



HAL
open science

An Architecture Framework for Virtualization of IoT Network

Toufik Ahmed, Abdelhamid Alleg, Nicolas Marie-Magdelaine

► **To cite this version:**

Toufik Ahmed, Abdelhamid Alleg, Nicolas Marie-Magdelaine. An Architecture Framework for Virtualization of IoT Network. IEEE Conference on Network Softwarization NetSoft, Jun 2019, Paris, France. hal-02140526

HAL Id: hal-02140526

<https://hal.science/hal-02140526>

Submitted on 27 May 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

An Architecture Framework for Virtualization of IoT Network

Toufik Ahmed¹, Abdelhamid Alleg¹, Nicolas Marie-Magdelaine^{1,2}

Univ. Bordeaux, Bordeaux INP, CNRS, LaBRI, UMR5800, F-33400 Talence, France

²Lectra SA, 23 Chemin de Marticot, 33610 Cestas, France

Email: tad@labri.fr, aalleg@labri.fr, nicolas.marie-magdelaine@labri.fr

Abstract—The *Internet of Things* (IoT) market is constantly opening new business and market opportunities to connect the digital and physical worlds in many sectors such as smart-city, industry, agriculture, building, transportation, energy, etc. To support this major growth, we anticipate the adoption of mainstream Software Defined Networking SDN, Software Defined Radio (SDR) and Network Function Virtualization (NFV) technologies within the IoT systems to allow providing new ways of offering services in a more flexible, agile and cost-effective manner than today. This paper investigates the provision of NFV in the Internet of Things through the virtualization of full stack IoT functions, from radio processing functions to processing capabilities, that can be orchestrated dynamically and deployed on generic IoT devices. We provide a proof of concept to demonstrate the provisioning of end-to-end virtualized service function chain using the GNU Radio framework and dockers containers for isolation and VNF deployment and execution

Keywords—Internet of Things (IoT), Network Function Virtualization (NFV), Software Defined Radio (SDR), Software Defined Networking (SDN).

I. INTRODUCTION

The Internet of Things industry is constantly looking for new improvement features that support cutting-edge service innovations with deployment velocity and business agility. As major key drivers in the evolution of IoT, Software Defined Networking / Radio (SDN, SDR) and Network Function Virtualization (NFV) are being positioned as central technology enablers towards decoupling IoT hardware from service deployment, leveraging an increasing in the number and type of services supported over a single deployed IoT platform. The support of virtualization concept within the IoT devices is envisioned to achieve critical cost reduction in the service offering, and at the same time, being able to easily bring a new set of innovative service into the market. This allows IoT device vendor to open their platform to a great extent, and at the same time, it enables IoT service provider to avoid vendor lock-in with proprietary hardware and software technologies. The IoT service provider will gain greater control over the IoT devices by simplifying network management with centralized management and control of IoT devices from multiple vendors, and creating opportunities for collaboration and interoperability.

Adopting SDR, SDN and NFV offers endless expectations that covers many distinct operational areas, such as fine-grained radio control and monitoring, advanced signal processing, e.g. collaborative signal processing, distributed enforcement of QoS, network resources management, in-network processing and storage of data including fusion and detection of complex events, etc.

The path toward the commoditization of IoT devices into a generic *whitebox* devices able to support most of IoT functions extended with out-of-tree functions will result also in scalability benefits since the underlying resources can be mobilized when

needed according to the aggregated demands, reducing the amount of specialized IoT resources to be provisioned and making feasible other modes of feature evolution. This concept sheds up light on new ways of designing IoT devices in order to improve IoT service offering with the support of virtualization and decentralization, to allow multiple isolated environments to be executed in a single device. As a result, cooperation among heterogeneous devices from multiple vendors, within an IoT network, is facilitated and at the same time, it provides resource efficiency and cost-benefit.

In this paper, we propose an IoT NFV architecture framework that leverages the above-mentioned advantages and provides flexible and reconfigurable solutions to create and deploy new customized on-demand virtualized IoT services. Our proposed architecture framework abstracts the underlying physical IoT network resources into a set of logical resources (radio, sensing capabilities, computational, memory and storage) used for service orchestration in terms of hosting, chaining and managing of IoT functions. Thus, this paper includes the following main contributions:

- Proposing an architecture framework to bring virtualization in IoT network based on introducing SDR, SDN and VNF paradigm shifting.
- Exploiting the virtualization paradigm introduced for IoT, to describe a set of use-cases and deployment scenarios.
- Presenting a proof of concept showing how the proposed architecture framework can be prototyped using Raspberry Pi acting as IoT devices and Docker containers as IoT functions.

The rest of this paper is organized as follows. Section II reviews some background knowledge and state of the art. Section III presents the proposed architecture that brings virtualization in IoT network and introduces the concept of Generic IoT device. Section IV presents some use-cases that can be derived by focusing on the advantages of the solutions in addressing well-known IoT challenges. Section V presents a proof of concept implementation. Finally, main conclusions are drawn in Section VI.

II. BACKGROUND AND STATE OF THE ART

IoT networks are complex and difficult to design, manage and operate in order to offer IoT services. These networks are composed of multiple, potentially heterogeneous IoT devices, acting as sensors and actuators, interconnected with each other and with IoT gateways. These latter offer connectivity to the rest of the Internet up to Cloud, where the service is exposed (stored, analyzed, presented and shared) to end-users. The IoT architecture involves a multitude of software and hardware components and calls for a wide range of protocols to allow device-to-device and device-to-cloud communications. Consequently, one of the major requirements for the successful and wide deployment of such environment to offer IoT services

concerns the efficient design, management and operation of IoT networks while addressing complex problems related to massively distributed applications, scalability support, and upgrading. Control architecture and virtualization using SDN and NFV are actually changing the way networks are designed. The efforts that led to these concepts are clearly related to the long research background on supporting programmable packet processing, active networks, in-band and out-band control, separation of control and data, network operating system, network virtualization, etc. Although network virtualization has played an important role in computer networks for many years, the virtualization concept for IoT is still in its infancy.

Many works suggested that the control plane for IoT is the appropriate place to introduce flexibility and programmability in order to achieve a multiservice environment with more reliability and guaranteed performance. Accordingly, the introduction of SDN in IoT has been proposed in many research papers [1][2][3][4][5][6] and various lessons have been learned for future deployments related to (1) seamless integration of wireless sensor networks and mobile networks, (2) ability to modify the network behavior according to user needs, (3) unified view on accessing, configuring and operating IoT cloud systems, (4) global optimizations with centralized methods, and (5) solving scalability issues within large IoT deployments, among many others.

A very little attention is given to IoT virtualization and the introduction of NFV concept for providing IoT service as a network slice offering an IoT Service deployment over virtualized IoT architecture. The survey [7] presented a set of works tackling SDN and virtualization solutions for IoT, but most of the items examined were concentrated on SDN applications. The study [8] suggested a new multi-layered IoT architecture involving SDN and NFV and proposed some use cases and founding principles for building an IoT infrastructure. In [9], we started an investigation on how NFV concept can be extended to IoT to allow decoupling IoT functionalities from specific dedicated devices and we proposed an energy efficient solution for the placement of IoT Service Function. However, the studied works lack a clear vision on the way to design a virtualization architecture framework for IoT network. In our case, we suggest the virtualization of full stack IoT functions using generic IoT device, from radio resources using SDR to network functions using SDN and NFV concepts. We also believe that the adoption of SDR, SDN and NFV over IoT networks will be a key enabler towards more flexible and agile end-to-end service provisioning that allows overcoming several existing limitations in terms of operational flexibility, evolvability and large-scale deployment.

III. PARADIGM SHIFT TOWARD GENERIC IOT DEVICE

Proprietary IoT devices and proprietary operating systems (OS) for IoT are dominating the market today, putting critical pressure on IoT service providers for always adopting vendor lock-in solutions which lack portability and interoperability. A paradigm shift has started to take place to adopt commercial off-the-shelf (COTS) devices with a shift to standards-based hardware such as the case of solutions from the Department of Defense (DoD) [10] used for military and tactical

communication. These COTS devices benefit from competitive pricing, interoperability and best practices while most of them are using open source OS [11], protocols and tools.

This paradigm shift opens a new perspective for leading IoT movement toward supporting Software Defined Radio / Networking (SDN, SDR) and Network Function Virtualization (NFV). The objective is to remove the dependency from dedicated and specialized physical IoT devices as well as abstracting physical resources into virtual resources that can be allocated when needed. With this approach, IoT service providers can concentrate on developing and optimizing software functions that can be dynamically orchestrated to respond more efficiently to changing market demands.

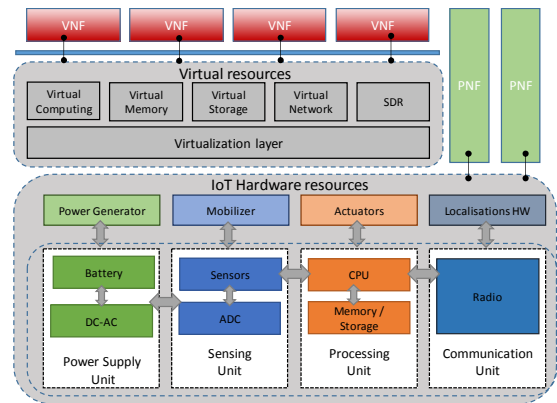


Figure 1: Paradigm shift for IoT devices

Figure 1 shows our proposal mapping of an IoT device to ETSI guideline for NFV reference architecture framework [12]. The decoupling of hardware and software exposes a new set of entities such as the virtualization layer, virtualized resources, the Virtualized Network Functions (VNFs). This mapping is augmented with SDR capabilities to allow access, control and management of the radio spectrum. VNFs can be chained with other VNFs and/or Physical Network Functions (PNFs) to realize a Network Service (NS). The PNFs are part of physical resources that remain as hardware components such as sensing capability that cannot be virtualized. The Network Services are created based on associated VNF Forwarding Graphs (VNFFGs), Virtual Links (VLs), PNFs, VNFs. In addition, the management and orchestration functions are not illustrated in this figure. Those functions are part of the IoT service provider infrastructure network.

The virtualization techniques used inside an IoT device are similar to those used for servers and cloud computing. The virtualization allows sharing of the physical resources so that multiple network functions (workloads in general) are executed while they are co-located with each other in a fully isolated environment. This represents a progressive manner to design, deploy and manage IoT Service.

The deployed IoT devices (generic IoT device, IoT gateway) form what we called, an **IoT network segment** that is usually connected to service provider cloud infrastructure (**cloud segment**) to deliver a full end-to-end IoT Service. The cloud runs a set of process for data aggregating, correlation, analysis, classification, visualization, etc. A fog computing or mobile

edge computing (MEC) can be deployed in the vicinity of the IoT network or within the backhaul to reduce the heavy burden on the cloud and to improve the service performance.

Figure 2 illustrates the architecture framework for virtualization of IoT network segment when connected to the cloud to deliver an end-to-end IoT service. This architecture is also aligned with MANO (Network Function Virtualization Management and Orchestration) framework. We focus on control and management planes only for IoT segment. In this architecture, we suppose that each IoT device follows the paradigm shift proposed in Figure 1. An IoT Service is represented by a sequence of PNFs and VNFs instances (IoT VNF and cloud-based VNF), chained together to compose an IoT Service (i.e. Network Service) that requires a particular amount of resources in terms of sensing capabilities, radio for communication, computational, memory and storage. In this context, the placement, management, chaining, and orchestration operations of these VNFs should be carefully considered to meet the required performances along with energy constraints to support diverse IoT services over the same shared infrastructure.

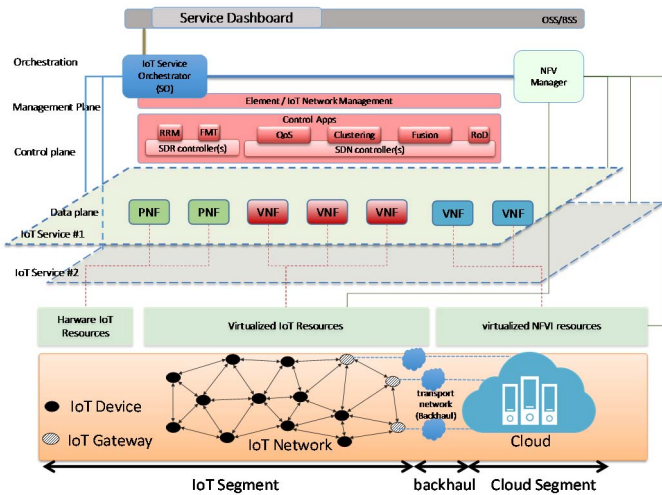


Figure 2: Virtualization architecture framework for IoT network

In particular, as illustrated in Figure 2, the following entities are considered as building blocks for deploying IoT services.

- IoT Physical Network Functions, essentially for sensing and radio communication and IoT Virtual Network Functions that represent the software part of an IoT device that runs as an isolated code on top of a virtualization layer.
- SDR-based applications and SDR controller for the realization of radio signal processing applications and radio-based control. In particular, using SDR for radio spectrum will allow an IoT device to be augmented with a set of wide range of software-based signal processing blocks connected to each other through flowgraphs that describe the data flow between those software blocks. The reconfigurability of the flowgraphs is a key feature provided by SDR. In this case, an IoT device will no more utilize a single specific radio with specific signal

processing and protocol, but instead, it will use a general-purpose radio that supports a wide-range of signal-processing applications that can be controlled through a centralized logic.

- SDN-based application and SDN controller for the realization of IoT-level control such as IoT routing, multipath forwarding, load balancing, clustering, in-network data fusion, etc.
- Element Management functions / IoT Network Management, which provide a package of management functions, e.g. Fault, Configuration, Accounting, Performance and Security (FCAPS) management, for IoT.

IV. USE CASES AND SCENARIOS

We present in the following some use cases that can be efficiently developed over the framework. The objective is to highlight some of the advantages of the solutions while addressing challenges related to deployment, configuration and operation, seamless integration with Internet, global optimizations, and scalability among others. Major advantages come with the use of generic IoT device augmented with full-stack virtualization capabilities that provide enhanced flexibility and re-configurability, software-based feature updates, simplified deployment procedures, and easy control of network topologies. These use cases are supported by a set of message sequence charts that are not illustrated for simplicity reasons.

A. Clustering

In the context of IoT and more generally **device-to-device communication**, the organization of the devices into clusters can be greatly enhanced with SDR, SDN, and NFV. There is a huge amount of work in the literature that tackled the problem of sensors network clustering to ensure low energy consumption as described by different surveys on this topic [13][14]. The clustering objectives for IoT are multiple: enhancing connectivity and communicating between nodes, providing load-balancing and fault-tolerance, and maintaining hierarchical topology to support scalability. However, as the cluster-head is selected based on some criteria, the selection process and the associated cluster-head functionalities should be provided in a flexible manner. Two relevant key features provided by our architecture framework related to clustering process could be (1) In-network processing, such as data aggregation and fusion, (2) Dynamic clustering, rotation and backup of cluster-head

B. Tracking

Target tracking techniques have been widely studied in the literature of wireless sensor networks [17]. The tracking consists of state-estimation of the target using techniques ranging from single-node to collaborative methods. The majority of these methods employs active prediction-based scheme coupled with selective activation of nodes activities. The nodes are then waked-up on-demand to follow the target path. In this context, the proposed virtualized IoT architecture framework can enhance the target tracking by extracting useful information using in-network processing, fine-grained radio control and monitoring, dynamic clustering, collaboration between nodes, coordination between communication-related and sensing-related operations, etc.

C. Tactical networking and high dynamic network

Tactical networks and high-dynamic networks are composed of extremely heterogeneous wireless and ad-hoc mobile nodes. They rely on a wide range of ground sensors, robots and UAVs aiming to provide mission-critical applications. These networks suffer from limited bandwidth and intermittent connectivity due to communication range, interference, mobility, and overhead induced by security requirements. As consequences, channel conditions and network topology vary over time. The service requirements are also extremely heterogeneous ranging from simple data dissemination to real-time communication with QoS support. This framework can be used to efficiently support service requirement for efficient tactical networks communication such as (1) Radio communication reprogramming and cognitive radio, (2) Network awareness and cross-layer networking.

V. PROOF ON CONCEPT IMPLEMENTATION

This section describes the primary implementation realized through a set of Raspberry PI 3 model B v1.2 acting as IoT devices. Each device is composed of single-board computer powered by a quad-core ARMv8 BCM2837B0 Cortex-A53 running at 1.2 GHz and equipped with a BCM43438 wireless LAN and Bluetooth Low Energy (BLE) chips. The board is also featured with 40 GPIO pins that can be controlled in software as input and output pin and used for wide range of purposes such as wiring many types of sensors and actuators. A list of 50 sensors is provided in [18]. Figure 3 illustrates our target application. The network is composed of devices having multiple sensing capabilities that can be activated when needed. The devices are deployed in a single building and we simulate a temperature measurement as an IoT service.

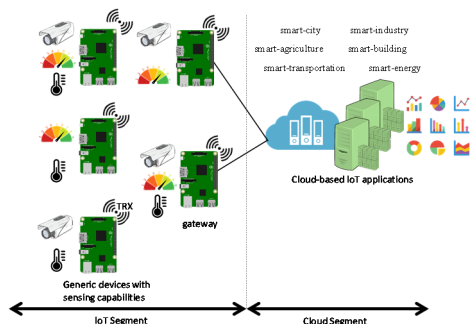


Figure 3: Network topology used as the target application

A. SDR Capabilities

Each device is augmented with a low-cost digital USB 2.0 stick with RTL2832U chipset initially designed for DVB-T reception but can support SDR capabilities. Another alternative is to use LimeSDR-Mini [19] USB stick which provides Transmission (TX) and Reception (RX) capabilities with SDR.

There exist a large set of general-purpose software supporting SDR such as SoapySDR [20], CubicSDR [21], GQRX [22] and Gnu Radio [23], which are used for monitoring and analyzing radio signal reception. In particular, GNU Radio is an open source toolkit for software development that provides building blocks for signal processing. The GNU Radio Companion (GRC) tool allows designing a flow graph by

connecting different blocks with input/output connections. Furthermore, tools such as rpitx [24] can allow the Raspberry Pi to transmit over a wide range of frequencies from 5 KHz up to 1500 MHz. This is achieved by connecting the GPIO 4 (Pin 7) with a band-pass filter and a wire acting as an antenna. This setup turns the Raspberry Pi as a general radio frequency transmitter. The rpitx tool can accept an I/Q signal as an input and can transmit I/Q signal back on a specific frequency. The combination of rpitx and Gnu Radio opens new capabilities and perspectives to ensure flexible radio manipulation and reconfiguration while helping to increase scalability, agility and enabling better use of radio resources. Our first setup allows us to have a generic IoT device that is easy to configure towards a new transmission technology and protocols, so it can adapt dynamically according to needs and scenarios. This proposed generic IoT device offers unmatched levels of programmability with the support of a high range of frequency bands and communication standards.

B. VNF Capabilities

The VNF capabilities that allow multiple isolated environments to be executed on a single Raspberry Pi device are provided by the virtualization layer based on containers technology using Docker [25]. This represents a fast way of creating, installing and running independent manageable Linux containers. Furthermore, to define, run and manage multi-container applications, Docker Compose is used to provide simple service composition facilities. We create an online repository from Docker Hub which is, in our case, the place where the IoT service provider is supposed to host the image of the different VNFs available to be on-boarded on the device. We show in the next section a scenario of service chain composition for clustering management for our temperature measurement as an IoT service.

C. Service Function Chain Composition and Clustering Management

We suppose that based on the battery depletion level a specific Cluster Head (CH) is elected. This CH is in charge of collecting the temperature level from all the Cluster Members (CM). Two VNFs are defined as docker images: Temperature_CM_VNF and Temperature_CH_VNF. The first VNF for the CM reads periodically the actual temperature from the PNF sensor. The second VNF for CH, performs a data fusion by providing the mean value (average) of the received temperature. The orchestration capability is done by a simple script based on CLI (Command Line Interface). However, it is possible to use advanced tools such as Kubernetes (k8s) which is an open-source software for automating deployment, scaling, and management of containerized applications. The logic sequence for the orchestration in our case is (1) A CLI for Docker client contacts the Docker daemon running locally in each device, (2) The Docker daemon pulls the appropriate image from the Docker Hub either Temperature_CM_VNF (for cluster member) or Temperature_CH_VNF (for cluster head), (3) The Docker daemon creates a new container from that image and runs the executable that produces the output, and (4) The Docker daemon streams that output to the Docker client, and then forwards the output to the next VNF in the SFC. Figure 4 shows the SFC composition for temperature measurement as an IoT service. This SFC is arranged as a PNF, two IoT VNFs and one

cloud VNF used for displaying the data. It is worth noting that an IoT device is able to manage multiple containers at the same time and one device can participate in multiple IoT services in a parallel manner.

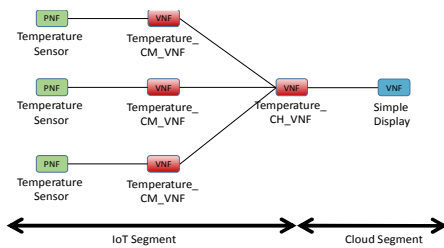


Figure 4: SFC composition for temperature measurement as an IoT service.

When the CH is updated according to the energy criteria, an updated SFC composition is triggered by the service orchestrator in order to install new image or to migrate existing container from Docker images (Temperature_CM_VNF and Temperature_CH_VNF) on the appropriate member (either CH or CM). In our case, the temperature service is stateless and there is no need to synchronize the different contexts of VNFs. Context and state synchronization can be solved using Docker Swarm so that the new CH joins the “Cluster head swarm” before the old CH leaves it. Our implementation demonstrates the ability of our solution to create and deploy new customized on-demand virtualized IoT services by managing an IoT network dynamically through the virtualization of full stack IoT functions, from radio communication processing functions to high-level processing capabilities. This result in a more flexible and reconfigurable network that suits a variety of IoT business and market applications.

VI. CONCLUSION

In this paper, we proposed an IoT NFV architecture framework that leverages existing key enablers technologies, namely SDR, SDN and NFV, in providing flexible and reconfigurable solutions for creating and deploying new customized on-demand virtualized IoT services. We supported the introduction of generic IoT device capable of accepting new transmission technology by (re)programming the radio functions and offering multiple isolated environments to execute multiple softwarized functions in a virtualized environment. We identified a set of relevant use cases and we realized a proof of concept implementation to demonstrate the feasibility of the proposed architecture framework. Wide acceptance of this concept will open a new era of opportunities for IoT.

ACKNOWLEDGMENT

This study has been carried out with financial support from the French State, managed by the French National Research Agency (ANR) in the frame of the "Investments for the future" Programme IdEx Bordeaux - SysNum (ANR-10-IDEX-03-02).

REFERENCES

[1] Zhu, Qian, Ruicong Wang, Qi Chen, Yan Liu, and Weijun Qin. "IoT gateway: Bridging wireless sensor networks into internet of things." In 2010 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, pp. 347-352. Ieee, 2010.

[2] Valdivieso Caraguay, Angel Leonardo, Alberto Benito Peral, Lorena Isabel Barona Lopez, and Luis Javier Garcia Villalba. "SDN: Evolution and opportunities in the development IoT applications." *International Journal of Distributed Sensor Networks* 10, no. 5 (2014): 735142.

[3] Nastic, Stefan, Sanjin Sehic, Duc-Hung Le, Hong-Linh Truong, and Schahram Dustdar. "Provisioning software-defined IoT cloud systems." In 2014 international conference on future internet of things and cloud, pp. 288-295. IEEE, 2014.

[4] Thubert, Pascal, Maria Rita Palattella, and Thomas Engel. "6TiSCH centralized scheduling: When SDN meet IoT." In *Proc. of IEEE Conf. on Standards for Communications & Networking (CSCN'15)*, 2015.

[5] Bera, Samaresh, Sudip Misra, and Athanasios V. Vasilakos. "Software-defined networking for internet of things: A survey." *IEEE Internet of Things Journal* 4, no. 6 (2017): 1994-2008.

[6] Mavromatis, Alex, Aloizio Pereira Da Silva, Koteswararao Kondepu, Dimitrios Gkounis, Reza Nejabati, and Dimitra Simeonidou. "A Software Defined Device Provisioning Framework Facilitating Scalability in Internet of Things." In 2018 IEEE 5G World Forum (5GWF), pp. 446-451. IEEE, 2018.

[7] Bizanis, Nikos, and Fernando A. Kuipers. "SDN and virtualization solutions for the Internet of Things: A survey." *IEEE Access* 4 (2016): 5591-5606.

[8] Omnes, Nathalie, Marc Bouillon, Gael Fromentoux, and Olivier Le Grand. "A programmable and virtualized network & IT infrastructure for the internet of things: How can NFV & SDN help for facing the upcoming challenges." In 2015 18th International Conference on Intelligence in Next Generation Networks, pp. 64-69. IEEE, 2015.

[9] Kouah, Riad, Abdelhamid Alleg, Abir Laraba, and Toufik Ahmed. "Energy-Aware Placement for IoT-Service Function Chain." In 2018 IEEE 23rd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), pp. 1-7. IEEE, 2018.

[10] Thomas J. Kelly, The shift to standards based hardware for military communications: What role will COTS systems play? *Industry Spotlight in Military Embedded systems* Volume 10 | Number 8 Nov / Dec 2014.

[11] Wearables Operating Systems and Platforms available at <https://www.abiresearch.com/market-research/product/1030999-wearables-operating-systems-and-platforms/>, Q3 2018.

[12] ETSI guideline for NFV reference architecture framework, V1.1.1 10, 2013.

[13] Abbasi, Ameer Ahmed, and Mohamed Younis. "A survey on clustering algorithms for wireless sensor networks." *Computer communications* 30, no. 14-15 (2007): 2826-2841.

[14] Boyinbode, Olutayo, Hanh Le, Audrey Mbogho, Makoto Takizawa, and Ravi Poliah. "A survey on clustering algorithms for wireless sensor networks." In 2010 13th International Conference on Network-Based Information Systems, pp. 358-364. IEEE, 2010.

[15] Zhao, Feng, Jie Liu, Juan Liu, Leonidas Guibas, and James Reich. "Collaborative signal and information processing: an information-directed approach." *Proceedings of the IEEE* 91, no. 8 (2003): 1199-1209.

[16] Ochiai, Hideki, Patrick Mitran, H. Vincent Poor, and Vahid Tarokh. "Collaborative beamforming for distributed wireless ad hoc sensor networks." *IEEE Transactions on Signal Processing* 53, no. 11 (2005): 4110-4124.

[17] Demigha, Oualid, Walid-Khaled Hidouci, and Toufik Ahmed. "On energy efficiency in collaborative target tracking in wireless sensor network: A review." *IEEE Communications Surveys & Tutorials* 15, no. 3 (2013): 1210-1222.

[18] "50 of the most important Raspberry Pi Sensors and Components", available at: <https://tutorials-raspberrypi.com/raspberry-pi-sensors-overview-50-important-components/>

[19] Lime Microsystems, Software defined radio technology for wireless networks. available at: <https://limemicro.com/>

[20] SoapySDR project available at: <https://github.com/pothosware/SoapySDR/wiki>

[21] CubicSDR Cross platform and open source SDR Application available at: <https://cubicsdr.com/>

[22] Gqrx SDR, Open source software defined radio by Alexandru Csete OZ9AEC, available at: <http://gqrx.dk/>

[23] GNU Radio home site available at: <https://www.gnuradio.org/>

[24] RF transmitter for Raspberry Pi available at: <https://github.com/F5OEO/rpitx>

[25] Docker home page available at: <https://www.docker.com/>