Nano Communication Networks 17 (2018) 45–62

Contents lists available at ScienceDirect

Nano Communication Networks

journal homepage: www.elsevier.com/locate/nanocomnet

Nano-networks communication architecture: Modeling and functions

Akram Galal *, Xavier Hesselbach

Department of Network Engineering, Universitat Politècnica de Catalunya (UPC) BARCELONATECH, 08034 Barcelona, Spain

ARTICLE INFO

Article history:
Received 27 February 2018
Received in revised form 7 June 2018
Accepted 15 July 2018
Available online 29 July 2018

Keywords:
Nano-machine
Nano-network
Internet of nano-things
Software defined networking
Fog computing
Network function virtualization

ABSTRACT

Nano-network is a communication network at the Nano-scale between Nano-devices. Nano-devices face certain challenges in functionalities, because of limitations in their processing capabilities and power management. Hence, these devices are expected to perform simple tasks, which require different and novel approaches. In order to exploit different functionalities of Nano-machines, we need to manage and control a set of Nano-devices in a full Nano-network using an appropriate architecture. This step will enable unrivaled applications in the biomedical, environmental and industrial fields. By the arrival of Internet of Things (IoT) the use of the Internet has transformed, where various types of objects, sensors and devices can interact making our future networks connect nearly everything from traditional network devices to people. In this paper, we provide an unified architectural model of Nano-network communication with a layered approach combining Software Defined Network (SDN), Network Function Virtualization (NFV) and IoT technologies and present how this combination can help in Nano-networks' context. Consequently, we propose a set of functions and use cases that can be implemented by Nano-devices and discuss the significant challenges in implementing these functions with Nano-technology paradigm and the open research issues that need to be addressed.

1. Introduction

Nano-technology is providing a new set of tools to the engineering community to design Nano-machines. Nano-machines are the basic functional Nano-devices that able to perform very simple tasks. A standalone Nano-device is highly constrained by limited energy, processing and communication range. Besides Nano-network is the network of Nano-machines, it expands the capabilities of a single Nano-device by providing a way to cooperate and share information. In general way, Nano-communication is the exchange of information at the Nano-scale on the basis of any wired or wireless interconnection of Nano-machines in a Nano-network [1]. Nano-network has different applications in various areas, extending from environment monitoring, industrial manufacturing and building labs-on-a-chip to an enormous number of applications in medicine, like drug delivery, diagnostics, tissue regeneration and surgical operations [2,3]. In the healthcare domain, Nano-network can collect vital patient information and provide it to computing systems making more accurate and efficient way of health monitoring. The implementation of the Internet of Nano-Things (IoNT) in healthcare systems will provide diagnostic and aid in the treatment of patients through accurate and localized drug delivery, in addition to tumor detection process.

Nano-network holds greater communication and processing potential that overcomes the limitations of standalone Nano-machine through Nano-devices cooperation. A Nano-network can carry the data to an external device such as a smartphone or a gateway enabling Nano-devices to wirelessly communicate with powerful external processing devices. This gateway can be connected to Internet. When the Nano-networks connect to an Internet gateway, they enable a new network paradigm called the Internet of Nano-Things (IoNT) [4]. With the continuous growth of IoT, a strain on traditional networks has been released. It is not just a matter of more bandwidth, IoT data is fundamentally different from the voice and video packets that comprise much of the traffic on modern networks. IoT traffic can be hard to predict, requiring additional network appliances [5]. It is worth remarking that Software Defined Networking (SDN) and Network Function Virtualization (NFV) are two useful technologies for IoT services. By outlining the way of combining SDN, NFV, IoT and Fog computing technologies, we can study how these technologies will be used in Nano-networks' context.

Nano-devices are not capable of handling complicated communication protocols, because of their limitations of computational capabilities in processing and power management. The functionalities of Nano-network can be powered by capacities provided by additional technologies, in order to be able to organize and optimize the behavior of the Nano-devices by means of decisions taken from strategies and algorithms executed with High Performance Computing (HPC). Therefore, a hierarchical architecture is

---

* Corresponding author.
E-mail addresses: akram.galal@upc.edu (A. Galal), xavier.hesselbach@upc.edu (X. Hesselbach).

https://doi.org/10.1016/j.nancom.2018.07.001

required in order to manage the Nano-devices and run externally the complex computations with their required decisions, hence enabling a controlled large-scale Nano-network. Moreover, using a hierarchical architecture will shift the processing complexity from Nano-devices to more powerful layers, where each layer has advanced capabilities that ease the complexity of any used protocol, algorithm or operation through the Nano-network by using the state-of-art networking technologies. In addition, A bottom-up layered architecture can be used to explain how the performance requirements of Nano-networks have been traditionally solved, i.e., the functionalities of the physical layer (information modulation, coding and transmission), the data link layer (medium access control, error control), the network layer (information forwarding, SDN, IoT, Fog Computing, NFV).

The objective of this paper is to provide an architecture approach for Nano-network communication using the current network technologies to manage Nano-machines. Also, to propose a set of functions and use cases of Nano-devices and show how these use cases can be useful in different Nano-scale applications. The rest of the paper is organized as follows. The literature review and related work are presented in Section 2. The proposed unified architecture model of Nano-networks is presented in Section 3. The proposed set of functions for Nano-networks is presented in Section 4. Functions application in real Nano-networks is presented in Section 5. Performance evaluation issues of the proposed computational architecture of Nano-networks are illustrated in Section 6. Then, the open research challenges are presented in Section 7, followed by the conclusions and future work in Section 8.

2. Literature review and related work

In this section, firstly, we discuss the current state-of-art in material architecture of Nano-devices, models of Nano-antennas, channel modeling, modulation technique, medium access control and energy consumption in Nano-communication paradigm. Secondly, we discuss the concept of Nano-network, besides the required networking technologies to manage and control a group of Nano-devices like Software Defined Network (SDN), Internet of Things (IoT), Fog Computing, Network Function Virtualization (NFV) and Virtual Network Embedding (VNE).

2.1. Material architecture of Nano-devices

There are several challenges to develop Nano-machines. However fully functional and efficacious Nano-machines have not been fabricated to date, there are various solutions had been prototyped and tested [6]. The connectivity between Nano-devices leads to the idea of Nano-networks, which presents Nano-communication definition. Nano-communication will expand the capabilities of Nano-devices to promote their features and expand the range of their applications. One of the promising techniques for data exchange between Nano-devices is Electromagnetic (EM) communication at terahertz band, which uses the electromagnetic wave as the carrier and its properties like amplitude, phase and delay are used to encode/decode the information [7]. The possibility of EM communication is studied based on the fact that terahertz band can be used as the operational frequency range for future EM Nano-transceivers, because of the emerging novel materials like Carbon Nano-Tube (CNT) and Graphene [6]. Graphene is one-atom-thick layer of carbon in a honeycomb crystal lattice, which has unique physical, electrical and optical properties. Graphene has many special properties. It is very strong, thin, light and highly bendable material. In addition, it has very good conduction properties with very high electron mobility. These unique properties enable the development of Nano-processors, Nano-memories, Nano-batteries and Nano-sensors, which are the main building blocks to fabricate fully operational Nano-machine [6]. The development of these novel materials, which can work at THz frequencies allows injecting Nano-devices inside the human body. One of the emerging areas of research is analyzing the propagation of terahertz electromagnetic waves through the tissues to develop some diagnostic tools for early detection and treatment of tumors [7].

2.2. Nano-antennas

The communication between Nano-devices is a major challenge, which is related to the development of Nano-antennas and the corresponding electromagnetic transceivers. Reducing the size of the traditional antenna to a few hundreds of Nano-meters would lead to extremely high operating frequencies. At THz-band frequencies, the very large available bandwidth leads to much higher path-loss than that of lower frequency bands. Nano-antenna can be made of metallic material like silver, aluminum, chromium, gold and copper or it can be made of novel materials like CNTs and graphene, which are attractive choice for Nano-antennas. It had been proved in [7] that the size and communication constraints could be overcome by using the graphene to fabricate the antennas, because the wave propagation velocity in CNTs and graphene can be up to one hundred times below the speed of light in vacuum depending on the structure geometry and temperature, in addition the resonant frequency of Nano-antennas based on graphene can be up to two orders of magnitude below that of Nano-antennas fabricated with other materials. Recent studies have proved that CNT/graphene antenna can work at the THz band (0.1–10 THz), which is the band of interest of EM communication [7].

2.3. Channel modeling for Nano-devices

In order to exploit and increase the potentials of Nano-devices, EM waveform propagation and channel model knowledge is necessary to build reliable and optimized Nano-communication system. The Terahertz band channel supports the transmission at very high bit-rates (up to Terabits per second), but over very short distances (below one meter) [8].

A novel channel model was presented in [9], where Jornet et al. developed a new channel model for Terahertz band communications and showed how the absorption from several molecules in the medium attenuates and distorts the traveling waves and introduced colored Gaussian noise. In [10], Jornet et al. described the Rate Division Time Spread On–Off Keying (RD TS-OOK), which is a new modulation and channel access mechanism for Nano-devices based on the asynchronous exchange of femtosecond-long pulses, which are transmitted following an on–off keying modulation spread in time. A logical "1" is transmitted by using a femtosecond-long pulse and a logical "0" is transmitted as silence, so the Nano-device remains silent when a logical zero is transmitted. An On–Off Keying (OOK) modulation is chosen instead of a binary Pulse Amplitude Modulation (PAM) or Pulse Position Modulation (PPM), because of the particular behavior of the molecular absorption noise. The time between transmissions is fixed and longer than the pulse duration, so multiple users can share the channel without affecting each other, with very low collision probability between pulses [11]. Due to the size and energy constraints of Nano-devices, it is not feasible to generate a high-power carrier signal in the Nano-scale at Terahertz frequencies. As a result, classical communication paradigms based on the transmission of continuous signals cannot be used. So, very short pulses can be generated and efficiently radiated in the Nano-scale. In particular, femtosecond-long pulses, which have their main frequency components in the Terahertz band are being used in several Nano-scale applications such as Nano-scale spectroscopy and biological imaging [10].

In [11], Jornet et al. studied the Terahertz channel behavior and provided analytical models for the path-loss, molecular absorption noise and interference in the Terahertz band, which is the frequency range of operation of graphene-based Nano-devices. Also, they illustrated the total path-loss for a traveling wave in the Terahertz band, which is defined as the addition of the spreading loss and the molecular absorption loss. The spreading loss accounts for the attenuation due to the expansion of the wave as it propagates through the medium and it depends on the frequency and the transmission distance. The absorption loss accounts for the attenuation that a propagating wave suffers because of molecular absorption, i.e., the process by which part of the wave energy is converted into internal kinetic energy to some of the molecules, which are found in the channel. The predicted channel capacity of the Terahertz band is promisingly very large, up to a few hundreds of Terabit/second. Meanwhile, the noise in the Terahertz band is mainly contributed by the molecular absorption noise. Vibrating molecules reradiate part of the energy that they have previously absorbed and this is conventionally modeled as a noise factor [11].

2.4. Medium access control

In traditional communication networks, the main use of the access layer is posed by the limited available bandwidth, which forces nodes to contend to the channel or follow time scheduling schemes. In Nano-networks, THz-band channel provides huge bandwidth, which results very high bit-rates, very short transmission times and minimum collision probability [7]. Although low transmission power of Nano-transceivers, high path-loss at THz channel and limited energy of Nano-machines, the use of MAC protocol is mandatory to regulate the link behavior, arrange the channel access and coordinate transmissions between Nano-machines [7].

In [8] Wang et al. proposed a centralized MAC protocol for Nano-networks, in which a Nano-controller would determine the best communication parameters for the Nano-devices. A transmitter-initiated hand-shake was required, which would result into a low channel utilization.

In [10] Jornet et al. proposed the Physical-layer Aware MAC Protocol for Electromagnetic Nano-networks (PHLAME). This protocol was built based on a novel pulse-based communication scheme for Nano-devices, where the transmitting and receiving Nano-devices have selected the optimal communication parameters and the channel coding scheme, which maximize the probability of successful message decoding and minimize the generated interference. The protocol was built on top of RD TSOOK and it was split in two stages, the handshaking process stage and the data transmission process stage [10]. The aim of the handshaking process is to allows a receiver to coordinate multiple simultaneous transmissions, in addition it facilitates the selection of the transmission symbol rate and the channel coding scheme, which make the data transmission more reliable. Firstly, when a Nano-device needs to send a message, it generates a Transmission Request (TR) packet, which contains the Synchronization Trailer, the Transmitter ID, the Receiver ID, the Packet ID, the transmitting Data Symbol Rate (DSR) and the Error Detecting Code (EDC). The DSR field specifies the symbol rate that will be used to transmit the data packet. The strength of RD TS-OOK against collisions increases when different Nano-devices transmit at different rates. In PHLAME, every transmitting node randomly selects a symbol rate from a set of rates, which had been shown to minimize the probability of collisions [10].

2.5. Energy consumption of Nano-devices

Energy limitations of Nano-devices require energy harvesting systems at Nano-scale, where the energy is captured from external sources and stored for continuous operations. The amount of energy that can be stored in the Nano-batteries is extremely low. As a result, Nano-devices can only complete a very simple task with a single battery charge. It is impossible to manually recharge or replace the batteries of the Nano-devices, so novel energy harvesting Nano-systems had been developed [8].

In [8] Wang et al. developed energy and spectrum-aware MAC protocols for Wireless Nano-sensor Networks (WNSNs) with the objective to achieve fair throughput and lifetime optimal channel access by jointly optimizing the energy harvesting and consumption processes. The limited energy that can be stored in Nano-batteries provides a major challenge for the development of useful applications of Nano-networks. For this, novel Nano-scale energy harvesting systems were being developed [12–14]. One of the most promising technologies to date is based on the use of piezoelectric Nano-generators based on Zinc Oxide (ZnO) Nano-wires, where a small amount of charge is generated at the tips of the Nano-wire each time that these are bent or released. This charge can be used to recharge an ultra Nano-capacitor. Several prototypes had already been developed [12]. The goal is to make a self-powered Nano-system that can operate wirelessly and independently. Harvesting energy from the environment is a choice for powering Nano-systems.

In [12], Wang illustrated the potential of converting mechanical energy (such as body movement, muscle stretching, blood pressure), vibrational energy (such as acoustic/ultrasonic waves) and hydraulic energy (such as flow of body fluids, blood flow, contraction of blood vessels, dynamic fluid in nature) into electrical energy by using Nano-generator, which converts mechanical waves into electricity that may be sufficient for self-powering Nano-devices and Nano-systems. This had been developed using the piezoelectric effect, where a mechanical stress/strain could be converted into electrical voltage.

In [6], Jornet proposed an energy model for self-powered Nano-machines, which successfully captured the correlation between the energy harvesting by means of a piezoelectric Nano-generator and the energy consumption process due to communication with TS-OOK.

2.6. Nano-networks

With the massive growth and progress in the area of Nanotechnology, the demand for communication between Nano-machines has arisen naturally. A Nano-device is considered the most basic functional unit. The definition of Nano-device is a mechanical or electromechanical device, whose dimensions are measured in Nano-meters. It can be also very tiny components consisting of an arranged set of molecules. Nano devices are called Nano-machines or Nano-nodes. The limited size of Nano-machines with their restricted capabilities make the ability to perform more complex actions depend on the cooperation between Nano-machines in large groups, which is called Nano-network [15].

Akyildiz et al. [1] provided an overview of the two main alternatives for Nano-communication, that are Electromagnetic Communications in the Terahertz Band and Molecular Communications. They provided a better understanding of the current research issues in this emerging field and paved the way of future research in Nano-networks. They also reviewed the state-of-art in the design and manufacturing of Nano-machines, discussed the different alternatives for communication in the Nano-scale and described the research challenges in the design of protocols for Nano-networks.

Balasubramaniam et al. stated in [16] two main ways of communications between Nano-devices. They are Electromagnetic Communications in Terahertz Band (Ultra-broadband) and Molecular Communications, which is consistent with the work provided in [1]. Terahertz Band is a very large transmission window that

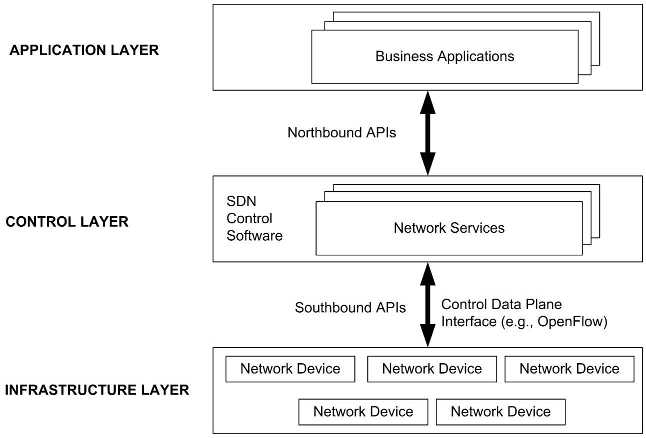

Fig. 1. Software Defined Network architecture [23].

can support very high transmission rates in a short range. Having a very large available bandwidth introduces major changes in classical networking protocols. In molecular communications, cells and many living organisms exchange information by means of molecular communication. They use molecules to encode, transmit and receive information.

Kulakowski et al. [2] focused on Nano-communications via Forster resonance energy transfer (FRET), which was found to be a technique with a very high signal propagation speed and discussed how to route signals through Nano-networks. They introduced five new routing mechanisms, based on biological properties of specific molecules and experimentally validated one of these mechanisms. In addition, they analyzed the open research issues showing the technical challenges for signal transmission and routing in FRET-based Nano-communications.

Ali et al. [4] outlined the different network models of Internet of Nano-things (IoNT) and the architectural requirements for implementation. They highlighted the main applications of IoNT and the significant challenges faced in implementing this technology in healthcare. They also discussed the communication and networking aspects of IoNT examining two paradigms, layer-based and non-layer-based models including the comprised layers in the layered model. Also, they provided a comparison between the two models and discussed the advantages, disadvantages of both.

Nakano et al. [17] developed the layered architecture of molecular communication and described research issues that molecular communication faces at each layer of the architecture with the open research issues that need to be addressed at each layer. In addition, they provided design example of targeted drug delivery, which is a Nano-medical application to illustrate how the layered architecture helps design an application of molecular communication.

Kuscu et al. [18] provided a detailed architectural view of Nano-communication focused on its fundamental principles and design requirements by surveying theoretical and experimental ideas. They gave an overview of networking opportunities offered by the intrinsic capabilities of fluorophores under the concept of Internet of Molecular Things. Also, they presented some prospective applications, theoretical modeling approaches, experimental opportunities and implementation challenges.

Stelzner et al. [19] were looking at the combination of in-body Nano-communication with the Internet of Things (IoT) especially Body Area Networks (BAN) and the resulting research challenges in the Internet of Nano-Things (IoNT). Moreover, they provided a concept for Function Centric Networking presented an approach to deal with these challenges by addressing specific groups of interchangeable and replaceable Nano-machines. Several communication paradigms can be used in Nano-networks depending on the technology used to manufacture the Nano-machines and the targeted application.

Chude-Okonkwo et al. [20] had provided a visionary survey of the application of Nano-medicine to Targeted Drug Delivery (TDD) from the perspective of a molecular communication scientist. The main objective of their survey was to provide researchers interested in exploring the idea of TDD on the Molecular Communication (MC) platforms with the necessary background information and motivational guide.

Tsioliaridou et al. [21] introduced a joint coordinate and routing system for Nano-networks, which could be deployed dynamically on an ad-hoc Nano-network.

Tsioliaridou et al. [22] introduced the N3, which is an addressing and routing scheme for 3D Nano-networks.

2.7. Software Defined Network (SDN)

Software Defined Networking (SDN) is an emerging network architecture, where network control is decoupled from forwarding and is directly programmable. This migration of control enables the underlying infrastructure to be abstracted for applications and network services, which can treat the network as a logical or virtual entity.

In [23], Open Networking Foundation (ONF) explained the need for new network architecture and standardizing critical elements of the SDN architecture such as the OpenFlow protocol, which structures the communication between the control and data planes of supported network devices. SDN architectures support a set of Application Programming Interfaces (APIs) that make it possible to implement common network services, including routing, multicast, security, bandwidth management, traffic engineering, quality of service and storage optimization. Fig. 1 illustrates the architecture of Software Defined Network (SDN).

In [24], Cisco Systems explained the technological forces that have given rise to SDN. They provided an easy-to-understand look at a new network architecture made possible with SDN and showed how the new model compared with a traditional network architecture. They also discussed why SDN is such a monumental change and enumerate the benefits it can bring to the network.

Bizanis et al. [25] surveyed the state-of-art on the application of SDN and Network Virtualization (NV) to Internet of Things (IoT). They provided a comprehensive description of every possible IoT implementation aspect for the two technologies. They started by outlining the ways of combining SDN and NV. Subsequently, they presented how the two technologies could be used in the mobile and cellular context, with emphasis on forthcoming 5G networks. In addition, they reviewed some general SDN-NV-enabled IoT architectures, along with real-life deployments and use cases.

OpenFlow is the first standard communications interface defined between the control and forwarding layers of an SDN architecture. It allows direct access to the forwarding plane of network devices such as switches and routers, both physical and virtual (hypervisor-based). A hypervisor is a software which abstracts/isolates the operating systems and applications from the computer hardware. This abstraction allows the machine hardware to independently operate one or more virtual machines as guests, allowing multiple VMs to effectively share the system's physical compute resources, such as processor, memory, network bandwidth and so on. A hypervisor is sometimes also called a virtual machine monitor. VMware application is an example of hypervisor. Network hypervisors allow to create virtual networks that are completely decoupled and independent from the physical network [23]. It also enables segments of a virtual network to be managed and provisioned independently. A network hypervisor

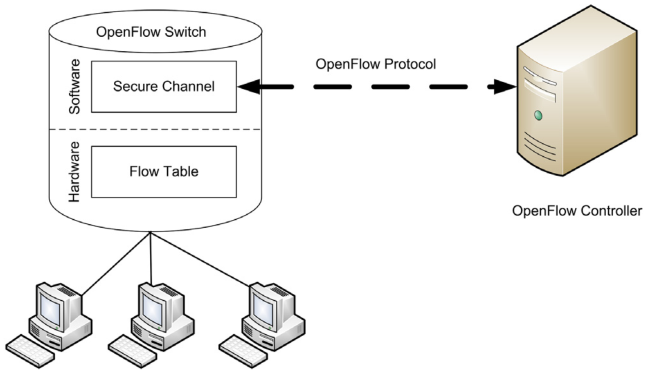

Fig. 2. OpenFlow protocol [26].

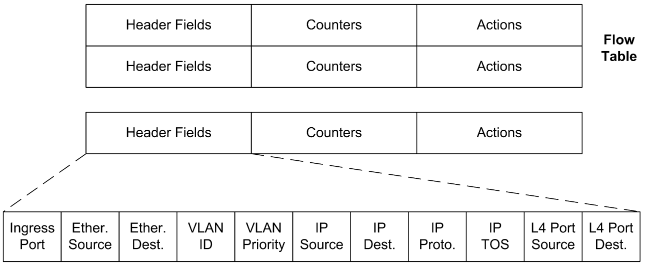

Fig. 3. Fields in the OpenFlow protocol [26].

is used at the border between the cloud and access network to translate network requirements to specific tasks in the cloud. It acts as centralized management and control of networking devices from multiple vendors and it improves automation and management by using common APIs to abstract networking details from the orchestration and provisioning systems and applications. Other OpenFlow benefit is the ability to deliver new network capabilities and services without the need to configure individual devices or wait for vendor releases. It increases network reliability and security as a result of centralized and automated management of network devices, as it provides uniform policy enforcement and fewer configuration errors. Also, it provides more network control with the ability to apply comprehensive and wide-ranging policies at the session, user, device and application levels [26].

In December 2011, the ONF board approved OpenFlow version 1.2 and published it in February 2012. The current version of OpenFlow is 1.5.1. Despite version 1.6 has been available since September 2016, it is accessible only to ONF's members [23]. Fig. 2 shows an OpenFlow switch and an OpenFlow controller. The term "switch" is used for the OpenFlow nodes because, as we shall see, paths are usually determined by the controller [26].

The OpenFlow has two parts: first part contains the queues, frames transmitted/received with the associated flow tables and second part that communicate with the controller using the OpenFlow signaling protocol. Fig. 3 describes the flow table. The first field enables matching between a row in the table and the flow. The flow is described by the ingress port number and the Ethernet addresses, VLAN number, VLAN priority, IP addresses, protocol and type of service fields and the port numbers of the TCP or UDP transport layer. The flow table is supplemented by values of the frame/byte counters giving an indication of the statistics of flows across all ports in the network. Therefore, the controller has a complete view of the whole network under its management. Amongst the possible actions which can be transported by OpenFlow signaling that includes: sending a packet over a list of ports, adding/rejecting/modifying a VLAN Tag, destroying a packet and sending a packet to the controller [26].

In SDN architecture, the northbound application program interfaces (APIs) are usually SDN Rest APIs used to communicate between the SDN controller and the services running over the network. The northbound APIs can be used to facilitate innovation, enable efficient orchestration and automation of the network. SDN northbound APIs are also used to integrate the SDN controller with automation stacks, such as Puppet, Chef, SaltStack, Ansible and CFEngine, as well as orchestration platforms, such as OpenStack and the open source CloudStack [27]. The southbound application program interfaces (APIs) are used to communicate between the SDN controller and the switches/routers of the network. While OpenFlow is the most well-known of the SDN protocols for southbound APIs, it is not the only one available or in development. The Network Configuration Protocol (NetConf) uses an Extensible Markup Language (XML) to communicate with the switches and routers to install and make configuration changes. Also, Extensible Messaging and Presence Protocol (XMPP) is a protocol based on Extensible Markup Language. Its intended use is for instant messaging and online presence detection. The protocol function is to facilitate real-time operations among servers. Open vSwitch Database Management Protocol (OVSDB) is an OpenFlow configuration protocol that is meant to manage Open vSwitch implementations. Open vSwitch is a virtual switch that enables network automation and the support of standard management interfaces and protocols. In addition, there are more established networking protocols finding ways to run in an SDN environment, such as Open Shortest Path First (OSPF), Multiprotocol Label Switching (MPLS), Border Gateway Protocol (BGP) and Intermediate System to Intermediate System (IS–IS) [28].

2.8. *Internet of Things (IoT)*

The Internet of Things (IoT) is the network of physical devices, vehicles and other items embedded with electronics, software, sensors, actuators and network connectivity, which enable these objects to collect and exchange data. The scope of IoT is enormous and will affect every aspect of our lives. IoT allows objects to be sensed or controlled remotely across existing network infrastructure creating opportunities for more direct integration of the physical world into computer-based systems. These devices or things connect to the network to provide information they gather from the environment through sensors or to allow other systems to reach out and act on the world through actuators. Each of them is able to convert valuable information from the real world into digital data that provides visibility into how the users interact with products, services or applications. Many technical communities are vigorously pursuing research topics that contribute to IoT. Nowadays, as sensing, actuation, communication and control become even more sophisticated and ubiquitous, there is a significant overlap in these communities, sometimes from slightly different perspectives, so more cooperation between communities is encouraged [29,30].

Chiang et al. [29] presented the range of new challenges in emerging IoT and the difficulty to address these challenges with today's computing and networking models, then they discussed why we would need a new architecture using Fog technology for simplicity and how it could fill the technology gaps and create new business opportunities.

In order to provide a basis for discussing open research problems in IoT, Stankovic [31] presented a vision for how IoT could change the world in the distant future. Then, eight key research topics were enumerated and research problems within these topics were discussed. Jafarey discussed in [32] the Internet of Things

(IoT) with IP address needs and illustrated how Fog computing concept could support the addressing needs of IoT devices using Radio Frequency Identification (RFID) tags.

2.9. Fog computing

Fog computing or Fog networking (Fogging) is an architecture that allows computing, storage and communications resources to be placed in a continuum between the cloud and the edge. Fog computing is a distributed edge computing paradigm, which extends the cloud to where the things are by analyzing the most time-sensitive data at the network edge, close to where it is generated instead of sending vast amounts of IoT data to the cloud. It also acts on IoT data in milliseconds and sends selected data to the cloud for historical analysis and longer-term storage. Some machine-to-machine computations and storage take place locally in the Fog, then there is a periodic communication with the cloud. Using Fog computing minimizes the latency, conserve network bandwidth and move the data to the best place for processing. The most time sensitive data are analyzed on the Fog node closest to the things generating the data. Data that can wait seconds or minutes for action is passed along to an aggregation node for analysis and action, while data that is less time sensitive is sent to the cloud for historical analysis, big data analytics and long-term storage. Fog computing also provides support for mobility and real-time interactions, where many Fog applications need to communicate directly with mobile devices.

Chiang et al. [29] summarized the opportunities and challenges of Fog, focused primarily on the networking context of IoT. Dastjerdi et al. [30] discussed Fog computing characteristics, components, applications and how it helped the Internet of Things (IoT) to realize its potential.

In [33], Cisco Systems illustrated how Fog computing solutions could capture the power of the IoT, which required a solution with comprehensive set of products for deploying, accelerating value and innovating with the Internet of Things.

In [34], Cisco Systems illustrated how Fog computing gave the cloud a companion to handle exabytes of data generated daily from the Internet of Things. They also showed how processing the data closer to where it is produced solves the challenges of exploding data volume, variety and velocity.

Bonomi et al. [35] outlined the vision and key characteristics of Fog computing. They envisioned the Fog to be a unifying platform, rich enough to deliver this new generation of emerging services and enable the development of new applications. Fig. 4 shows how Fog data services coordinate the movement of data from the Fog to the cloud. The local devices that interact with Fog layer may communicate without connection to the Internet, then the local collected data could bridge to the cloud [32].

2.10. Network Function Virtualization (NFV)

NFV has emerged in order to increase the flexibility of network services deployment and integration within operator's networks. This is achieved through the implementation of any network function, which is a function used for the network itself such as firewalls, encryption, filtering, load balancing, etc. via software modules. Transforming network functions in software allows them to run on different general-purpose equipment that could be located in a variety of telecom infrastructure, including data centers, network nodes and even in end-user facilities [36].

NFV provides an alternative hardware centric network infrastructure development, management and expansion. NFV along with SDN allows control plane functions to run as virtual network functions (VNFs). This is a major departure from the costly legacy procedure of purchasing, managing and maintaining several different devices to address different applications. VNFs can replace multiple dedicated appliances for more scalable IoT networks. Application acceleration, load balancing, policy management, network optimizations, intrusion detection and prevention system (IDS/IPS), distributed denial of service (DDoS), other web-based protections and firewalls are all applications that can be virtualized on such platforms. One reason NFV is cheaper, faster and more flexible than traditional network implementations is the availability of open-source network virtualization tools that define an information model, set of APIs and control protocols such as OpenStack and OpenDaylight (ODL) [5].

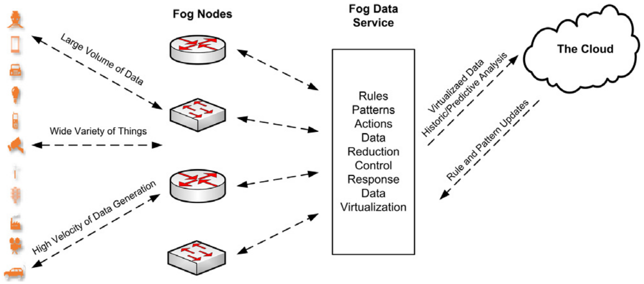

Fig. 4. Fog data services coordinate the movement of data from fog to cloud [33].

Mijumbi et al. [36] discussed NFV and its relationship with complementary fields of SDN and cloud computing. Also, they surveyed the state-of-art in NFV and provided an overview of NFV key projects, standardization efforts, early implementations, use cases and commercial products.

In [37], Cisco Systems discussed network optimization through virtualization and what does this mean for a virtualized infrastructure.

In [38], ETSI NFVISG illustrated Network Functions Virtualization (NFV) architectural framework as shown in Fig. 5. There are three main working domains are identified in NFV: (a) Virtualized Network Function (VNF), as the software implementation of a network function which is capable of running over the NFVI. (b) NFV Infrastructure (NFVI), which including the diversity of physical resources and how these can be virtualized. NFVI supports the execution of the VNFs. (c) NFV Management and Orchestration, which covers the orchestration and lifecycle management of physical and/or software resources that support the infrastructure virtualization and the lifecycle management of VNFs. NFV Management and Orchestration focus on all virtualization specific management tasks necessary in the NFV framework.

Network services in NFV is an end-to-end network service (e.g., mobile voice/data, Internet access, virtual private network) that can be described by an NF Forwarding Graph of interconnected Network Functions (NFs) and end points [38]. Fig. 6 illustrates an end-to-end network service that includes: NF Forwarding Graph as indicated by the network function block nodes in the middle of the figure interconnected by logical links. The end points are connected to network functions via network infrastructure (wired or wireless), resulting in a logical interface between the end point and a network function. These logical interfaces are represented in Fig. 6 with dotted lines. These network functions can be implemented in a single operator network or interwork between different operator networks. The end points and the network functions of the service are represented as nodes and correspond to devices, applications and/or physical server applications. NF Forwarding Graph can have network function nodes connected by logical links that can be unidirectional, bidirectional, multicast and/or broadcast. An example of a forwarding graph is a chain of network functions, while a

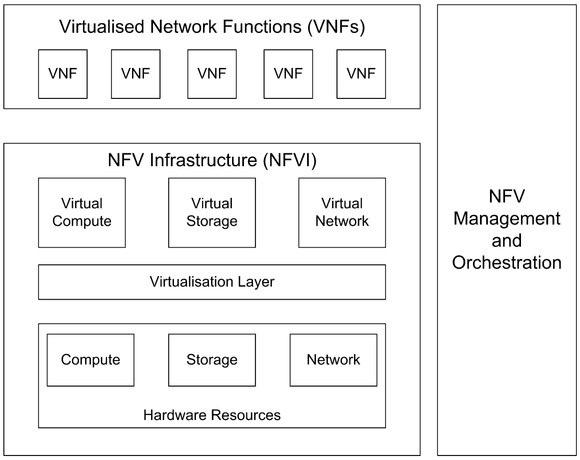

Fig. 5. High-level NFV framework [38].

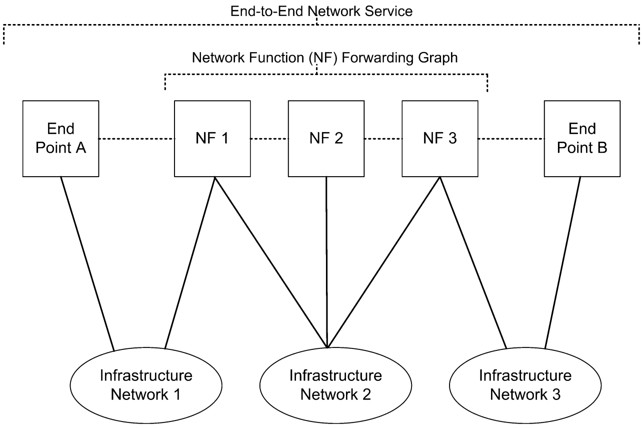

Fig. 6. Graph representation of an end-to-end network service [38].

smartphone, a wireless network, a firewall and load balancer can be an example of such end-to-end network service [38].

Fig. 7 illustrates an example of an end-to-end network service with VNFs and nested forwarding graphs. An end-to-end network service can be composed of only VNFs and two end points. The decoupling of hardware and software in network functions virtualization is realized by a virtualization layer, which abstracts hardware resources of the NFV Infrastructure. The NFVI-PoPs includes resources for computation, storage and networking deployed by a network operator. Virtualized Network Functions run on top of the virtualization layer, which is part of the NFVI, as indicated by the arrow labeled "virtualization". The interface between NFs and/or VNFs and the infrastructure is a multi-vendor environment based upon accepted standards. NFV emphasizes the fact that the physical deployment of a VNF is not visible from the end-to-end service perspective, however VNF and their supporting infrastructure need to be visible for configuration, diagnostic and troubleshooting purposes [38].

The NFV architectural framework identifies functional blocks and the main reference points between such blocks. As illustrated in Fig. 8, the functional blocks are: Virtualized Network Function (VNF), Element Management (EM), NFV Infrastructure, Virtualization Layer, Virtualized Infrastructure Manager(s), NFV Orchestrator, VNF Manager(s), Service, VNF and Infrastructure Description, Operations and Business Support Systems (OSS/BSS) [38].

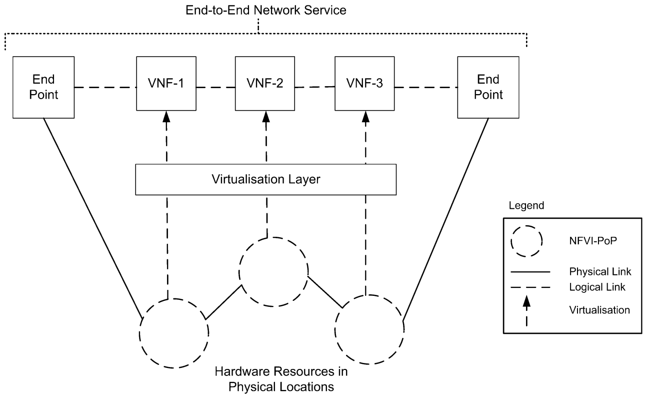

Fig. 7. Example of an end-to-end network service with VNFs and nested forwarding graphs [38].

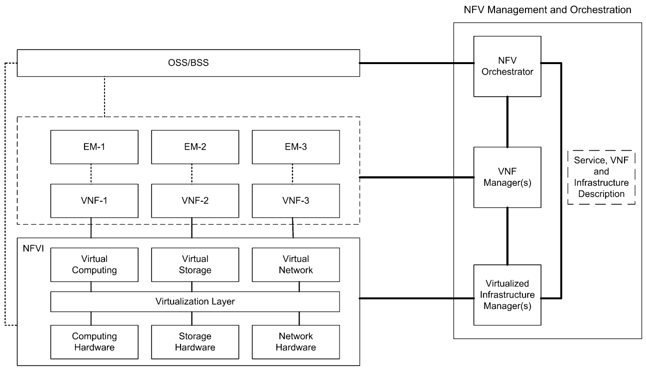

Fig. 8. NFV reference architectural framework [38].

Virtualized Network Function (VNF) is a virtualization of a network function in a non-virtualized network. Examples of NFs are evolved packet core network elements, serving gateway, packet data network gateway, elements in a home network, Dynamic Host Configuration Protocol (DHCP) servers, firewalls, etc. Functional behavior and state of a NF are independent of whether the NF is virtualized or not. The functional behavior and the external operational interfaces of a Physical Network Function (PNF) and a VNF are expected to be the same. A VNF can be deployed over multiple VMs, where each VM hosts a single component of the VNF. However, in other cases, the whole VNF can be deployed in a single VM as well.

Element Management (EM) performs the typical management functionality for one or several VNFs.

NFV Infrastructure is all hardware and software components which build up the environment in which VNFs are deployed, managed and executed. The NFV Infrastructure can span across several locations, i.e., NFVI-PoPs. The virtualization layer and the hardware resources look like a single unit providing the VNF with resources.

Hardware Resources are the physical hardware resources include computing, storage and network that provide processing, storage and connectivity to VNFs through the virtualization layer (e.g., hypervisor). Computing and storage resources are commonly pooled, while network resources are comprised of switching functions, e.g., routers and wired or wireless links.

Virtualization Layer and Virtualized Resources are responsible for abstracting and logically partitioning physical resources, commonly as a hardware abstraction layer, therefore the software can be deployed on different physical hardware resources. Also, enabling the software that implements the VNF to use the underlying virtualized infrastructure and providing virtualized resources to the VNF, so that the latter can be executed.

Virtualized Infrastructure Manager contains the functionalities that are used to control and manage the interaction of a VNF with network resources. It performs resource management and allocation (e.g., increase resources to VMs and improve energy efficiency). It also provides a root cause analysis of performance issues from the NFV infrastructure perspective and collects infrastructure fault information. In addition, it collects information for capacity planning, monitoring and optimization. Multiple Virtualized Infrastructure Managers may be deployed.

NFV Orchestrator is in charge of the orchestration and management of NFV infrastructure and software resources and realizing network services on NFVI.

VNF Manager(s) is responsible for VNF lifecycle management (e.g., instantiation, update, query, scaling, termination).

Service, VNF and Infrastructure Description, this data-set provides information regarding the VNF deployment template, VNF Forwarding Graph, service related information and NFV infrastructure information models. These templates are used internally within NFV Management and Orchestration. The NFV Management and Orchestration functional blocks handle information contained in the templates.

Operations Support Systems and Business Support Systems (OSS/BSS) refers to the OSS/BSS of an Operator [38].

Each of NFV, SDN and Cloud Computing technologies is an abstraction of different resources, compute for cloud computing, network for SDN and functions for NFV. The advantages that accrue from each of them are similar (i.e., agility, cost reduction, dynamism, automation, resource and scaling) [38].

2.11. *Virtual Network Embedding (VNE)*

The Virtual Network (VN) is a combination of active and passive network elements (network nodes and network links) on top of a Substrate Network (SN). Virtual nodes are interconnected through virtual links, forming a virtual topology/network. By using the virtualizing concept, multiple virtual networks with different characteristics can be created and co-hosted on the same physical hardware. Virtual Network Embedding (VNE) is the allocation process of the physical resources for virtual nodes/links to form a virtual network. Therefore, it can be divided to Virtual Node Mapping, where virtual nodes have to be allocated in physical nodes and Virtual Link Mapping, where virtual links connecting these virtual nodes have to be mapped to paths connecting the corresponding nodes. This is typically modeled by a Virtual Network Request (VNR) with node and link demands. The problem of mapping virtual resources to substrate resources in an optimal way is commonly known as the Virtual Network Embedding problem, which is NP-hard problem. It is computationally difficult for a number of important metrics, as it is related to the multi-way separator problem. All VNE approaches proposed in the literature can be categorized according to whether they are Static or Dynamic, Centralized or Distributed and Concise or Redundant [39].

2.11.1. *Static vs. Dynamic*

In most real-world situations, VNE has to be an online problem, where VNRs will not be known in advance. Instead, they arrive to the system dynamically and can stay in the network for a certain amount of time. VNE algorithm has to handle the VNRs as they arrive, rather than attending a set of VNRs at once (offline VNE). Static VNE does not provide the possibility of remapping VNRs to improve the performance of the embedding in the SN, while dynamic VNE provides tries to reconfigure the mapped VNRs in order to reorganize the resource allocation and optimize the utilization of SN resources [39].

2.11.2. *Centralized vs. Distributed*

The VNE problem can be solved in either a centralized or in a distributed way. In a centralized approach there will be one entity, which is responsible for performing the embedding. This can be a dedicated machine computes optimal solution to the problem. The advantage of this approach is that this machine is aware of the overall situation of the network, which facilitates more optimal embedding. However, it is a centralized entity that presents a single point of failure. In a distributed approach, multiple entities for computing the embedding are used. There may be some internal organization in how the mapping is distributed among the participating entities. The advantage of such an approach lies in its better scalability. Since the load is distributed among several nodes, each individual node will be better able to cope with the embedding. However, its disadvantage is the synchronization overhead, where each node needs sufficient information about the global state of the network [39].

2.11.3. *Concise vs. Redundant*

A failure of a single substrate entity will affect all virtual entities that are mapped upon it. Therefore, in environments where fault-sensitive applications are used, it is advised to set-up backup resources that can be used in case the corresponding primary resources fail. To do that, the embedding result itself can be redundant regarding node/link failures. Otherwise, the embedding result is referred to be concise. VNE algorithm can be centralized, dynamic and redundant at the same time. So, each algorithm denoted with a code that allows for quick categorization. VNE algorithm can be centralized, dynamic and redundant at the same time. So, each algorithm denoted with a code that allows for quick categorization [39].

2.12. *Summary*

As we have seen in this section, Nano-devices are able to transmit and receive binary bit streams. This is achieved by electromagnetic communication among Nano-machines in the Terahertz Band (0.1–10 THz) using the graphene-based Nano-antenna transceivers. It is obvious that this communication system needs complex computations to achieve a successful end-to-end packet delivery. Some of these complex computations span femtosecond-long pulse-based modulations, channel coding schemes, error detection and correction mechanisms, medium access control, addressing schemes and different routing strategies. Taking into consideration the limited energy consumption of a single Nano-machine, besides its limited computational capabilities to handle this huge amount of data with its complex communication. As a result, it is radical to implement Nano-communication architecture able to manage and control Nano-devices in a proper way, also to extend these complex computations to powerful layers rather than the Nano-machine itself. This can be accomplished by the aid of current networking technologies such as SDN, IoT, Fog computing and NFV. This is a fundamental step toward designing a consolidated Nano-network architecture supports a set of functions to represent the Nano-network demand.

3. Proposed unified architecture model of Nano-networks

In this section, an unified flexible architecture model of Nano-networks is proposed based on a horizontal platform of SDN, IoT, Fog Computing and NFV technologies. The architecture is divided into five layers as described and illustrated in both Figs. 9 and 10.

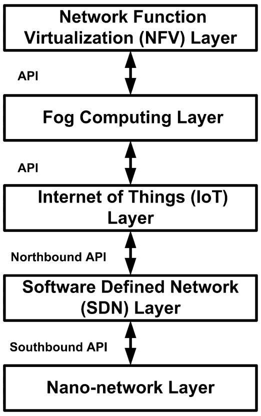

Fig. 9. Schematic high-level diagram of unified architecture model of Nano-networks.

3.1. Nano-network layer

It is called end devices layer and contains Nano-devices/Nano-machines, Nano-sensors and Nano-actuators. It also includes Micro/Nano-gateways that transmits the signal to/from Nano-devices to perform a specific action. It is the physical Layer, which contains the physical hardware, such as sensors and gateways that connect Nano-devices to the backbone network. Its main function is the delivery of data/information from one device to another. This layer does not possess any intelligence and it leaves the decision-making to the control layers that lay above.

3.2. Software Defined Network (SDN) Layer

SDN layer is the communication layer that contains the SDN-enabled switches, which forward data to/from Nano-devices according to the commands set by the SDN controller. SDN layer is used between the edge and the core network and it gives the ability to control the traffic, in addition to the flexibility to configure, manage, secure and optimize Nano-network resources via dynamic automated SDN programs. It is also used to facilitate the upper IoT layer functionalities, where specific service requirements are translated by a central controller into network requirements using the OpenFlow signaling protocol in the southbound API. This layer is able to make an intelligent delivery of data, by means of the stored configured routing tables and the corresponding decision taken in the controller.

3.3. Internet of Things (IoT) Layer

Internet of Nano-Things (IoNT) can be a special form of IoT. IoT layer acts as a service layer, which is used by the IoT applications to give instructions to the SDN controller, which in turn will translate them to specific network commands. Also, it acts as an application layer, which is used to build IoT developing applications by using the exposed Applications Programming Interface (APIs). In addition, IoT layer provides device management like provisioning, operating, updating, logging and monitoring of various events through dashboards and alert mechanisms, which can keep Nano-devices running in a proper way. In this layer, information which comes from each Nano-device is sent to whether the cloud or Fog computing layer, where it is processed, stored and combined with data from other devices.

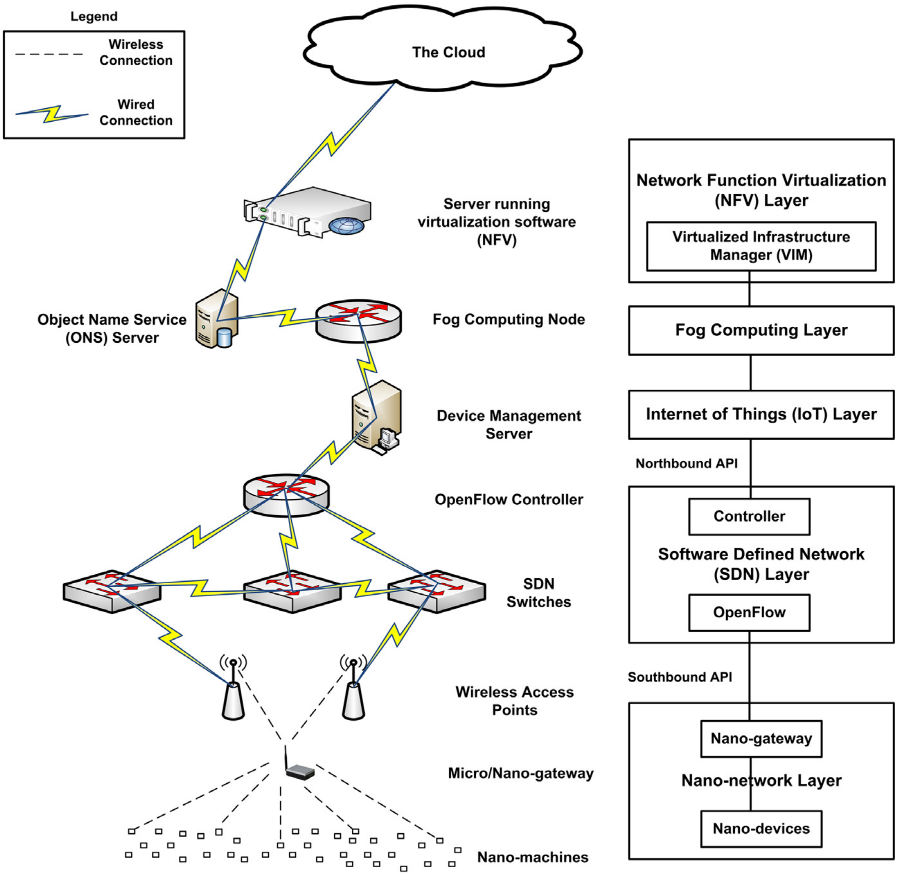

Fig. 10. Schematic low-level diagram of unified architecture model of Nano-networks.

Each Nano-device can provide or consume various types of information called device metadata. Metadata contains information like device identifier or tag ID, device type, device model, date manufactured, hardware serial number, status of the device whether it is ON/OFF and commands or actions performed by a Nano-device. The command mechanism can include a return value or might rely on the confirmation being made through a separate return message or by reflecting the expected change in the state data. These commands might include a time-to-live (TTL) or other expiration value acting as an indication. Operational information is another type of metadata that is most relevant to the operation of the Nano-device. This might include things such as operating temperature and battery life to maintain the operating state, such as responding to breakages and correcting performance degradation [40].

3.4. Fog computing layer

Fog Computing layer provides computing services, with limited capacity compared to the computing services provided from NFV layer, however Fog Computing layer is more agile in terms of the responsiveness for the Nano-network. It provides data aggregation and processing analyses, as it acts on the most time-sensitive IoNT data at the network edge, close to where it is generated instead of sending vast amounts of data to the cloud. So, it represents the local computation. Also, it is responsible of Nano-devices addressing scheme, as it includes the Object Name Service (ONS) server, where Nano-devices identification tags are stored. ONS server performs the look-up system process, which takes an Electronic Product Code (EPC) and returns the Nano-device ID with information about its location.

3.5. Network Function Virtualization (NFV) layer

NFV layer is the brain of the architecture. It is the controller of the system, where decisions and actions related to Nano-devices' functions will take place (i.e., how to act and when) with the assistance of Virtualized Infrastructure manager (VIM) by means of high performance computers (i.e., Data Centers). So, NFV represents the global computation. NFV layer will analyze how Nano-network's infrastructure can be orchestrated according to NFV Management and Orchestration (MANO). Network Virtualization (NV) has been proposed as an enabling technology for the network architecture, in order to facilitate the deployment of new Internet services. NV allows multiple heterogeneous, logical networks to cohabit on a shared substrate network (SN). The Virtual Network Embedding (VNE) problem refers to the embedding of virtual networks in a substrate network under a set of constraints with the objective of optimize the performance according to specific defined metrics (such as maximum throughput, minimum delay or minimum power consumption). Since the Nano-network is the substrate network where actions are performed, a Nano-device can be mapped to a virtual node, while the channel model of Terahertz band can be mapped to a virtual link. Therefore, different functionalities will be provided for a single substrate Nano-network. So, VNE concept has an important role in terms of providing a logical separation of Nano-devices' functions under a controlled behavior from a steering decision center such as NFV layer.

3.6. Summary

The proposed architecture model of Nano-networks is divided into five layers, the first one is the Nano-network layer, which represents the physical layer that contains all physical Nano-devices. The second layer is SDN layer, which is responsible of the intelligent delivery of data between Nano-devices. The third one is IoT layer, which is responsible of data management and provisioning. The fourth layer is Fog computing layer, which is responsible for addressing and local computations. The fifth layer is NFV layer, which is responsible for the global computations and decision-making process. Based on the type of information coming from Nano-things, it can be computed in devices like Micro/Nano-gateways, which represent Fog nodes of a Nano-network, therefore it is not necessary to reach data centers or high-performance computers for processing, trying to avoid more computations occurred on NFV layer. Hence, Fog computing can be represented by the gateway, while NFV can be represented by data center.

4. Proposed set of functions for Nano-networks

In this section, we will propose a set of functions for Nano-networks, besides a set of use cases which illustrates how these functions will be concatenated to form a network demand in the Nano-network paradigm.

Nano-machines can provide many functions and tasks according to its application. For example, in the health care systems Nano-machines can provide an encapsulation system that protects the enclosed drugs in a slow and controlled manner, also they can rearrange bonds and tissues holding molecules together or pick up and move individual atoms, representing self-replicating or self-assembler vision. Likewise, Nano-devices in an in-body network can circulate through the patient's blood and collect measurements where/when-ever necessary and communicate their results to outside medical personnel [19]. Furthermore, Internet of Bio-Nano-Things (IoBNT) stands as a novel molecular communication network concept for a prospective application domain, where very tiny biocompatible devices can communicate through the Internet. The medical person sends out the applicable signal through the Internet or any other appropriate network. This signal is received and relayed by a bio-cyber signaling system on the patient's body or a wearable device that hangs around the patient's hand. The bio-cyber system responds by sending the appropriate signal to the designated set of Nano-devices in the target Nano-network through the body. The molecular information source is essentially a transceiver that responds to a trigger signal (i.e., change in its environment) by releasing molecular information, which can be drug molecules or signaling molecules. The environmental change may be due to a change in concentration level of certain molecules, pH, temperature, light, ultrasound, enzymes, magnetic and illumination fields in the extracellular environment. Hence, all we need to do is to change the appropriate environmental condition of the Nano-transmitter and it releases the required molecules [20]. A security hypothesis shall be guaranteed, i.e., the security of Nano-devices is mandatory and they could not be accessed from any location other than the corresponding access point or intelligent gateway. Fig. 11 illustrates a schematic representation of IoBNT.

Nano-devices face challenges in functionalities, because of their limitations in processing and power capabilities. So, we proposed a set of simple functions described the way of communication between Nano-devices and the controller. The functions will be implemented in the southbound of SDN layer and Nano-network layer, however decisions and reactions to the functions will take place in IoT, Fog computing and NFV layers. This depends on the specifications and objectives of the function. Table 1 illustrates the proposed set of functions of Nano-networks and their corresponding descriptions. They describe the way of communication between the Nano-devices up to the controller and vice versa.

A use case is a methodology used to identify, clarify and organize the Nano-system requirements. It shows the interaction of the corresponding layers, when the controller in the NFV layer sends a specific signal or a command to a Nano-device/group of Nano-devices in the Nano-network layer or when a Nano-device/group of Nano-devices reply to the controller. In addition,

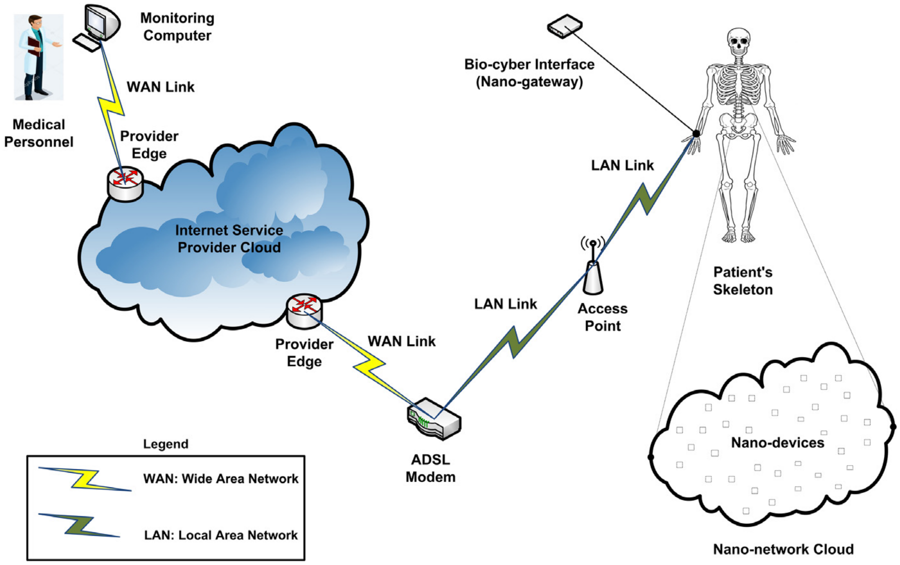

Fig. 11. Schematic representation of IoBNT [20].

Table 1
Set of functions of Nano-networks.

Function	Description
void **ACTIVATE** (activation-time, ID)	This function is generated by the controller to trigger the Nano-device by sending a signal to a certain Nano-device's ID to make it active on a specific time. Triggering process can be represented by sending a sequence of bits or it can be performed by stimulating the Nano-device/molecular by any signal coming from the Nano-sensor such as pH, temperature, light, ultrasound, frequency, magnetic field and enzymes. As a result, Nano-device's status will be transited from OFF-state to ON-state. Nano-device's ID can be a specific device or a group of devices, i.e., it can be unicast address, multicast address, broadcast address or anycast address.
void **DEACTIVATE** (deactivation-time, ID)	This function is generated by the controller to deactivate the Nano-device by sending a signal to a certain Nano-device's ID to deactivate it on a specific time. As a result, Nano-device's status will be transited from ON-state to OFF-state, which can represent a sleeping state to save energy consumption.
void **OPERATE** (execution-time, ID)	This function is generated by the controller and it sends an order of operation to a certain Nano-device's ID to start a specific operation during a certain execution time. Particles releasing, capturing, detecting, measuring or assembling can be an example for an operation.
void **MOVE** (ID, location)	This function is generated by the controller and sent to the Nano-device. It drives a Nano-device with a certain ID to a deterministic location.
void **ABORT** (ID)	This function is generated by the controller and sent to the Nano-device. It can be used to exit the injected Nano-device from the network and terminate its functionality after a certain process has been accomplished in the area of interest.
data, ID **LISTEN** (void)	This function is generated by the Nano-device and it is received by the controller to listen the possible coming messages from a Nano-device with a specific ID.
ID **ACK** (message)	This function is generated by the Nano-device with a specific ID and it represents an acknowledgment message coming from the activated Nano-device to the controller.
location **LOCATE** (ID)	This function is generated by the Nano-device and it is received by the controller. It is used to send the location of a certain Nano-device with a specific ID to the controller.

it demonstrates the sequence of messaging exchange between the layers and shows the execution ordering of functions across the computational architecture. We are proposing five use cases: general activation, advanced activation, operation request, localized operation request and responsive operation request.

4.1. General activation use case

The simplest considered use case is a group of Nano-devices, which are able to receive an activation signal from the NFV controller, with no feedback or acknowledgment message sent from

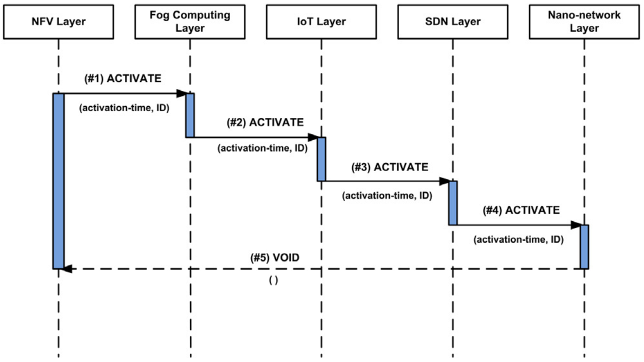

Fig. 12. Sequence diagram of general activation use case.

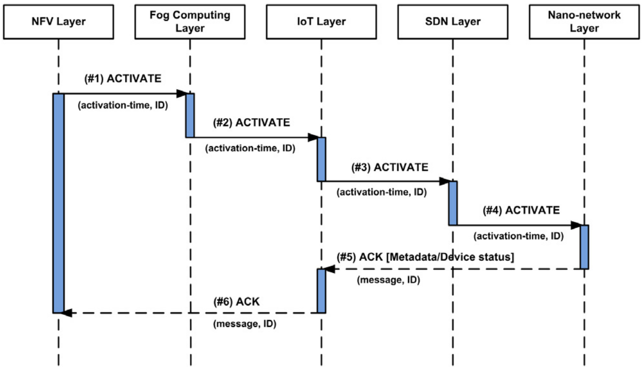

Fig. 13. Sequence diagram of advanced activation use case.

these Nano-devices to the controller. As illustrated in Fig. 12, the controller will send an activation signal to a certain Nano-device's ID to make it active on a specific time. Triggering process can be represented by sending a sequence of bits or it can be performed by stimulating the Nano-device by any signal coming from the Nano-sensor such as pH, temperature, light, ultrasound, frequency, magnetic field and enzymes. As a result, Nano-device's status will be transited from OFF-state to ON-state. Nano-device's ID can be a specific device or a group of devices, i.e., it can be unicast address, multicast address, broadcast address or anycast address. The proposed ACTIVATE function will be used in this use case and it will cross all the architecture's layers reaching the Nano-network layer. Despite the decision of the function is at NFV layer and its implementation is at the Nano-network layer, there will be an additional delay to the execution time caused by passing the function by other layers of the architecture.

4.2. *Advanced activation use case*

In the advanced activation use case, a modification on the general activation case can be added, where the NFV controller will receive a feedback from the Nano-devices to determine their status whether they are activated or not (i.e., ON/OFF status). So, ACK proposed function will be used besides ACTIVATE function. Fig. 13 shows the sequence diagram of the advanced activation use case. Once Nano-device receives the activation signal, it will reply with acknowledge message informing its status. This type of data coming from the Nano-device is considered as metadata, which will be sent to IoT layer for checking, then another acknowledgment message from IoT layer will be sent to NFV layer. Hence, the connection between the controller and Nano-device is established and acknowledged.

4.3. *Operation request use case*

In the operation request use case, the NFV controller will request a specific action from the Nano-devices. After an initial communication between the NFV controller and Nano-devices (i.e., activation and acknowledgment), the controller will request a specific operation from the Nano-device using the proposed OPERATE function. The controller will send an order of operation to a certain Nano-device's ID to start a specific operation during a certain execution time. Particles releasing, capturing, detecting, measuring or assembling can be an example for an operation. After the required operation has been accomplished, the controller will send an order of termination to the Nano-devices using ABORT function and Nano-devices will acknowledge receiving this function accordingly. This will be used to exit the Nano-device from the network and terminate its functionality. So, in this use case ACTIVATE, ACK, OPERATE and ABORT proposed functions will be used. Fig. 14 illustrates the sequence diagram of the operation request use case.

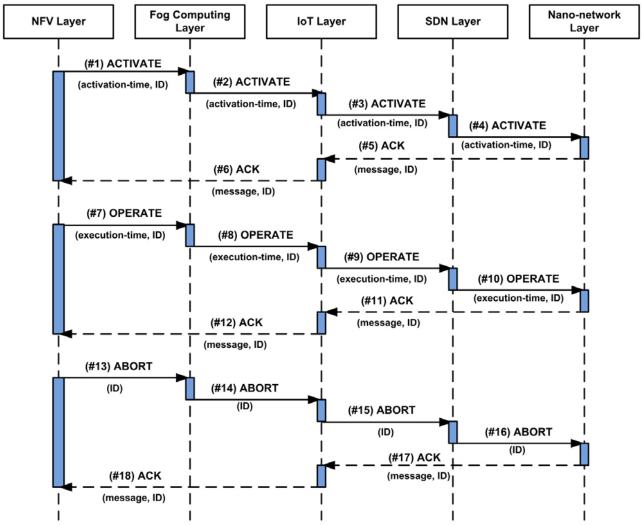

Fig. 14. Sequence diagram of operation request use case.

4.4. *Localized operation request use case*

The aim of localized operation request use case is to carry out an operation in a specific location of interest. As illustrated in Fig. 15, ACTIVATE, ACK, MOVE, LOCATE, OPERATE and ABORT proposed functions will be used in such use case. After an initial communication between the NFV controller and Nano-devices (i.e., activation and acknowledgment), the controller will drive a certain Nano-device's ID to a deterministic location using MOVE function. Once the Nano-devices receive the move signal they will act accordingly and send a locate message using LOCATE function with their specific ID and location. A location checkup process will be occurred in the Fog computing layer using ONS server to confirm that the Nano-devices have reached the accurate position, then an acknowledge message will be sent to the controller in NFV layer. Hence, the controller will send an order using OPERATE function to Nano-devices with a certain execution time, which will cross all layers to reach Nano-devices. As a result, the Nano-devices will acknowledge the operation message and start the required process. When the process execution time has been expired, the controller will send an ABORT message to the Nano-devices across all layers to terminate their functionalities and the Nano-devices will acknowledge receiving this function.

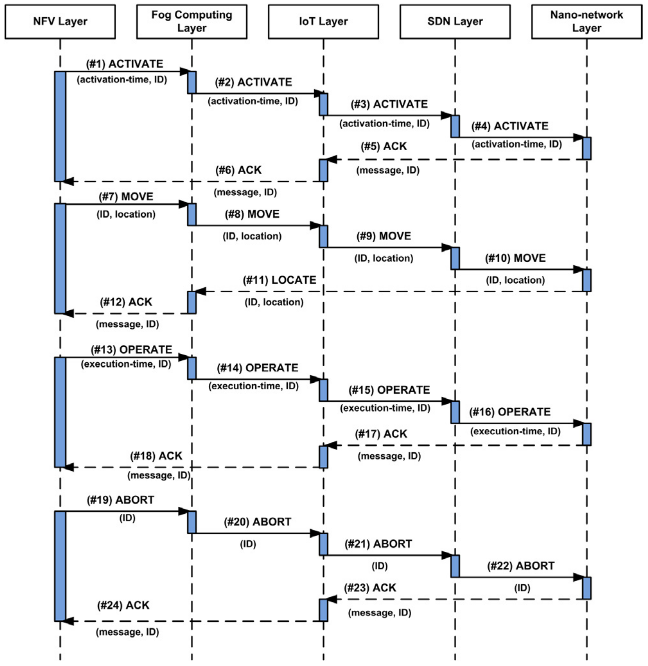

Fig. 15. Sequence diagram of localized operation request use case.

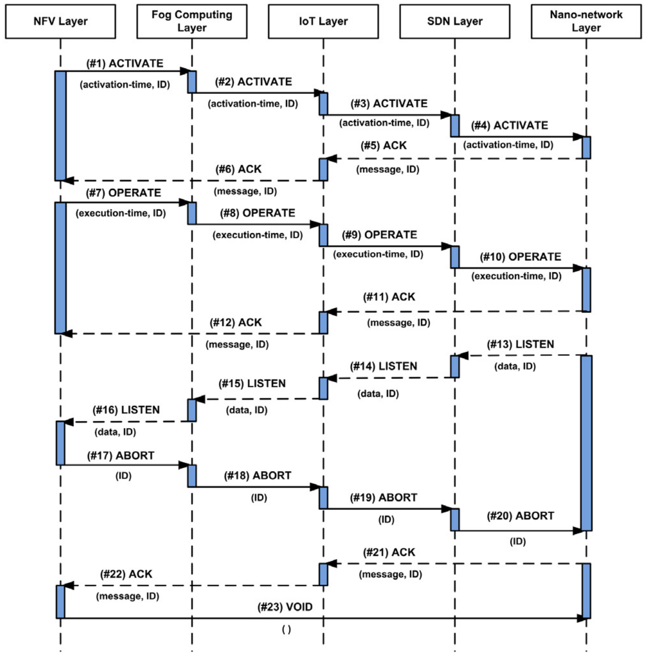

Fig. 16. Sequence diagram of responsive operation request use case.

4.5. *Responsive operation request use case*

In the responsive operation request use case, the NFV controller is waiting for a certain response from the Nano-devices. It can be considered as a feedback operational use case, where ACTIVATE, ACK, OPERATE, LISTEN and ABORT proposed functions will be used in such use case as showed in Fig. 16. After an initial communication between the NFV controller and Nano-devices (i.e., activation and acknowledgment), the controller will send an order using OPERATE function to Nano-devices. During the operational time, the controller will be waiting for a certain response from the Nano-devices carrying information. So, LISTEN function will appear on the scene providing the required gathered information back to the controller. Based on the received data, the controller shall preform some analyses according to the system's application, as a result it will send an ABORT message to the Nano-devices across all layers to stop the operation, hence the Nano-devices will acknowledge receiving this function.

5. Functions application in real Nano-networks

Coordination and information sharing among several Nano-machines will expand the potential applications of individual devices both in terms of complexity and range of operations. The resulting Nano-networks will be able to cover larger areas and reach hard-to-reach locations. Moreover, the interconnection of Nano-scale devices with classical networks defines a new networking paradigm refers as the Internet of Nano-things [41]. When it arrives, the Internet of Nano-scale things could provide much more detailed, inexpensive and up-to-date pictures of our cities, homes, factories even our bodies. Today traffic lights, wearables or surveillance cameras are getting connected to the Internet with billions of expected sensors harvesting huge amounts of real-time information and beaming it up to the cloud [41]. In this section, we will introduce some of potential applications of Nano-networks using the proposed use cases illustrated in Section 4 showing that Nano-technology has the ability to create new applications in different fields such as medical field, environmental field and industrial field.

5.1. *Medical field*

5.1.1. *Targeted drug delivery*

In intra-body networks, using the legalization process by the corresponding medical and health organizations, Nano-machines such as Nano-sensors and Nano-actuators can be injected inside the human body and controlled over the Internet by an external user such as a health care provider. Nano-devices can be used in medical treatment applications of in-body networks, where Targeted Drug Delivery (TDD) is one of these major applications [4,20]. By using the localized operation request use case showed in Fig. 15, the drug delivery operation can be processed. Firstly, the patient's body will be injected with Nano-devices/capsules carrying the required drug. Once the activation signal is received and acknowledged by the Nano-capsules, the controller will send MOVE signal to the Nano-capsules to move to the specific location, where drugs shall be injected into cells. Once the Nano-capsules receive MOVE signal, they will act accordingly and swim to the desired location inside the body. When they reach their location, they will send LOCATE message with their specific ID and location. A location checkup process will be occurred in the Fog computing layer using ONS server to confirm that the Nano-capsules have reached the accurate position, then an acknowledge message will be sent to the controller in NFV layer. Hence, the controller will send OPERATE command to the Nano-capsules with a certain execution time, which will cross all layers to reach Nano-devices. As a result, the Nano-capsules will acknowledge the operation message and release the carried drugs. When the process execution time has been expired, the controller will send ABORT message to the Nano-capsules across all layers, hence the Nano-devices will acknowledge accordingly to end up their functionalities and exit from the human body.

5.1.2. *Wearable health tracking system*

Nano-devices can be used in health tracking systems. By hooking up to wearable health trackers, medical persons can monitor real-time measurements of the human body like heart beats rate, blood pressure, breath tests and get an early detection of sickness [20,41]. The responsive operation request use case illustrated in Fig. 16 can be used in this application, where ACTIVATE, ACK, OPERATE, LISTEN proposed functions will be used for such use case. After the injection process of the Nano-devices into the human body, the activation signal from the controller to the Nano-devices will be sent across all layers and an acknowledged reply by Nano-devices will be sent to IoT layer then to NFV layer respectively. The controller will send an order to Nano-devices using OPERATE function to track the body measurements. Nano-devices will send acknowledge message to the controller, then start the tracking process. Once the Nano-devices collect the required information, they will send a message to the controller using LISTEN function carrying the required data. Once the controller in NFV layer receives the corresponding data from the Nano-devices, medical analyses will be occurred on the body measurements. In addition, it will send a termination message using ABORT function to the Nano-devices, which will acknowledge accordingly to end up the process if needed.

5.2. *Environmental field*

5.2.1. *Spread detection of viruses and diseases*

By releasing Nano-machines into the air, Nano-devices can play an important role in monitoring the spread of viruses and diseases in public locations [41]. Using the responsive operation request use case showed in Fig. 16, ACTIVATE, ACK, OPERATE, LISTEN and ABORT proposed functions will be used. Nano-devices will be released into the air using a controlled manner. After an activation signal has been sent from the controller to the Nano-devices across all layers and an acknowledged reply by Nano-devices has been sent to IoT layer then to NFV layer respectively, the controller will send an order to Nano-devices using OPERATE function to detect viruses and diseases, which spread into air. Nano-devices will send ACK message to the controller, then start the detecting process. Once the Nano-devices collect the required information, they send a message to the controller using LISTEN function carrying the required data. Once the controller in NFV layer receives the corresponding data from Nano-devices, some analyses will be occurred based on the collected data. As a result, the controller will send a termination message using ABORT function to Nano-devices, which will acknowledge accordingly to end up the detecting process if needed.

5.2.2. *Fertilizers and pesticides delivery in agriculture*

Nano-technology has the potential to bring important changes in agricultural industry. Some of these challenges include the increasing threats to agricultural production and risks of plant-related diseases. The agricultural sector will benefit greatly from Nano-technology tools to detect diseases in a rapid manner, improve the ability of plants to absorb nutrients and promote molecular treatment of diseases [42]. By using the localized operation request use case showed in Fig. 15, releasing fertilizers and pesticides in a controlled fashion can be achieved ensuring that they reach the intended destination and limiting their environmental impact. Firstly, the soil will be injected with Nano-devices/capsules carrying the required fertilizers and pesticides. Once the activation signal is received and acknowledged by the Nano-capsules, the controller will send MOVE signal to the Nano-capsules to move to the specific location inside the soil, where fertilizers and pesticides shall be injected. Once the Nano-capsules receive MOVE signal, they will act accordingly and swim to the desired location inside the soil. When they reach their location, they will send LOCATE message with their specific ID and location. A location checkup process will be occurred in the Fog computing layer using ONS server to confirm that the Nano-capsules have reached the accurate position, then an acknowledge message will be sent to the controller in NFV layer. Hence, the controller will send OPERATE command to the Nano-capsules with a certain execution time, which will cross all layers to reach the Nano-devices. As a result, the Nano-capsules will acknowledge the operation message and release the carried fertilizers and pesticides. When the process execution time has been expired, the controller will send ABORT message to the Nano-capsules across all layers, hence the Nano-devices will acknowledge accordingly to end up their functionalities.

5.3. *Industrial field*

5.3.1. *Food packing industry*

Nano-technology can transform the food industry by bringing changes in the production, processing, packaging and consumption of food. Usage of Nano-technology in these processes ensures safety of food products, creates a healthy food and enhances the nutritional quality. Smart food packaging systems can be developed using Nano-technology that in turn increases the shelf-life of food products by developing active anti-microbic surfaces. A number of companies have started to develop smart packaging systems to prevent drying of food content and protect it from oxygen and moisture [42]. The operation request use case showed in Fig. 14 can demonstrate the Nano-devices role in food packing industry, where ACTIVATE, ACK and OPERATE proposed functions are used. Nano-devices/molecules are added into food products in order to deliver nutrients and increase the absorption of oxygen and moisture. After activation and acknowledge functions are implemented, the NFV controller sends an operational order for the Nano-devices/molecules to start releasing nutrients and increasing the absorption of oxygen and moisture in the packed food. Nano-devices/molecules will reply with an acknowledge message and start the required process.

5.3.2. *Assembling ions in device manufacturing*

The assembly of components is a key feature of device manufacturing. Nano-technology can play a vital role in rapid pattern devices by directly assembling ions and materials without expensive physical masks and complex processes [43]. The operation request use case showed in Fig. 14 can represent the application of assembling ions and materials in industrial field for a certain product, where ACTIVATE, ACK, OPERATE and ABORT functions are used. After activation and acknowledge functions are implemented, the NFV controller sends an operational order for the Nano-devices to start the assembling process. Nano-devices will reply with an acknowledge message and start assembling the ions and materials. Once the process has been accomplished, the controller will send ABORT message to the Nano-devices, which will acknowledge accordingly to terminate the process.

6. Performance evaluation issues of the proposed computational architecture of Nano-networks

In general term, the added value of the proposed computational architecture of Nano-networks is to provide a high-level functionality for Nano-devices and reduce the communication gap between user's application and Nano-machines. The performance evaluation of the computational system is robustly required in order to provide a clear idea of the system efficiency, especially with real-time applications, which require fast, immediate and effective actions. In this section, we will discuss some issues regarding the performance evaluation of the proposed computational architecture of Nano-networks by stating some challenges that face Nano-networks and show how the proposed computational architecture will provide an added value to overcome these challenges. As illustrated in Table 2, complex computations, functions

Table 2
Performance evaluation issues of the proposed computational architecture of Nano-networks.

Category	Current challenge	Proposed solution's added value
Complex computations	– Nano-network topology might be variable, random and dynamic with time, because of abrupt environmental or medium conditions, which will affect the reliable transmission of data. So, Nano-devices within the same Nano-network topology might not have the same topology information. – With limited memory, storage and computational processing capabilities, Nano-devices face limitations to store routing tables, protocol codes and algorithms, as they can only produce packets with a small number of bits.	– With a single-hope transmission to Nano/Micro-gateway, High-Performance Computing (HPC) architecture will handle all the required actions/decisions to perform such complex computations instead of processing them on a Nano-device. – Micro/Nano-gateway, which is fully controller by SDN controller, has full knowledge of the network topology and routing tables, so it has the capacity and ability to reconfigure the Nano-network behavior accordingly.
Functions and services	– Standalone Nano-device can do very simple tasks, as a result there is a gap between Nano-devices and end-user's applications. – Current service-oriented architectures cannot deal properly with the large variety of Nano-network's services.	– The proposed computational architecture represents the middleware to abstract the futuristic Nano-network functionalities and allow end user's applications to access these data coming from Nano-devices to preform data analysis and programing activities using softwarized network technologies. i.e., SDN and NFV. – Using the simplest functions that can be done by Nano-devices to be collaborated altogether using the proposed computational architecture will create different complex operations and different applications use-cases. – Using distributed interaction layers to divide the process/service into different layers (i.e., access layer, data collection layer, computing layer and application layer) each layer has specific service and function, which will be more relevant.
Data analysis	– Data propagation between Micro/Nano-gateways and Nano-devices could result some delays for messages, so transmission of information could take considerable time, especially when it expects feedback (i.e., look up addresses or perform path calculations). – Large quantities of data processing and storage considering thousands, millions or billions of Nano-machines.	– Using Fog computing with Internet of things technologies in the proposed computational architecture will play a vital role in local computations and data analysis, making the Nano-network bandwidth and delay more conservative.
Energy conservation	– When Micro/Nano-gateway processes information coming to/from each Nano machine, considering thousands, millions or billions of Nano-machines, the harvested energy of Nano-gateway and/or Nano-machines will be rapidly consumed.	– Multiple OpenFlow switches will be used externally, where each switch will communicate with hundreds of Micro/Nano-gateways. – Dynamic timing synchronization of SDN controller and NFV controller with Micro/Nano-gateways and Nano-devices will make it possible to determine when to force a sleeping state to conserve energy. So, SDN controller, which fully controls Nano-gateways will take the lead to adjust the sleeping states of each Nano-gateway and/or Nano-machines using ACTIVATE and DEACTIVATE proposed functions.

and service, data analysis and energy conservation are considered vital challenges that face Nano-networks at present. Furthermore, numerical performance evaluation can be studied in details for each use case proposed in Section 4 by calculating the total number of actions/decisions required with the total time duration to implement a complete use case, which will be studied in our future work.

7. Open research challenges

This section highlights the future research directions that may propel the Nano-networks research in the near future. They are identified as Nano-devices addressing scheme, Nano-network system scalability and Nano-network system complexity.

7.1. Nano-devices addressing scheme

As Nano-devices have limited energy and computational processing, addresses with reasonable size are needed to consume less processing power. Moreover, the address domain shall be adequate to support the expected enormous number of fabricated Nano-machines. IPv4 provides 32-bit address space of four billion address, however it is not even enough to give each person on earth a unique identifier. By addressing Internet of things using IPv6 concept, there is no need to fear that there will not be enough IP addresses for things. IPv6 is providing 128-bit addresses, this makes the address needs of IoNT will be sufficient. The real question is whether "everything" needs its own IP address. The answer of this question is no, because in today's Internet, things are mostly servers and switches, firewalls, routers, laptops, phones and tablets with IP to IP connectivity. When we start talking about refrigerators, clothing, thermostats, light bulbs and Nano-devices/machines, they do not need to be directly on the Internet with an IP address [32]. By using Radio Frequency Identification (RFID) tags and readers, it can be an addressing mechanism for each Nano-device.

RFID systems are composed of one or more readers and several electronic tags. These tags are characterized by unique identifier that takes the form of binary number. These tags are applied to objects and even persons or animals. Physically, RFID tag is a small microchip attached to an antenna that is used for both receiving the reader signal and transmitting the tag ID. By using IPv6 addressing scheme, we can map tag IDs to network addresses. In the Internet of things, all objects, virtual as well as physical, are interconnected and reachable via IPv6 in combination with RFID technology [44]. Essentially, the tag connects to a physical Nano-device that we want to authenticate and track, when it comes in contact with the reader. A reader can read tag's information, while the back-end database keeps information related to different tags/readers. The reader triggers the tag's transmission by generating an appropriate signal, which represents a query for the possible presence of tags in the surrounding area and for the reception of their identification codes/IDs [44].

RFID tags can be online when they are in the electric field of a reader. Electronic Product Code (EPC) has been developed to support the extensive use of RFID in modern and IoT networks. EPC is used as a pointer to find additional data about a certain object to which it attaches. This additional data should be stored on a server connected to the enterprise network and/or Internet.

Bits	48 (or more)	16 (or fewer)	64
Field	Routing Prefix	Subnet ID	Interface Identifier

Fig. 17. General IPV6 unicast address format.

Bits	8	4	4	112
Field	Prefix	Flags	Scope	Group ID

Old Format

Bits	8	4	4	4	4	8	64	32
Field	Prefix	ff1	Scope	ff2	reserved	Plen	Network Prefix	Group ID

New Format

Fig. 18. General IPV6 multicast address format.

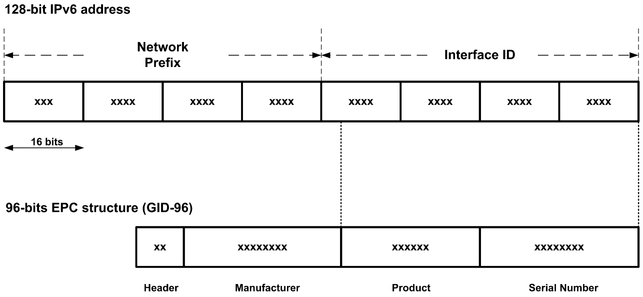

Fig. 19. Mapping an EPC code into IPv6 address format [44].

The server is identified via a look-up system, which is called Object Name Service (ONS). This server acts as a directory service to look up devices IDs. The ONS resolution process takes the EPC code and returns device's information and/or location [44].

IPv6 address format is a 128-bit address that takes the form of a 64-bit network prefix and 64-bit host/interface. There is no broadcast in IPV6 and it is replaced by multicast. Unicast, anycast and multicast address general formats are illustrated in both Figs. 17 and 18, where the routing prefix is a value assigned to a cluster of subnets and its prefix is designed to be structured by Internet Service Providers (ISPs).

As the data layout of an EPC code is 96-bit, Fig. 19 shows the IPv6 address format and gives an example on how 96-bit EPC address can be mapped to an IPv6 network address.

It can be observed that not all 96 bits of the EPC code can be fitted within the host suffix/interface identifier of an IPv6 address 64 bits. Hence, a problem in the mapping of a tag ID to an IPv6 address has been raised, because there are different passive RFID standards that have no common ID structure for tags [44]. There are some proposed solutions to overcome this problem, they are classified based on EPC code length. For EPC code with a longer length of 64 bits, identifying if there are some bits in the longer EPC header can be removed without affecting the uniqueness property of the tags. Another proposed solution is that EPC bits can be hashed into a length of 64 bits and then use the direct mapping method, however the risk of this operation is requiring a separate system to perform the mapping with an extra computational power as the process is based on cryptography. For other EPC codes with a shorter length of 64 bits, a zero-padding technique to the tag ID from left to right can be used to complete 64 bits size. This operation can provide a solution; however, the risk is there might be an address collision with another auto-configured network system use the same IP address [44]. It is concluded that there is a trade-off between a convenient addressing scheme of the Nano-device and the constraints we are facing in Nano-network paradigm, which need more research investigations in the near future. Some of the potential research directions in this area is to find a compatible address format with the existing Nano-network system or searching for a new addressing scheme such as the bridged networking (i.e., address translator) to perform direct translation from one side of the network (Intranet) to the other side (Internet) and vice versa. It is important in case of the Nano-machine needs to communicate globally over Internet. Another research direction can be investigating Network Address Translation (NAT) for the Nano-networks, which is a special case of the bridged networking.

7.2. Nano-network system scalability

Scalability is the capability of a system to handle a growing number of tasks. A system is considered to be scalable, if it is capable of increasing its total output under an increased load when resources are added. A Nano-network system is considered scalable, when its performance improves after adding additional Nano-devices with additional computational functions. A Nano-network system will be suitably efficient and practical, when it applied to large situations (e.g., a large input data set, a large number of outputs or a large number of participating Nano-machines/devices). If the system fails when a quantity increases, it will not be a scalable system. Fully integration between Nano-machines has not been built to date. As a result, many of the developed solutions cannot be experimentally validated. However, very advanced emulation tools are available for the partial validation of analytical models. In [6], an emulation of a one-to-one Nano-link between two Nano-machines had been developed, hence scalability of Nano-network system is an open research area, so the research community shall target how to manipulate and control thousands, millions or billions of Nano-devices forming a Nano-network.

7.3. Nano-network system complexity

Computational complexity theory introduces a mathematical model of computation to study the problems of a Nano-network system and quantify the amount of resources needed to solve them such as time, storage and the amount of communication. More precisely, computational complexity theory tries to classify problems that can or cannot be solved with appropriately restricted resources. Reference to the Nano-network architecture model proposed in Section 3, calculating the number of actions and decisions needed to run a certain function and the corresponding time cycle from its start at NFV layer till its end at Nano-network layer or vice versa will provide a clear indication how far the complexity of the system would be. Complexity of Nano-network system is still an open research area, so more cooperation between research communities is encouraged.

8. Conclusion and future work

Developing networks of Nano-scale devices with very limited operational capabilities needs different strategies with a new communication paradigm. One of these strategies is to implement an architecture able to control and manage a set of Nano-devices. In this paper, firstly we proposed an unified architectural model of

Table 3
The proposed changes related to Nano-machines communication domain.

Category	The proposed change
Infrastructure	The traditional network infrastructure is converted into millions or billions of Nano-machines.
Control plane	SDN controller represents the control plane.
Data plane	Micro/Nano-gateways with OpenFlow switches represent the forwarding/data plane.
Communication	SDN controller communicates with Micro/Nano-gateway. SDN controller speaks OpenFlow with the Micro/Nano-gateway. The communication between the Micro/Nano-gateway and Nano-machines by using different triggering techniques or sending stream of bits.
Synchronization	It is a must to keep full synchronization and orchestration between both controllers of SDN and NFV to fully implement Nano-network's functions and use cases.

Nano-networks communication with a layered approach combining Software Defined Network (SDN), Internet of Things (IoT), Fog Computing, Network Function Virtualization (NFV) technologies and present how these combination can control and manage the Nano-network in terms of the provided functionalities from a high-level perspective. Secondly, we proposed a set of functions and use cases that can be implemented by Nano-devices with some real relevant applications and discuss the significant challenges and open research problems we are facing in implementing these functions with Nano-technology paradigm. Table 3 summarizes the proposed changes related to Nano-machines communication domain, while using the proposed unified computing architecture.

Our future work will study the scalability and complexity of the proposed computational architecture, providing theoretical modeling approach and numerical performance evaluation of the system. Also, we will study an appropriate addressing scheme for the Nano-devices taking into consideration the limitation of processing and power capabilities. In addition, we will check how to implement some functions of Nano-devices form the proposed set and create a real test-bed for this implementation trying to study the performance of Nano-devices in an emulated Nano-network environment.

Acknowledgment

This work has been supported by the "Ministerio de Economía y Competitividad" of the Spanish Government under project TEC2016-76795-C6-1-R and AEI/FEDER, UE.

References

[1] I.F. Akyildiz, J.M. Jornet, M. Pierobon, Nanonetworks: A new frontier in communications, Commun. ACM 54 (11) (2011) 84–89.
[2] P. Kulakowski, K. Solarczyk, K. Wojcik, Routing in fret-based nanonetworks, IEEE Commun. Mag. 55 (9) (2017) 218–224.
[3] W.B. AZoNano, What is nanotechnology and what can it do? (2017) [cited 13.01.18]. URL http://www.azonano.com/article.aspx?ArticleID=1134.
[4] N.A. Ali, W. Aleyadeh, M. AbuElkhair, Internet of nano-things network models and medical applications, in: Wireless Communications and Mobile Computing Conference (IWCMC), 2016 International, IEEE, 2016, pp. 211–215.
[5] A. Paultre, Using virtualization to empower iot network infrastructure (2017) [cited 10.01.18]. URL http://www.insight.tech/communications/using-virtualization-to-empower-iot-network-infrastructure.
[6] J.M. Jornet Montana, Fundamentals of Electromagnetic Nanonetworks in the Terahertz Band, (Ph.D. thesis), Georgia Institute of Technology, 2013.
[7] Q.H. Abbasi, K. Yang, N. Chopra, J.M. Jornet, N.A. Abuali, K.A. Qaraqe, A. Alomainy, Nano-communication for biomedical applications: A review on the state-of-the-art from physical layers to novel networking concepts, IEEE Access 4 (2016) 3920–3935.
[8] P. Wang, J.M. Jornet, M.A. Malik, N. Akkari, I.F. Akyildiz, Energy and spectrum-aware mac protocol for perpetual wireless nanosensor networks in the terahertz band, Ad Hoc Networks 11 (8) (2013) 2541–2555.
[9] J.M. Jornet, I.F. Akyildiz, Channel capacity of electromagnetic nanonetworks in the terahertz band, in: 2010 IEEE International Conference on Communications, (ICC), IEEE, 2010, pp. 1–6.
[10] J.M. Jornet, J.C. Pujol, J.S. Pareta, Phlame: A physical layer aware mac protocol for electromagnetic nanonetworks in the terahertz band, Nano Commun. Netw. 3 (1) (2012) 74–81.
[11] J.M. Jornet, I.F. Akyildiz, Information capacity of pulse-based wireless nanosensor networks, in: 2011 8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, (SECON), IEEE, 2011, pp. 80–88.
[12] Z.L. Wang, Towards self-powered nanosystems: from nanogenerators to nanopiezotronics, Adv. Funct. Mater. 18 (22) (2008) 3553–3567.
[13] F. Cottone, H. Vocca, L. Gammaitoni, Nonlinear energy harvesting, Phys. Rev. Lett. 102 (8) (2009) 080601.
[14] L. Gammaitoni, I. Neri, H. Vocca, Nonlinear oscillators for vibration energy harvesting, Appl. Phys. Lett. 94 (16) (2009) 164102.
[15] I.F. Akyildiz, F. Brunetti, C. Blázquez, Nanonetworks: A new communication paradigm, Comput. Netw. 52 (12) (2008) 2260–2279.
[16] S. Balasubramaniam, J. Kangasharju, Realizing the internet of nano things: challenges, solutions, and applications, Computer 46 (2) (2013) 62–68.
[17] T. Nakano, T. Suda, Y. Okaie, M.J. Moore, A.V. Vasilakos, Molecular communication among biological nanomachines: A layered architecture and research issues, IEEE Trans. Nanobiosci. 13 (3) (2014) 169–197.
[18] M. Kuscu, O.B. Akan, The Internet of molecular things based on fret, IEEE Internet Things J. 3 (1) (2016) 4–17.
[19] M. Stelzner, F. Dressler, S. Fischer, Function centric networking: an approach for addressing in in-body nano networks, in: Proceedings of the 3rd ACM International Conference on Nanoscale Computing and Communication, ACM, 2016, p. 38.
[20] U.A. Chude-Okonkwo, R. Malekian, B.T. Maharaj, A.V. Vasilakos, Molecular communication and nanonetwork for targeted drug delivery: A survey, IEEE Commun. Surv. Tutor. 19 (4) (2017) 3046–3096.
[21] A. Tsioliaridou, C. Liaskos, S. Ioannidis, A. Pitsillides, Corona: A coordinate and routing system for nanonetworks, in: Proceedings of the Second Annual International Conference on Nanoscale Computing and Communication, ACM, 2015, p. 18.
[22] A. Tsioliaridou, C. Liaskos, L. Pachis, S. Ioannidis, A. Pitsillides, N3: Addressing and routing in 3d nanonetworks, in: 2016 23rd International Conference on Telecommunications, (ICT), IEEE, 2016, pp. 1–6.
[23] O.N. Fundation, Software-defined networking: The new norm for networks, ONF White Paper, 2 (2012) 2–6.
[24] C. Systems, Demystifying SDN for the Network Engineer, Cisco, CA, USA, White Paper Edition (2017).
[25] N. Bizanis, F.A. Kuipers, Sdn and virtualization solutions for the internet of things: A survey, IEEE Access 4 (2016) 5591–5606.
[26] G. Pujolle, Software Networks: Virtualization, SDN, 5G, Security, no. ISBN 978-1-84821-694-5, John Wiley & Sons, 2015.
[27] SDXCentral, What are sdn northbound apis (and sdn rest apis)? (2018) [cited 05.02.18]. URL http://www.sdxcentral.com/sdn/definitions/north-bound-interfaces-api/.
[28] SDXCentral, What are sdn southbound apis? (2018) [cited 05.02.18]. URL https://www.sdxcentral.com/sdn/definitions/southbound-interface-api/.

[29] M. Chiang, T. Zhang, Fog and iot: an overview of research opportunities, IEEE Internet Things J. 3 (6) (2016) 854–864.

[30] A.V. Dastjerdi, R. Buyya, Fog computing: Helping the internet of things realize its potential, Computer 49 (8) (2016) 112–116.

[31] J.A. Stankovic, Research directions for the internet of things, IEEE Internet Things J. 1 (1) (2014) 3–9.

[32] A. Jafarey, The internet of things and ip address needs (2015) [cited 15.11.17]. URL http://www.networkcomputing.com/networking/internet-things-ip-address-needs/1170065007.

[33] C. Systems, Unleash the power of the Internet of Things, Cisco, CA, USA, White Paper Edition (2015).

[34] C. Systems, the Internet of Things: Extend the Cloud to Where the Things Are., Cisco, CA, USA, White Paper Edition (2015).

[35] F. Bonomi, R. Milito, J. Zhu, S. Addepalli, Fog computing and its role in the internet of things, in: Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, ACM, 2012, pp. 13–16.

[36] R. Mijumbi, J. Serrat, J.L. Gorricho, N. Bouten, F. De Turck, R. Boutaba, Network function virtualization: State-of-the-art and research challenges, IEEE Commun. Surv. Tutor. 18 (1) (2016) 236–262.

[37] C. Systems, Network Optimization Through Virtualization: Where, When, What, and How?, Cisco, CA, USA, White Paper Edition (2016).

[38] G. ETSI, Network functions virtualisation (nfv): Architectural framework, ETSI Gs NFV 2(2) (2013) V1.

[39] A. Fischer, J.F. Botero, M.T. Beck, H. De Meer, X. Hesselbach, Virtual network embedding: A survey, IEEE Commun. Surv. Tutor. 15 (4) (2013) 1888–1906.

[40] G. Cloud, Overview of internet of things —solutions —google cloud platform. (n.d.) (2017) [cited 13.11.17]. URL https://cloud.google.com/solutions/iot-overview.

[41] P. Tracy, What is the internet of things at nanoscale? (2016) [cited 10.11.17]. URL http://www.rcrwireless.com/20160912/big-data-analytics/nano-scale-iot-tag31-tag99.

[42] W. Soutter, Nanotechnology in agriculture (2013) [cited 22.01.18]. URL http://www.azonano.com/article.aspx?ArticleID=3141.

[43] N. Liu, F. Wang, L. Liu, H. Yu, S. Xie, J. Wang, Y. Wang, G.B. Lee, W.J. Li, Rapidly patterning micro/nano devices by directly assembling ions and nanomaterials, Sci. Rep. 6 (2016) 32106.

[44] M.B.I. Reaz, Radio Frequency Identification from System to Applications, no. ISBN 978-953-51-1143-6, InTech, 2014.

Akram Galal received the B.Sc. in Electronics and Communication Engineering from University of Alexandria, Egypt, and the M.Sc. degree in Electronics and Communication Engineering from Arab Academy for science & Technology and Maritime Transport, Egypt in 2010 and 2017 respectively. He received a post graduate diploma in Computer Networks from Information Technology Institute (ITI), Cairo, Egypt in 2011. From 2011 to 2014, he served as an Enterprise Networks Engineer at TE Data, Egypt. From 2014 to 2017, he served as a Solution Design Consultant at Tawasul Telecom, Kuwait. In 2017, Akram joined Design, modeling and evaluation of broadband networks (BAMPLA) research group pursuing his Ph.D. in Network Engineering department at Universitat Politècnica de Catalunya (UPC). His research interests span Software Defined Networks, Internet of Things, Fog Computing, Network Function Virtualization and Nano-network.

Xavier Hesselbach is an Associate Professor at the Department of Network Engineering (Enginyeria Telematica) at the UPC, and IEEE Senior Member (http://www-entel.upc.edu/xavierh/). He received the M.Sc. degree with honors in Telecommunications Engineering in 1994, and the Ph.D. degree with honors in 1999, from the Universitat Politècnica de Catalunya (UPC). In 1993, he joined the Design, modeling and evaluation of broadband networks group in the Network Engineering Department of UPC. His research interests include networks virtualization, resources management, broadband networks, quality of service and green networking. He has been involved in several national and international projects, and he is author of 4 books and more than 100 national and international publications in conferences and journals. In 1994, he received the award from the COIT/AEIT of Spain for the best Master Thesis on Networks and Telecommunication Services. He has participated in the technical program committees of several conferences, also he has been the Information Systems and Internet Chair in Infocom 2006, and guest editor of the Ad Hoc Networks Journal and the Journal of Electrical and Computer Engineering. He has taken part in several European and Spanish research projects, such as the EuroNGI/FGI/NF Network of Excellence, COST293, Mantychore and All4Green being main UPC researcher in the Mantychore and All4Green projects.