Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

*Digital Object Identifier xx.xxxx/ACCESS.xxxx.DOI*

# Smart Virtualization Packets Forwarding During Handover for Beyond 5G Networks

**FOUAD A. YASEEN[1], HAMED S. AL-RAWESHIDY.[1], ( Senior Member, IEEE)**
[1]Wireless Networks and Communications Centre (WNCC), Department of Electronic and Computer Engineering, College of Engineering, Design and Physical Sciences, Brunel University London, London, UB83PH, U.K.

Corresponding author: Fouad A. Yaseen (e-mail: Fouad.Yaseen@ brunel.ac.uk).

**ABSTRACT** Keeping a connection continuity during the movement of a Mobile Node (MN) between access points without any suspension of provided services is one of the most pressing issues should be solved. Long handover processing causes interruptions in session connection, high rate of data loss, and long End-to-End delay time. *Smart Virtualization* means cooperation of different virtualization technologies with novel ideas. In this paper, we proposed a mobile network architecture compatible with cloud computing of 5G and beyond networks. We invented a new idea to create a tag to be used as an MN's identity, which consists of the standard E.164 numbering and MAC address. Based on the uniqueness of E.164 numbering and MAC which are processed together to generate the MN tag ($T_H$). The $T_H$ is used to handle the packets inside the mobile networks. The Software Defined Networking (SDN) provides a capability of separating the Control Plane from the Data Plane. This decoupling is a suitable candidate to exploit it in our proposed system which uses SDN and other virtualization technologies. The requirements of 5G and beyond for future mobile communications encouraged us to think in a novel packet forwarding during the handover to keep real-time connection continuity for an MN. Our proposed system has been simulated and performed by MATLAB and Mininet platforms. The results showed that the packet loss rate decreased to 4% of that were lost during the handover delay time or while packets re-direction mechanism. At the same time, the MN could receive 96.4% of the data that was lost during the handover process.

**INDEX TERMS** 5G, Control Plane, Communication Networks, SDN, NFV.

## I. INTRODUCTION

THE most important feature of modern life is to be connected to the Internet whether by fixed or mobile devices. The smartphone is the most significant of these devices which burst into a broad spread in whole the world. As a result, the IPv4 addresses are depleted due to the growing number of the smart devices. Consequently, Internet service providers can not provide enough IP addresses to the continually increasing in the number of these devices. Using IPv6 service providers can accommodate a tremendous amount of IP addresses which can be assigned to all devices and networks associated with the IP address [1]. The IP addresses enable the communication between mobile devices and their wireless access points. Hence, wireless devices can identify themselves by their IP addresses. Moreover, it is used as a pointer of the MN's location as well as a device identifier. Additionally, the IP address binds the MN's identity with its running applications to the current position of the MN on Internet networks. The enormous increase in the number of mobile devices makes the binding is difficult to support the mobility through the Internet and the wireless networks due to increasing the size of the routing table, depletion IP addresses and bandwidth, more power consumption, and packets handover delay.

There are several protocols for mobility based on the separation of the mobile IP into Locator and Identifier. Reference [2] explained the Mobile IPv6 (MIPv6) mechanism. This protocol used a permanent IP address called Home of Address (HoA) for an MN as an identifier and changeable Care of Address (CoA) as a locator. These routable IPs are managed by Home Agent (HA) which maintains the mapping between the HoA and CoA. Another protocol called Proxy MIPv6(PMIPv6) [3] which used the Home Address (HoA) for identifying the MN and Proxy CoA (PCoA) for the locator. Managing the binding information of HoA and PCoA is done by Local Mobility Anchor (LMA) [4].

In the last decade, several protocols emerged based on the separation of IP address into locator and identifier. These protocols suggested two concepts of separation. The first one is host centric such as Host Identifier Protocol (HIP) [5], Site Multihoming by IPv6 Intermediation (Shim6) protocol [6], Mobile-Oriented Future Internet (MOFI) [7] and Locator-Identifier Separation Protocol host (LISP-host) [8]. The second concept is network-centric such as Locator-Identifier Separation Protocol Distributed Mobility Control (LISP-DMC) [9], and Distributed Hash Table (DHT)-based identifier-to-locator mapping [10]. All these protocols use tunneling techniques to deliver packets. These tunneling techniques deplete a significant amount of networks bandwidth and increase processing delay.

The expected communications system should support the 5G and beyond mobile networks that guarantee to satisfy the requirements such as:

- Low control latency (less than 1ms).
- High-speed mobility up to 500 km/h.
- Traffic density up to 1000 folds than today.
- Almost 100% coverage.
- Less network management and administration.
- Separation of the data plane from the control plane of the network traffic.
- Flexible sharing of network resources.

At the same, these networks should not increase infrastructure cost and power consumption. Besides, the self-organization network functions should present to manage the systems [11]. Emerging virtualization of networking technologies such as Software Defined Networking (SDN) and Network Functions Virtualization (NFV) has been helping to make mobile and Internet networks to be more flexible and agiler. A network architecture based on SDN technology depends on the separation of the data layer from the control layer. The data layer involves physical switches with high performance to deliver the data, while, the control layer is represented by the SDN controller (SDNc), which is centralized in the logical software substance. The idea of SDN based on four pillars [12]–[14]., they are:

- Separation of the control plane from the data plane.
- Forwarding decisions are made by SDNc which is placed away from the forwarding data plane devices.
- Packet forwarding based on the flow rather than the destination address.
- Programmable software of the network functions interacts with the forwarding data plane devices via application program interface (API) under the management of SDNc.

We have focused and worked on the delay reduction issue of handling the packets pass through the network devices during the handover for 5G networks. To address the problems mentioned above, we propose an MN centric network-based SDN and network function virtualization (NFV). The virtualization technology actively empowers the SDN. The fundamental structure of the SDN network depends on decoupling the Control Plane (CP) (which represents controlling packets that are created by the SDNc) from Data Plane (DP) (which involves physical switches with high performance to deliver a pure data packets of a network). The CP can be implemented by a physical or virtual machine away from the DP [15]. Moreover, in the SDN environment, the control packets do not utilize the standard IP routing only, because it could use different algorithms and mechanisms according to which task is wanted to be implemented by the algorithm. The idea of this paper based on our previous proposal that entitled Smart Virtual eNB (SVeNB) [16]. The SVeNB has suggested of using SDN and NFV. The SDN harmonize with NFV technology which enables building a new virtualized mobile networks underpin by employing virtualization technology standards to consolidate the different network devices.

1) Creating a new identifier for an MN to be used as a local identity within the mobile operator networks.
2) Adopting continuous, seamless packets delivery during the handover and a new mobility management mechanism based-SDN and NFV.

Our paper is organized as follows. Section III-A explains the proposed tag generating ($T_H$) and cause of using it. The system architecture is detailed in Section III-B. Section III-C describes the mobility management procedures. In Section III-D illustration of handover procedures, handover delay and the causes of the delay. Section IV presents a comparison between the traditional and our proposed schemes. The simulation and performance evaluation of the proposed scheme is given in Section V, and finally, Section VI contains the conclusion.

## II. BACKGROUND

### A. SOFTWARE DEFINED NETWORKING

The SDN concept has emerged to overcome the different types of network devices that have been producing by many different companies. The SDN has been developed and standardized by the Open Networking Foundation (ONF), which is a nonprofit consortium. It defined the SDN architecture as a separation of the control plane from the data plane. The intellect of the network is logically gathered in a separate place, and the implicit network devices are extracted away from the applications [13].

In general, the SDN concept architecture composed of four principal elements as shown in Fig 1. Each of these elements is explained briefly below [12], [14]:

1) Applications layer involves the applications software that exchanges the controlling data with the SDNc, which collects the extracted information from that the constructed by the application layer about the network infrastructure.
2) Control layer contains the principal element of the SDN model, (i.e., SDNc), which manages and makes all the forwarding rules and decisions of the network data. The

SDNc is the creative element that is responsible for directing and making the decisions regarding the flows that enter the underlying SDN infrastructure through northbound and southbound APIs.

3) In universal terms, The API is a group of defined rules of communication between various software parts. It is a collection of routines, protocols, and tools for creating software applications. Radically, an API specifies how software segments should interact. Three types of APIs work with the SDN concept.
   a) Northbound API connects the control layer with the applications layer. It communicates between the network management station running its network applications and the SDNc.
   b) Southbound API provides an efficient controlling of the network devices and permits the SDNc dynamically to make modifications based on real-time demands and needs. It connects the SDNc with the real infrastructure devices of the network.
   c) East-West APIs define the communication of different controllers in the same domain or adjacent domains to interact with each other.
4) Forwarding layer represents the physical devices which forward the data packets according to the rules and actions that are sent by SDNc via southbound APIs to govern the flows of forwarding devices.

The OpenFlow is a well-known protocol that links the SDNc and network forwarding devices. The ONF standardized the OpenFlow protocol to be the significant southbound API which can be an open standard or user's proprietary. The switches and routers should support the OpenFlow protocol to transfer controlling information with the SDNc. The southbound APIs can be customized by the user to achieve an appropriate task [13], [17], [18].

### B. RELATED WORKS

Reducing the handover and mobility management delays in mobile networks have been researched by different proposals. Most of these proposals suggested modification either in hardware such as access points or software like protocols [19]. To review the related works, we focused on the papers and articles that were proposed the separation of IPv6 address into locator and identifier concept and binding this concept with SDN to present our idea. Shim6 is a host-based multihoming layer 3 protocol. It can provide more than one IPv6 addresses for each host. A host employs Shim6 can use more than one prefix of IPv6 addresses if the host has more than one interface for networks attachment points [20]. The HIP protocol proposed a new layer called the host identity layer. Host identity layer was inserted between layer 3 and layer 4 to identify the host. This layer maintains the mapping information of identifier and locator [5].

The MOFI protocol proposed a local Locator (local LOC) and Host ID (HID) for recognizing the locator and host. It proposed ID-LOC mapping to control HID-LOC information. The binding between the HIDs and LOCs was achieved by access router which has a hash table, an HID-LOC register, and local mapping controller [21]. The LISP-MN protocol supported node mobility. Two subsets of a standard are used called the Egress Tunnel Router (ETR) and Ingress Tunnel Router (ITR) functionality in an MN. A centralized mapping in LISP-MN was performed by a server that works as a mobility anchor for providing information on ID-LOC mapping. The shortcomings in LISP-MN are the double encapsulation required and triangle routing which caused problems of the path stretch [8]. Authors in [22], proposed an improvement to solve the problem of double encapsulation by localized the local LOC and Local Map Server (LMS) of mobility controller. All the protocols aforementioned are host-based. These protocols needed to modify the MN either by hardware or software. Besides, they were difficult to deploy on mobile networks. Moreover, they provided a single point of failure due to utilizing the centralized map server [23].

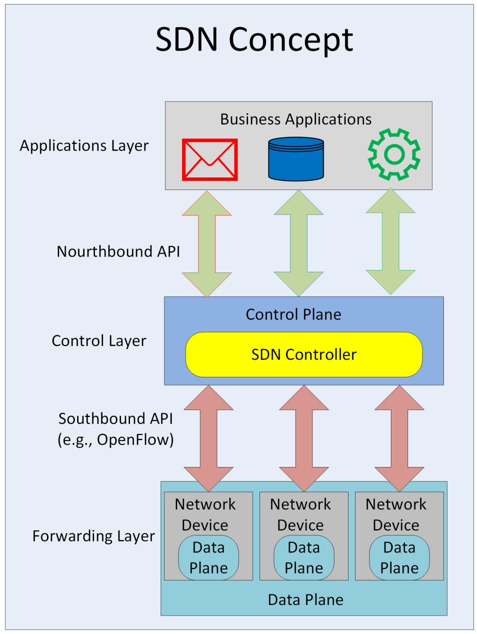


FIGURE 1: SDN Concept Architecture

The second group of protocols adopted the network-based method. The LISP protocol used a mechanism for alternating the IP addresses with two separated namespaces. First part is the Routing LOCator(RLOC), which was used by Internet network devices. The second part was the Endpoint ID (EID), which was used by the site service operators. The LISP has mapped EIDs to RLOCs through the Map-Resolvers and Map-Servers [24]. The LISP-AR-DMC protocol provided scalability and flexibility of packet routing. Also, it solved the LISP-MN protocol issues. The Access Routers (ARs) worked as a Tunnel Router (TR) functionality, which enables multicast communications that were used by the ARs for mapping the ID-LOC. The DHT based on a resolver LOC/ID, mapping approach has been suggested to solve the problem of the locator of a flat ID. DHT supposed every autonomous system manages the EID-LOC mapping information. It utilized a modified Content Addressable Network (CAN) which was applied "keys" onto "values" mapping [25]. Items are registered by the resolver to a CAN to refer that the EID-LOC mapping. However, all these protocols based host or network used encapsulation and tunneling which caused depleting bandwidth, increasing power consumption and demanding more processing due to overhead data.

In mobile communications, the location of an MN must determine geographically and topologically to maintain the connection continuity. The mobile IPv6 address can be split into two parts. The first part is called prefix ID which represents the Network Identifier (NID) or topology ID ( also known as locator ID) which consists of both the network and the subnet IDs. Prefix ID consists of 64 bits. The least significant bits (16 bits) of that 64 bits were assigned to subnets and known as Subnet Identifier (SID). The second part is the host location identifier which related to the IID or host identifier. In each MN the MAC address is bound

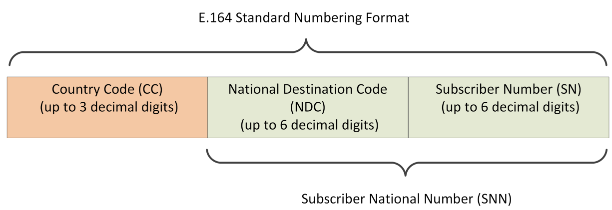


FIGURE 2: The Structure of the E.164 Standard

to each a specific interface of a network attachment point. Hence, all packets have to include a MAC address of the source and the destination. A packet never crosses to the network layer unless the MAC address is checked by the physical layer first. If the destination MAC address matched the conditions, then the physical layer will forward the packet to the upper layers, the packet will be dropped if there is no matching. This concept is the foundation of our proposed idea. The smart virtual eNB (SVeNB) is suitable to be the most candidate for our proposal. The SVeNB consists of several virtual machines (VMs), one of these VMs serves as an S/P-GW server, which supports the users within the coverage area of SVeNB to make a connection.

The E.164 standard is defined by the International Telecommunications Union for Telecommunications (ITU-T). The ITU-T defines the international public telecommunication numbering plans and telephone number formats. The E.164 standard numbering has a maximum of 15 decimal digits. The first (one to three digits) of the telephone number is the Country Code (CC), the second (up to six digits) is the National Destination Code (NDC), and the last part (six digits) is the Subscriber Number (SN). The SN and NDC together are called the Subscriber National Number (SNN) [26]. Figure 2 shows the structure of the E.164 standard.

## III. PROPOSED SYSTEM ARCHITECTURE

To give a clear view of the proposed idea, we illustrate how to generate a tag as a new identity for an MN and the entire proposed SDN network has been presented with details in the next Sections.

### A. GENERATING MOBILE NODE TAG

We suggested a novel *identity* for an MN consists of the Organizationally Unique Identifier (OUI) which is 24 bits and the Subscriber National Number (SNN). The SNN can cover all probabilities of the mobile subscriber numbers assigned to users. The likelihood of the most significant mobile number that can be formed by the SNN (12 decimal digits) when all the digits are 9s and that number can be represented over 40 bits. By combining the SNN and OUI, we can create a local identifier within the mobile operator networks. This identifier is 64 bits long, and it is compliant with the IPv6. Moreover, it can be used as an alternative to the IID part. Figure 3 shows the format of the generated tag.

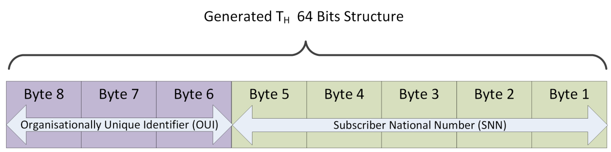


FIGURE 3: The Structure Format of the Generated Tag

Using the SNN and OUI as an MN ID is for security reason and to distinguish the MN within specific mobile operator networks. The created tag is the objective to achieve as a local identifier ($T_H$) for operator networks only, i.e., $T_H$ is well known for mobile operator network devices (SDN controllers, OpenFlow switches, and SVeNBs). The $T_H$ should be generated by the MN. We proposed an approach to creating the $T_H$ by combining the OUI part of the MAC address of the wireless attachment interface of the MN and the SNN. The generated $T_H$ inherits the global uniqueness feature from OUI and local uniqueness of the SNN.

The process of generating $T_H$ is executed by the MN once only. Each MN retains its generated $T_H$. When the MN moves amongst the SVeNBs, it uses its $T_H$ to contact with the new SVeNB as MN identity. The $T_H$ remains the same for that MN as long as it belongs to the same mobile operator networks. The $T_H$ must be regenerated if the SNN is changed. In this case, a new $T_H$ should be created by the MN with the new SNN. The generated tag is unique due to the uniqueness of both the OUI of the MN and the SNN in the domains of a specific mobile operator network. In other words, the $T_H$ is permanent for an MN even when it moves amongst the mobile operator network domains. Any router placed outside the domains of the mobile operator networks cannot utilize the $T_H$. Therefore, packets are routed by the standard IPv6 protocols on the Internet networks and other networks located outside the mobile operator domains.

### B. PROPOSED MOBILE NETWORK ARCHITECTURE

The proposed system architecture as shown in Figure 4 consists of two main parts. The first part represents the domains control layer. The second part is the domains themselves. Domains control layer comprises an Edge Software Defined Network controller (ESDNc) which controls the edge OpenFlow switch (EOFS) and other local SDNc (LSDNc)s and OpenFlow switches (OFS)s in each domain. Based on the information saved in the ESDNc lookup table about the links which connect all devices included in the network, the ESDNc dictates the transfer data flow from/to the mobile operator network through the EOFS and OFSs. According to that information, the ESDNc constructs and maintains its lookup table and makes the rules and actions, that are sent by the OpenFlow protocol to the EOFS and other LSDNc and OFS.

The primary task of the ESDNc is the mobility management of the MNs amongst the domains. Also, it directs the flows of the data traffic that enters the EOFS (notably, in case of packet handover). Besides to the ESDNc, this layer comprises the EOFS which acts as an edge point of aggregation and distribution of data streams from/to the OFSs in each domain. The responsibility of the EOFS is to forward the packets based on the rules and actions in its flow tables. These flow tables are built and modified by the ESDNc.

The suggested network architecture for each domain consists of the LSDNc, SVeNB, and at least one OFS. These domains are complementary with the most candidate 5G networks C-RAN architecture. The functions of the mobile network resources are virtualized to be hosted by the SVeNBs. In the beginning, each SVeNB declares its link address to the LSDNc to receive packets of their users. The following sections explain the idea behind this proposed architecture.

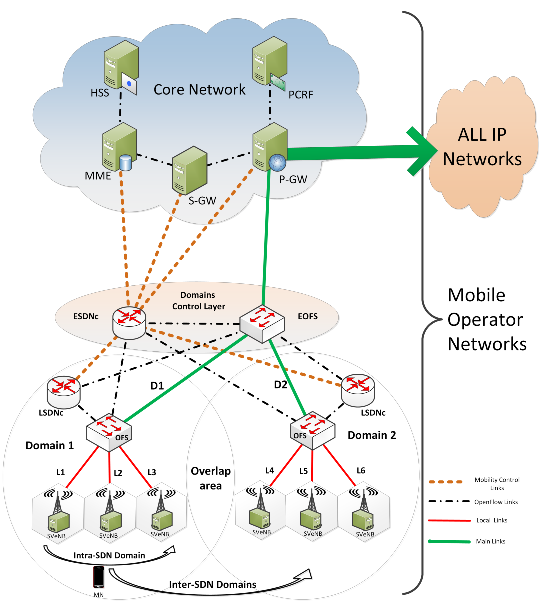


FIGURE 4: The System Architecture

TABLE 1: The ESDNc entries lookup table

| Main Link | Local Link | Prefix ID | MN ID Tags |
|---|---|---|---|
| D1 | L1 | 2001:0DB8:ACAD:0001::/64 | $T_{H_{i_1}, i=1\ldots N_1}$ |
| | L2 | 2001:0DB8:ACAD:0002::/64 | $T_{H_{i_2}, i=1\ldots N_2}$ |
| | L3 | 2001:0DB8:ACAD:0003::/64 | $T_{H_{i_3}, i=1\ldots N_3}$ |
| D2 | L4 | 2001:0DB8:ACAD:0004::/64 | $T_{H_{i_4}, i=1\ldots N_4}$ |
| | L5 | 2001:0DB8:ACAD:0005::/64 | $T_{H_{i_5}, i=1\ldots N_5}$ |
| | L6 | 2001:0DB8:ACAD:0006::/64 | $T_{H_{i_6}, i=1\ldots N_6}$ |

#### 1) Domains Control Layer

This layer contains the ESDNc and EOFS devices. The ESDNc connects to the mobile operator core network, EOFS, OFSs, and LSDNcs in each domain. The principal duties of the ESDNc are the packet steering during the handover and mobility management of the MN among the domains and filtering the data traffic. The ESDNc receives the information from Mobility Management Entity (MME), Serving Gateway (S-GW), and Packet Gateway (P-GW) of the Core Network (CN), LSDNcs, EOFS, and OFSs to update its lookup tables. According to that information, the ESDNc makes decisions (rules and actions) and sends these decisions to EOFS, OFSs, and LSDNcs. The second device in this layer is EOFS which receives the data traffic from the P-GW server of the CN. The EOFS tests the incoming packets with the entries of its flow tables to forward the flow to the destination target. .

The ESDNc is the central brain of made forwarding decisions in the domains control layer. It is responsible for making the decisions that control the EOFS to guide the data forwarding by specifying the main links ($D1$ or $D2$), which connect the EOFS to the OFSs in domain 1 or 2 respectively. Table 1 illustrates the contents of the lookup table of ESDNc such as main links, routing prefix ID, local links and $T_H$s. $i$ represents *ith* tag of the $MN_i$ that belongs to link *ith* and $N$ represents the number of users at each link.

#### 2) Domain Entities Task And Function

In order to illustrate the roles of each entity in the domain, we discuss every entity's tasks and what functions can be executed by it.

- The SVeNB [16] represents as a macrocell base station. It serves as a virtual eNB with the ability to host several VMs to mimic the functionalities of a mobile operator core network entities. These entities such as MME, S-GW, P-GW are virtualized by a multi VMs into the SVeNBs. The virtual MME (vMME) which serves as a local vMME entity, virtual serving/packet (vSP-GW) which implements as local virtual serving/packet gateway and so on. The VMs partially perform the functions of the core network, due to the profiles of users were manipulated and updated in the core network and sent to the SVeNB. Therefore, the VMs can serve the local users that are covered by SVeNB without contacting the core network again when an MN tries to make a connection with another MN within the coverage area of an SVeNB [16]. To move from one SVeNB to another (i.e., from one subnet to another) the vMME and vSP-GW VMs play the primary role to bind the IP address of an MN and determine its location. The VMs of SVeNB can achieve these tasks. In other words, the SVeNB can tie an MN's IPv6 address (prefix and EUI-64) with $T_H$ in the routing table which is built by the vSP-GW in the SVeNB as shown in Figure 5.

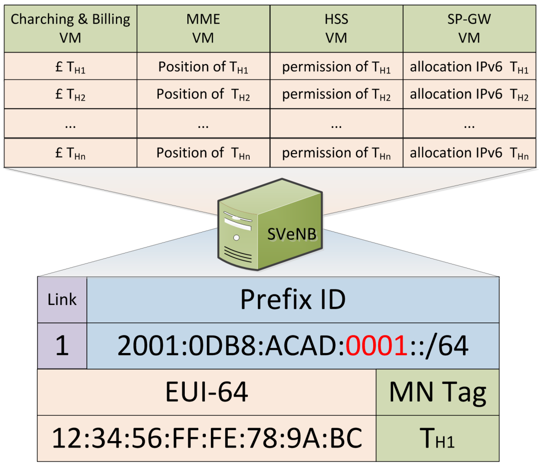


FIGURE 5: SVeNB VMs and Routing Table Based on the $T_H$

- The OFSs steer the packet forwarding within a domain. The domain could have more than one switch. We assumed using a single OFS for every area in Figure 4 for easy understanding. The OFS requests the forwarding decisions from the LSDNc to direct the packet flows to the target SVeNB. Then these decisions are cached in its forward table for a given time at the forwarding device. The OFS could contain more than one flow tables, which consist of forwarding entries. These entries restrict how the packet will be rerouted and processed according to the records of the flow table.
  The typical entries of the flow table are *(1) matching rules, or match fields* contain information to be matched with those in the header of arrived packets, metadata, and ingress port. *(2) counters* collect the statistics such as the number of bytes, the number of arrived packets, and the period of a certain flow. *(3) actions* apply a set of instructions on the received packets by the OFS to dictate how to forward the matching data [27].
- The LSDNc is the main brain of forwarding decisions of packets that are incoming the domain. It is responsible for making the decisions to the OFS to forward the data to a specific SVeNB which is serving the destination MN. These taken decisions are based on the lookup table entries that were saved and updated by the LSDNc. The lookup table consists of the SID, the $T_H$, and the link which represents the port ID of a specific SVeNB. Table 2 shows the entries of the lookup table. The subnet ID part (*16 bits with red color*) represents the local link topology of the mobile operator's networks (domains).
  The connection between the LSDNc and OFS utilizes the OpenFlow protocol [28]. By using the OpenFlow protocol, the LSDNc can add, remove, or update flow entries of the OFS flow tables to support the MN to receive packets when it moves amongst the SVeNBs belong to a domain.

TABLE 2: The LSDNc entries lookup table

| Local Link | Prefix ID | MN ID Tags |
|---|---|---|
| L1 | 2001:0DB8:ACAD:0001::/64 | $T_{H_{i_1}}, i=1 \ldots N_1$ |
| L2 | 2001:0DB8:ACAD:0002::/64 | $T_{H_{i_2}}, i=1 \ldots N_2$ |
| L3 | 2001:0DB8:ACAD:0003::/64 | $T_{H_{i_3}}, i=1 \ldots N_3$ |

### C. MOBILITY MANAGEMENT

The most significant feature of the SDNc is the ability to manage the mobility for each flow [29]. This feature enables the forwarding, load balancing, and packets handover in both intra-domain and inter-domain. Movement of an MN from a one SVeNB to another or from one domain to another, this movement needs the new attachment point to receive the information of the MN from the old SVeNB or from the MN itself which involves in the handover. The SDNcs, OFSs, and the SVeNBs should establish new binding tables depending on this information. Figure 4 shows the mobility of an MN into the intra-SDN domain and inter-SDN domains respectively.

#### 1) Inter-SDN Domains Mobility Management

The domains control layer is regarded as the first line to filter incoming packets from the CN due to the operation of checking which is done by the ESDNc and EOFS. This filtering process can be considered as packets sifting. Figure 6 illustrates the proposed procedures which are implemented by EOFS and ESDNc, i.e., domains control layer. ESDNc can manage the horizontal handover (as defined in Section III-D) amongst the domains. This approach begins when the MN enters the overlap area, i.e., passes from the hosted domain to a new one. Horizontal handover begins when the MN detects the signal of the visited SVeNB at the overlap area to register to the new SVeNB and sends its $T_H$. At the same time, the MN keep the connection with the previous SVeNB to send and receive data. The new SVeNB binds the $T_H$ in its routing tables and triggers the $T_H$ to the LSDNc which in turn sends the $T_H$ to the ESDNc to modify forwarding tables of the EOFS. The EOFS sends the last packet on the previous link until the EOFS executes the modification on its flow table.

#### 2) Intra-SDN Domain Mobility Management

When an MN moves from one SVeNB to another within one domain, it should declare its $T_H$ to the new SVeNB which requests the profile of that MN either from the old SVeNB or from the mobile operator CN. The new SVeNB advertises the $T_H$ of the MN to the LSDNc to bind with its prefix ID and link. Figure 7 shows the mobility management by the

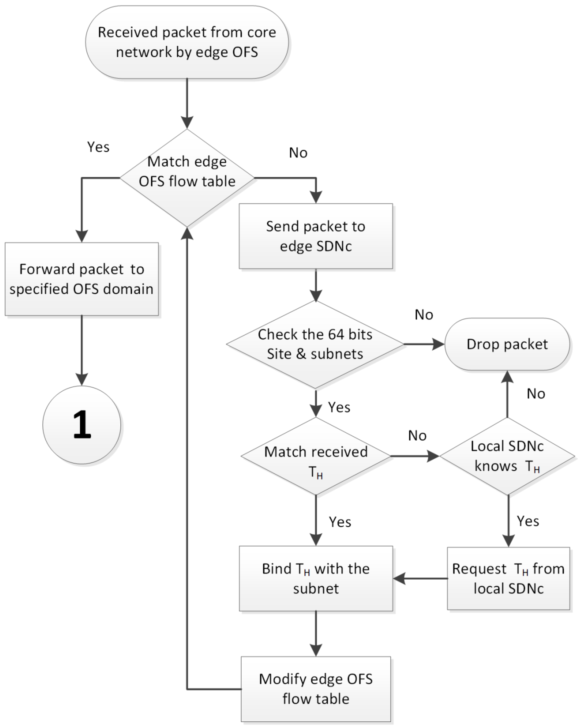


FIGURE 6: Flowchart of Forwarding Packet in Domains Control Layer.

LSDNc. The packet is received by the OFS which checks its flow table to forward the packet. If the OFS finds a match for that packet, it immediately sends it to the target SVeNB. If the OFS does not find a match, then it forwards that packet ( step 1) to the LSDNc. The decision regarding that packet is replied by the LSDNc (step 2) whether modification the flow entries or dropping that packet.

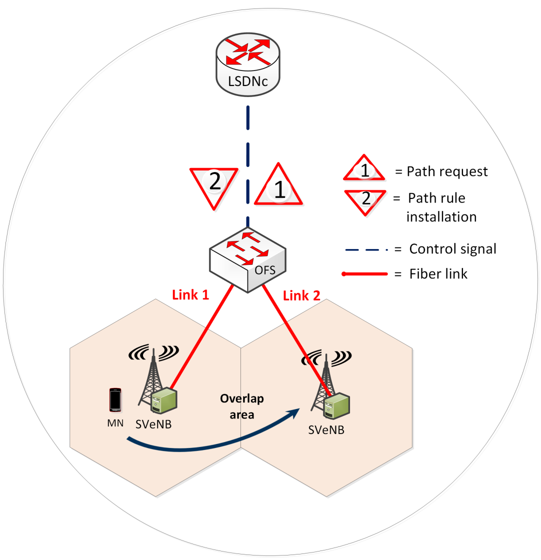


FIGURE 7: Intra-SDN Domain Mobility

- In the beginning, the MN generates its $T_H$ through the procedure mentioned in Section III-A. The MN requests a radio frequency bearer after it detects the signal of the SVeNB. The VMs that were installed into the SVeNB manipulates the session establishment. This process is known as an access stratum connection. Meanwhile, the MN sends its $T_H$ to the SVeNB that covers the MN. A standout of the majority advantages of using IPv6 is its capability with auto-configuration addressing. An MN can configure its IPv6 address according to the link-local prefix ID for each interface. This procedure is known as a stateless auto-configuration IPv6 creation which depends on the IID of an MN's EUI-64 (based on MAC address) and link prefix to form a global or local address [30]. The installed VMs (P/S-GW) into SVeNB bind $T_H$ with the IPv6 address of the MN into the forwarding or routing table which is used by the SVeNB to deliver the packets to the MN. Also, this table is used by the vMME to locate the position of that MN. Figure 8 shows the messages between the MN and SVeNB for establishing the connection. The necessary

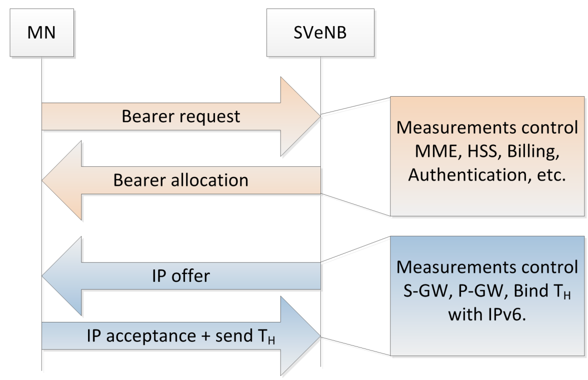


FIGURE 8: The Messages Between MN and SVeNB

measurements such as authentication, mobility, user service permissions, and other measurments should be accomplished by the installed VMs into the SVeNB

- After finishing the MN its registration by the SVeNB, the $T_H$ is directly sent to the LSDNc to update its lookup table. Then the LSDNc sends the new rules and actions to the OFS which updates its forwarding table. At the same time, it triggers the received $T_H$ to the ESDNc to know the new locator (hosted subnet) of the MN. Figure 5 illustrates the installed VMs, the profiles of the users, and routing table based on the $T_H$. The SVeNB plays a significant role in connecting all users that are under the coverage area of that SVeNB. Besides, it receives the data from the source through the OFS to deliver it to the destination MN. The vMME maintains the mobility of the MN within the coverage zone, i.e., it controls the local movement of all users under the tent of the SVeNB.
- The LSDNc receives packets that should contain SVeNB prefix ID (locator) and the MN host ID ($T_H$) from the attached SVeNB. These packets are used by the LSDNc to update its lookup table and maintain

the mobility of that MN. In other words, the locator represents the subnetting topology as well as indicates to the geographic place of the SVeNB, which is already identified by the LSDNc, and the $T_H$ represents the position of the MN that should be bound with that locator to be known by LSDNc. These positions awareness can consider as domain mobility management, due to determination the topology identifier (prefix ID) and the MN identifier ($T_H$), which represent as the locator and position of the MN respectively. The algorithm 1 illustrates the procedures that are taken by the LSDNc to make decisions and update its lookup table. Figure 9 illustrates the proposed checking and processing in the SDN domain.

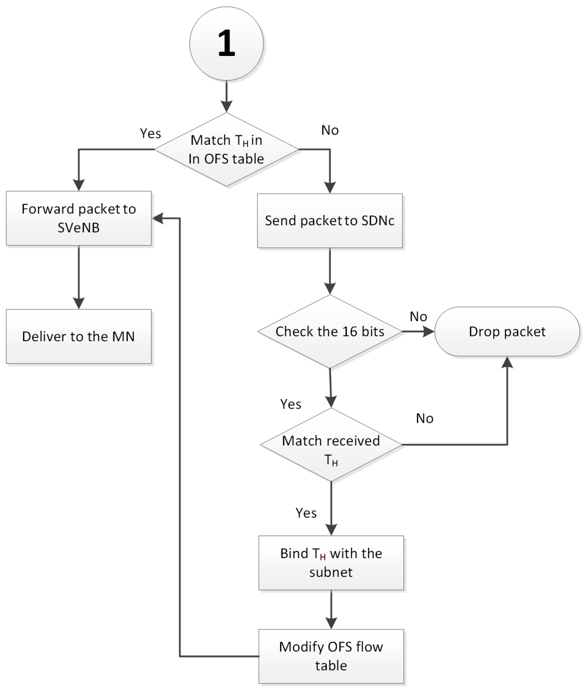


FIGURE 9: Flowchart of Forwarding Packet in SDN Domain

### D. HANDOVER PROCEDURE

The mobility management has emerged to solve the roaming problems of the MN among wireless mobile networks. Additionally, it preserves the continuity of the MN connection, when it alters the attachment point to a new network, this is called the handover management. Furthermore, mobility management enables the MN to receive packets from serving networks at different access points of the network attachments, this is known as location management. [31]. There are two kinds of handover the first one is the vertical handover which means, the MN can connect to different technologies of wireless access points. For example, WiFi, WiMAX, LTE, etc. The vertical handover can be done within one geographic region has a diversity of wireless coverage connectivity. The second type is the horizontal handover which refers to the MN when it moves within the same technology in different geographic places. Each IPv6 address carries a network identifier (prefix ID) which consists of 64 bits of the IP address and a host identifier or an interface identifier (IID) which consists of the other 64 bits of the IPv6 address. The prefix ID has topological importance due to the routers use the prefix ID to forward the packets among different networks, i.e., at the network layer, while the IID is topologically important at the target subnet to delivering a data to the MN belongs to that subnet. In our proposal the $T_H$ is equivalent to the IID indicates the position of the MN at a specific subnet [32]. Our system has proposed using the $T_H$ within the domains control layer and the domains themselves only. As the communication between the MN and its sender node uses the standard IPv6 to keep receiving and sending packets from/to backbone networks, the sender node is not aware of the MN's location and what standard of IP address was used by the MN as well. Consequently, the continuous and uninterrupted connection leads to a seamless and very low handover delay, also to almost zero packet loss rate. Figure 10 shows the links type according to use the standard or non-standard IPv6 routing schemes. The handover between the domains starts at the ESDNc after receiving information about the new binding of the $T_H$ and the visited subnet from the LSDNc. The ESDNc makes modifications and changes on its routing flow tables. These changes and modifications are sent to the EOFS to change the exit link from $D1$ to $D2$ to forward packets as shown in Figure 10. That happens when the MN enters the overlap area and after registering to the visited domain.

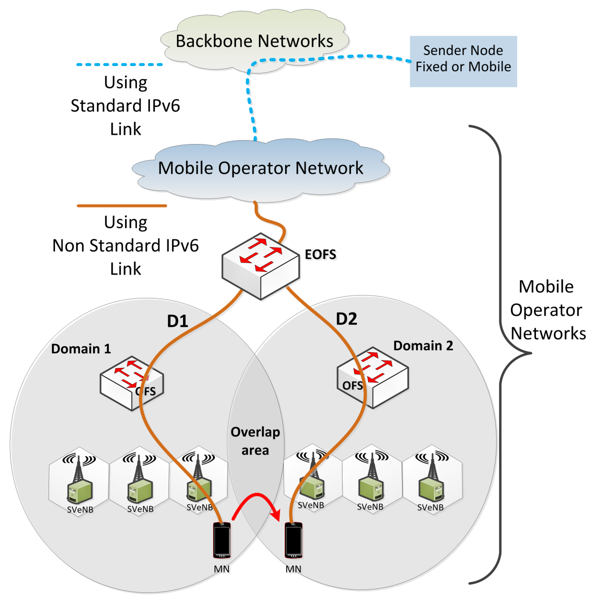


FIGURE 10: The Link Type Based on Using Standard or Non Standard IPv6

#### 1) Handover Delay

Currently, packets pass via the HA and FA to deliver the data between the MN and sender node. That needs more procedures, such as create tunneling between the MN and the sender node to keep the connection continuity. The longest delay time happens for the packets handover is due to the processing procedures to modify the of the packets. If we consider in the C-RAN system, there are many physical servers each one in charge to achieve a specific task, and some of these servers located in one geographic area, and the other located away from each other. Moreover, some servers depend on the decisions of the other to complete its task. In both cases, all servers should process incoming data also process the preparation to send that data. Furthermore, queuing delay which depends on the amount of data on the link that transfers the data between two points, also depends on the hardware specifications of the servers.

### 2) Packets Path Decision Delay

The proposed system suggests using non-standard IPv6 routing scheme to forward packets through the system is based on the separation of the CP from the DP. This feature of separation is supported by using the OpenFlow protocol [29]. The MN declares its $T_H$ to the SVeNB after acquiring prefix ID from it, which updates and binds the information of that MN in SVeBV's routing table. In the same time, the SVeNB advertises the $T_H$ to the LSDNc to updates its lookup table and sends the modified entries of the flow table to the OFS. There are two probabilities for the received packet by the LSDNc. Firstly, the received packet already has been bound with the subnet and the link in the lookup table. In this case, the decisions are sent to the OFS to forward that data and keep that *matching rules* and *applying actions* for all packets which match that rules. Secondly, the packet is received by the LSDNc for the first time; it will check the subnet of the prefix ID *(only the 16 bits)*. This checking is executed by the LSDNc to know if this packet belongs to one of its subnets or not. If the answer was yes, the LSDNc scans its lookup table to see whether the $T_H$ within that subnet or not. If the answer was no, then there are two likelihoods, the first one is the LSDNc will send a request to the SVeNB and ESDNc about that $T_H$ to making a decision concern it. The second possibility is that dropping all packets for which the LSDNc, SVeNB or ESDNc are not known about their $T_H$. In other words, the LSDNc drops all packets that are not matching or that are not known their $T_H$ or subnet by SVeNB, ESDNc, and LSDNc. Algorithm 1 shows the procedures have adopted to forward packets within the proposed system.

**Algorithm 1:** ESDNc and LSDNc Algorithm to Make a Flow Decision

```
1  Packets received from th core network
2  if Received packet prefix (64 bits) match then
3      if T_H bound with main link then
4          Send packet through the specific main link
5      else
6          Request T_H from LSDNc
7          Update tables of ESDNc and EOFS
8          if Subnet (16 bits) match then
9              if T_H belongs to a subnet then
10                 Send to SVeNB belongs to that subnet
11                 Deliver to the MN
12             else
13                 Request T_H from SVeNB
14                 Update tables of LSDNc and OFS
15                 Send trigger of T_H to ESDNc
16                 Go to step 7
17             end
18         else
19             Drop packet
20         end
21     end
22 end
```

### 3) Processing and Queuing Delay

To calculate the packets handover delay we need to determine the processing and queuing delays for each server. Suppose every server achieves one task. Consider the queuing of the proposed system is M/M/1 with Poisson process. Let $\mathcal{S}$ is the number of servers, $\mu$ is the packet transmission rate of the control messages and $\lambda$ is the Poisson arrival process rate (packet/sec) at each server which can provide a traffic load as follows.

$$\rho_i = \frac{\lambda_i}{\mu_i}, \qquad i = 1, 2, ..., \mathcal{S} \tag{1}$$

where, $\rho_i$ and $\lambda_i$ are the utilization factor and arrival rate of $ith$ server respectively [33]. The total delay of the expected queuing equals to the summation of the expected queues at every server. So, it is expressed as.

$$E[X] = \sum_{i=1}^{\mathcal{S}} E_i \left[\frac{1}{\mu_i}\right] \tag{2}$$

where, $X$ is the service delay of a server and equals to $1/\mu$, and based on the first-come-first-serve, the inter-arrival times, the service times are independent, and by using Markov chain then the probability ($Prob_{cm}$) of control message packets be in the queue is.

$$Prob_{cm} = \rho^B(1-\rho) \tag{3}$$

where, $B$ is the number of the control messages that transferred in a channel, and $\rho = \lambda/\mu$, from this we can get $B = \rho/(1-\rho)$. So the delay for each server will be.

$$D_{qs} = \frac{B}{\lambda} = \frac{\rho}{\lambda(1-\rho)} = \frac{1}{(\mu-\lambda)} \tag{4}$$

Then the overall delay due to the queuing in C-RAN could be given as follows.

$$D_{qT_t} = \sum_{i=1}^{\mathcal{S}} \frac{1}{(\mu_i - \lambda_i)} \tag{5}$$

whereas for the proposed scheme, we adopted the queuing system of M/M/m [34], and the equation of the total queuing delay will be.

$$D_{qProp_t} = \sum_{i=1}^{\mathcal{D}} \frac{1}{(m\mu_i - \lambda_i)} \quad (6)$$

$\mathcal{D}$ is the number of physical servers in proposed architecture, and $m$ is the number of performed tasks simultaneously by the $ith$ server . The queuing delay time depends on several physical parameters including a transmission line capacity and the specifications of server components. We named the processing delay for all elements that cause delays due to processing the data. Assume the packet length is $L_p$, the machine word size is $W_m$, the arrived word size is $W_a$, the number of the packet in each control message is $P_{cm}$, and the CPU architecture of a server (e.g.,32 or 64 bit) is $CPU_x$. Besides, the lookup delay of memory access is assumed almost 100 nsec [16]. Hence the processing delay equation can be modified and written as follows.

$$D_{ps} = 100\frac{W_a}{W_s} \times \left[log_{sys}P_{cm} + \frac{L_p}{CPU_x}\right] \quad (7)$$

Equation 7 is used for finding the processing delay of one physical server. Thus, the total delay for all the servers in the traditional system is the summation of the individual delay for each server that involved in making decisions for packets routing.

$$D_{pt} = \sum_{i=1}^{\mathcal{S}}\sum_{u=1}^{U} 100\frac{W_{aiu}}{W_{si}} \times \left[log_{sys}P_{cmiu} + \frac{L_{p_u}}{CPU_{xi}}\right] \quad (8)$$

where, $\mathcal{S}$ is the number of servers, $U$ is the number of control messages of the user $uth$ to be processed by the specific $ith$ server, and $sys$ is the system constant.

In our proposed scheme, the number of physical servers is much lower than in the C-RAN system due to the installed VMs in the SVeNBs, which perform the functions of real servers of the CN to support the CP for any communication. Moreover, using SDNc and OFS give the ability to decrease the number of control messages of each packet flow. These essential pros can be observed through our proposal for enhancing the packet handover process and reducing the handover delay time. Figure 11 shows the results of the packet loss probability versus increasing in the number of users. From the figure, the traditional system suffered from a higher rate of packet loss than the SDN system. In the traditional scheme, the network devices execute steps of the queuing, processing, decapsulation, and encapsulation on every packet enters these devices to determine its destination. Whereas in the SDN network packet forwarding process is executed by the SDNc and the data is sent through the OpenFlow switch as flows. Each flow consists of a set of data packets. These flows are directed based on the decisions of the forwarding process, which is done only on the first packet of every flow. In other words, all the packets of a flow track the first

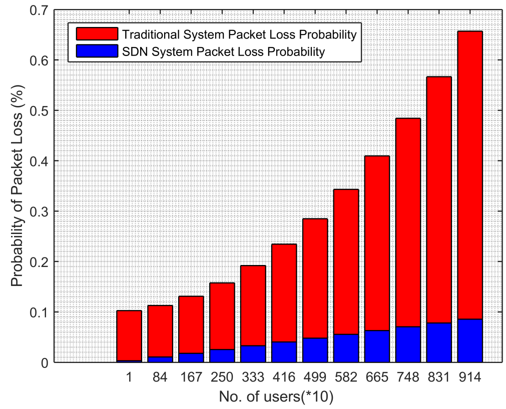


FIGURE 11: The Probability of Packet Loss

packet to reach their destination. This leads to keeping the probability of packet loss rate in the SDN network almost in range of one-seventh of that in the traditional system.

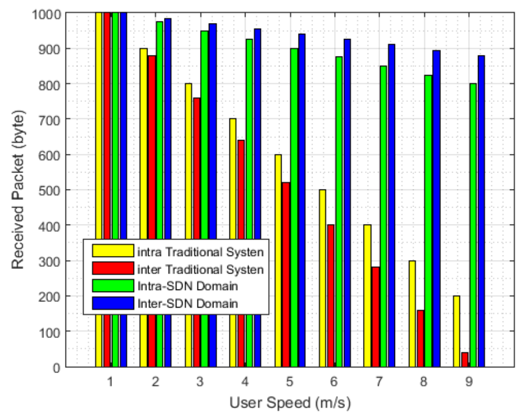


FIGURE 12: The Received Packets During MN Movement

Figure 12 shows the results of the received packets during the mobility of the MN between network attachment points. As shown in Figure 12 when the MN moved slowly, the performance of both systems was a high, due to both systems considered the MN as a fixed node. Accordingly, the MN could receive packets with the minimum probability of a packet loss. That is, the data packet did not need to change or modify its route for enough time to be accurately directed during the MN's slow movement. Thus, the lost packets are at a lower rate, while the packet loss increases to the highest value when the MN moves quickly between the network access points. At increasing the MN's speed, the proposed scheme recorded a higher performance than the traditional system. Moreover, the intra-SDN domain handover scenario performed higher rate in receiving of packets than the inter-SDN domains handover scenario. This is because of the LSDNc handled the packets within one domain. However, in the case of inter-SDN domains, the performance was slightly less than of the intra-SDN domain. This degradation in performance is due to the packets have been directed by the ESDNc and LSDNc. In general, the performance of both intra-SDN and inter-SDN domains are higher than of the traditional method when increasing the MN's speed. Considering more than one network devices govern the packet forwarding in the traditional scheme and each device makes its forwarding decisions independently (each device gathered the CP and DP processing).

## IV. COMPARISON WITH OTHER SCHEMES

The similarity and difference features between the suggested scheme and other schemes have been summarized by Table 3. We can recognize from Table 3 that the unique and shared points of our proposed and the other schemes. Host-based requires to amend the protocol stack of a host, so it leads to more cost and deployment problem [24]. Whereas, network-based does not need much modification in a protocol stack. Therefore, it is more acceptable and cost-efficient of engaging with SDN environments.

The centralized management suffers from the traffic burden, single point of failure, and the centralized mapping, to overcome these obstacles there are two choices. The first one either by adding more devices with super specifications to contribute in data processing and this leads to the high cost or distributing the services amongst more than one device with reasonable specifications to afford the new devices' cost. The second choice by utilizing SDN technology to distribute the tasks between two or more devices support the SDN-enabled technique with reasonable specifications. The data processing delay is decreased by the SDN technology, due to the jobs are treated in parallel at the same time. Particularly, when separate the DP from the CP into separate devices (i.e., flow forwarding and flow decision maker respectively) [15].

Our proposed system provides new features that collaborate with SDN technology in order to reduce the End-to-End delay, the packet loss, and low handover latency through:

1) Depending on decoupling the CP from the DP in forwarding and directing packets within an SDN mobile network has been used.
2) Maintaining mobility per flow or per packet to carry data packets to an MN has been adopted.
3) Using direct forwarding to the MN based on the generated $T_H$ instead of standard IP routing scheme has been implemented.
4) Performing the flow forwarding instead of the routing or switching mechanism.

The above are the substantial differences between our proposed scheme and the other schemes.

## V. SIMULATION AND PERFORMANCE EVALUATION

### A. MININET SIMULATOR

To implement and perform the proposed scheme, the Mininet simulation has been used to create the network. It is a network emulator to simulate the functions of the network devices (servers, routers, switches, hosts, and links). Also, it is a great tool to work on underly the open sources software such as SDN, NFV, and systems virtualization. Moreover, it can be used by researchers to design a virtual network, that has the same properties and performance of the real network elements and could be run by a single physical or virtual machine. Mininet allows creating custom topologies and gives the ability to create and configure controllers, switches, and hosts through:

- Interactive user interface (IUI).
- Command line interface (CLI).
- Programming Languages such as Java, Python, etc.

The Mininet simulator has been used to implement our SDN network system. The simulation scenario consisted of three OpenFlow enabled switches (EOFS, OFS1, OFS2), and two sets, each set with three hosts. The first set connected with OFS1 and the second set connected with OFS2. The hosts represent the SVeNBs to mimic the stationary parts of the mobile network. The Python language has been used to configure the APIs of the simulation scenario. Figures 13

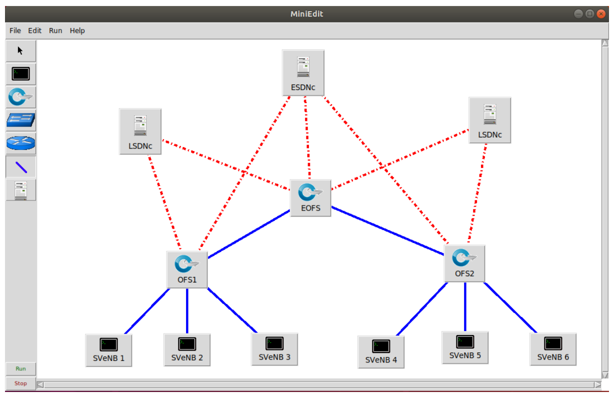


FIGURE 13: The Mininet Proposed Scenario Setup

and 14 present the setup of the proposed network and its execution under Linux operating system.

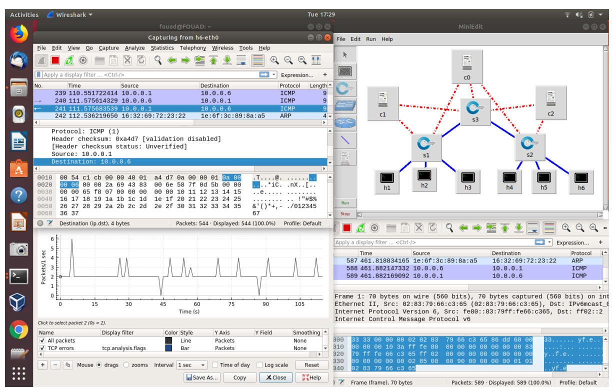

FIGURE 14: The Proposed Scenario Execution

### B. PERFORMANCE EVALUATION

The performance evaluation comparisons of the packets handover, address mapping, links switch, packet loss, and

TABLE 3: Comparison Between The Proposed Scheme And Other Protocols Scheme

| Protocols | HIP | Shim6 | LISP-MN-Local | MOFI | LISP-AR-DMC | DHT-MAP | Proposed Scheme |
|---|---|---|---|---|---|---|---|
| Centric Type | Host-based | Host-based | Host-based | Host-based | Network-based | Network-based | Network-based |
| Mapping Type | Centralized | Centralized | Centralized | Distributed | Distributed | Distributed | Distributed |
| Management Manner | Rendezvous Server | DNS Server | LMS | LMCs | LMS | Rendezvous | SDNc |
| Decoupling CP & DP | No | No | No | No | No | No | Yes |
| Deployment Cost | High | High | High | High | Low | Low | Low |
| Packet Forwarding | Tunneling | Tunneling | Tunneling | Tunneling | Tunneling | Tunneling | Based-Fow Table |
| Direct ID Forwarding | No | No | No | No | No | No | Based-$T_H$ |
| Dispatch Path | Routing | Routing | Routing | Routing | Routing | Routing | Flow Forwarding |

TABLE 4: The System Parameters

| Parameters | Value |
|---|---|
| No. of Users | 10k |
| No. of Packets | 1000 |
| User Speed | Up to 9 m/sec |
| Packet Size | 1522 Byte |
| Max. Control Messages | 120 message |
| Min. Control Message Size | 50 Byte |
| Delay Per Link | 0.1 msec |
| Processing Delay | 0.05 msec |
| No. of Re-directions | 8 |
| OFS Modifying Delay | 0.005 msec |
| SDN Modifying Delay | 0.001 msec |
| Virtual S/P-GW Delay | 0.001 msec |
| Virtual MME Delay | 0.001 msec |
| RF Intra Registration | 1 sec |
| RF Inter Registration | 2 sec |
| Simulation Rounds | 12000 |

processing delay of CP consideration for both traditional (C-RAN) and the proposed systems. The MATLAB platform was used to collect the datasets which were prepared and pre-processed for implementing in this system. Additionally, it has been used to test the performance of the proposed algorithms. Table 4 contains the relevant simulation's parameters. The extracted data has been injected to evaluate the performance measurements of our system behavior.

Figure 15 shows the delay time difference between the proposed and traditional schemes for setting up a flow path connection between the MN and the CN. From the results presented in the figure, the conventional scheme network devices need more time for queuing and processing to determine a path for the arrived packets. This delay time is replicated for each data packet to determine its destination. While in our proposal, the flow path determination demands much less time for path resolution due to making decisions having been executed in one or two servers each one with multiple VMs. For example, the Figure shows that at 200 packets, the delay is 0.182 ms in the proposed system, while for the same amount of data in the traditional scheme is 0.2 ms. The delay time at 1000 packets is 0.62 ms, however for the similar amount of data in the traditional system scheme, the delay time is 1.4 ms. It is clear that the delay time is lesser in the proposed system than that in the traditional schemes. This reduction in the delay is due to the parallel processing that has been applied by the VMs on the arrived packets in our proposal.

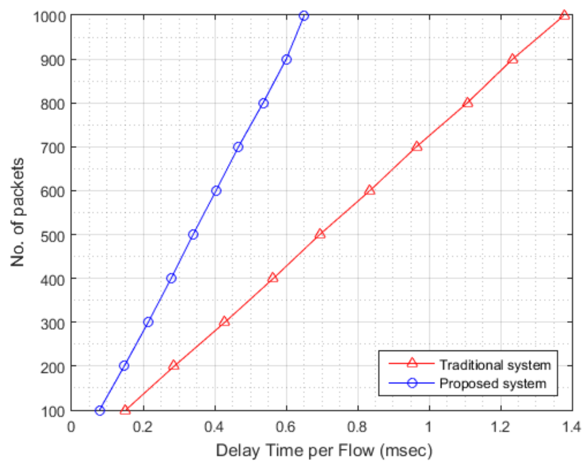


FIGURE 15: The Delay Time Comparison Between SDN And Traditional of Forwarding And Routing Schemes Respectively

Figure 16 illustrates the addresses of mapping delay versus the control messages. Each control message contains many packets. In the traditional system, some control messages can be considered as $MasterMessages$ are generated and sent by a server to another server as a complementary control message as $SlaveMessage$. Therefore, the line graph of the traditional system is exponential, and the delay grows by increasing the number of control messages. However, in the proposed scheme the same messages $MasterMessage$ and $SlaveMessage$ could be processed in parallel into the same physical server through the several VMs. This method leads to a decrease in the required time to build of the addresses mapping tables. Also, Figure 16 illustrates the SDN system could build its addresses mapping tables in less than one-tenth of the time needed by the traditional networks for the same number of the control messages and the number of users.

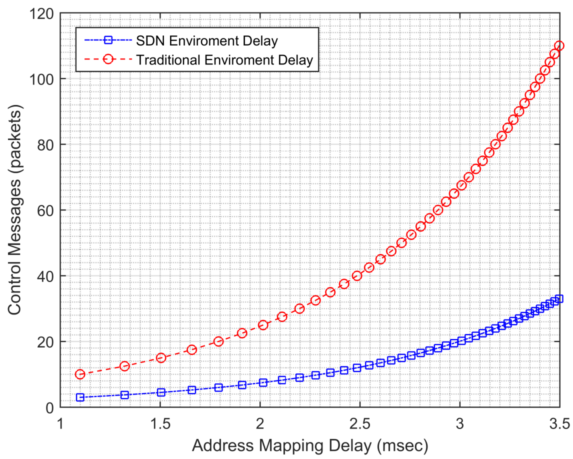


FIGURE 16: The Delay Time to Create Addresses Mapping Tables

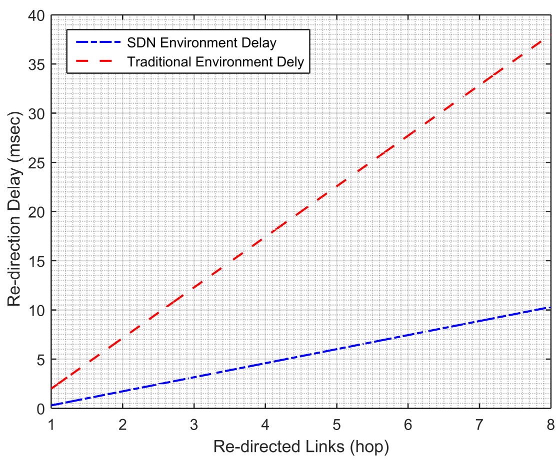


FIGURE 17: The Delay of Links Switch

Figure 17 represents the delay time of the packet flow re-direction (number of hops or routers between the sender and the receiver) that should the packets pass through them to reach the target MN. As shown in the figure, the SDN environment needed almost 12.5% of control messages that were required by the traditional schemes. That because the link switch mechanism depends on tagging which was used by the proposed system. At the instant of receiving the $T_H$ by the ESDNc, the lookup table entries will be amended by the ESDNc according to the location of the MN. After that, the modified entries are sent by the ESDNc to the EOFS to change the packet flow from $D1$ to $D2$. These links connect the EOFS to the OFSs in domain 1 and domain 2, as shown in Figure 17.

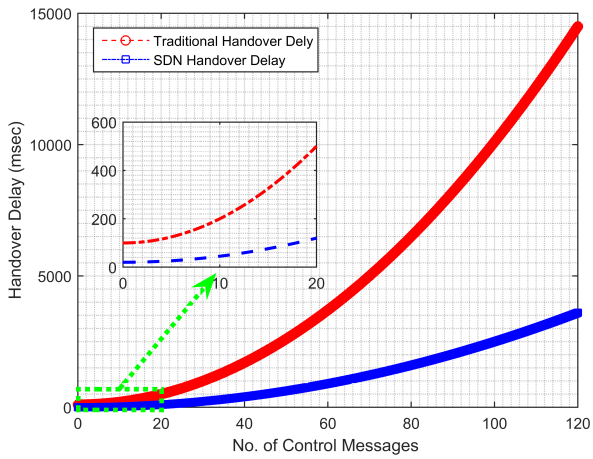


FIGURE 18: The Overall Delay of Handover And Required Control Messages

Packets re-directing process delay plays a vital role in the session continuity of an MN connection. The registration, getting the CoA, and the tunneling are the main parameters cause the packets handover delay in mobile networks. From Figure 18 we can see that the packets handover in the proposed scheme needed fewer control messages to make decisions and change the flow paths of data packets. While in the traditional system that took more processing and time, thus led to losing packets during the handover.

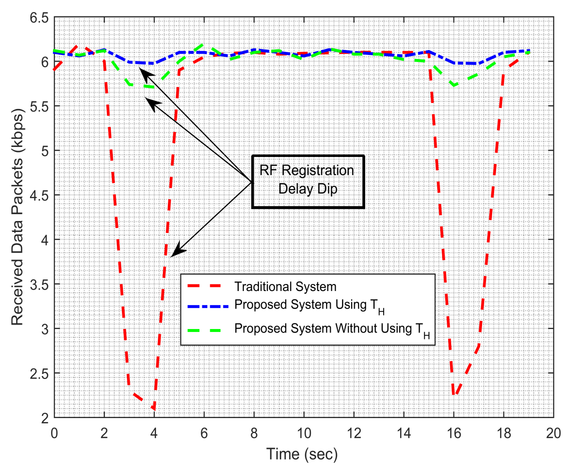


FIGURE 19: The Received Data During The Handover Procedure

Figure 19 shows the received data during the packets handover processing in both the traditional and the proposed systems. From Figure 19, we can observe that the proposed system kept the average of received data almost at the high level during the packets handover procedure. Whereas, the obtained data dropped to the lower amount in the traditional system during the packets re-directing. The results as mentioned earlier were based on the assumption of an MN moves at the same speed in both states (proposed and traditional systems). Moreover, the maximum values of received packets in the proposed network were almost 5.98 kb (when the SDN network used the $T_H$) and 5.7 kb (when the SDN did not use the $T_H$) of 6.2 kb of the unbuffered transmitted live stream respectively. In contrast, the minimum values of the received packets in the traditional system were around 2.25 kb of 6.2 kb of the unbuffered transmitted of the live stream, this means, the MN could receive 96.4% and 91.9% of the packets that have transmitted during the handover. This improvement was achieved by the SDN network based on the proposed scheme for packet forwarding and re-directing mechanism with the support of $T_H$. Whereas, the traditional packets routing mechanism the percentage of the received packets during the handover process period is 36.3%. This low percentage is due to the packets re-directing process into each device, which has made forwarding decisions for every received packet by it. This procedure leads to losing the packets during the handover processing period. Also, the figure shows the proposed scheme could retrieve almost three times of that lost packets in the traditional system with neglecting the RF registration delay time for both systems.

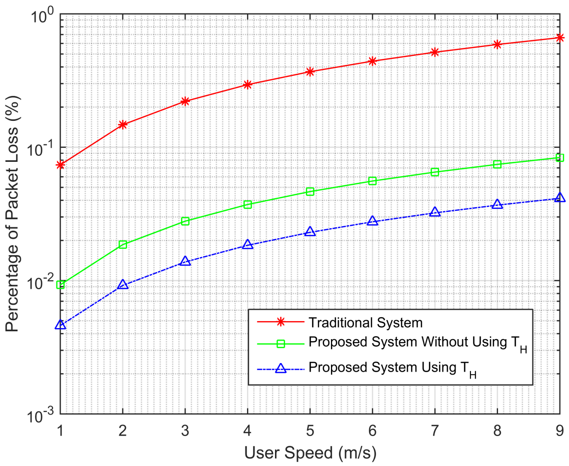


FIGURE 20: Percentage of Average Packet Loss During Handover Process

Figure 20 indicates the percentage of average packet loss against the MN speed. As expected, the SDN networks overcome the conventional networks in reducing the values of packet loss for unbuffered streams. This reduction in lost packets rate is due to the decrease in the required processing time to forward and re-direct packets into the SDN network (CP messages exchange). The performance of the SDN network has been enhanced by using the $T_H$, as shown in the Figure 20, where the lowest value of lost packets was around 4% through using the proposed scheme with supporting of $T_H$ and almost 8% without using the $T_H$ in SDN network, while the value of the lost packet was nearly 34% in conventional network scheme.

## VI. CONCLUSION

Applying Smart Virtualization architecture in mobile communication networks as a paradigm can impact on their performance. In general, the future of mobile communication networks. In this paper, we put forward a novel proposed system of smart virtualization for packets delivery, mobility management, and handover procedure comes down to the network-based. The SDN and its integral OpenFlow protocol are used to separate the CP and DP of network flow. This separation enables the mobile operator to control the infrastructure, reduce the operational and capital costs, and fulfill horizontal packets handover optimization. The SDN could achieve the same duties and tasks that were accomplished by many physical devices can be performed and implemented by virtual networking environments. The SDN is perfect for simplifying the management of IPv6 due to the potential of IPv6 such as the vast address space and the stateless auto-configuration. Moreover, the IPv6 in mobile communications is not only used for routing purposes, but it can be accepted as a locator identifier and host identifier as well. These concepts are utilized by the proposed system which separates IPv6 into prefix ID and IID which are equivalent to locator ID and host ID respectively Our proposed system has suggested an approach to generate a host tag to be employed as an indicator of MN movement between subnetworks. Consequently, horizontal packet handover can be achieved seamlessly with a very low rate of the packet loss and minimum delay time. All advantages aforementioned meets the 5G and beyond mobile networks for future mobile communications

• • •

## REFERENCES


[1] J. Hyun, J. Li, H. Kim, J.-H. Yoo, and J. W.-K. Hong, "Ipv4 and ipv6 performance comparison in ipv6 lte network," in Network Operations and Management Symposium (APNOMS), 2015 17th Asia-Pacific. IEEE, 2015, pp. 145–150.

[2] C. Mugga, D. Sun, and D. Ilie, "Performance comparison of ipv6 multihoming and mobility protocols," in Thirteenth International Conference on Networks (ICN). IARIA XPS Press, 2014.

[3] I. Kim, Y. Jung, and Y.-T. Kim, "Low latency proactive handover scheme for proxy mipv6 with mih," Challenges for next generation network operations and service management, pp. 344–353, 2008.

[4] S. Gundavelli, "K. leung, v. devarapalli, k. chowdhury, and b. patil,?" Proxy Mobile IPv6, 2008.

[5] R. Moskowitz, P. Nikander, and P. Jokela, "and t. henderson," host identity protocol," RFC 5201, April, Tech. Rep., 2008.

[6] E. Nordmark and M. Bagnulo, "Shim6: Level 3 multihoming shim protocol for ipv6," Tech. Rep., 2009.

[7] H. Jung and S. J. Koh, "Mobile-oriented future internet (mofi): Architecture and protocols," Release, vol. 2, no. 2, 2010.

[8] D. Farinacci, D. Lewis, D. Meyer, and V. Fuller, "The locator/id separation protocol (lisp)," 2013.

[9] M. Gohar and S.-J. Koh, "A distributed mobility control scheme in lisp networks," Wirel. Netw., vol. 20, no. 2, pp. 245–259, Feb. 2014. [Online]. Available: http://dx.doi.org/10.1007/s11276-013-0605-x

[10] H. Luo, Y. Qin, and H. Zhang, "A dht-based identifier-to-locator mapping approach for a scalable internet," IEEE Transactions on Parallel and Distributed Systems, vol. 20, no. 12, pp. 1790–1802, 2009.

[11] R. Wang, H. Hu, and X. Yang, "Potentials and challenges of c-ran supporting multi-rats toward 5g mobile networks," IEEE Access, vol. 2, pp. 1187–1195, 2014.

[12] R. Jain and S. Paul, “Network virtualization and software defined networking for cloud computing: a survey,” IEEE Communications Magazine, vol. 51, no. 11, pp. 24–31, 2013.
[13] J. Doherty, SDN and NFV simplified: a visual guide to understanding software defined networks and network function virtualization. Addison-Wesley Professional, 2016.
[14] B. R. Al-Kaseem and H. S. Al-Raweshidyhamed, “Sd-nfv as an energy efficient approach for m2m networks using cloud-based 6lowpan testbed,” IEEE Internet of Things Journal, vol. 4, no. 5, pp. 1787–1797, 2017.
[15] G. Pujolle, Software Networks: Virtualization, SDN, 5G, Security. John Wiley & Sons, 2015.
[16] F. A. Yaseen, N. A. Al-Khalidi, and H. S. Al-Raweshidy, “Smart virtual enb (svenb) for 5g mobile communication,” in Fog and Mobile Edge Computing (FMEC), 2017 Second International Conference on. IEEE, 2017, pp. 88–93.
[17] P. A. Morreale and J. M. Anderson, Software defined networking: Design and deployment. CRC Press, 2014.
[18] J. Costa-Requena, J. L. Santos, V. F. Guasch, K. Ahokas, G. Premsankar, S. Luukkainen, O. L. Pérez, M. U. Itzazelaia, I. Ahmad, M. Liyanage et al., “Sdn and nfv integration in generalized mobile network architecture,” in Networks and Communications (EuCNC), 2015 European Conference on. IEEE, 2015, pp. 154–158.
[19] S. Yan, X. Huang, M. Ma, P. Zhang, and Y. Ma, “A novel efficient address mutation scheme for ipv6 networks,” IEEE Access, vol. 5, pp. 7724–7736, 2017.
[20] S. Barré, J. Ronan, and O. Bonaventure, “Implementation and evaluation of the shim6 protocol in the linux kernel,” Computer Communications, vol. 34, no. 14, pp. 1685–1695, 2011.
[21] J.-I. Kim, H. Jung, and S.-J. Koh, “Mobile oriented future internet (mofi): Architectural design and implementations,” ETRI Journal, vol. 35, no. 4, pp. 666–676, 2013.
[22] M. Menth, D. Klein, and M. Hartmann, “Improvements to lisp mobile node,” in Teletraffic Congress (ITC), 2010 22nd International. IEEE, 2010, pp. 1–8.
[23] M. Gohar and S. J. Koh, “Network-based distributed mobility control in localized mobile lisp networks,” IEEE Communications Letters, vol. 16, no. 1, pp. 104–107, 2012.
[24] C. White, D. Lewis, D. Meyer, and D. Farinacci, “Lisp mobile node,” IETF Internet draft, Oct. 2011.[Online]. Available: http://tools. ietf. org/html/draft-meyer-lisp-mn-06, Tech. Rep., 2014.
[25] R. Escriva, B. Wong, and E. G. Sirer, “Hyperdex: A distributed, searchable key-value store,” in Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication. ACM, 2012, pp. 25–36.
[26] S. K. Afshar, N. Conley, K. Kiser, W. J. Leighton III, D. N. Lokhande, P. E. Mccrink, S. Neshatfar, B. J. Olszowy, R. Patel, S. Rajamannar et al., “Paradigm in multimedia services creation methodology, and new service creation and service execution enviroments,” Nov. 22 2016, uS Patent 9,501,266.
[27] N. McKeown, “Software-defined networking,” INFOCOM keynote talk, vol. 17, no. 2, pp. 30–32, 2009.
[28] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, “Openflow: enabling innovation in campus networks,” ACM SIGCOMM Computer Communication Review, vol. 38, no. 2, pp. 69–74, 2008.
[29] M. Idri, “Mobility management based sdn-ipv6 routing header,” in Software Defined Systems (SDS), 2017 Fourth International Conference on. IEEE, 2017, pp. 150–155.
[30] J. Davies, Understanding ipv6. Pearson Education, 2012.
[31] J. Sen, “Mobility and handoff management in wireless networks,” arXiv preprint arXiv:1011.1956, 2010.
[32] R. Atkinson, S. Bhatti, and S. Hailes, “Evolving the internet architecture through naming,” IEEE Journal on Selected Areas in Communications, vol. 28, no. 8, pp. 1319–1325, 2010.
[33] K. De Turck, E. De Cuypere, S. Wittevrongel, and D. Fiems, “Algorithmic approach to series expansions around transient markov chains with applications to paired queuing systems,” in Performance Evaluation Methodologies and Tools (VALUETOOLS), 2012 6th International Conference on. IEEE, 2012, pp. 38–44.
[34] M. Harchol-Balter, Performance modeling and design of computer systems: queueing theory in action. Cambridge University Press, 2013.

FOUAD ALI YASEEN received the BSc degree from the University of Technology, Baghdad, Iraq, in 1994. He received the MSc degrees from University of Baghdad, Baghdad, Iraq, in 2010, both in electronic and communications engineering. He is currently doing his PhD at Brunel University London since 2016. His research interests include wireless communication networks and mobile communication systems. He is a student member of the IEEE.

HAMED S. AL-RAWESHIDY received the B.Eng. and M.Sc. degrees from the University of Technology, Baghdad, Iraq, in 1977 and 1980, respectively, the Post Graduate Diploma degree from Glasgow University, Glasgow, U.K., in 1987, and the Ph.D. degree from the University of Strathclyde, Glasgow, in 1991. He was with the Space and Astronomy Research Centre, Baghdad, PerkinElmer, Waltham, MA, USA, British Telecom, London, U.K., Oxford University, Oxford, U.K., Manchester Metropolitan University, Manchester, U.K., and Kent University, Canterbury, U.K. He is currently the Director of the Wireless Network Communications Centre, Brunel University London.