

Article

Development of Intelligent Core Network for Tactile Internet and Future Smart Systems

Abdelhamied A. Ateya ^{1,2} , Ammar Muthanna ^{1,*} , Irina Gudkova ³,
Abdelrahman Abuarqoub ⁴ , Anastasia Vybornova ¹ and Andrey Koucheryavy ¹

¹ Telecommunication Networks and Data Transmission, The Bonch-Bruевич State University of Telecommunication, 193232 Sankt-Peterburg, Russia; a_ashraf@zu.edu.eg (A.A.A.); a.vybornova@gmail.com (A.V.); akouch@mail.ru (A.K.)

² Electronics and Communications Engineering, Zagazig University, Ash Sharqia Governorate 44519, Egypt

³ Applied Probability and Informatics, Peoples' Friendship University of Russia (RUDN University), 117198 Moskva, Russia; gudkova_ia@pfur.ru

⁴ Faculty of Information Technology, Middle East University Amman, 383 Amman 11831, Jordan; Aabuarqoub@meu.edu.jo

* Correspondence: ammarexpress@gmail.com; Tel.: +7-952-210-4486

Received: 31 October 2017; Accepted: 11 December 2017; Published: 2 January 2018



Abstract: One of the main design aspects of the Tactile Internet system is the 1 ms end-to-end latency, which is considered as being the main challenge with the system realization. Forced by recent development and capabilities of the fifth generation (5G) cellular system, the Tactile Internet will become a real. One way to overcome the 1 ms latency is to employ a centralized controller in the core of the network with a global knowledge of the system, together with the concept of network function virtualization (NFV). This is the idea behind the software defined networking (SDN). This paper introduces a Tactile Internet system structure, which employs SDN in the core of the cellular network and mobile edge computing (MEC) in multi-levels. The work is mainly concerned with the structure of the core network. The system is simulated over a reliable environment and introduces a round trip latency of orders of 1 ms. This can be interpreted by the reduction of intermediate nodes that are involved in the communication process.

Keywords: Tactile Internet; 5G; latency; NFV; SDN; MEC

1. Introduction

Tactile Internet is expected to be a novel approach in the human-to-machine (H2M) communication by moving from the content delivery to the skill-set delivery. The main application that is supported by the Tactile Internet system will be the haptic communication in real time [1]. It will be a revolution in the area of information and communication technology with enormous applications in many fields [2]. Powered by the 5G cellular network, Tactile Internet will provide a way for human to transfer their touch and actuation in real time form.

Tactile Internet is one of the main use cases of the near future 5G cellular system, as announced by International Telecommunication Union (ITU) [3]. Through the Tactile Internet, physical habits will be communicated remotely. The construction of the Tactile Internet system can be viewed as three main parts as presented in Figure 1 [4]. The network part connects the master and the slave parts through a huge infrastructure of fiber cables and network elements. Here, we consider this part to enable live time communication between transmitter and receiver by means of employing modern technologies throughout the cellular network.

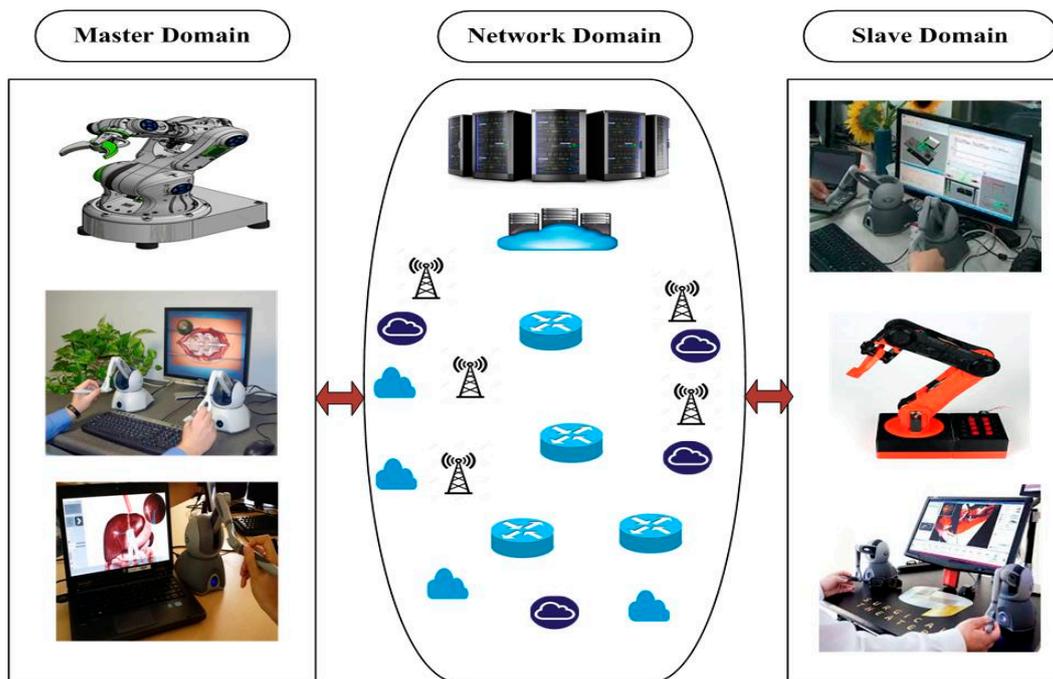


Figure 1. General view of Tactile Internet System.

Designing Tactile Internet system and realizing it meets some challenges that are presented in [5]. The main challenge is the 1 ms end-to-end latency. The end-to-end latency differ from the user plane latency and can be defined as the time duration starts from the transmission of a small data packet from the transmitter’s application layer and ends by the reception of the data by the receiver’s application layer, including the response feedback that is dedicated by the communication process [6]. Thus, the end-to-end latency depends on the number of network nodes that are involved in the communication process. In order to reduce the round trip latency and achieve the 1 ms latency requirement for the Tactile Internet system, the number of network nodes involved in the communication process should be reduced and bring them as near as possible to the user equipment. This is can be achieved by employing software defined networking (SDN), network function virtualization (NFV), and mobile edge computing (MEC) with the 5G cellular system.

SDN, NFV, and MEC technologies provide powerful solutions for the challenges that are associated with the design of 5G cellular system and Tactile Internet system, specially the 1 ms latency. Thus, the academia and the industry pay a great attention to the research and development in these areas. In today’s cellular networks, the network services are handled by means of various network functions that are connected in static manner [7]. Thus, it is not easy to add new services, this is because the cost, energy and integration difficulties of hardware required. To overcome these problems NFV and SDN are developed.

NFV provides fixed network functions by means of software run on a virtualized environment, which increases the flexibility of the overall system. NFV employs virtualization technologies on programmable hardware, such as storage devices, general purpose servers, and switches to separate the software implementation of network functions from the dedicated hardware. In another word, NFV set up the network functions by means of software virtualization techniques on commodity hardware [8]. Several projects are launched, for developing open source, commercial solutions, and standardizations for NFV. The most important projects are the Service Programming and Orchestration for Virtualized Software Networks (SONATA) project and Management and Orchestration (MANO) project. SONATA is constructed with a vision of increasing the programmability and flexibility of 5G cellular system. It aims to make the service platforms easier and modular to be suitable for the requirements of the

different service providers. Also, it stands with the network function orchestration and provides a development model for the developers [9]. MANO is an open source project that was developed by European Telecommunications Standards Institute (ETSI) [10].

SDN is a new approach that provides a dynamic and cost effective network structure by the physical separation of data forwarding plane and control plane. The control plane is the network part that gives the appropriate decision to handle traffic [11–13]. The data plane is the other part of the network, which forward traffic in respond to the control plane. The control signal moves from the control plane to the data plane by means of an open standard interface protocol, such as OpenFlow protocol and ForCES protocol [14]. SDN allows for the network operator to configure, control, and manage the network through customized software known as application programming interfaces (APIs). SDN is mainly presented to simplify the network hardware and increase the network flexibility [15].

Dawn of the software networks is supported by the recent development in NFV and SDN technologies. NFV and SDN technologies are not reliant on each other, but they complete each other [16]. Both technologies arise the advantage of employing inexpensive programmable hardware. The early generations of SDN technology were corresponded with data centers, campus networks, and private networks. By the time that SDN find its way to the core of cellular networks. On the other hand, NFV aims to replace the Middleboxes that are used in cellular networks from hardware appliance to software running on commodity hardware (i.e., less expensive general server hardware).

Employing SDN and NFV provides a way for performing the recent important concept of network slicing. The network slice may be viewed as a group of network functions worked together with a specific radio access technology (RAT) to achieve a certain use case of the network [17]. In another word, a network slice is a way of supporting communication service by means of a special connection. Network slicing allows network operator to set up multiple logical networks (each for a certain use case) on the same physical infrastructure [18].

One way to improve the cellular network efficiency is to offload network operations to cloud units that are employed at the edge of the cellular system. This is the idea behind the recent paradigm launched by the cellular network operators and known as MEC. MEC is the way of carrying cloud computing capabilities to the edge of the cellular network one hop away from the user equipment. European Telecommunications Standards Institute (ETSI) is one of the leading organizations in the research of MEC and the standardization of this technology. The latest progress in MEC platforms and their cellular uses are summed in [19]. The main benefit of employing MEC in cellular networks is the reduction of end to end system latency. Moreover, it provides a higher system bandwidth and reduces the network congestion by providing away for offloading data.

In this paper, we introduce a network structure for the Tactile Internet system employs SDN at the core of the network and enable NFV. The system uses MEC in multilevel hierarchical. In (Section 2, the related works to the proposed system is discussed. In Section 3, the system structure is proposed. In Section 4, the system is simulated over a reliable environment. Finally, Section 5 gives the conclusion.

2. Related Works

Moving from traditional Evolved Packet Core networks to next generation of cellular systems requires the involvement of a smart controller at the core of the network. The controller manages all of the operations inside the network. In this part, we present the main works that are concerned with the development of a centralized controller for delay sensitive networks. We analyze each work and provide similarities and differences between each work and our system.

In [20], the authors present a SoftCell cellular system that enables the system operator to achieve service policies at high levels. The proposed system reduces the forwarding table's size and aggregates traffic in multiple dimensions at various switches in the network. The system employs an access switches at the base stations, which are software switches. Access switches can classify packets that have arrived at the base station and handle the required state and bandwidth. SoftCell employs the

components of the traditional Evolved Packet Core cellular network with a controller at the core network with certain functions. The controller installs the switch rules and implements the signaling rules that are used between hosts. The system is suggested for deployment to the core of LTE existing networks. The system improves the flexibility and scalability of the LTE cellular system. The main problem with the system is the capability of the core network controller and it still mainly depends on the gateways (SGW, PGW) at the core network.

Recently, there are a number several approaches speak about employing SDN at the core network of cellular system. These are the most relates works. In [21], the problems and transport challenges to realize the 5G system. The authors suggested a structure for the core network based on SDN and employ the edge computing. In [22], a network coding is developed and employed with SDN to reduce Latency in 5G cellular system. The network coding is achieved through a software router that acts as a virtual network function. The main problem with the system is that they did not consider the MEC; however, they mainly concerned with the coding and SDN. The system is mainly introduced for 5G and the Internet of Things [23–25].

In [26], an optimized framework for a virtual network is introduced to reduce end-to-end delay in LTE-A cellular networks. A central controller is employed in the core network and used mainly for slicing the physical resources. The authors employ Virtual Network Embedding (VNE) algorithm to map the virtualized networks on substrate network. The system model consists of destination edge routers and only single gateway. The edge router is responsible for the mobility management based on a distributed mobility management (DMM) scheme that is applied to the system. The system mainly concerned with the path optimization and network virtualization. The system achieves better latency performance and increases the user mobility. MEC is not involved in the structure and the system may be seen as a modification for the LTE-A cellular system.

In [27], a 5G based SDN architecture is introduced, with the dense deployment of small cells. Employing small cell concept raises the challenge of frequent handover and the latency dedicated with the handover process. Authors provide a system structure to overcome these challenges by using SDN controller at the core network. Another main function of the controller is the allocation of radio access resources and overcoming challenges that are associated with the deployment of multiple radio access technologies. The system is mainly concerned with the problem of latency of the handover process. SDN controller defines three types of programmed interfaces that make the system able to predict the user is movement, and thus handle the handover process in less time.

Each work of the previous works employs a controller at the core network for certain functions. All of these functions provide system with a high efficiency in terms of bit rate and latency. Since the main requirement of the Tactile Internet system is the ultra-low latency, intelligent controller should be deployed at the core of the network to achieve the desired latency.

3. System Structure

Tactile Internet system requires a very high availability and reliability, and ultra-low latency. This put high constraints on system components in terms of context, content, and mobility. Tactile Internet system requires an end-to-end latency of millisecond. This contains both the transmission delay (including reaction transmission (feedback)) and the processing delay. The time between stimulation and response should be in one millisecond. In another word, the end-to-end latency should be less than the human reaction time [28]. In this work, we provide a frame work for the Tactile Internet system, which can solve the problem of millisecond latency and achieves other requirements from the Tactile Internet system.

Our Tactile Internet system may be generally viewed as a three layer system that is based on the proposed 5G system structure, as suggested by Next Generation Mobile Networks (NGMN) [6]. As illustrated in Figure 2, the three layer system is based on the three main technologies; MEC, SDN, and NFV. The whole structure is based on the 5G cellular system, which decouples the hardware and software, and provides an APIs to facilitate the control and management of the system.

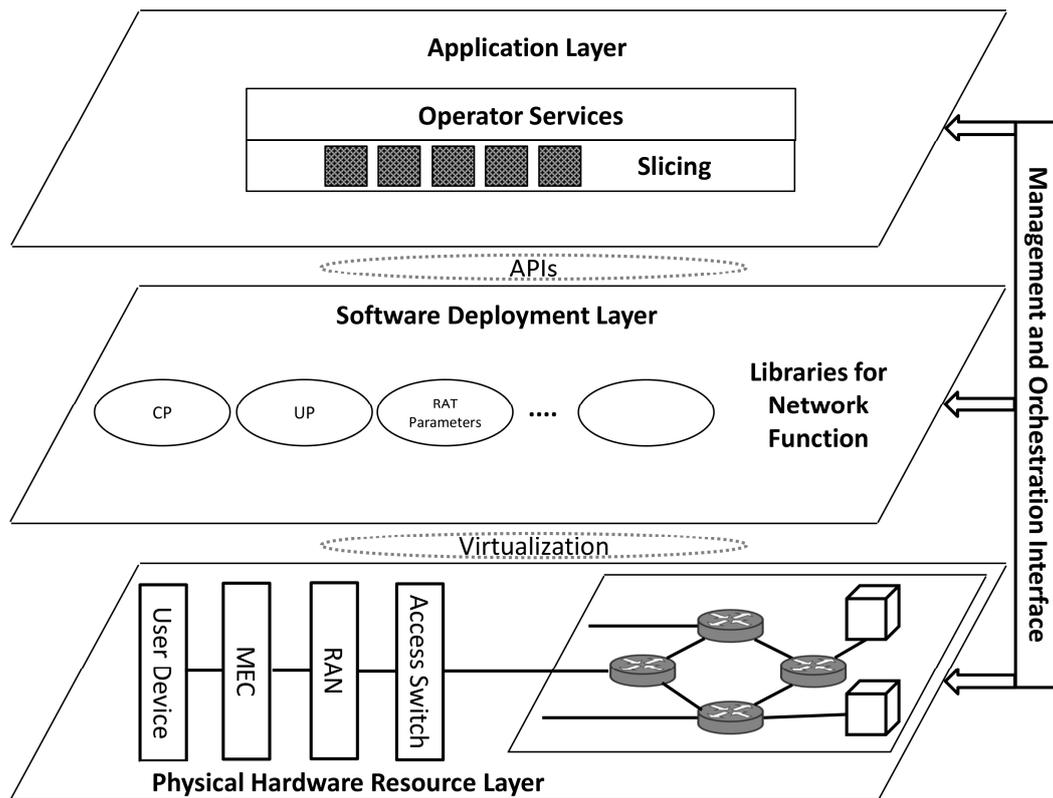


Figure 2. System structure viewed as a three layer system.

The first layer is the physical hardware resource layer, which includes the cellular network infrastructure (5G/Tactile user devices, fiber cables, cloud units, and networking nodes).

User devices may be a master robot or a 5G smart phone that have enough capabilities to be configured in the network [29]. The master robot is a haptic device that represents the human interface system. Haptic device has the ability to translate the human input to a haptic input by means of coding techniques [30]. Using haptic devices users can feel and touch objects in remote environments and also can control them.

Cloud units are employed at the edge of the network based on the concept of MEC. We built a multi-level cloud based Tactile Internet system in [5,31], which can be used as the cloud part in this layer. The multi-level cloud system moves from the idea of centralized cloud to the heterogeneous distributed cloud units. Cloud units are employed in levels, the first level includes Micro-clouds with small storage and processing capabilities connected to each cellular base station (eNB) [5]. The second level employs more efficient cloud units with higher processing and storage capabilities, known as Mini-clouds. Each Mini-cloud unit connects and controls a group of Micro-cloud units through a very high speed fiber connections. The final level of cloud units is the main cloud unit with powerful storage and processing capabilities centered at the core network. Main cloud unit connects, controls, and monitors all Mini-cloud units connected to the core network. Also it acts as the gateway to the huge central cloud units that are employed faraway. Figure 3 illustrates the different levels of cloud units [5].

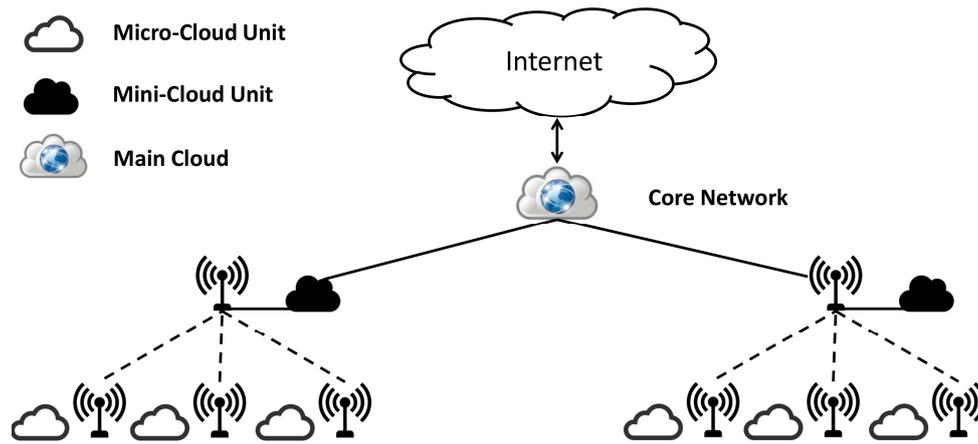


Figure 3. Multi-level cloud layers for Tactile Internet system.

Introducing multi-level cloud units to the Tactile Internet infrastructure, achieves variety of benefits that can be summarized in the following points [31]:

- 1- provides the offloading for the workload of the end user,
- 2- reduces the traffic load flow to the core network,
- 3- achieves better efficiency in terms of network congestion,
- 4- provides new services by accessing the network parameters through the BS,
- 5- provides higher security, and
- 6- reduces the end-to-end latency.

Reducing the end-to-end latency is achieved by reducing the communication path, as the cloud is one hop away from the user’s equipment. The time passed in the core of the network has not existed, as there is no data passed to the core network.

The second layer is the software deployment layer, which maintains libraries for all of the functions that are needed in the network. These libraries are software functions and modules that can be used in the desired location in the network. Also, the radio access parameters and other configuration parameters of the network are located in this layer. The final layer is the application layer, which defines the services and applications that are required by system operators.

The three layers of the system are connected through the system management and orchestration interface. This interface is responsible for managing the operation of the first two layers based on the business application in the higher layer. The management and control tasks of the interface are performed through APIs. The management and orchestration interface is connected to the first layer through APIs to perform, the system configuration and monitoring of status and performance. Also, the software and parameters of the second layer can be fetched at any time by the managing interface through APIs. The link between the application layer and the management and orchestration interface allows the interface to perform network slicing for each application or map it to an existing slice [6].

In order to perform these critical and various tasks, the management and orchestration interface employs SDN/NFV technologies. SDN technology is one way that is used to overcome the problem of one millisecond round trip delay concerned with the Tactile Internet system and the future 5G cellular system. Moreover, SDN provides efficient solutions for achieving network flexibility and efficiency. The core network of cellular network will be based on SDN technology [32]. SDN works based on the separation of the data plane and the control plane. The two plans are linked through the OpenFlow interface protocol.

Figure 4 illustrates the end-to-end structure of our proposed Tactile Internet system. The end-to-end system structure consists of user devices, RAN (eNBs), cloud units, access switches,

OpenFlow switches, Middleboxes, and, finally, SDN controller. Each base station (eNB) is connected to the network through an access switch, which powerfully performs packet classification on traffic from user devices. The access switches is software switches such as Open vSwitch [33].

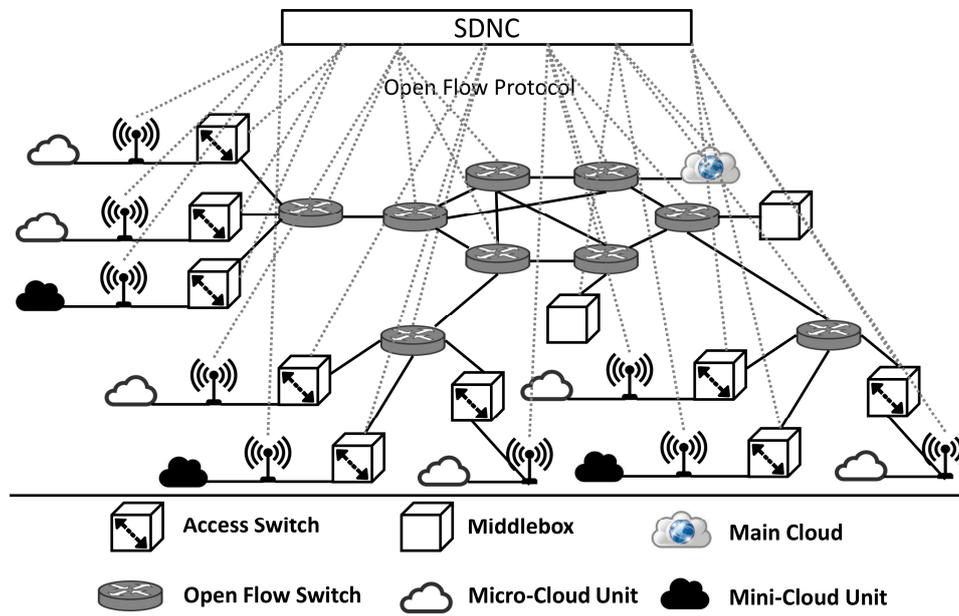


Figure 4. End-to-end system structure.

The whole network is connected through OpenFlow switches that manage data packets and forward traffic based on their flow tables. Middleboxes are commodity hardware that represents a way for the network operators to add extra functionalities, such as firewall and network address translation. The major requirements of the functions and services that are introduced by these Middleboxes are the efficient use of resources and the system protection from attacks. All of these elements represent the data plane of the network.

The last part of the system is the central controller (SDNC), which represents the control plane of the network. By means of programming and control logic, SDNC can execute functions and operations in the control plane. SDNC maintains the global information of the edge and core network devices include; OpenFlow switches, access switches, Middleboxes, RAN, and cloud units. SDNC manages and controls the edge and core network devices mentioned through the OpenFlow protocol. Table 1 contains some of the existing controllers that can be used as an SDNC with the OpenFlow protocol.

Functionally, switches that are used in the tactile system can be viewed as a packet forwarding devices with single or multiple flow tables [34]. Two types of switches are deployed in the OpenFlow based network, pure switches, and hybrid switches. Pure switches support only OpenFlow and fully depended on the SDNC. In the other side, hybrid switches, which are more commercial, have the ability to support OpenFlow with the traditional operations and protocols.

OpenFlow protocol is the signaling standard used for the communication between SDNC and OpenFlow switches. In other words, the instructions from SDNC to other network devices are transferred through the OpenFlow protocol [35]. Each OpenFlow node communicates with the SDNC via a secured OpenFlow channel. OpenFlow switches build their forwarding tables and build the packet processing rules based on the instructions delivered from the SDNC via the OpenFlow protocol. These instructions are deduced from the flow table received from the SDNC. The flow table can be viewed as a set of flow entries. Each of these entries holds a set of instructions to the node [36].

There are only three types of exchanged messages between SDNC and OpenFlow switches that are legalized by the OpenFlow protocol. The first kind is the controller-to-switch messages. This type

refers to all messages initiated and moved from the controller to any switch, for the issues that are related to Handshaking, configuring the switch, checking status of the switch, and even modifying the switch. A summary of all available controller-to-switch messages supported by OpenFlow 1.4 are collected in Table 2, with the fields and entities of each one [37].

The second type represents all messages transferred from the switch to the SDNC; these messages are referred to as asynchronous messages. Table 3 summarizes all of the asynchronous messages used by OpenFlow 1.4; also, the function and structure of these messages are pointed [37]. The third type is the message exchanged in both directions between SDNC and a switch. These messages are referred to as symmetric messages and have various forms and functions. One of the main symmetric messages is the echo request/response messages that are used by the switch or the SDNC to check latency and measure bandwidth. Another important symmetric message is the error messages that are used to report a connection problem. Error messages are mainly used by the switch to report a request failure.

SDNC improve system performance in many terms, such as system latency and user's mobility. One important aspect to reduce the round trip latency is to reduce the number of intermediate nodes that are involved in the communication process and this is introduced by employing SDNC. Moreover, SDNC can foresee the user's mobility as it has access to all features of the user device (including; device type, location, billing information, etc.), which improves the critical radio access processes, such as handover. In other words, employing SDNC reduces the round trip latency, achieves handover process easily, and also reduces the handover latency.

One way to illustrate the benefits of our system is to compare it with the latest traditional Evolved Packet Core cellular network [38]. In the traditional Evolved Packet Core networks, all data traffic flow through the Evolved Packet Core network including the Packet data Gateway (PGW) and the Serving Gateway (SGW). This represents a load on these gateways and leads to an increase in round trip latency. Unlike the traditional systems, our proposed system employs SDN, which removes this barrier and reduces the round trip latency by reducing the number of intermediate nodes that are involved in the communication process.

Figure 5 illustrates this comparison between two systems. While, in Figure 5b, the two eNBs belong to the same Mini-cloud unit, but in Figure 5c, the two eNBs belong to different Mini-cloud units. Another important aspect is that the SDN structure allows and facilitates adding or changing network functions, which is not offered by the traditional systems. Moreover, network slicing can be done through the SDNC which will be the main feature of the Tactile Internet and 5G systems.

Summing up, the advantage of the proposed structure can be summarized in the following points:

- 1- the system is more flexible, as routing can be established easily through SDNC,
- 2- the system is reliable as it relay on the open flow protocol and other standardized technologies,
- 3- the network function virtualization can be easily implemented powered by the use of SDNC,
- 4- the requirements for achieving a certain quality of service (QoS) of the system can be supported by SDN. As the system can define and implement new rules through SDN switches and Middleboxes and
- 5- the system provides higher scalability than traditional packet core networks. This is because the SDNC takes into account the control process only and not involved in data flow.

Table 1. Features of available open source software defined networking (SDN) controllers.

Controller	Implementation Language	Developer	Year of Release	Review
NOX [39]	C++/Python	Nicira	2008	<ul style="list-style-type: none"> - The first SDN controller. - considered as network operating system than being a controller.
Maestro [40]	Java	Rice University	2011	<ul style="list-style-type: none"> - Allows applications to access and modify the network state. - considered as network operating system than being a controller. - Multi-thread platform. - Portable and scalable.
POX [41]	Python	Nicira	2013	<ul style="list-style-type: none"> - become more common than the old NOX. - Modern controller that enables rapid development and prototyping.
Beacon [42]	Java	Stanford University	2013	<ul style="list-style-type: none"> - Dynamic and cross-platform. - used to increase productivity of developers. - enables the start and stop of applications and introducing newer ones.
Floodlight [43]	Java	Big Switch Networks	2013	<ul style="list-style-type: none"> - One of the leading SDN controllers. - Hybrid controller that can support both OpenFlow and non OpenFlow networks. - High performance multi-thread controller.
OpenDaylight [44]	Java	OpenDaylight Foundation	2013	<ul style="list-style-type: none"> - efficient for any size networks. - provides various commercial use cases. - Main concern is the network programmability.
IRIS [45]	Java	ETRI	2014	<ul style="list-style-type: none"> - Built on Beacon. - Simpler architecture. - reconfigure some features in Floodlight.
ONOS [46]	Java	Open Network Operating System project	2014	<ul style="list-style-type: none"> - ONOS stands for Open Network Operating System. - Main concern is the scalability and the availability. - has various commercial applications.
DEFO [47]	Java	DEFO project	2015	<ul style="list-style-type: none"> - Stands for Declarative and Expressive Forwarding Optimizer. - Increase the connectivity. - Provides scalability and high response.

Table 2. Controller-to-Switch messages supported by OpenFlow.

Message Type	Message Structure	Message	Purpose	Reply
Handshake	- ofp_header - (no body)	OFPT_FEATURES_REQUEST	Used by the controller to identify the switch and its features.	OFPT_FEATURES_REPLY: - ofp_header - pad - n_buffers - n_tables-datapath_id - auxiliary_id
Switch configuration	- ofp_header - (no body)	OFPT_GET_CONFIG_REQUEST	Used by the controller to set the configuration parameter of the switch.	OFPT_GET_CONFIG_REPLY: - ofp_header - flags - miss_send_len
Modify state messages	- ofp_header - table_id - pad - config	OFF_TABLE_MOD	Used by the controller to perform the dynamic state configuration of a switch.	
	- ofp_header - cookie - cookie_mask - table_id - command - idle_timeout - hard_timeout - priority - buffer_id - out_port - out_group - flags - importance - ofp_match	OFPT_FLOW_MOD	Used by the controller to modify a flow table.	No main response message.
	- ofp_header - command - type - pad - group_id - ofp_bucket	OFPT_GROUP_MOD	Used by the controller to modify a group table.	

Table 2. Cont.

Message Type	Message Structure	Message	Purpose	Reply
	<ul style="list-style-type: none"> - ofp_header - port_no - pad - hw_addr - pad2 - config - mask 	OFPT_PORT_MOD	Used by the controller to modify the properties of a port.	
	<ul style="list-style-type: none"> - ofp_header - command - flags - meter_id 	OFPT_METER_MOD	Used by the controller to modify a meter and a virtual meter.	
	<ul style="list-style-type: none"> - ofp_header - type - flags - pad - body 	OFPT_MULTIPART_REQUEST	Used by the controller to encode large requests and replies messages that can't be sent in one message.	OFPT_MULTIPART_REPLY: <ul style="list-style-type: none"> - ofp_header - type - flags - pad - body
	<ul style="list-style-type: none"> - ofp_header - (no body) 	OFPMP_DESC (Request)	Used by the controller to request information about the switch (i.e., switch manufacturer, information about the used software and serial number).	OFPMP_DESC (Reply): <ul style="list-style-type: none"> - mfr_desc - hw_desc - sw_desc - serial_num - dp_desc
Multipart messages	<ul style="list-style-type: none"> - table_id - pad - out_port - out_group - pad2 - cookie - cookie_mask - ofp_match 	OFPMP_FLOW (Request)	Used by the controller to request information about flow entries.	OFPMP_FLOW (Reply): <ul style="list-style-type: none"> - length - table_id - pad2 - duration_sec - duration_nsec - pad - priority - ofp_match - idle_timeout - hard_timeout - flags - importance - cookie - packet_count - byte_count

Table 2. Cont.

Message Type	Message Structure	Message	Purpose	Reply
- table_id - pad - out_port - out_group - pad2 - cookie - cookie_mask - ofp_match		OFPMP_AGGREGATE (Request)	Used by the controller for the information aggregation of multiple flow entries.	OFPMP_AGGREGATE (Reply): - packet_count - byte_count - flow_count - pad
- ofp_header (no body)		OFPMP_TABLE (Request)	Used by the controller to get main information about the tables.	OFPMP_TABLE (Reply): - table_id - pad - active_count - lookup_count - matched_count
- ofp_header (no body)		OFPMP_TABLE_DESC (Request)	Used by the controller to get the current tables configuration of a switch.	OFPMP_TABLE_DESC (Reply): - length - table_id - pad - config - properties
- port_no - queue_id		FPMP_QUEUE_STATS (Request)	Used by the controller to get the queue information of one or more port.	FPMP_QUEUE_STATS (Reply): - length - queue_id - pad - duration_sec - tx_packets-tx_bytes - tx_errors - port_no - duration_nsec
- group_id - pad		OFPMP_GROUP (Request)	Used by the controller to get information about one or more groups.	OFPMP_GROUP (Reply): - length - group_id - pad - packet_count - ref_count - pad2 - byte_count - duration_sec - duration_nsec

Table 2. Cont.

Message Type	Message Structure	Message	Purpose	Reply
	<ul style="list-style-type: none"> - ofp_header - (no body) 	OFPMPP_PORT_DESCRIPTION (Request)	Used by the controller to get knowledge of all ports support OpenFlow.	OFPMPP_PORT_DESCRIPTION(Reply): <ul style="list-style-type: none"> - port_no - length - pad-hw_addr - pad2 - name - config - state
	<ul style="list-style-type: none"> - meter_id - pad 	OFPMT_METER (Request)	Used by the controller to request meter statistics.	OFPMT_METER (Reply): <ul style="list-style-type: none"> - meter_id - pad - byte_in_count - length - flow_count - duration_nsec - packet_in_count - duration_sec
Packet-out message	<ul style="list-style-type: none"> - ofp_header - buffer_id - in_port-pad - actions_len - ofp_action_header - data 	OFPT_PACKET_OUT	Used by the controller when intended to send a packet out.	No main answer.
Barrier message	<ul style="list-style-type: none"> - ofp_header - (no body) 	OFPT_BARRIER_REQUEST	Used by the controller to ask for the completed operations.	OFPT_BARRIER_REPLY
Role request message	<ul style="list-style-type: none"> - ofp_header - role - pad - generation_id 	OFPT_ROLE_REQUEST	Used by the controller to change its role.	OFPT_ROLE_REPLY (if no errors occur) <ul style="list-style-type: none"> - ofp_header - role - pad - generation_id

Table 3. Asynchronous Messages supported by OpenFlow.

Message type	Message Structure	Message	Purpose
Packet-in message	<ul style="list-style-type: none"> - ofp_header-data - reason - -cookie-table_id - total_len - ofp_match-buffer_id-pad 	OFPT_PACKET_IN	Used by the switch to inform controller with packet received.
Flow removed message	<ul style="list-style-type: none"> - ofp_header - cookie - reason - -priority-table_id - ofp_match - idle_timeout - duration_sec - hard_timeout - packet_count - byte_count - duration_nsec 	OFPT_FLOW_REMOVED	Used by the switch to notify the controller with deleted flow entries.
Port status message	<ul style="list-style-type: none"> - ofp_header – reason - pad - ofp_port desc 	OFPT_PORT_STATUS	Used by the switch to notify the controller with the ports added, removed or modified.
Controller role status message	<ul style="list-style-type: none"> - ofp_header - role - reason - pad - generation_id 	OFPT_ROLE_STATUS	Used by the switch to change the controller role.
Table status message	<ul style="list-style-type: none"> - ofp_header - reason - pad - ofp_table_desc table 	OFPT_TABLE_STATUS	Used by the switch to inform controller with the change of table status.

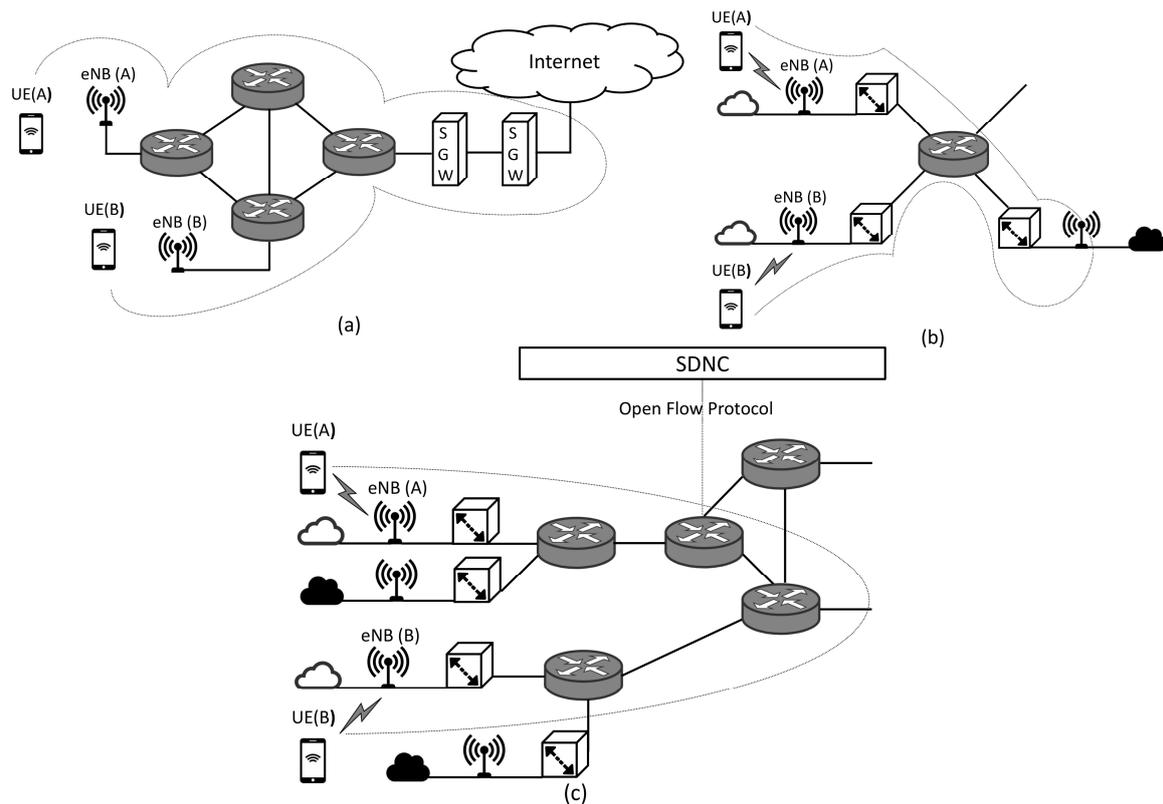


Figure 5. (a) Evolved Packet Core cellular network; (b) First case of the proposed system; (c) Second case of the proposed system.

4. Simulation and Results

In this section, the proposed structure for the Tactile Internet system is simulated over a reliable environment, and the results are discussed.

4.1. Simulation Environment and Experiment Set Up

There are a number of efficient simulation environments and frameworks that are used to simulate and evaluate the performance and attributes of SDN and MEC based networks. One of the most efficient, reliable, and powerful frameworks is the CloudSim environment and its related extension projects. CloudSim is a Java based simulation framework that enables simulation, experimentation, and modeling for cloud based networks [48]. CloudSimSDN is a Java based simulation framework that is built on top of CloudSim [49]. This framework is developed mainly for simulation purposes of SDN and cloud based systems.

In CloudSimSDN, the SDNC is programmable and it allows for the testing of VM management policies and workload scheduling algorithms [50]. Furthermore, the centralized controller is responsible for dynamically managing resources in data centers. Based on [49], the authors prove that CloudSimSDN simulator provides much features and facilities with better or at least the same performance as Mininet simulation environment. Also, it gives the ability for the modifications and extensions.

We consider two simulation cases with the topology illustrated in Figure 5b,c. In the first case, the two hosts belongs two different base stations (eNBs), but the two eNBs are connected to the same Mini-cloud. In the second case, the two hosts are in different cells and the cells are connected to different Mini-cloud units. The round trip delay is considered to be the performance of our system. The simulation process is repeated multi-times, with different bandwidth for each time but with the data size is the same in each case. All important simulation parameters used are illustrated in Table 4.

Table 4. Simulation parameters.

Simulation Parameter	Value
OpenFlow Switch processing delay	5 μ s
SDN Controller processing delay	0.5 μ s
Arrival rate of the Micro-cloud unit λ_i	15
The communication latency inside the cellular cell	100 μ s
Bandwidth	Variable
Micro-cloud RAM, Storage	1024 Mb, 1 Gb
Mini-cloud RAM, Storage	2048 Mb, 5 Gb

4.2. Simulation Results and Analyses

Figures 6 and 7 illustrate the results for the first and the second simulation cases. From simulation results, the round trip delay is decreased with the increase of the system bandwidth. For the second case, the round trip delay is higher than that of the first case, this is because the core network controller is not involved in the communication process as the Mini-cloud unit manages and performs the communication process. Based on [6,51] the user bit rate for the future 5G cellular system will range from 1 Gbps to 10 Gbps. For the least expected bit rate (1 Gbps), the proposed tactile system affords a round trip delay of 0.95 ms for the first simulation case and 1.22 ms for the second case. Moreover, as the bit rate is raised above 1 Gbps the round trip delay gets below the previous values which make the proposed structure get ride off the challenge of 1 ms round trip latency.

Based on the 3GPP release 13, the LTE system achieves round trip latency of 16 ms for the 100 Mbps bit rate and 8 ms for 1 Gbps based on release 15 [52,53]. Thus, the proposed system can improve the latency efficiency and provides a way for Tactile Internet realization. When compared to the traditional Evolved Packet Core network based on the 3GPP release 15, the proposed system achieves 88% reduction of the round trip latency for the first case and 84% for the second case, which is the worst.

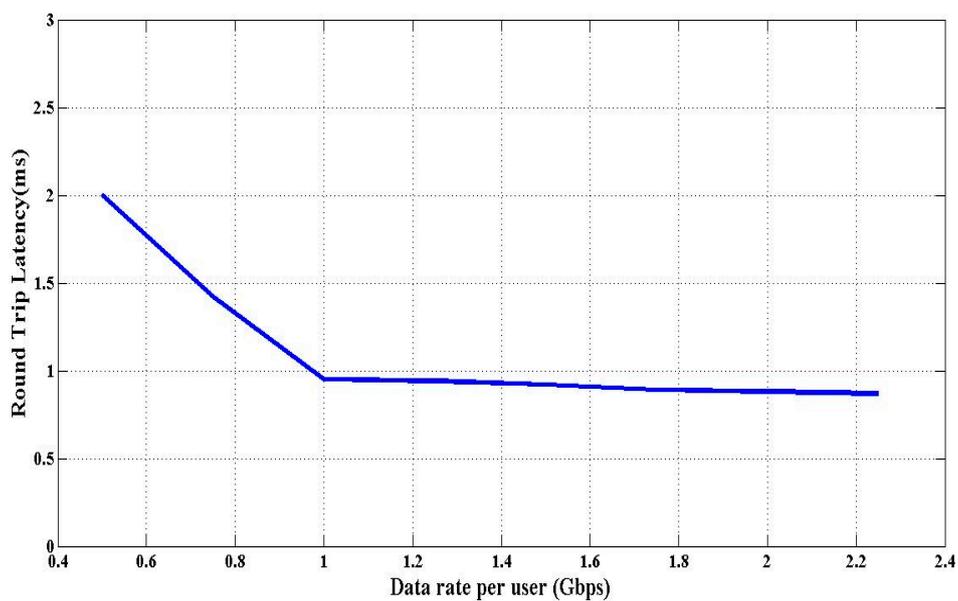


Figure 6. Simulation results for the first case.

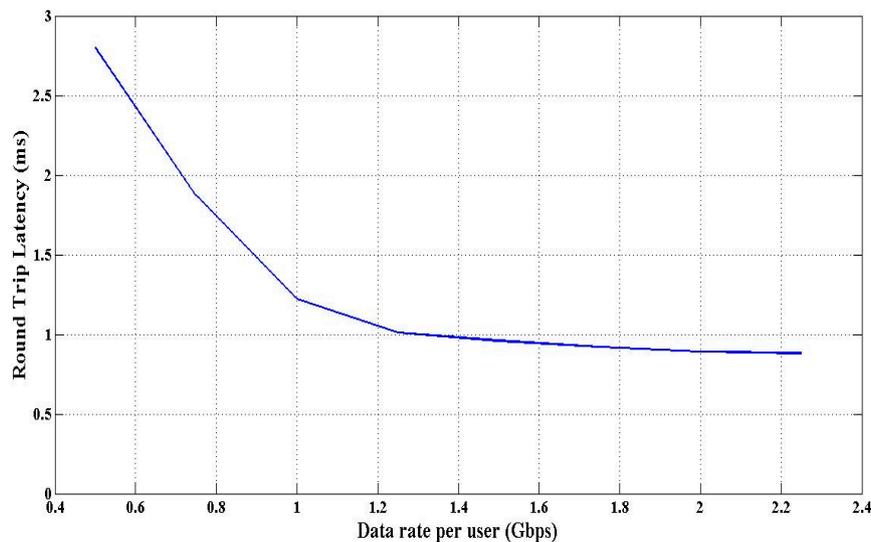


Figure 7. Simulation results for the second case.

5. Conclusions

SDN, NFV, and MEC technologies are solutions to overcome the challenges associated with the realization of Tactile Internet system, especially the 1 ms round trip latency. Employing SDN at the core network of cellular system reduces the latency and efficiently improves the system performance. This is because SDNC can efficiently manage and establish an efficient and flexible routing path between any two end points. Thus, it can reduce the number of intermediate nodes that are involved in communication process. The proposed structure for the Tactile Internet system, which employs SDN at the core of the network and multi-level cloud units, poses a round trip latency of orders of ms. Thus, this structure is helpful and effective for building the Tactile Internet system.

Acknowledgments: The publication was financially supported by the Ministry of Education and Science of the Russian Federation (the Agreement number 02.a03.21.0008). The authors are grateful to the Middle East University, Amman, Jordan for the financial support granted to cover the publication fee of this research article.

Author Contributions: Abdelhamied A. Ateya and Ammar Muthanna built the network model and perform the simulation process. Irina Gudkova and Abdelrahman Abuarqoub analyze the existing methods and the issues related to the OpenFlow protocol. Anastasia Vybornova and Andrey Koucheryavy wrote the paper and revise the whole work.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Aijaz, A. Towards 5G-enabled tactile internet: Radio resource allocation for haptic communications. In Proceedings of the 2016 IEEE Wireless Communications and Networking Conference (WCNC), Doha, Qatar, 3–6 April 2016; pp. 1–6.
2. Simsek, M.; Aijaz, A.; Dohler, M.; Sachs, J.; Fettweis, G. 5G-enabled tactile internet. *IEEE J. Sel. Areas Commun.* **2016**, *34*, 460–473. [[CrossRef](#)]
3. *ITU-T Technology Watch Report—The Tactile Internet*; International Telecommunication Union: Paris, France, 2014.
4. Maier, M.; Chowdhury, M.; Rimal, B.P.; van, D.P. The tactile internet: Vision, recent progress, and open challenges. *IEEE Commun. Mag.* **2016**, *54*, 138–145. [[CrossRef](#)]
5. Ateya, A.A.; Vybornova, A.; Kirichek, R.; Koucheryavy, A. Multilevel cloud based Tactile Internet system. In Proceedings of the 19th International Conference on Advanced Communication Technology (ICACT), Bongpyeong, Korea, 19–22 February 2017; pp. 105–110.
6. Alliance, N.G.M.N. *5G White Paper*; Next Generation Mobile Networks: Frankfurt, Germany, 2017.

7. Liu, Y.; Liu, G. User-centric wireless network for 5G. In *5G Mobile Communications*; Springer International Publishing: Cham, Switzerland, 2017; pp. 457–473.
8. Han, B.; Gopalakrishnan, V.; Ji, L.; Lee, S. Network function virtualization: Challenges and opportunities for innovations. *IEEE Commun. Mag.* **2015**, *53*, 90–97. [[CrossRef](#)]
9. Karl, H.; Dräxler, S.; Peuster, M.; Galis, A.; Bredel, M.; Ramos, A.; Martrat, J.; Siddiqui, M.S.; van Rossem, S.; Tavernier, W.; et al. DevOps for network function virtualisation: An architectural approach. *Trans. Emerg. Telecommun. Technol.* **2016**, *27*, 1206–1215. [[CrossRef](#)]
10. Blanco, B.; Fajardo, J.O.; Giannoulakis, I.; Kafetzakis, E.; Peng, S.; Pérez-Romero, J.; Trajkovska, I.; Khodashenas, P.S.; Goratti, L.; Paolino, M.; et al. Technology pillars in the architecture of future 5G mobile networks: NFV, MEC and SDN. *Comput. Stand. Interfaces* **2017**, *54*, 216–228. [[CrossRef](#)]
11. Kirichek, R.; Vladyko, A.; Paramonov, A.; Koucheryavy, A. Software-defined architecture for flying ubiquitous sensor networking. In Proceedings of the 2017 19th International Conference on Advanced Communication Technology (ICACT), Bongpyeong, Korea, 19–22 February 2017; pp. 158–162.
12. Hammoudeh, M.; Newman, R. Information extraction from sensor networks using the Watershed transform algorithm. *Inf. Fusion* **2015**, *22*, 39–49. [[CrossRef](#)]
13. Abuarqoub, A.; Hammoudeh, M.; Adebisi, B.; Jabbar, S.; Bounceur, A.; Al-Bashar, H. Dynamic clustering and management of mobile wireless sensor networks. *Comput. Netw.* **2017**, *117*, 62–75. [[CrossRef](#)]
14. Athmiya, N.S.; Shobha, K.R.; Sarimela, V. Feasibility study and implementation of openflow based SDN controller for tactical scenario. In Proceedings of the IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), Bangalore, India, 20–21 May 2016; pp. 789–794.
15. Muthanna, A.; Masek, P.; Hosek, J.; Fujdiak, R.; Hussein, O.; Paramonov, A.; Koucheryavy, A. Analytical evaluation of D2D connectivity potential in 5G wireless systems. In Proceedings of the International Conference on Next Generation Wired/Wireless Networking, St. Petersburg, Russia, 26–28 September 2016; Springer International Publishing: Cham, Switzerland, 2016; pp. 395–403.
16. Shariat, M.; Gutierrez-Estevez, D.M.; Vijay, A.; Safjan, K.; Rugeland, P.; Silva, I.; Lorca, J.; Widmer, J.; Fresia, M.; Li, Y.; et al. 5G radio access above 6 GHz. *Trans. Emerg. Telecommun. Technol.* **2016**, *27*, 1160–1167. [[CrossRef](#)]
17. Jiang, M.; Condoluci, M.; Mahmoodi, T. Network slicing management & prioritization in 5G mobile systems. In Proceedings of the 22th European Wireless Conference European Wireless (VDE), Oulu, Finland, 18–20 May 2016; pp. 1–6.
18. An, X.; Zhou, C.; Trivisonno, R.; Guerzoni, R.; Kaloxylos, A.; Soldani, D.; Hecker, A. On end to end network slicing for 5G communication systems. *Trans. Emerg. Telecommun. Technol.* **2017**, *28*. [[CrossRef](#)]
19. Satyanarayanan, M. The Emergence of Edge Computing. *Computer* **2017**, *50*, 30–39. [[CrossRef](#)]
20. Jin, X.; Li, L.E.; Vanbever, L.; Rexford, J. Softcell: Scalable and flexible cellular core network architecture. In Proceedings of the Ninth ACM Conference on Emerging Networking Experiments and Technologies, Santa Barbara, CA, USA, 9–12 December 2013; pp. 163–174.
21. Iovanna, P.; Ubaldi, F. SDN solutions for 5G transport networks. In Proceedings of the 2015 International Conference on Photonics in Switching (PS), Florence, Italy, 22–25 September 2015; pp. 297–299.
22. Szabo, D.; Gulyas, A.; Fitzek, F.H.; Lucani, D.E. Towards the tactile internet: Decreasing communication latency with network coding and software defined networking. In Proceedings of the 21th European Wireless Conference European Wireless, Budapest, Hungary, 20–22 May 2015; pp. 1–6.
23. Hammoudeh, M.; Newman, R. Interpolation techniques for building a continuous map from discrete wireless sensor network data. *Wirel. Commun. Mob. Comput.* **2013**, *13*, 809–827. [[CrossRef](#)]
24. Abuarqoub, A.; Abusaimh, H.; Hammoudeh, M.; Uliyan, D.; Abu-Hashem, M.A.; Murad, S.; Al-Jarrah, M.; Al-Fayez, F. A survey on internet of things enabled smart campus applications. In Proceedings of the International Conference on Future Networks and Distributed Systems, Cambridge, UK, 19–20 July 2017.
25. Hammoudeh, M. Putting the lab on the map: A wireless sensor network system for border security and surveillance. In Proceedings of the International Conference on Internet of things and Cloud Computing, Cambridge, UK, 22–23 March 2016.
26. Chochlidakis, G.; Friderikos, V. Low latency virtual network embedding for mobile networks. In Proceedings of the 2016 IEEE International Conference on Communications (ICC), Kuala Lumpur, Malaysia, 22–27 May 2016; pp. 1–6.

27. Wang, K.; Wang, Y.; Zeng, D.; Guo, S. An SDN-based architecture for next-generation wireless networks. *IEEE Wirel. Commun.* **2017**, *24*, 25–31. [[CrossRef](#)]
28. Aijaz, A.; Simsek, M.; Dohler, M.; Fettweis, G. Shaping 5G for the Tactile Inter-net. In *5G Mobile Communications*; Springer International Publishing: Cham, Switzerland, 2017; pp. 677–691.
29. Chih-Lin, I.; Han, S.; Xu, Z.; Sun, Q.; Pan, Z. 5G: Rethink mobile communications for 2020+. *Phil. Trans. R. Soc. A* **2016**, *374*. [[CrossRef](#)] [[PubMed](#)]
30. Pance, A.; Webb, U.; Mayo, S.A.; Apple Inc. Haptic Feedback Device. U.S. Patent 9,710,061, 18 July 2017.
31. Ateya, A.A.; Vybornova, A.; Samouylov, K.; Koucheryavy, A. System model for multi-level cloud based tactile internet system. In Proceedings of the International Conference on Wired/Wireless Internet Communication, St. Petersburg, Russia, 21–23 June 2017; Springer: Cham, Switzerland, 2017; pp. 77–86.
32. Sahoo, K.S.; Mohanty, S.; Tiwary, M.; Mishra, B.K.; Sahoo, B. A comprehensive tutorial on software defined network: The driving force for the future internet technology. In Proceedings of the International Conference on Advances in Information Communication Technology & Computing, Bikaner, India, 12–13 August 2016; p. 114.
33. Open vSwitch. Available online: <http://openvswitch.org> (accessed on 20 September 2017).
34. Taleb, T.; Samdanis, K.; Mada, B.; Flinck, H.; Dutta, S.; Sabella, D. On multi-access edge computing: A survey of the emerging 5G network edge architecture & orchestration. *IEEE Commun. Surv. Tutor.* **2017**. [[CrossRef](#)]
35. Open Networking Foundation (ONF). Available online: <https://www.opennetworking.org/> (accessed on 10 October 2017).
36. Karakus, M.; Durresi, A. Quality of Service (QoS) in software defined networking (SDN): A survey. *J. Netw. Comput. Appl.* **2017**, *80*, 200–218. [[CrossRef](#)]
37. OpenFlow v1.4 Specification White Paper. Available online: <https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.4.0.pdf> (accessed on 12 December 2017).
38. Basta, A.; Kellerer, W.; Hoffmann, M.; Hoffmann, K.; Schmidt, E.D. A virtual SDN-enabled LTE EPC architecture: A case study for S-/P-gateways functions. In Proceedings of the 2013 IEEE SDN for Future Networks and Services (SDN4FNS), Trento, Italy, 11–13 November 2013; pp. 1–7.
39. Gude, N.; Koponen, T.; Pettit, J.; Pfaff, B.; Casado, M.; McKeown, N.; Shenker, S. NOX: Towards an operating system for networks. *ACM SIGCOMM Comput. Commun. Rev.* **2008**, *38*, 105–110. [[CrossRef](#)]
40. Ng, E.; Cai, Z.; Cox, A.L. *Maestro: A System for Scalable Openflow Control*; TSEN Maestro-Technical Report TR10-08; Rice University: Houston, TX, USA, 2010.
41. NOXrepo.org. Available online: <https://github.com/noxrepo/> (accessed on 1 October 2017).
42. Erickson, D. The beacon openflow controller. In Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, Hong Kong, China, 16 August 2013; pp. 13–18.
43. Project Floodlight: Open Source Software for Building Software-Defined Networks. Available online: <http://www.projectfloodlight.org/floodlight/> (accessed on 25 September 2017).
44. OpenDaylight. Available online: <http://www.opendaylight.org/> (accessed on 25 September 2017).
45. OpenIRIS Project. Available online: <https://github.com/openiris/IRIS/wiki> (accessed on 5 October 2017).
46. Onosproject. Available online: <https://onosproject.org/software/> (accessed on 7 October 2017).
47. Hartert, R.; Vissicchio, S.; Schaus, P.; Bonaventure, O.; Filsfils, C.; Telkamp, T.; Francois, P. A Declarative and Expressive Approach to Control Forwarding Paths in Carrier-Grade Networks. In Proceedings of the ACM SIGCOMM Computer Communication Review, London, UK, 17–21 August 2015; Volume 45, pp. 15–28. Available online: <https://sites.uclouvain.be/defo/#publications> (accessed on 7 October 2017).
48. Kumar, R.; Sahoo, G. Cloud computing simulation using CloudSim. *arXiv*, 2014.
49. Son, J.; Dastjerdi, A.V.; Calheiros, R.N.; Ji, X.; Yoon, Y.; Buyya, R. Cloudsimsdn: Modeling and simulation of software-defined cloud data centers. In Proceedings of the 2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), Shenzhen, China, 4–7 May 2015; pp. 475–484.
50. CloudSimSDN Project. Available online: <https://github.com/jayjmin/cloudsimsdn> (accessed on 12 December 2017).
51. Zheng, K.; Zhao, L.; Mei, J.; Dohler, M.; Xiang, W.; Peng, Y. 10 Gb/s hetsnets with millimeter-wave communications: Access and networking-challenges and protocols. *IEEE Commun. Mag.* **2015**, *53*, 222–231. [[CrossRef](#)]

52. Arenas, J.; Dudda, T.; Falconetti, L. Ultra-Low Latency in Next Generation LTE Radio Access. In Proceedings of the SCC 2017 11th International ITG Conference on Systems, Communications and Coding, Hamburg, Germany, 6–9 February 2017; pp. 1–6.
53. Lauridsen, M.; Gimenez, L.C.; Rodriguez, I.; Sorensen, T.B.; Mogensen, P. From LTE to 5G for Connected Mobility. *IEEE Commun. Mag.* **2017**, *55*, 156–162. [[CrossRef](#)]



© 2018 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 (<http://creativecommons.org/licenses/by/4.0/>).