor, 6 GB of RAM and built-in wireless network interface cards¹¹. In the second scenario,

we analyze the two proposed virtual network functions, previously described. Thus, first we compare the performance of three classification algorithms based on machine learning to identify malicious traffic and legitimate traffic. Next, we evaluate the effectiveness of the proposed network virtualization infrastructure to protect the access network of IoT devices from attacks; and to provide quality of service to priority traffic. For the second scenario, we have also used a wireless IP camera D-Link DCS-5020L.

5.1 Access gateway performance

In the first experiment, we verify the performance of the connection between the network of IoT devices and the network virtualization infrastructure in emulating different delay values for the connection between the *gateway* and the NFVI. The idea of adding delay to the connection between the *gateway* and the NFVI is to simulate a connection over a wide-area network (WAN). The extra delay is added to the connection with a 10% variation using the τ_C ¹² (Traffic Control) tool. Therefore, this experiment measures the maximum rate of packets sent by IoT devices and received by the data consumers. Figure 3 compares the maximum rates achieved for sending 10 thousand, 50 thousand, and 100 thousand packets/s, for small packets of 64 B and for large packets of 1200 B. It is worth mentioning that in our prototype the maximum transfer unit (MTU) was set to 1280 B due to overloads with encapsulations. Traffic generation was accomplished by creating UDP packet flows at constant rates of packets per second.

In Fig. 3a, we observe that the maximum packet rate arriving at the *gateway* is 4000 packets/s, regardless of the sending rate of the IoT device. Another interesting result is that the delay between the *gateway* and the NFVI has



little influence on the packet arrival rate at the NFVI, as the rate of received packets remains almost the same while the delay varies from 0 to 100 ms. The results reinforce the idea of placing a gateway away from the NFVI does not affect the capability of the NFVI to process and to forward almost the same packet rate. When using a larger packet size, 1200 B, the limiting factor is the transmission rate achieved by the *gateway* wireless network card. Checking the actual baud rate achieved by the network card, set to operate in the IEEE 802.11g mode, the obtained rate was approximately 18 Mb/s.

Next we evaluate the virtualization overhead of the wireless network interface. The evaluation considers four simple scenarios for applying the virtualization infrastructure to IoT. It is important to highlight that we use laptops to emulate the traffic of several IoT devices. In the first scenario, a computer is connected to a virtual access point (*c1ap1*); in the second, two computers connect to a single virtual access point (*c2ap1*); the third, a computer joins one of two virtual access points (*c1ap2*); and finally, in the last scenario, two computers connect to two distinct virtual access points (*c2ap2*). Figure 4a shows the impact of virtualizing the wireless network access point on the communication delay experienced by IoT devices. We add a connection delay, from 0 – 100 ms, between the *gateway* and the NFVI to emulate different types of networks. Clearly, when no delay is added (0 ms), the impact of the virtualization delay is more important but is limited to 10 ms. As the connection delay between the *gateway* and the NFVI increases, the impact of wireless access virtualization is smoothed.

We also measured the aggregate bandwidth reached by IoT devices in each scenario, Fig. 4b. The aggregated bandwidth is not affected by the wireless network virtualization for 1200 B packets. First, we observe that with just one computer, adding a virtual AP does not impact the aggregate bandwidth. But most interesting, when we compare the scenarios with two computers, each one connecting to an isolated virtual AP, the aggregate bandwidth increases, instead of two computers sharing a single AP. The aggregate bandwidth increases in the scenario with two access points and two computers. Such behavior is due to the fact that when performing the virtualization, the scheduling of wireless nodes is performed by the kernel of the *gateway* operating system instead of performing the scheduling by the controller of the wireless network card. The wireless network card features a controller with low processing power since it is a low-cost wireless card.

5.2 Virtual network function performance

As mentioned above, we have deployed two virtual network functions as use cases to evaluate the effectiveness of our infrastructure in providing complex network services for IoT applications.

In this scenario, we consider that IoT devices shares their access network with other wireless users, smartphones, and devices with Internet access, such as an IP camera. We also assume that all IoT devices are susceptible to attacks and malware infection. Thus, a possible network protection is the instantiation of a virtual network function able to classify legitimate traffic and malicious traffic, and subsequently to enforce conformance policies to the traffic.

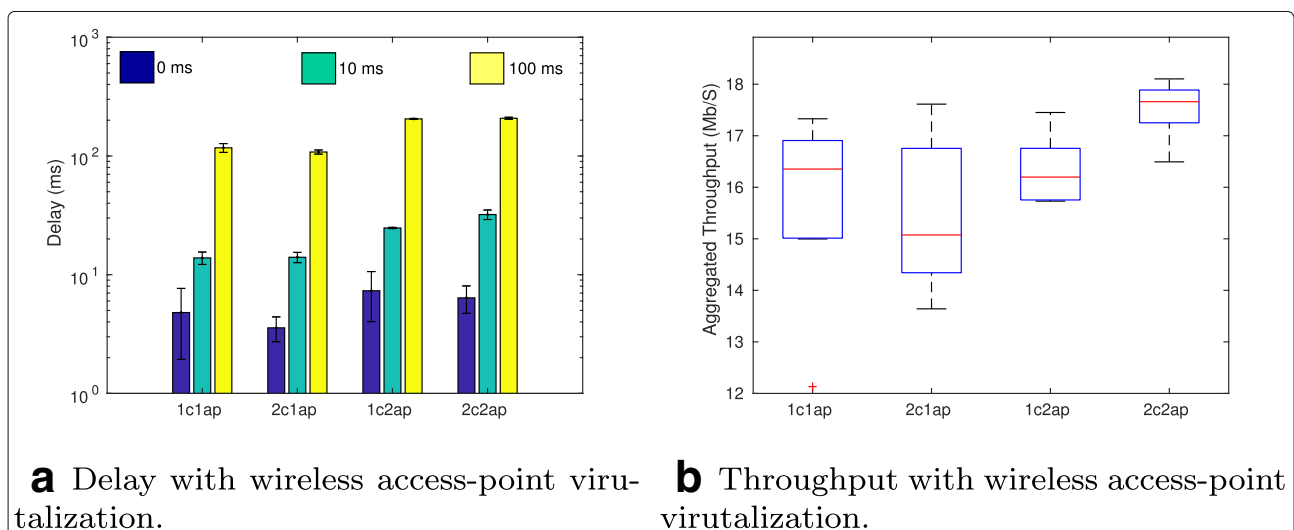


Fig. 4 Virtualization of the wireless link. Evaluation of scenarios with one computer and a one single virtual access point (1c1ap); two computers and one access point (2c1ap); two computers and one access point (2c1ap); and two computers and two access points (2c2ap). **a** The added delay which is perceived by the devices is lower than 10 ms, even when the delay between the *gateway* and the NFVI is 100 ms. **b** The aggregated bandwidth when using two computers and two access points is greater than in other scenarios

In the first experiment, we compare the performance of three classification algorithms based on machine learning to identify malicious traffic and legitimate traffic from devices connected to the IoT network. The traffic classification between legitimate and malicious flows depends on the training and evaluation of classification algorithms. For this purpose, a training and testing dataset, composed of legitimate data and labeled attack data, was created¹³. Legitimate data were collected during the daily use of IP cameras and wireless users in the laboratory of Grupo de Teleinformática e Automação (GTA/UFRJ). Particularly, the cameras were accessed to generate video streaming data, accesses to FTP servers for video and photo transfer, and date synchronization via NTP. The attack data were obtained from the data collected by Garcia et al. in a study on the behavior of *botnets* [24]. The dataset consists of flows identified by a tuple composed of the source and destination IP address, source and destination transport ports, and transport protocol [25]. Our compiled dataset counts with traffic from 15 devices both from captured laboratory usage traffic and the *botnet* dataset. The features that were used for generating the dataset are a subset of the numerical features provided by the *flowtbag*¹⁴ network characterization application. We also added some tagging features which indicates whether the flow comes from one of the top 10 most accessed services, in terms of the number of flows.

For traffic classification, we model the flows' data using time-related and size-related features, such as flow duration, flow size in the forward direction, flow size in the backward direction, average idle time, average packet size. The 10 most accessed services are the services that concentrate about 90% of all data traffic volume but comprise

just 1% of all services that were accessed in the dataset. In order to enhance the classification performance, we enrich the flows' data adding features that show whether the flow belongs to the most common services on the network. We added 10 new features into the dataset, that marks if the flow belongs to the specific service. It is important to use tagging features to keep some information about the service whereas we still are able to calculate correlation and to apply the principal components analysis over the dataset [25].

Figure 5a compares the accuracy of three classification algorithms: probabilistic neural network (PNN); multilayer perceptron neural network (MLP) and Boosted Decision Trees [26]. We applied the algorithms for all features and for the scenarios in which the dimensionality of the problem was reduced using Principal Component Analysis (PCA). Note that classification using boosted decision trees with the *gradient boosting* algorithm presents the best accuracy when compared to neural networks. Such behavior is expected given the discrete nature of the network flows' data, where binary data segmentation, as done by the decision tree algorithm, leads to accurate classification rules. The relation between true positive and false positive rates of the classification algorithms, the ROC curve¹⁵ shown in Fig. 5b, reveals that decision trees have an area under the curve (AUC) very close to 1, signaling that data classification incurs low rate of false positives and false negatives for the attack class. Reducing the dataset dimensionality with PCA for only 8 major components kept 99% of the dataset information and leads to a classification problem of which accuracy is almost equal to the full set of features classification problem. Table 1¹⁶ shows the performance of the decision

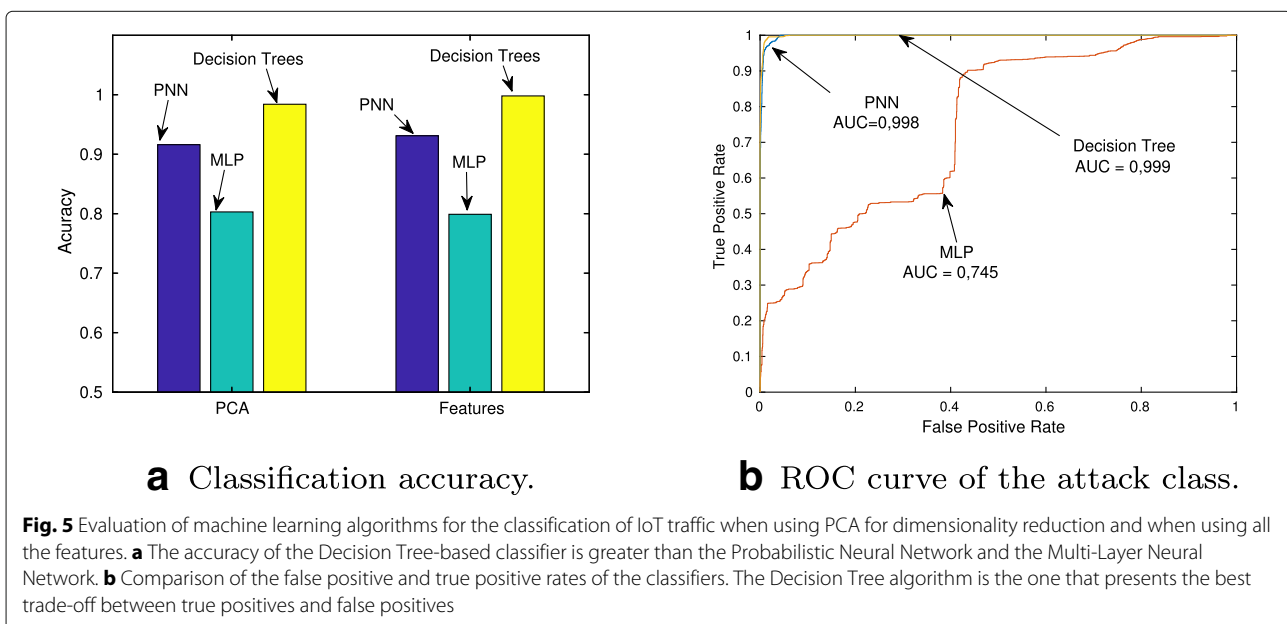


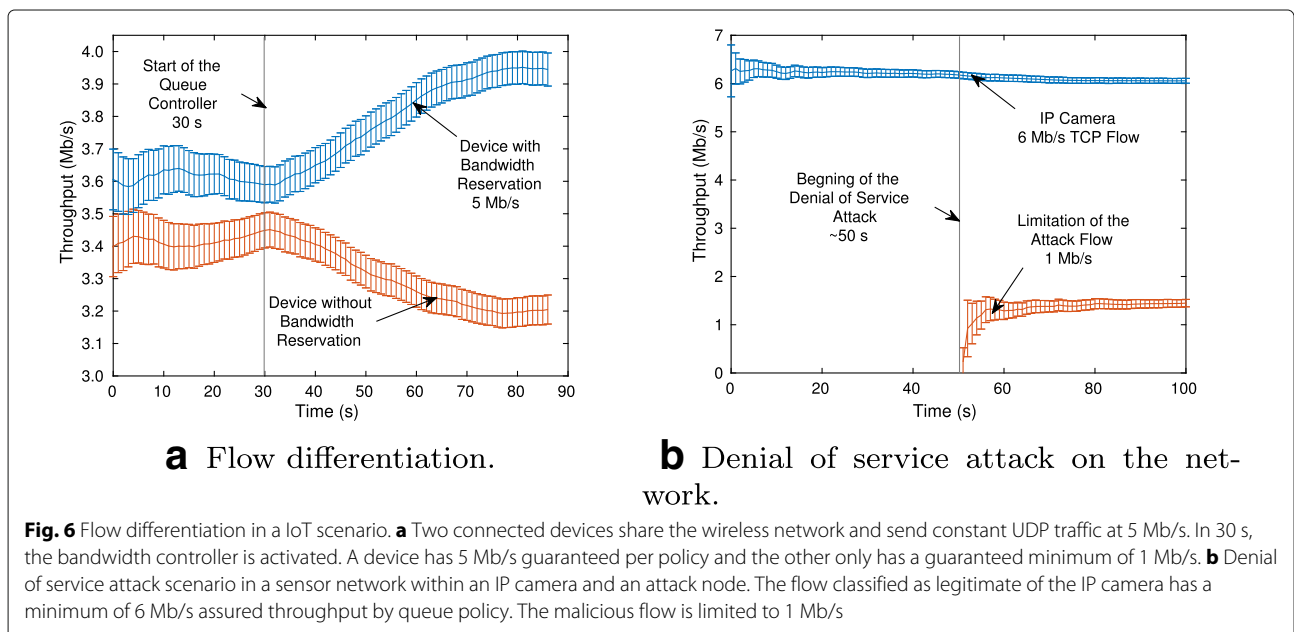
Table 1 Decision tree with dimensionality reduction classification

| | TP | FP | TN | FN | Precision | Sens. | Spec. |
|-----------|--------|------|--------|------|-----------|-------|-------|
| Malicious | 14,208 | 290 | 3,995 | 0.00 | 0.98 | 1.00 | 0.93 |
| Normal | 3995 | 0.00 | 14,208 | 290 | 1.00 | 0.93 | 1.00 |

trees, with dimensionality reduced by PCA, in a cross-evaluation in 10 rounds. The nominal accuracy was 0.998, with a sensitivity of 1.0 in malicious traffic.

In the next experiment, we evaluate the ability of our virtualization infrastructure at providing quality of service (QoS), taking countermeasures and applying policies to packet flows. For this purpose, we use two notebooks connected to the wireless network to emulate the traffic of two sets of IoT devices with different QoS requirements when accessing the network. Service differentiation is provided by a VNF that re-directs the flows from IoT devices to distinct queues, which are implemented in a software switch *Open vSwitch*. During the experiment, both nodes execute a UDP flow at their maximum rate. One of the nodes is routed to a queue with minimum bandwidth, assured at 5 Mb/s, which characterizes a high priority queue. The other node has a minimum bandwidth assured at 1 Mb/s. Figure 6a points out that after 30 s, the queues are configured with their bandwidth constraints and the flows become conformed to their limits. Figure 6a shows that the traffic that flows through the queue with the most reserved bandwidth reaches a total throughput, useful more encapsulation, up to 5 Mb/s, to the detriment of the other node that does not have enough dedicated resources.

The last experiment aims at evaluating the performance of the proposed infrastructure under a Denial of Service (DoS) attack. Hence, we combine the two virtual network functions to protect IoT applications from DoS attacks. It is worthy to note that our queue-based approach differs from classical approach since our proposal performs traffic policy enforcement in a virtual network function running on a centralized NFVI, unlike other proposals that enforce traffic policies on the access gateways. This difference where applying the queue limits enables the outsourcing of gateway functions to the NFVI but implies an indirect bandwidth control on the access gateway. Figure 6b shows the scenario where an IP camera is sending a continuous TCP video stream of approximately 6 Mb/s when a DoS attack occurs. The denial of service attack is configured as a UDP flow of 20 Mb/s. Although the camera hardware is a more resource-restricted configuration, compared to the attacking notebook computer, the virtual network function assures a stable transmission rate for the camera, while limiting the malicious traffic to approximately 1 Mb/s. Therefore, we avoid the degradation of the service provided by the camera. The protection provided by our infrastructure is accomplished due to the existence of a special *gateway* that virtualizes the wireless network, and the availability of processing and memory isolated resources in the infrastructure to handle the amount of traffic of the DoS attacker. It is worth mentioning that our proposal is able to correctly classify the malicious traffic in real-time, which allows redirecting it to the service function chain where it can be shaped with a strict queue policy.



6 Related work

Petrolo et al. argue that smart cities are an application of the Internet of Things in the domain of cities and they highlight the benefits of integrating various connected devices [2]. The authors define the concept of Cloud of Things (CoT) that consists of using cloud computing environments to provide a platform for integrating data silos of IoT. Santana et al. ratify the idea of smart cities composed of connected devices and argue that the amount of data generated in smart cities is large and requires the use of specific techniques for Big Data processing [3]. Thus, Santana et al. envision the architecture of smart cities as a clustering of cloud processing with cyber-physical systems (CPS).

Zhang et al. identify several applications that run on smart city platforms such as smart energy, smart environment, smart industry, smart home and smart utility [4]. The authors argue that it is necessary to provide a management and data processing architecture aware of data security and privacy requirements.

Atzori et al. compare the characteristics of radio frequency identification (RFID) systems, wireless sensor networks, and RFID networks [9]. The authors point out that RFID systems are small, low cost and energy is not a limiting factor. Wireless sensor networks have high radio coverage and communication does not require the presence of a reader, while RFID, reader and sensor systems are asymmetric. RFID sensor networks enable detection, computation, and communication in a passive system. In turn, Adelantado et al. investigate the limitations of the LoRaWAN [27] standard. Thus, each different access network for the IoT devices presents distinct characteristics and network requirements.

Quin et al. propose an architecture of a *middleware* based on software-defined networking for the Internet of Things. The idea is to provide multiple network environments for IoT to meet network demands with different requirements [6]. The proposal is a *middleware* capable of monitoring the network and adapting it according to the needs of the executed tasks. Hence, the proposal monitors the network and uses network calculus to predict the change in performance. The BlackSDN proposal, in turn, proposes the use of software-defined networking to provide security in IoT networks by encrypting both content and packet headers [28]. To this end, the network packet routing is performed by the network controller that is a trust anchor and, therefore, holds the key to decipher the packets and install rules in the network according to the correct routing.

Bizanis and Kuipers investigate the use of virtualization and software-defined networking in the Internet of Things environments [7]. The authors conclude that network virtualization and software-defined networking are enablers of the Internet of Things, but they just focus on

the use of these technologies as a way of managing the flows generated by IoT devices. Ojo et al. argue that traditional network architectures and protocols are inefficient to support the high level of scalability, the large amount of traffic, and the mobility of IoT devices. In addition, it is difficult to manage the amount of generated data, which can cause problems and disruptions in the network service [8]. Ojo et al. propose to apply plane separation paradigm to the IoT packet forwarding. They propose to run the data plane on software-defined networking, which interfaces with IoT devices through an SDN-enabled gateway. The control plane runs on an NFV-enabled environment. Although they propose to use SDN and NFV technologies, only SDN devices perform packet forwarding, and the NFVI does not perform complex networking services or isolate different IoT domains. Nevertheless, the authors do not focus on the creation of isolated IoT domains covering the network service from the node access to the data consumption.

ONAP¹⁷ provides a platform for real-time orchestration and automation of physical and virtual network functions that enable the quick automation and support full lifecycle management of services. ONAP is expanding the ETSI NFV Management and Orchestration (MANO) specifications with additional telecommunication capabilities. ONAP includes infrastructure controllers, such as software-defined networking (SDN) and NETCONF network controllers [29]. CORD¹⁸ (Central Office Re-architected as a Datacenter) combines NFV, SDN and commodity cloud elasticity technologies to bring the datacenter architecture and cloud agility to the telecom environment. CORD's design is hard coupled between its functional elements, which makes it less flexible to upgrade or use new component plugins when compared to OPNFV [29]. Both technologies are complementary, and we map our proposal on an ONAP and CORD environment as policy directives defined on ONAP orchestrator, which are deployed on the telco network as VNFs on top of CORD's VNF manager and as applications on top of the ONOS SDN controller.

The transport network infrastructure proposed in this paper is based on a simple *gateway* that routes all packets of IoT devices to a network function virtualization environment. Unlike previous proposals, this work focuses on the data transport network, regardless of the use of a specific middleware or an IoT platform. The functions carried out by the transport network include packet routing, policy enforcement, and data adaptation, such as protocol translations. In addition, another contribution is the reaction against network anomalies in real time through the execution of machine learning mechanisms coupled with virtual network functions, which was deployed in our platform.

7 Conclusion

The number of devices connected to the Internet of Things is increasing and the criticality of the transported data is also growing. However, the data transport network of the Internet of Things is deprecated in comparison to the acquisition and processing of data layers. In this paper, we proposed a network function virtualization infrastructure that allows the agile and effective deployment of virtual functions for outsourcing network tasks from the IoT devices to the cloud. The proposal developed the idea of a virtualized access *gateway* node, capable of creating independent and isolated domains of connected devices. In each domain independent, isolated and tailored network functions are applied to respond to the performance and security requirements of each Internet of Things application. A prototype of the proposed infrastructure was developed and evaluated. The results demonstrated that the virtualized access *gateway* node does not introduce performance losses for the access of the IoT devices. It has further been found that latency between the access node and the infrastructure can be substantially high without performance losses. The results of the experiments show that even while performing the virtual queue policy function in a virtualized environment, the effects of policy enforcement are visible on the physical interface, avoiding the overuse of resources. Since the proposal does not apply protocols that execute flow control, the variation in the delay on the communication between the gateway and network function virtualization infrastructure (NFVI) does not interfere with the received packet rate and generates little influence for the processing of the packets in the virtualized environment. Another important point is that the proposal is based on traditional, consolidated and widely developed tools, such as the use of GRE, which allows higher performance than experimental tools that are still in development, such as NSH. However, the central idea of outsourcing network services from the IoT gateway to a central processing infrastructure remains true even in new virtualization proposals for network function virtualization infrastructures that rely on orchestrators and complex and federated platforms service providers.. Finally, a use case of the proposed infrastructure was developed, in which a virtual network function was able to classify traffic between legitimate or threats with 99.8% accuracy, and then another virtual function assured quality of service and limited a denial of service attack by preventing the essential service of an IP camera from being degraded.

Future work intends to develop new virtual network functions, such as application protocol adaptation functions and intrusion detection systems based on Deep Packet Inspection for the Internet of Things. Moreover, as future work, we intend to port our deployment of the *gateway* to other Linux-based platforms, such as OpenWRT¹⁹,

which which targets low-cost resource-constrained access points.

Endnotes

- ¹ Available at <https://tools.ietf.org/html/rfc2784>.
- ² Available at <https://w1.fi/hostapd/>.
- ³ Available at <http://drvbp1.linux-foundation.org/~mcgrof/rel-html/iw/>.
- ⁴ Available at <http://www.opnfv.org/>.
- ⁵ Available at <http://www.openstack.org/>.
- ⁶ Available at <https://www.opendaylight.org/>.
- ⁷ Available at <http://www.linux-kvm.org/>.
- ⁸ Available at <http://ceph.com/>.
- ⁹ Available at <https://www.opnfv.org/software/downloads/release-archives/danube-3-0>.
- ¹⁰ Disponível em <http://docs.openvswitch.org/>
- ¹¹ The notebooks are equipped with hardware more powerful than ordinary IoT devices. For the sake of experiment reproducibility, the notebooks replay the IoT traffic because they are controllable by out off-band channels, and they have enough memory to store the dataset used to reproduce the experiments for several turns. Thus, the notebooks simulate the traffic of an ensemble of 15 devices.
- ¹² Documentation at <http://lartc.org/manpages/tc.txt>.
- ¹³ The dataset can be obtained through contact with the authors.
- ¹⁴ <https://github.com/DanielArndt/flowtbag>
- ¹⁵ The receiver operating curve, ROC curve, plots the true positive rate (TPR) against the false positive rate (FPR) for various discrimination thresholds of a binary classifier system. It shows the system ability to classify data as a binary classification.
- ¹⁶ TP: True Positive; FP: False Positive; TN: True Negative; FN: False Negative; Sens.: Sensitivity; Spec.: Specificity.
- ¹⁷ Available at <https://www.onap.org/>.
- ¹⁸ Available at <https://opencord.org/>.
- ¹⁹ Available at <https://openwrt.org/>.

Acknowledgements

We acknowledge colleagues of the Grupo de Teleinformática e Automação (GTA/UFRJ) and MídiaCom (UFF) for their incentives and suggestions.

Funding

This research is partially funded by CNPq, CAPES, FAPERJ, and FAPESP (2015/24514-9, 2015/24485-9, and 2014/50937-1).

Availability of data and materials

Please contact author for data requests.

Authors' contributions

All authors read and approved the final manuscript.

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 19 October 2018 Accepted: 10 February 2019

Published online: 15 March 2019

References

1. United Nations. World urbanization prospects: The 2014 revision, highlights. Department of Economic and Social Affairs. Popul Div U Nation. 2014. Available from <https://esa.un.org/unpd/wup/publications/files/wup2014-highlights.pdf>. Accessed 19 Feb 2019.
2. Petrolo R, Loscri V, Mitton N. Towards a smart city based on cloud of things, a survey on the smart city vision and paradigms. *Trans Emerg Telecommun Technol*. 2017;28(1).
3. Santana EFZ, Chaves AP, Gerosa MA, Kon F, Milojicic DS. Software platforms for smart cities: concepts, requirements, challenges, and a unified reference architecture. *ACM Comput Surv*. 2018;50(6):78:1–78:37. <https://doi.org/10.1145/3124391>. <http://doi.acm.org/10.1145/3124391>.
4. Zhang K, Ni J, Yang K, Liang X, Ren J, Shen XS. Security and privacy in Smart City applications: Challenges and solutions. *IEEE Commun Mag*. 2017;55(1):122–9.
5. Velasquez K, Abreu D, Assis M, Senna C, Aranha D, Bittencourt L, et al. Fog orchestration for the internet of everything: state-of-the-art and research challenges. *J Internet Serv Appl*. 2018;9(1):14.
6. Qin Z, Denker G, Giannelli C, Bellavista P, Venkatasubramanian N. A software defined networking architecture for the Internet-of-Things. In: 2014 IEEE Network Operations and Management Symposium (NOMS). Krakow: IEEE; 2014. p. 1–9.
7. Bizanis N, Kuipers FA. SDN and virtualization solutions for the Internet of Things: A survey. *IEEE Access*. 2016;4:5591–606.
8. Ojo M, Adami D, Giordano S. A SDN-IoT architecture with NFV implementation. In: 2016 IEEE Globecom Workshops (GC Wkshps). Washington, DC: IEEE; 2016. p. 1–6.
9. Atzori L, Iera A, Morabito G. The Internet of Things: A survey. *Comput Netw*. 2010;54(15):2787–805.
10. Dos Santos X, Rafael J, Wauters T, Volckaert B, De Turck F. Resource provisioning for IoT application services in smart cities. In: CNSM2017, the 13e International Conference on Network and Service Management. Tokyo: IEEE; 2017. p. 1–9.
11. Flinta C, Johnsson A, Ahmed J, Moradi F, Pasquini R, Stadler R. Real-time resource prediction engine for cloud management. In: 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM). Lisbon: IEEE; 2017. p. 877–8.
12. Yibo C, Hou KM, Zhou H, Shi HI, Liu X, Diao X, et al. 6LoWPAN stacks: A survey. In: 2011 7th International Conference on Wireless Communications, Networking and Mobile Computing. Wuhan: IEEE; 2011. p. 1–4.
13. Suo H, Wan J, Zou C, Liu J. Security in the internet of things: A review. In: 2012 International Conference on Computer Science and Electronics Engineering. vol. 3. Hangzhou: IEEE; 2012. p. 648–651.
14. Internet of things. Applications and challenges in technology and standardization. *Wirel Pers Commun*. 2011;58(1):49–69.
15. Omnes N, Bouillon M, Fromentoux G, Grand OL. A programmable and virtualized network IT infrastructure for the Internet of Things: How can NFV SDN help for facing the upcoming challenges. In: 2015 18th International Conference on Intelligence in Next Generation Networks. Paris: IEEE; 2015. p. 64–9.
16. Sanz I, Mattos D, Duarte O. SFCPerf: An automatic performance evaluation framework for service function chaining. In: NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium. Taipei: IEEE; 2018.
17. Salah K, Calyam P, Boutaba R. Analytical model for elastic scaling of cloud-based firewalls. *IEEE Trans Netw Serv Manag*. 2017;14(1):136–46.
18. Checko A, Christiansen HL, Yan Y, Scolari L, Kardaras G, Berger MS, et al. Cloud RAN for mobile networks: A technology overview. *IEEE Commun Surv Tutor*. 2015;17(1):405–26.
19. Medhat AM, Taleb T, Elmangoush A, Carella GA, Covaci S, Magedanz T. Service function chaining in next generation networks: State of the art and research challenges. *IEEE Commun Mag*. 2017;55(2):216–23.
20. ETSI. ETSI GS NFV-MAN 001: Network functions virtualisation; management and orchestration. Tech rep. 2014. Available: http://www.etsi.org/deliver/etsi_gs/NFVMAN/001_099/001/01.01.01_60/gs_NFV-MAN001v010101p.pdf. Accessed 18 Oct 2018.
21. Andreoni Lopez M, Mattos DMF, Duarte OCM. Evaluating allocation heuristics for an efficient virtual network function chaining. In: 7th International Conference on the Network of the Future (NoF). Buzios: IEEE; 2016. p. 1–5.
22. Kulkarni S, Arumathurai M, Ramakrishnan KK, Fu X. Neo-NSH: Towards scalable and efficient dynamic service function chaining of elastic network functions. In: 20th Conference on Innovations in Clouds, Internet and Networks (ICIN). Paris: IEEE; 2017. p. 308–12.
23. Quinn P, Elzur U. "Network service header," Working Draft, IETF Secretariat, Internet-Draft draft-ietf-sfc-nsh-12; 2017. Available: <http://www.ietf.org/internet-drafts/draft-ietf-sfc-nsh-12.txt>. Accessed 18 Oct 2018.
24. Garcia S, Grill M, Stiborek J, Zunino A. An empirical comparison of botnet detection methods. *Comput Secur*. 2014;45(Supplement C):100–23.
25. Andreoni Lopez M, Silva R, Alvarenga I, Rebello G, Sanz I, Lobato A, et al. Collecting and characterizing a real broadband access network traffic dataset. In: 2017 1st Cyber Security in Networking Conference (CSNet'17). Rio de Janeiro: IEEE; 2017.
26. Boutaba R, Salahuddin M, Limam N, Ayoubi S, Shahriar N, Estrada-Solano F, et al. A comprehensive survey on machine learning for networking: evolution, applications and research opportunities. *J Internet Serv Appl*. 2018;9(1):16.
27. Adelantado F, Vilajosana X, Tuset-Peiro P, Martinez B, Melia-Segui J, Watteyne T. Understanding the limits of LoRaWAN. *IEEE Communications Magazine*. 2017;55(9):34–40. <https://doi.org/10.1109/MCOM.2017.1600613>.
28. Chakrabarty S, Engels DW, Thathapudi S. Black SDN for the Internet of Things. In: 2015 IEEE 12th International Conference on Mobile Ad Hoc and Sensor Systems. Dallas: IEEE; 2015. p. 190–8.
29. Rossem SV, Sayadi B, Roullet L, Mimidis A, Paolino M, Veitch P, et al. A vision for the next generation platform-as-a-service. In: 2018 IEEE 5G World Forum (5GWF). Silicon Valley: IEEE; 2018. p. 14–19.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com