



## Article

# An Integrated Software-Defined Networking–Network Function Virtualization Architecture for 5G RAN–Multi-Access Edge Computing Slice Management in the Internet of Industrial Things

Francesco Chiti <sup>\*,†</sup> , Simone Morosi <sup>†</sup> and Claudio Bartoli <sup>†</sup>

Department of Information Engineering, University of Florence, Via di S. Marta 3, 50139 Firenze, Italy; simone.morosi@unifi.it (S.M.); claudio.bartoli@edu.unifi.it (C.B.)

\* Correspondence: francesco.chiti@unifi.it; Tel.: +39-055-2758588

<sup>†</sup> These authors contributed equally to this work.

**Abstract:** The Internet of Things (IoT), namely, the set of intelligent devices equipped with sensors and actuators and capable of connecting to the Internet, has now become an integral part of the most competitive industries, as it enables optimization of production processes and reduction in operating costs and maintenance time, together with improving the quality of products and services. More specifically, the term Industrial Internet of Things (IIoT) identifies the system which consists of advanced Internet-connected equipment and analytics platforms specialized for industrial activities, where IIoT devices range from small environmental sensors to complex industrial robots. This paper presents an integrated high-level SDN-NFV architecture enabling clusters of smart devices to interconnect and manage the exchange of data with distributed control processes and databases. In particular, it is focused on 5G RAN-MEC slice management in the IIoT context. The proposed system is emulated by means of two distinct real-time frameworks, demonstrating improvements in connectivity, energy efficiency, end-to-end latency and throughput. In addition, its scalability, modularity and flexibility are assessed, making this framework suitable to test advanced and more applications.

**Keywords:** Internet of Industrial Things; software-defined networking; network function virtualization; 5G slice management



**Citation:** Chiti, F.; Morosi, S.; Bartoli, C. An Integrated Software-Defined Networking–Network Function Virtualization Architecture for 5G RAN–Multi-Access Edge Computing Slice Management in the Internet of Industrial Things. *Computers* **2024**, *13*, 226. <https://doi.org/10.3390/computers13090226>

Academic Editor: Kannan Govindarajan

Received: 29 July 2024

Revised: 4 September 2024

Accepted: 6 September 2024

Published: 9 September 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Industrial Internet of Things (IIoT) is a recently introduced concept [1–3] which leads to the design of viable solutions for the disruptive Industry 4.0 paradigm, especially in light of the technological advancements of 5G cellular systems that are going to open a novel perspective [4,5]. With the aim to provide an abstract IIoT platform for general distributed application, the adoption of the so-called Web of Things (WoT) approach has been investigated [6]. The information exchange occurring as a WoT domain interacts with the physical world is characterized by specific features, mainly dictated by the constrained devices which suffer from limited processing power, network bandwidth and intermittent connectivity. In addition, communications are asynchronous, often event-triggered by updates as well as multiple bursty and correlating data flows.

Considering a typical industrial case study, heterogeneous, complex and time-varying QoS requirements, even within the same factory, represent an open challenge to be faced by 5G mobile networks: for example, it may be necessary to simply collect data originating from a cluster of sensors for machinery monitoring and at the same time to track the joint movements of a swarm of robots or even to control the trajectory of self-driving vehicles by interacting in real time with specific services running in a Cloud-to-Edge continuum data center [7,8].

As anticipated, since heterogeneous technologies are usually adopted to provide local connectivity to wireless sensor and actuator (WSAN) subnetworks, a widely accepted approach to enhance interoperability on a wide area basis is represented by the 5G infrastructure. This involves logical and virtual radio resource management: 5G systems, indeed, permit flexible management of resources on a service basis, allowing optimal and real-time resource allocation, with particular attention given to the whole end-to-end resource chaining, this meaning both radio, forwarding, storage and computing modules. This disrupting feature with respect to the previous 3G and 4G networks can be afforded by introducing the Software-Defined Networking (SDN) paradigm, which is able to open a new perspective in the IIoT domain. In particular, the separation of the Control from the Data Planes allows us to optimize traditional networks' operation management. Specifically, the SDN Controller achieves a global vision of the network of an IIoT domain, allowing real-time device configuration and data flow handling on an end-to-end basis [9].

This paper presents the design of a high-level network architecture for managing information flows generated by multiple clusters of sensors and actuators. Specifically, it is based on a Network Slicing approach and applied to an Industry 4.0 context where wireless subnetworks are interconnected via 5G Radio Access and Core Networks (RACNs), while data routing is managed by an SDN Controller. The proposed system is emulated by means of two distinct real-time frameworks, one for 5G networks and the other for SDN network simulation.

This paper focuses on the interconnecting subnetworks of mobile industrial things by means of a high-level network architecture for managing bidirectional and heterogeneous data flows involving multiple clusters of sensors and actuators. The proposed architecture is based on a Network Slicing approach (i.e., managing both radio and computing resources), and it is further applied to an Industry 4.0 context where wireless subnetworks are interconnected via 5G access points and data routing is managed by an SDN Controller. Moreover, an integrated framework is characterized to emulate and test the proposed system, both in terms of 5G network components and SDN procedure.

The most relevant contributions of this paper could be summarized as follows:

- The design of an original overall 5G-IIoT network architecture comprising macro/micro 5G base stations with heterogeneous sensors and actuator clusters;
- Integration with the SDN and NFV paradigms to effectively apply an RAN-MEC Network Slicing management framework;
- The development and testing of a novel and functionally complete network simulator composed of two distinct environments (i.e., Simu5G and Mininet);
- Accurate performance evaluation, addressing scalability, modularity and metrics optimization.

This paper's structure is as follows: after introducing the specific features and the open issue of the IIoT domain in Section 1, the most relevant enabling technologies are reviewed and discussed in Section 2. The SD-IIoT reference architecture is presented in Section 3, in terms of both devices and management procedure and control protocols. The overall performance is investigated in Section 4 for different case studies, pointing out a remarkable gain in terms of connectivity, energy efficiency, end-to-end latency and throughput. Finally, the conclusions are drawn in Section 5.

## 2. Overview of Enabling Technologies

### 2.1. SDN

Software-Defined Networking (SDN) is one of the most promising paradigms enabling a new generation of 5G networks [10,11]. According to this innovative approach, the control functions usually embedded in network devices are remotely decoupled from them, while only the packet forwarding needs to be executed on board, leading to the Control Plane (CP) and Data Plane (DP) separation. In this way, network devices such as routers and Switches are relieved of their burdensome management responsibilities, keeping the only task of relaying packets to pre-established destinations. Conversely, the management of the entire network is delegated to the so-called SDN Controller; this software module is requested

to harmonize the underlying infrastructure by means of appropriate APIs, also including a possible intermediate abstraction level in order to present to the higher layer a logical network view (i.e., a connectivity graph). Therefore, the SDN paradigm has been gradually introduced in IoT network management to provide the overall system with agility and flexibility in facing run-time changes in the network environment [12,13]. In addition, it has been extended towards security challenges arising at the various layers of a generic IoT network, pointing out that SDN properties, such as programmability, global visibility and manageability, can overcome the constraints of the conventional IoT networks [14].

When applied to a typical IIoT domain, the SDN Controller is able to divide the underlying infrastructure into a series of subnetworks (*slices*) based on the policies expressed by the Application/Management Plane. Each of these slices is associated with a certain context (service), and it is therefore possible to dynamically manage the resources necessary to achieve the requirements required in that particular context [15].

## 2.2. NFV

Network functions are usually implemented by appropriately combining a variety of hardware and vendor-specific software, each performing specific tasks (e.g., routing or security). Network Function Virtualization (NFV) is a reference architecture that aims, instead, to decouple network functions from the hardware on which they run. The primary objectives of NFV technology are (i) to decouple hardware and software, (ii) to flexibly deploy network functions and (iii) to reduce implementation and operation/maintenance costs.

These goals are achieved by being able to develop software components separately from hardware components; in addition, the use of commercial-off-the-shelf (COTS) hardware is encouraged, which ensures less time-to-market and, consequently, reduced capital expenditure (CapEx) and operational expenditure (OpEx). The high-level NFV architecture consists of VNF (Virtualized Network Function), NFVI (NFV Infrastructure) and NFV MANO (NFV Management and Orchestration) modules.

Virtual network functions are classic network functions, such as those ones related to routing, load balancing or mobility management, which, instead of being performed by specialized hardware and software, are executed on virtual machines housed on general-purpose and therefore low-cost hardware. These virtual machines are managed by a hypervisor, which in turn relies on an operating system that provides a level of abstraction from the physical resources that are actually used. The NFV MANO framework manages the various components of the system and provides allocation and release of virtual resources, performance monitoring, lifecycle management of virtual network functions and general supervision of the system as a whole [16].

A special application case in the IIoT domain is represented by those strategies capable of monitoring, protecting and reacting to IoT security threats, which have been investigated in [17], and it was proven to be achievable by the joint application of NFV and SDN approaches.

## 2.3. Multi-Access Edge Computing

Multi-access Edge Computing (MEC) is an emerging architectural model that aims to move the storage and computing infrastructures that support IT services from the Cloud into close proximity with RANs, so as to enable new services that need extremely low latency and a very high level of interactivity and QoE in general [18]. The MEC approach is complementary to NFV: particularly, while NFV focuses on Virtualized Network Functions (NFs), MEC enables their execution in the Edge. In addition, their on-demand orchestration requires an underlying SDN platform as discussed in [13].

The need to develop MEC-based services arises from the increasing expansion of the IoT; in fact, the presence of an ever-increasing number of connected devices generates congestion of core networks, with direct repercussions on the quality of Cloud services. The possibility of bringing these services closer to the end user strongly mitigates the drawbacks related to excessive bandwidth consumption in the core, where these resources

can be reserved for other services. The novel services enabled by the MEC meet the vision of Industry 4.0 in which every aspect of the production cycle is managed by processes in the Edge, which provide extremely faster response times and allow for real-time, remote control of machinery and devices, without requiring a human operator [19,20].

#### 2.4. 5G Verticals and Network Slicing

Next-generation networks come into being with the aim of satisfying a wide range of highly differentiated use cases usually referred to as *verticals*, for example, Vehicle-to-Everything (V2X), video surveillance, remote health monitoring, remote surgery, big data analytics and IIoT. To enable these verticals, 5G networks are designed to be easily decomposed into logical disjointed and complete subnets, that is, capable of covering the entire end-to-end service, according to what is known as the Network Slicing technique [21]. Moreover, a comprehensive analysis of the Network Slicing concept's applications for IIoT system realization is addressed in [22], highlighting that various networking demands dictated by heterogeneous IIoT applications can be matched via dedicated slices.

The possibility of resource partitioning (physical, logical, virtual) on a service basis is enabled precisely by the massive use of the previously discussed SDN and NFV technologies which allow virtual network functions to work through known interfaces (APIs) and which can be managed dynamically as demand changes (NFV-MANO). The goal is to create logical end-to-end network instances (slice instances) that provide all the services of a complete network: access, transport and service. These slice instances are created on demand and contain all the resources necessary to perform the required operations. Once their task is completed, the resources reserved for them can be released so that they become available again for other uses [23].

As mentioned, there are many verticals, and new ones are expected to emerge in the future with requirements that are difficult to predict, so the IMT-2020 Focus Group defined three macro-areas that represent all possible case studies: Enhanced Mobile Broadband (eMBB), Ultra-Reliable and Low-Latency Communications (URLLC) and Massive Machine-Type Communications (mMTC) [24].

### 3. Software-Defined IIoT Proposed Architecture Design

In this section, we propose a high-level software-defined network architecture to support QoS-constrained data flows occurring within a 5G IIoT system. We focus on a scenario consisting of a macro base station (gNB) integrated with one or more micro base stations (small gNBs) with the goal of ensuring ubiquitous and continued coverage, as represented in Figure 1. Sensor and actuator clusters embodied in the User Equipment (UE) module are interconnected by means of 5G infrastructure via specific gateways.

The different BSs are controlled by one (or more) SDN Controllers placed in the Edge (or in the Cloud) according to the needs of the different applications. They are responsible for routing packets to/from the gateways. The latter are intended to avoid forwarding messages to the base stations by adopting a D2D mode, through what is called Sidelink (SL), in order to further reduce delays, e.g., in case of emergency management following a failure. Such an infrastructure allows for local management of information flows with the ability to perform the necessary processing in the Edge in case low latency and high availability are required, such as in the case of the motion control process for the robot, as pointed out in Figure 2.

For other types of services, QoS requirements may be relaxed, so that proper management of a slice is critical for efficient use of network resources combined with high performance. If we consider the data traffic generated by sensors, we observe that this is intermittent, with low rate and large idle time intervals. As a consequence, stringent latency constraints are not necessary, while a great number of simultaneous connections need to be supported.



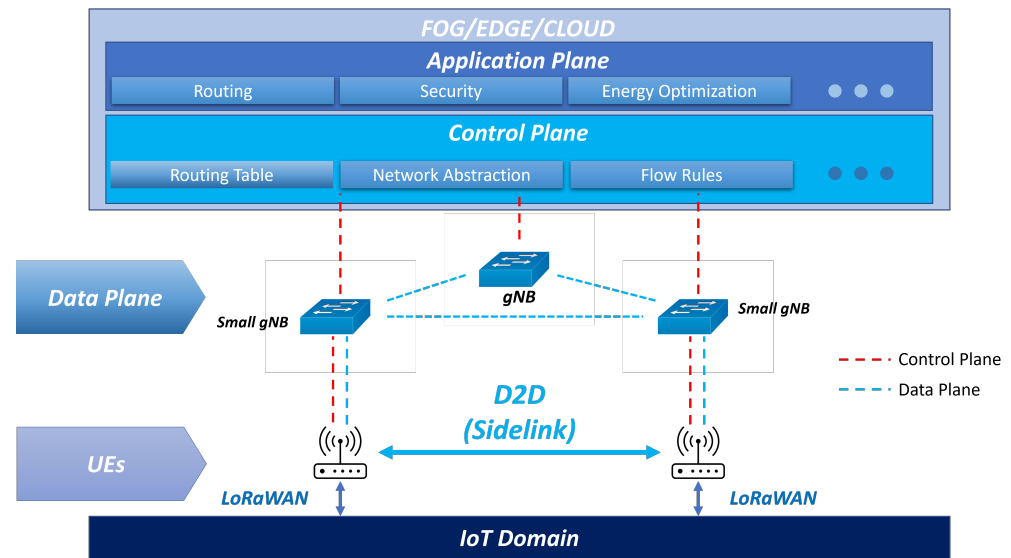


Figure 1. Proposed network architecture.

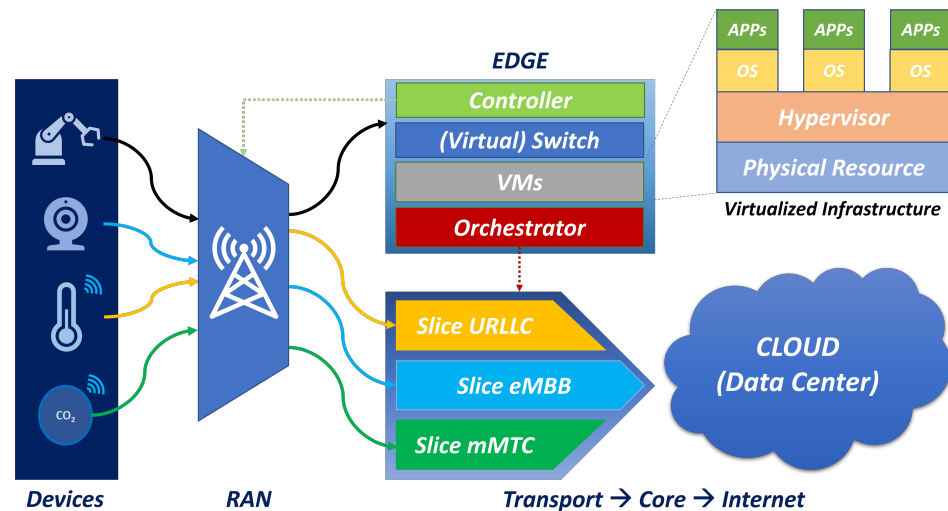


Figure 2. Radio Access Network and Edge Computing slice integration and orchestration.

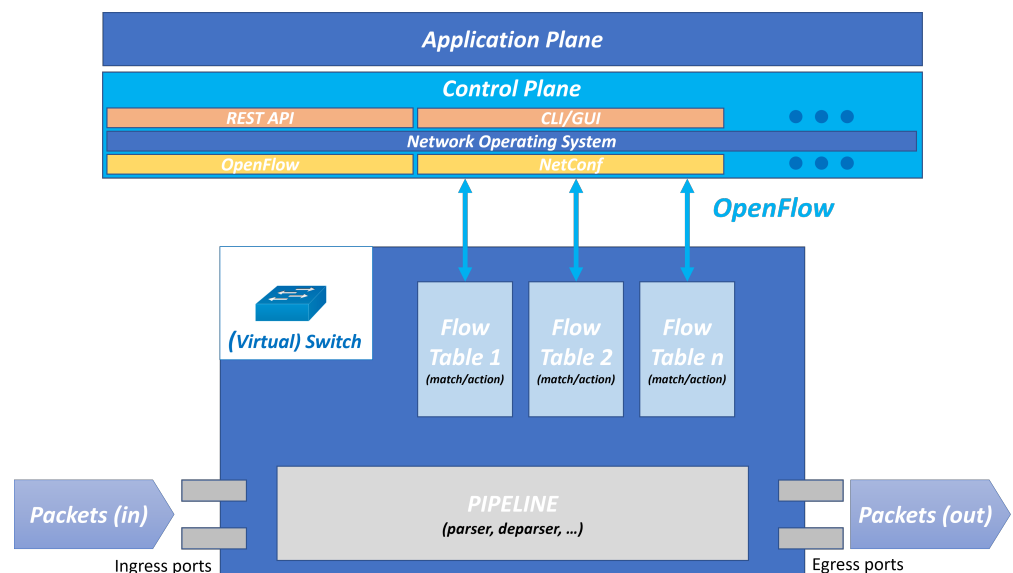
The dynamic selection and scheduling of (physical or virtual) resources is required for the creation of a complete end-to-end service, i.e., the creation of a slice, and is the responsibility of the NFV Orchestrator, which provides the SDN Controller with an abstract representation of the resources that it physically connects by determining the optimal paths according to the policies specified by the Application Plane. The involved resources range from bandwidth in the access network, which is RAN Slicing, to virtualized resources within MEC servers (storing, computing, networking) or core network resources (network functions). It is worth pointing out that in our approach a functional network slice basically consists of an RAN and core sub-slices, according to the widely adopted architecture summarized in [22]. The resource allocation problem has not been addressed in depth, since the specific QoS requirements have been managed by (i) a resource mapping into a predefined set of slices and (ii) user data flow prioritization. The SDN Control Plane has been introduced to route and control a particular data flow without affecting other slices. Moreover, by means of adopting standard northbound and southbound interfaces, the proposed system architecture is able to support interoperability among networks that are realized by different providers.

The Data Plane comprises sensor nodes interconnected by a specific wireless technology (LoRaWAN in our case) and deployed in a wide area including indoor or outdoor factories. A gateway is able to interconnect these subnetworks to the 5G Access Point which eventually connects to the Internet.

The Control Plane consists of the SDN Controller, whose typical functional block architecture is presented in Figure 3. In particular, through the appropriate interfaces and protocols (typically derived from the OpenFlow protocol), it is able to program the behavior of the devices belonging to the Data Plane, i.e., the SDN Switches, either physical or virtual.

The SDN Controller determines the paths for a packet's flow by means of configuring the flow tables of the SDN Switches with specific rules (flow rules) necessary so that they can be forwarded to the proper destinations, with the aim of chaining basic service modules in order to provide the requested service. The individual rules (flow entries) within the flow tables implement match/action operations, i.e., based on the value found within a particular packet header field, a specific operation is performed on it, such as forwarding to a specific output port or even discarding the packet. As previously explained, these (virtual) SDN Switches are responsible for forwarding packets to the proper destination, and according to the proposed reference architecture, they are required to be mapped onto the specific network device, as it is pointed out in Section 4. In particular, they could be installed both on gNB premises and on the servers where the various applications and control processes are running.

The main contribution of the proposed architecture relies on functional block modeling and the definition of standard interfaces compliant with de facto standards. As a consequence, the focus has been put on SDN Controller application to dynamic data flow forwarding, while NFV management of Network Slices has been generalized, without providing a specific characterization of NFs' orchestration and resource allocation algorithms so that, in evaluating the service level performance, the impact of the overhead is not taken into account.



**Figure 3.** (Virtual) Switch configuration and control via OpenFlow interface.

## 4. System Emulation and Performance Analysis

### 4.1. Proposed Testing Methodology

In order to evaluate the performance of the proposed system architecture described in Section 3, we introduce a system emulation environment in which services, forwarding strategies, standard interfaces and messaging have been implemented and mapped to the specific 5G network devices. To provide a more detailed and realistic system develop-

ment and testing, two distinct and specific frameworks have been integrated, which are the following:

- Simu5G: a 3GPP-compliant network simulator built by the Computer Networking Group of the University of Pisa in collaboration with Intel Corporation. It is an extension of the previous SimuLTE and is released as a library for the well-known OMNeT++ network simulator [25].
- Mininet: a network simulator that allows for creating virtual networks containing hosts, Switches and SDN Controllers and modeling the links between these devices. Operation is based on typical networking tools offered by Linux platforms (Linux Kernel) in addition to the possibility of using the OpenFlow protocol to determine the behavior of Switches (e.g., Open vSwitch) according to the SDN paradigm [26].

#### 4.1.1. Simu5G

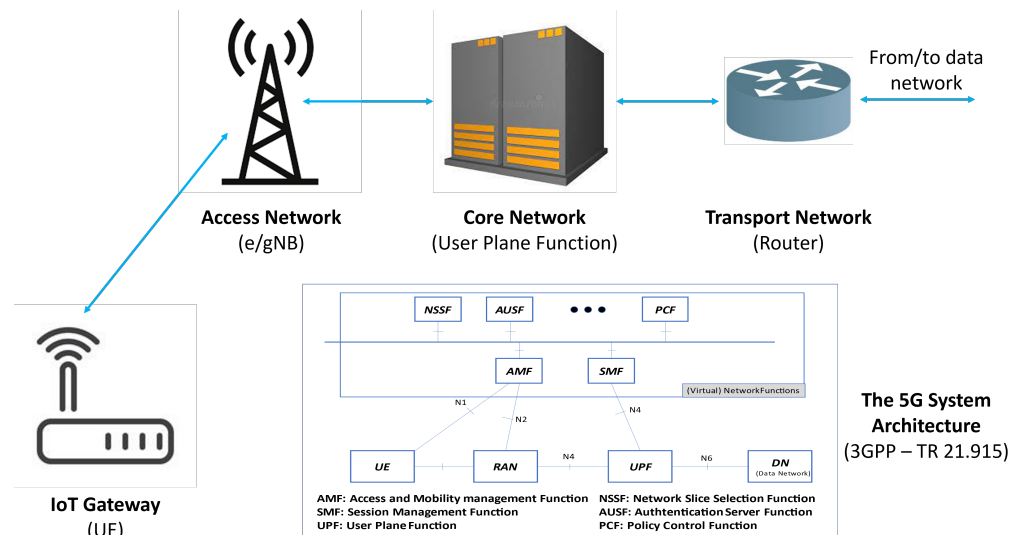
The Simu5G library is an extension of the previous SimuLTE library and takes the basic structure from it. This has allowed the developers to recover the previous work and extend it to the new context of 5G networks, which have been developed over the years in such a way as to not produce a breakaway from fourth-generation technologies because they were well aware that the transition had to be made as smooth as possible to allow users to be able to adapt to the new technologies without necessarily having to replace pre-existing infrastructure and devices. Simu5G enables the simulation of the 3GPP ENDC (E-UTRA New Radio Dual Connectivity): more specifically, Simu5G models the Data Plane of a 5G mobile network and makes it possible to assess the performance of the entire end-to-end chain, that is, from the user to the service provider. For this purpose, the entire protocol stack is modeled, also due to the integration with the INET library, from which all the typical Internet features, which can be summarized in the TCP/IP suite, are also inherited. Considering layers 1 (PHY) and 2 (MAC) of the OSI protocol stack, there are several configurable parameters within Simu5G, such as scheduling strategy (MAX C/I, Proportional Fair, Deficit Round Robin) and dynamic selection of modulation and coding schemes, to name a few. However, the complete list of capabilities comprises the following items:

- Transmit power;
- Transmission direction and antenna radiation pattern (micro- and pico-cell);
- Scheduling (MAX C/I, Proportional Fair, Deficit Round Robin);
- Modulation and coding schemes;
- Use of muMIMO (multi-user MIMO) technology, also known as mMIMO (massive MIMO);
- Dual Connectivity (ENDC);
- Channel Quality Indication (CQI);
- Interference in uplink/downlink.

In addition to this, inside Simu5G, there is a module named carrierAggregation which models the Carrier Aggregation mechanism through which radio transmissions are divided on different carriers, each with a certain number of Resource Blocks (RBs) reserved, according to what the channel access scheme named Orthogonal Frequency Division Multiple Access (OFDMA). The CarrierAggregation module maintains within it a vector of component carriers, characterized by parameters such as carrier frequency, number of RBs, Numerology Index and flag on the use of TDD or FDD duplexing techniques [27]. Moreover, it is allowed to characterize the indoor/outdoor communications scenario in a realistic way, even considering the New Radio (and/or LTE) interface. The main features present in Simu5G are Device-to-Device Communications (D2D), Dual Connectivity (ENDC), Cellular Vehicle-to-Everything (C-V2X) and Multi-access Edge Computing (MEC). A very interesting and highly useful aspect of Simu5G, inherited from OMNeT++, is the ability to perform real-time simulations for performance evaluation of a network based on the New Radio (and/or LTE) interface. This functionality is provided by the InetRealTimeScheduler module contained in the INET library which allows, under appropriate conditions, the

simulation time to be slowed down to run at the speed of real time, so an external device sees the network simulated with Simu5G exactly as if it were a real network.

Through the real-time emulation mechanism previously described, a simulation campaign has been performed in which two or more hosts external to the 5G network (e.g., Publisher, Subscriber and MQTT Broker) communicate with each other using the Simu5G internal network as the infrastructure that provides the necessary connectivity services, comparing different reference scenarios, as presented in Figure 4.



**Figure 4.** The 5G simulated network architecture compliant with 3GPP (TR 21.915).

#### 4.1.2. Mininet

Mininet is the best-known SDN-oriented network simulator and is also released in a version dedicated to Mininet-WiFi wireless networks. The secret of its success lies in its simplicity, as well as the fact that it implements code that is fully compatible with typical networking systems, meaning that it can be integrated on hardware devices to assess in a field test the results obtained from simulations.

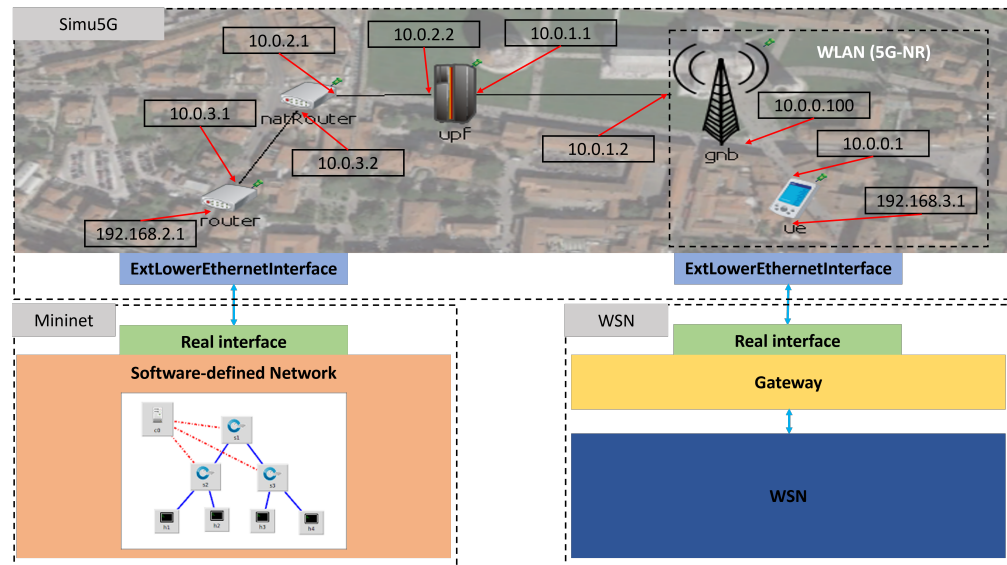
Mininet allows us to build even very complex virtual networks to easily apply the typical SDN mechanisms by means of the OpenFlow protocol.

The virtual devices are based on the Linux kernel and share all its typical features, and a CLI (Command Line Interface) is also made available through which one can interact with the network to check its link status, for example, or perform simple performance tests.

Although these features allow users to quickly become familiar with OpenFlow and SDN networks in general without further additions, Mininet provides Python APIs with which users can build their own applications, test custom topologies and more.

#### 4.1.3. Simu5G and Mininet Integration

Simu5G takes advantage of the functionality offered by the OMNeT++ Inet library, including the ability to create interfaces to outer environments. This makes it possible to connect a simulated device inside Simu5G with a real external device. With this feature, it is possible to leverage the virtual 5G network simulated in Simu5G as a real network to connect the sensor network gateway with a remote control process that can be instantiated in an SDN network that in turn is connected to the 5G network, as presented in Figure 5, where it points out a virtual machine (VM), in which runs the real-time emulated 5G network (Simu5G) including the virtual network devices with specific addresses, a second VM, where the SDN Controller is activated (Mininet) and an interface to a real (or even virtualized) Gateway process.



**Figure 5.** Proposed and developed overall emulation architecture integrating Simu5G (5G RAN), Mininet (Cloud/Fog/Edge) and wide area wireless sensor networks.

#### 4.2. Results Analysis

In the following, different case studies are tested by means of the proposed integrated emulation framework. In particular, the focus here is on a couple of opposite applications devoted to (i) background sensed data collection and (ii) mobile robot management and control.

The network resources required to manage an autonomous robot are different from those required for typical sensors and actuators, where a publish/subscribe communications protocol, such as MQTT, has been consequently adopted, since they are asynchronously activated to transmit or request small amounts of data without particular latency or reliability constraints. A different case is represented by an industrial robot, where there is a need for very accurate control of its movements in terms of both the correct execution of the machining and safety aspects. Under these circumstances, a persistent connection is required to ensure the proper operation of the closed-loop control process, i.e., characterized by high availability and reliability, as well as latency times, i.e., round-trip time, as low as possible since this is a time-sensitive control process.

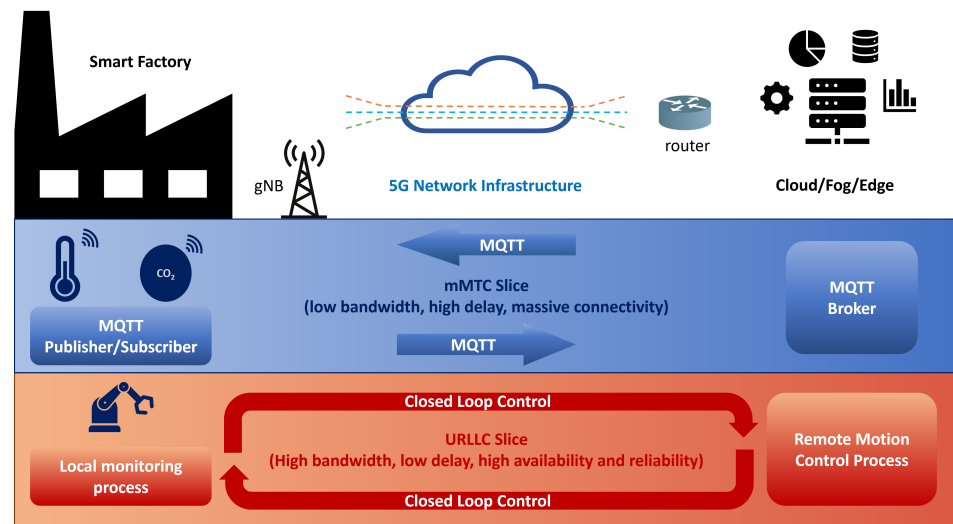
As a consequence, we adopted a service-based architecture in which different devices are assigned specific network resources to meet the requirements of individual applications, as depicted in Figure 6. For example, the gateways responsible for interconnecting with the MQTT Broker require lower resources in terms of bandwidth and less stringent latency constraints compared with the mobile robot needs. In addition, the mobile robot presents a reduced mobility range, while the gateways, even in the case where they are fixed, may be distributed over much larger areas with the need for sufficient coverage to be ensured by the involved gNBs.

Within Simu5G, it is possible to implement Network Slicing at the radio interface level (RAN Slicing) by exploiting the Carrier Aggregation mechanism and the structure of the module of the same name. As previously introduced, the Simu5G carrierAggregation module is, indeed, responsible to maintain a vector of componentCarriers for each UE to support its data flows, selected from the ones available at the corresponding gNB, which includes the following parameters:

- Carrier frequency;
- Number of resource blocks;
- TDD/FDD flag;
- Slot format (if TDD flag is asserted);
- Numerology Index.

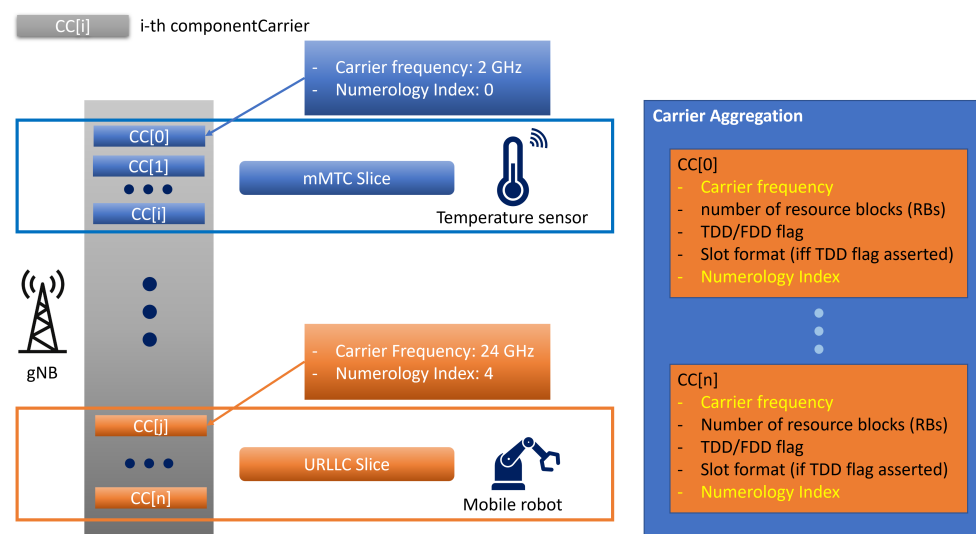


Finally, a channelModel is instantiated for each componentCarrier to evaluate the communication quality.



**Figure 6.** The 5G-IIoT service-based adopted architecture with a possible example of network deployment via Network Slicing.

In Figure 7, Network Slicing management at the RAN level, using the Carrier Aggregation mechanism previously described and making use of the componentCarrier, is sketched. In particular, the mMTC slice is assigned a carrier frequency equal to 2 GHz to achieve wide coverage and a Numerology Index equal to 0 since no special latency constraints are required, while the URLLC slice is assigned a carrier frequency belonging to the mmWave band and a Numerology Index equal to 4 to ensure very low latency and high reliability and availability to guarantee the proper operation of the control loop. In the latter case, it is worth mentioning that adopting higher frequencies, i.e., the mmWave band, requires close proximity to the terminal, in this case, the mobile robot, and the gNB to avoid fading and path loss effects.



**Figure 7.** Network Slicing management scheme applied at the RAN level by exploiting the Carrier Aggregation module provided in Simu5G.

The mMTC slice is reserved for the typical traffic of IoT systems, which are characterised by a massive number of simultaneous connections requiring limited resources

in terms of both bandwidth and reliability and latency. Typically, the frequency bands reserved for these types of communications are located at the lower end of the sub 6 GHz frequencies (Frequency Range 1), where bandwidths are limited but coverage over long distances is guaranteed.

The URLLC slice is, instead, reserved for traffic related to mission-critical applications, i.e., all those transmissions characterised by stringent constraints on reliability and latency, as they are crucial for delivering high-priority services, for example, to ensure safety in a workplace such as a factory where remotely controlled machinery and robots are present. In this case, the mmWave band (Frequency Range 2) can then be employed to exploit wider bandwidth, while coverage is reduced due to the greater impact of fading, leading to cells with an effective radius of a few tens of meters.

#### 4.2.1. MQTT System Set-Up

In this preliminary simulation, two clients and a Mosquitto MQTT Broker [28] are developed to verify the possibility of connecting a network of sensors/actuators, represented by a gateway that acts as Publisher/Subscriber to a remote Broker. To run this simulation, at least two distinct virtual machines (VMs) are needed:

- One VM (VM1) in which runs the real-time emulated 5G network (Simu5G), an MQTT Publisher and an MQTT Subscriber (Mosquitto Clients).
- A second VM (VM2) in which runs the MQTT Broker (Mosquitto Broker).

It is necessary, first of all, to select a network card with a bridge in the VM1 network settings and to activate a second network card as a host only. We select only one network card for the second VM. We open a terminal in VM1 and issue the following commands:

```
$ sudo ip link add veth0 type veth peer name veth1
$ sudo ip link add veth2 type veth peer name veth3
$ sudo ip addr add 192.168.2.2 dev veth1
$ sudo ip addr add 192.168.3.2 dev veth3
$ sudo ip link set veth0 up
$ sudo ip link set veth1 up
$ sudo ip link set veth2 up
$ sudo ip link set veth3 up
$ sudo route add -net 192.168.3.0 netmask 255.255.255.0 dev veth3
$ sudo route add -net 192.168.2.0 netmask 255.255.255.0 dev veth1
$ sudo route add -net 10.0.3.0 netmask 255.255.255.0 dev veth1
$ sudo route add -net 10.0.2.0 netmask 255.255.255.0 dev veth3
```

The other ends of the two virtual Ethernet pairs interfaces are assigned, respectively, to the router and to the UE inside the Simu5G network by indicating it in the configuration file as follows:

```
*.router.numEthernetInterfaces = 1
*.router.eth[0].typename = "ExtLowerEthernetInterface"
*.router.eth[0].device = "veth0"
*.ue.numEthernetInterfaces = 1
*.ue.eth[0].typename = "ExtLowerEthernetInterface"
*.ue.eth[0].device = "veth2"
*.ue.extHostAddress = "192.168.3.2"
*.ue.ipv4.forwarding = true
```

It is also necessary to set the NAT of the virtual machine in order to send the packets to the right destination and to make them able to come back even if the Broker's machine cannot be physically accessed to change the routing paths. To this end, the following commands must be issued in a terminal of VM1, as described in [29]:

```
$ sudo iptables -t nat -A POSTROUTING -d "Broker IP" -o enp0s8 -j MASQUERADE
$ sudo iptables-save
$ sudo iptables -L
```

The device called *natRouter* has the task of steering the passage of data through Simu5G when the client and server are active on the same machine. Basically, the IP address of one of the two interfaces of the *natRouter* is set as the destination IP address, so the packets will be securely sent within Simu5G. The *natRouter* performs the translation of the IP address by modifying both destination and source address: in place of the first, it puts the IP address one intends to reach, i.e., that of the Broker, while in place of the second, it puts the IP address of its other interface. The same action is performed in the other flow direction. This behaviour must be suitably specified in the configuration file (.ini).

Now, it is possible to activate the MQTT Broker in VM2:

```
$ sudo systemctl start mosquitto
```

The last step consists in starting two MQTT clients representing, respectively, the Publisher and the Subscriber, using two separate terminals of VM1, indicating the address of the *natRouter* 10.0.2.1 as the Broker address that replaces the the destination address, as described in Figure 8.

```

Publisher
simu5g@simu5g:~$ mosquitto_pub -h 10.0.2.1 -A 192.168.3.2 -t /factory/indoor/kiln -m "Kiln Temperature: 700°C"
simu5g@simu5g:~$ mosquitto_pub -h 10.0.2.1 -A 192.168.3.2 -t /factory/indoor/kiln -m "Kiln Temperature: 750°C"
simu5g@simu5g:~$ mosquitto_pub -h 10.0.2.1 -A 192.168.3.2 -t /factory/indoor/kiln -m "Kiln Temperature: 800°C"
simu5g@simu5g:~$ mosquitto_pub -h 10.0.2.1 -A 192.168.3.2 -t /factory/indoor/kiln -m "Kiln Temperature: 850°C"
simu5g@simu5g:~$ mosquitto_pub -h 10.0.2.1 -A 192.168.3.2 -t /factory/indoor/kiln -m "Kiln Temperature: 900°C"
simu5g@simu5g:~$ mosquitto_pub -h 10.0.2.1 -A 192.168.3.2 -t /factory/indoor/kiln -m "Kiln Temperature: 950°C"
simu5g@simu5g:~$ mosquitto_pub -h 10.0.2.1 -A 192.168.3.2 -t /factory/indoor/kiln -m "Kiln Temperature: 1000°C"
simu5g@simu5g:~$

Subscriber
simu5g@simu5g:~$ mosquitto_sub -h 10.0.2.1 -t /factory/indoor/kiln
Kiln Temperature: 700°C
Kiln Temperature: 750°C
Kiln Temperature: 800°C
Kiln Temperature: 850°C
Kiln Temperature: 900°C
Kiln Temperature: 950°C
Kiln Temperature: 1000°C

```

**Figure 8.** In the upper half of the image, a Client MQTT (Publisher) publishes values related to the topic “Kiln Temperature”, while in the lower half, another Client MQTT (Subscriber) subscribes to the same topic and receives the temperature values.

It should be noted that the argument of -A represents the interface through which traffic is sent to Simu5G, i.e., to the simulated 5G network, that of -t is the transmission topic and that of -m is the message one wants to transmit (payload).

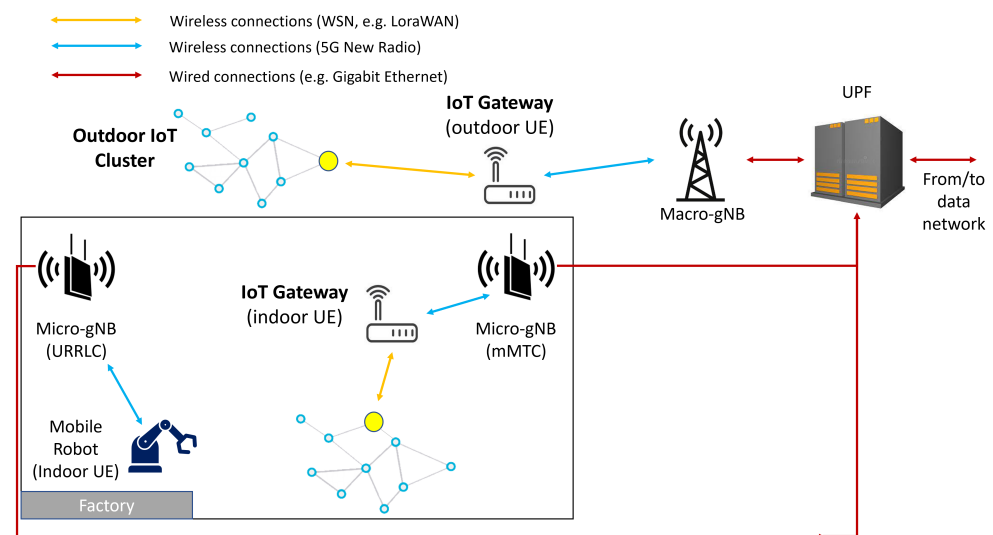
#### 4.2.2. Energy Efficiency

Regarding the power consumption, simulations are performed on two different scenarios. The first scenario, typical of new 5G architectures, comprises two small gNBs positioned internally at the plant in addition to an external gNB with wide coverage. Another reference scenario is characterized by a single traditional gNB positioned outside the plant in an elevated area and responsible for the management of a large cell.

With reference to Figure 9, the following elements are introduced and considered:

- IoT Cluster Indoor/Outdoor: a cluster of devices located inside/outside the facility (e.g., temperature sensors) connected to its afferent IoT gateway via some radio technology for sensor networks (e.g., LoraWAN);

- IoT Gateway (Indoor UE): a device that translates the protocol employed within its cluster (indoor) to the MQTT protocol for connection to the remote Broker;
- IoT Gateway (Outdoor UE): a device that translates the protocol employed within its own cluster (outdoor) into the MQTT protocol for connection to the remote Broker;
- Mobile robot (Indoor UE): a particular UE capable of moving inside the structure and with stringent requirements in terms of drive reliability;
- Macro-gNB: it represents the main base station in the area where the facility is located and is characterized by high coverage;
- Micro-gNB (URLLC): a base station located internally at the facility and working at higher frequencies to provide very wide bands with a reduced coverage;
- Micro-gNB (mMTC): a base station located internally within the facility and working at higher frequencies than macro-gNB to provide wider bands, but not enough to provide complete coverage within the entire facility;
- UPF: it represents the 5G core network and provides the necessary connectivity between the access network and data network.

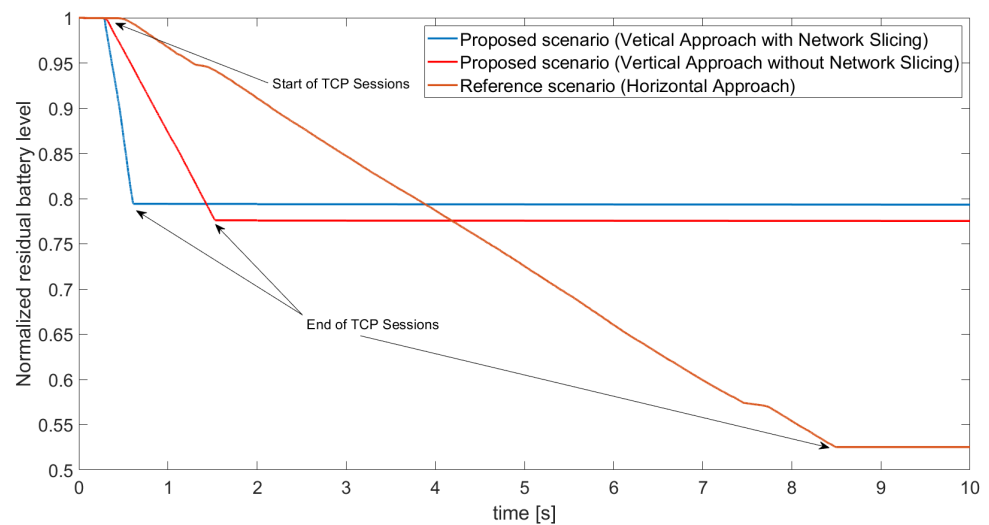


**Figure 9.** The 5G-IIoT deployment scenarios for performance evaluation, where multiple base stations (gNBs) are considered in order to facilitate RAN Slicing.

In order to evaluate the energy consumption, 10 MB of TCP traffic is sent by the mobile robot (Indoor UE) to a server in the presence of background burst traffic typical of IoT systems generated by the two IoT Gateways. In addition, to evaluate the specific performance achieved in the two scenarios described, two different approaches are compared:

- A horizontal approach typical of 3G and 4G networks, i.e., in which the same resources are divided among the various UEs regardless of the applications in which they are involved;
- A vertical approach that characterizes 5G networks and in which resources are allocated to UEs according to the service required.

As can be seen from Figure 10, energy consumption remarkably decreases when two small gNBs are used, with a performance gain of approximately 30%, as the efficiency of communications benefits from the proximity of the two terminals, since the signal-to-noise ratio increases and, consequently, the need to perform re-transmissions decreases. The consumption in the horizontal and vertical approach cases is comparable, but the traffic generated is transmitted with a higher rate in the second case: this is due to the wider bandwidth available. This positive aspect is clearly visible in the subsequent end-to-end delay analysis.

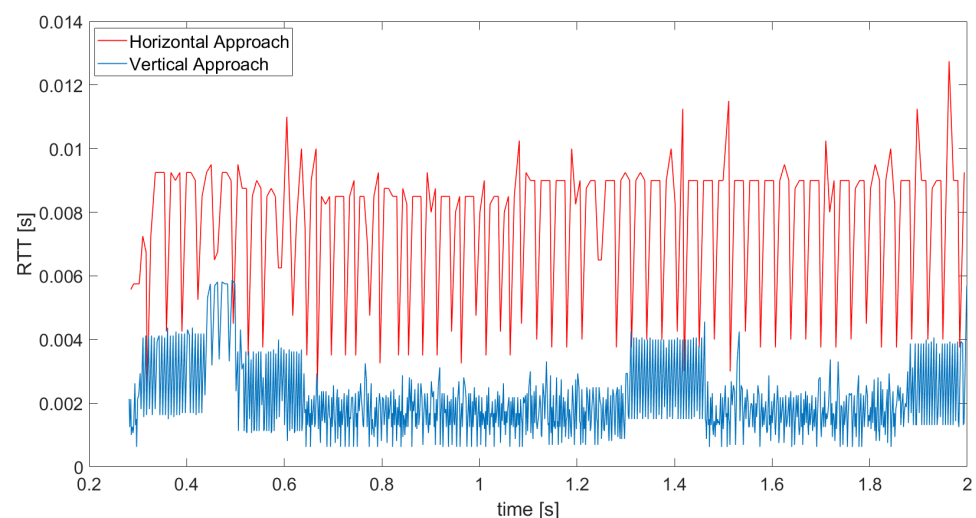


**Figure 10.** Normalized residual battery level of the mobile robot as a function of time.

#### 4.2.3. Round-Trip Time

With reference to the Round-Trip Time experienced by the mobile robot operating in the previously described scenarios, the following results are obtained, where we omit to report the case of a single gNB as delays are extremely higher than those ones achieved in the 5G-IIoT scenario (approximately by an order of magnitude).

In this case, the vertical approach assigns more bandwidth to the mobile robot by exploiting the mmWave band. In addition, it is possible to increase the Numerology Index  $\mu$ , which means reducing the duration of time slots in the OFDM plot, i.e., increasing the symbol-rate, as reported in Table 1. This increase in symbol-rate means an increase in the available bandwidth, which is why higher Numerology Indices are typically employed in the mmWave band. As a result, as pointed out in Figure 11, the latency decreases, while maintaining consumption in line with those ones obtained with a horizontal approach. Specifically, by performing a statistical analysis, the mean values are equal to 7.75 ms and 2.1 ms for the horizontal and vertical approaches, respectively, with a gain of approximately 70%. Obviously, this strategy offers good results when UE and gNB are close enough, since Frequency Range 2 frequencies are more affected by fading issues, as well as experiencing higher path loss.



**Figure 11.** Round-Trip Time comparison for the cases of vertical approach (Slicing) and horizontal approach (one size fits all).



**Table 1.** Slot duration with respect to the Numerology Index  $\mu$ .

| $\mu$              | 0 | 1   | 2    | 3     | 4      |
|--------------------|---|-----|------|-------|--------|
| Slot duration [ms] | 1 | 0.5 | 0.25 | 0.125 | 0.0625 |

#### 4.2.4. Qualitative Issues

As far as interoperability is concerned, the proposed architecture could be used for any kind of scenario given its generality and the possibility of being connected with any networking system, be it a private network or even the Internet. Expandability is guaranteed together with the capability to develop novel applications and, thus, simulate more complex scenarios. An example in this sense is the possibility of introducing specific (virtual) network functions, e.g., to place and enforce specific security policies. Some network functions required for certain core services such as authenticating a user or managing security can be modeled directly within the Simu5G access network, as well as for those typically hosted in the core network, such as firewall, NAT, deep-packet inspection (DPI), charging and many others. In addition, the presence of an SDN control network allows for the development of specific network functions that can be interconnected to realize complex services involving an SDN Controller to optimize the routing of packets.

### 5. Conclusions and Future Developments

This paper focuses on interconnecting subnetworks of mobile IIoT devices by a high-level network architecture for managing data flows generated by multiple clusters of sensors and actuators. The proposed framework is based on a Network Slicing approach (i.e., managing both radio and computing resources), and it is further applied to an Industry 4.0 context where 5G technology is adopted to improve interoperability. Specifically, the 5G-IIoT network architecture, comprising macro/micro 5G base stations with heterogeneous sensor and actuator clusters, is further integrated with the SDN and NFV paradigms to effectively apply a general RAN-MEC Network Slicing management scheme. Finally, the proposed system is emulated by means of integrating two distinct real-time frameworks, one for 5G networks and the other for SDN networks. The obtained results demonstrate that the proposed approach allows us to allocate the available resources more efficiently and afford improved performance in terms of connectivity, energy efficiency, end-to-end latency and throughput. In addition, its scalability, modularity and flexibility are assessed, making it a general tool to test novel applications and more complex and real-world IIoT deployment scenarios.

We want to highlight that several issues still need be addressed to improve the relevance of the proposed framework. First of all, an NF orchestration and resource allocation algorithm could be integrated to investigate the trade-off between latency and control overhead and to support multi-tenant scenarios. Additionally, more complex and dynamic applications and services are to be modelled and evaluated: this is the case of distributed machine learning, blockchain of things and, above all, on-demand security policies' composition, set-up and enforcement.

**Author Contributions:** Conceptualization, F.C. and S.M.; methodology, F.C. and C.B.; software, F.C. and C.B.; validation, F.C., S.M. and C.B.; investigation, F.C. and S.M.; writing—original draft preparation, F.C. and S.M.; writing—review and editing, F.C., S.M. and C.B.; supervision, F.C.; project administration, F.C.; funding acquisition, F.C. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was partially supported by the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, in partnership with “Telecommunications of the Future” (PE00000001—program “RESTART”), and by the Regione Toscana under the POR FESR 2014/2020, in partnership with the “RI.B.AT Riciclo integrato Batterie AutoTrazione” project.

**Data Availability Statement:** Data is contained within the article.

**Acknowledgments:** This work was partially supported by the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, in partnership with “Telecommunications of the Future” (PE00000001—program “RESTART”), and by the Regione Toscana under the POR FESR 2014/2020, in partnership with the “RI.B.AT Riciclo integrato Batterie AutoTrazione” project.

**Conflicts of Interest:** The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

## References

1. Wan, Z.; Gao, Z.; Di Renzo, M.; Hanzo, L. The Road to Industry 4.0 and Beyond: A Communications-, Information-, and Operation Technology Collaboration Perspective. *IEEE Network* **2022**, *36*, 157–164. [\[CrossRef\]](#)
2. Wollschlaeger, M.; Sauter, T.; Jasperneite, J. The Future of Industrial Communication: Automation Networks in the Era of the Internet of Things and Industry 4.0. *IEEE Ind. Electron. Mag.* **2017**, *11*, 17–27. [\[CrossRef\]](#)
3. Weyrich, M.; Ebert, C. Reference Architectures for the Internet of Things. *IEEE Softw.* **2016**, *33*, 112–116. [\[CrossRef\]](#)
4. Zhong, M.; Yang, Y.; Yao, H.; Fu, X.; Dobre, O.A.; Postolache, O. 5G and IoT: Towards a new era of communications and measurements. *IEEE Instrum. Meas. Mag.* **2019**, *22*, 18–26. [\[CrossRef\]](#)
5. Seres, G.; Schulz, D.; Dobrijevic, O.; Karaagaç, A.; Przybysz, H.; Nazari, A.; Chen, P.; Mikecz, M.L.; Szabó, Á.D. Creating programmable 5G systems for the Industrial IoT. *Ericsson Technol. Rev.* **2022**, *2022*, 2–12. [\[CrossRef\]](#)
6. Bartoli, C.; Bonanni, M.; Chiti, F.; Pierucci, L.; Cidronali, A.; Collodi, G.; Maddio, S. Toward the Web of Industrial Things: A Publish-Subscribe Oriented Architecture for Data and Power Management. *Sensors* **2022**, *22*, 4882. [\[CrossRef\]](#)
7. Borsatti, D.; Davoli, G.; Cerroni, W.; Raffaelli, C. Enabling Industrial IoT as a Service with Multi-Access Edge Computing. *IEEE Commun. Mag.* **2021**, *59*, 21–27. [\[CrossRef\]](#)
8. Bonadio, A.; Chiti, F.; Fantacci, R. Performance Analysis of an Edge Computing SaaS System for Mobile Users. *IEEE Trans. Veh. Technol.* **2020**, *69*, 2049–2057. [\[CrossRef\]](#)
9. Bonanni, M.; Chiti, F.; Fantacci, R.; Pierucci, L. Dynamic Control Architecture Based on Software Defined Networking for the Internet of Things. *Future Internet* **2021**, *13*, 113. [\[CrossRef\]](#)
10. Ameigeiras, P.; Ramos-munoz, J.J.; Schumacher, L.; Prados-Garzon, J.; Navarro-Ortiz, J.; Lopez-soler, J.M. Link-level access cloud architecture design based on SDN for 5G networks. *IEEE Network* **2015**, *29*, 24–31. [\[CrossRef\]](#)
11. Babbar, H.; Rani, S.; AlZubi, A.A.; Singh, A.; Nasser, N.; Ali, A. Role of Network Slicing in Software Defined Networking for 5G: Use Cases and Future Directions. *IEEE Wirel. Commun.* **2022**, *29*, 112–118. [\[CrossRef\]](#)
12. Siddiqui, S.; Hameed, S.; Shah, S.A.; Ahmad, I.; Aneiba, A.; Draheim, D.; Dustdar, S. Toward Software-Defined Networking-Based IoT Frameworks: A Systematic Literature Review, Taxonomy, Open Challenges and Prospects. *IEEE Access* **2022**, *10*, 70850–70901. [\[CrossRef\]](#)
13. Rafique, W.; Qi, L.; Yaqoob, I.; Imran, M.; Rasool, R.U.; Dou, W. Complementing IoT Services Through Software Defined Networking and Edge Computing: A Comprehensive Survey. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 1761–1804. [\[CrossRef\]](#)
14. Iqbal, W.; Abbas, H.; Daneshmand, M.; Rauf, B.; Bangash, Y.A. An In-Depth Analysis of IoT Security Requirements, Challenges, and Their Countermeasures via Software-Defined Security. *IEEE Internet Things J.* **2020**, *7*, 10250–10276. [\[CrossRef\]](#)
15. Ordonez-Lucena, J.; Ameigeiras, P.; Lopez, D.; Ramos-Munoz, J.J.; Lorca, J.; Folgueira, J. Network Slicing for 5G with SDN/NFV: Concepts, Architectures, and Challenges. *IEEE Commun. Mag.* **2017**, *55*, 80–87. [\[CrossRef\]](#)
16. Mijumbi, R.; Serrat, J.; Gorricho, J.I.; Latre, S.; Charalambides, M.; Lopez, D. Management and orchestration challenges in network functions virtualization. *IEEE Commun. Mag.* **2016**, *54*, 98–105. [\[CrossRef\]](#)
17. Farris, I.; Taleb, T.; Khettab, Y.; Song, J. A Survey on Emerging SDN and NFV Security Mechanisms for IoT Systems. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 812–837. [\[CrossRef\]](#)
18. Tran, T.X.; Hajisami, A.; Pandey, P.; Pompili, D. Collaborative Mobile Edge Computing in 5G Networks: New Paradigms, Scenarios, and Challenges. *IEEE Commun. Mag.* **2017**, *55*, 54–61. [\[CrossRef\]](#)
19. Sabella, D.; Vaillant, A.; Kuure, P.; Rauschenbach, U.; Giust, F. Mobile-Edge Computing Architecture: The role of MEC in the Internet of Things. *IEEE Consum. Electron. Mag.* **2016**, *5*, 84–91. [\[CrossRef\]](#)
20. Corcoran, P.; Datta, S.K. Mobile-Edge Computing and the Internet of Things for Consumers: Extending cloud computing and services to the edge of the network. *IEEE Consum. Electron. Mag.* **2016**, *5*, 73–74. [\[CrossRef\]](#)
21. Li, X.; Samaka, M.; Chan, H.A.; Bhamare, D.; Gupta, L.; Guo, C.; Jain, R. Network Slicing for 5G: Challenges and Opportunities. *IEEE Internet Comput.* **2017**, *21*, 20–27. [\[CrossRef\]](#)
22. Wijethilaka, S.; Liyanage, M. Survey on Network Slicing for Internet of Things Realization in 5G Networks. *IEEE Commun. Surv. Tutor.* **2021**, *23*, 957–994. [\[CrossRef\]](#)
23. Belikaidis, I.P.; Georgakopoulos, A.; Tsagkaris, K.; Altman, Z.; Ben Jemaa, S.; Michaloliakos, M.; Demestichas, P.; Mitrou, N. 5G Radio Access Network Slicing: System-Level Evaluation and Management. *IEEE Veh. Technol. Mag.* **2019**, *14*, 49–55. [\[CrossRef\]](#)
24. Guo, S.; Lu, B.; Wen, M.; Dang, S.; Saeed, N. Customized 5G and Beyond Private Networks with Integrated URLLC, eMBB, mMTC, and Positioning for Industrial Verticals. *IEEE Commun. Stand. Mag.* **2022**, *6*, 52–57. [\[CrossRef\]](#)

25. Nardini, G.; Sabella, D.; Stea, G.; Thakkar, P.; Viridis, A. Simu5G—An OMNeT++ Library for End-to-End Performance Evaluation of 5G Networks. *IEEE Access* **2020**, *8*, 181176–181191. [[CrossRef](#)]
26. Muelas, D.; Ramos, J.; Vergara, J.E.L.d. Assessing the Limits of Mininet-Based Environments for Network Experimentation. *IEEE Network* **2018**, *32*, 168–176. [[CrossRef](#)]
27. Islam Farooqui, M.N.; Mubashir Khan, M.; Farooq, F.; Arshad, J. A Comparative Analysis of 5G Network Slicing Simulators. In Proceedings of the 2022 3rd International Conference on Innovations in Computer Science and Software Engineering (ICONICS), Karachi, Pakistan, 14–15 December 2022; pp. 1–8.
28. Eclipse Mosquitto: An Open Source MQTT Broker. Available online: <https://github.com/eclipse/mosquitto> (accessed on 14 August 2024).
29. Viridis, A.; Nardini, G. Setting up an Emulation with Simu5G and OpenNESS. Available online: [https://simu5g.org/users-guide/emulation\\_openness.html](https://simu5g.org/users-guide/emulation_openness.html) (accessed on 14 August 2024).

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.