MultiPath TCP to Support User's Mobility in Future LTE Network

Morteza Karimzadeh[1], Luca Valtulina[1], Hans van den Berg[1,2] and Aiko Pras[1]
[1]University of Twente, [2]TNO, The Netherlands

Pablo Garrido Ortiz
Universidad de Cantabria, Spain

Ramin Sadre
Université Catholique de Louvain, Belgium

***Abstract*—Mobile networks have achieved a high-level of acceptance, mainly due to the ubiquitously deployment and ever growing connectivity through various wireless access technologies for new services, being available to mobile users. The core architecture of existing mobile networks does not scale well to accommodate future traffic demands owing to its highly centralized and hierarchical composition. Decentralization of the network architecture, is believed to be a sustainable trend against to the constantly growing of mobile data traffic. Given that LTE will handle substantial part of the worldwide mobile data traffic in the coming 5-10 years, in this paper we discuss an approach for realizing a decentralized LTE architecture. Next, we present a novel scheme to support user's mobility in this architecture. The proposed solution can handle seamless data traffic steering and session continuity for the users moving among the distributed anchor points, in a resource-efficient manner and with a minimal impact on the existing network implementation. Our approach is based on the MPTCP (*Multipath TCP*) which is also one of the most important candidate protocols to enhance end-to-end communication reliability, resiliency and bandwidth efficiency in the future mobile networks (5G). We used the NS3-LENA simulation software to implement a decentralized LTE network as well as the proposed mobility management scheme. The evaluation results show that the proposed solution efficiently fulfill the functionality and performance requirements related to mobility management.**

I. Introduction

With the ubiquitous deployment and rapid evolution of mobile networks, the demand of accessing the Internet for mobile users has been soared dramatically. The mobile devices (*e.g.,* smart-phones and tablets) generate a substantial part of the total Internet traffic, which is still increasing significantly. It is foreseen that the worldwide mobile data traffic will increase 10-13 fold between 2016 and 2021 [1,2]. Coping with such a demand in the current mobile networks is neither economically nor technically viable. The Radio Access Network (*RAN*) cannot be easily extended due to spectrum limitations. Furthermore, the core of mobile networks is highly centralized, resulting in scalability and reliability issues.

Mobile network operators increase RAN capacity by improving spectrum utilization in several ways, *e.g.,* deployment of small cells, and exploiting multi-carrier and multi-radio access approaches [3]. The major challenge related to the core networks (standardized by 3GPP, IETF) is due to the fact that a few high level network entities, entitled *anchor points*, handle both the *Data plane* (*e.g.,* routing, forwarding, tunneling) and the *Control plane* (*e.g.,* addressing, mobility management, monitoring) functions. In such a centralized architecture, Mobile Nodes (*MNs*) traffic must traverse the core anchor point and then go to the Corresponding service Node (*CN*), see Fig. 1(a). This may make the network prone to several limitations, *e.g.,* sub-optimal routing, low scalability, signaling overhead, and the lack of granularity on services [4,5].

The straightforward solution to cope with such an issue may consist of operators investment to upgrade the resources of the core network entities. Although this approach is technically feasible, network operators prefer the cost-effective and long-term solutions. Traffic offloading is an alternative approach to mitigate the traffic overhead on the limited resources of the core network. This can be achieved by placing small-scale anchor points in the proximity of the access network to locally handle MNs connections and traffic [6]. This essentially leads to a decentralized (flat) network architecture, see Fig. 1(b). Even though decentralization also requires further investments for network architecture changes and management, it seems in the long-term to be more cost-efficient than continuously extending capacity of the centralized architecture to cope with the future demands.

Following the concept of decentralization, relocation of the mobile devices' edge anchor points helps maintaining efficient routes for MNs' connections. However, it *demands additional mechanisms to maintain the MN's ongoing data session active, by enabling IP traffic continuity and steering the data packets towards the new anchor points*.

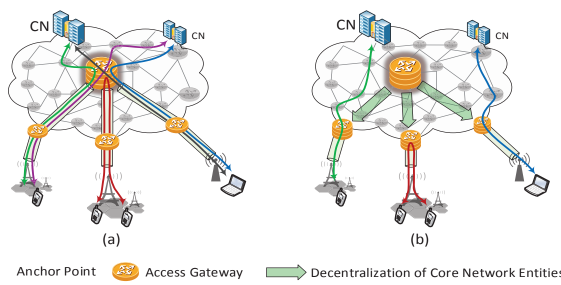


Fig. 1: A general view of the current (centralized) (a) and decentralized (b) mobile network architectures.

Long Term Evolution (*LTE*) is expected to be the leading mobile networking technology, handling considerable part (around 79%) of worldwide mobile data traffic in coming decade [1,2]. Therefore, in this paper we pay special attention to an approach for realizing a decentralized LTE system for 3GPP access, already discussed in detailed in our previous works [7,8], and introduce a novel solution to support MN's mobility and to handle traffic session continuity in such an architecture.

Various mobility management approaches may target performing at different layers of the protocol stack. The techniques based on the *Network layer* and *Transport layer* are the most widely studied. Generally speaking, the solutions corresponding to the network layer handle MN's mobility requirements in a transparent manner and hide any changes from upper layers. However, they often demand some infrastructural modifications and impose extra overhead at network [7,8]. Transport layer mobility management schemes keep the network infrastructure intact and implement the whole functions for supporting MN's mobility in the transport layer of ***only*** the end host entities. The solutions in this layer might have some limits on supporting the applications relying on other transport layer protocols.

Given that TCP is the most widely used protocol in the today Internet [9,10], in this paper we follow the latter approach and develop a new transport layer-based technique to handle MN's mobility in a decentralized LTE architecture. Our solution is based on the MPTCP, which is also one of the key candidate protocols to efficiently address the demands for faster connectivity, higher download rate and more reliable communication in the future mobile networks (5G) [11,12,13]. MPTCP has been already deployed on the Apple smart-phones and tablets (since iOS 7) to improve performance of the delay-sensitive applications (*e.g.,* Siri) [14], and it is also available for the smart-devices running Android 4.4 [15]. This makes the proposed approach easily feasible to implement in a large scale with meager overhead and complexity.

For other types of applications an additional mechanism may be required in order to translate the demanded data to the TCP-based traffic, *e.g.,* [16,17]. This topic is out of the scope of this paper.

Summarizing, our main contributions in this paper are as follow:

- We develop a MPTCP-based mobility management mechanism, enabling traffic session continuity for the MNs in 3GPP access at a decentralized LTE network.
- We extend the NS3-LENA simulation environment to support implementation of a decentralized LTE network as well as the proposed mobility support solution.
- Using this implementation, we verify the functionality and evaluate the performance of our MPTCP-based mobility support approach.

The remainder of this paper is organized as follows: § II reviews the literature relevant to the MPTCP and mobility management researches, and specifies how our work is distinct from the literature. § III provides concisely the necessary background regarding the current LTE network and its mobility management solutions. It also elaborates on the main modifications required in the current LTE system to realize an approach in decentralizing the LTE architecture, as well as to support the MNs' mobility in this architecture. § IV describes briefly the MPTCP concept, and presents our proposed mobility support solution and details its functionality. § V describes the implementation of the LTE's new architecture and our developed solution in the NS3-LENA. § VI defines the performance measures and presents the obtained simulation results. Finally, we conclude the paper at § VII.

II. Related Works

In this section we discuss some recent works on MPTCP, related to the mobility management and in particular characterize the novel aspects of our approach compared to the previous studies.

The work in [18] presents a mechanism relying on the cooperation of the MPTCP protocol and dynamic DNS. The proposed solution is based on the *ID/Locator* separation concept, where each MPTCP-enabled MN has a permanent IP address as ID, and several temporary IP addresses as locators. The IDs of MNs are used as the source and destination IP addresses (by the TCP applications), while the locators are used in the network layer for routing. During the traffic forwarding procedure, IDs of the data packets are replaced by locators, and for each MN the related mapping is stored in the DNS server. In order to support mobility functions upon moving to a new location, the MN updates its new locators on the DNS server. Next the CN, who has an initial TCP session with the MN, gets the MN's ID and locators through the DNS and accordingly sets up a new MPTCP connection. In [19] an experimental study has been performed to evaluate the performance of MPTCP in handling an MN's handover from the 3G to WiFi network. The authors in [20] discussed a seamless TCP session continuity, where multiple paths are provided through independent access technologies such as cellular and fixed wire-line networks. To avoid MPTCP's inherent deployment issue at the end points, they proposed a lightweight MPTCP proxy to be integrated into the mobile devices, network routers or data centers to handle TCP $\leftrightarrows$ MPTCP translation via straightforward packet-header rewriting. In the context of LTE network, Selected IP Traffic Offload (*SIPTO*) is introduced by 3GPP [21] in order to selectively breakout some of the IP traffic closer to the MN. SIPTO allows the selected IP traffic to be directly routed within the access networks, while non-offloaded traffic is routed towards the LTE core network. In the framework of SIPTO, traffic session continuity is not supported, as the IP address of the MN has been changed due to mobility. In [22] a MPTCP-based solution has been proposed to maintain a single session active for the MN, changing its connection from one SIPTO entity to another. In [23] a baseline architecture has been proposed to develop a distributed mobility management scheme on a 3GPP flat network. In this research the MPTCP is used to remove the chain of IP preservation of the current mobility management solutions, and instead to benefit from the available multiple paths to provide traffic session continuity to MNs.

The common assumption for all the previous works is the MN to be equipped with *multiple access interfaces* (*e.g.,* WiFi, 3GPP, Wired), enabling it to set up multiple parallel connections for a single TCP session. The MN may activate the interfaces and accordingly may establish the subflows simultaneously (to improve, *e.g.,* throughput or reliability) or one after the other (to handover among the heterogeneous wireless access networks) through the available channels towards a CN.

Our work differs from the literature, *Firstly* with respect to insightful discussion on the major modifications required in the current LTE system to realize a decentralized LTE network, as a key approach to cope with the future demand on mobile data traffic. *Secondly*, we develop a MPTCP-based scheme, ***without demanding an additional network interface*** (using only one 3GPP interface on MNs), to enable a seamless traffic session continuity for the moving MNs in a decentralized LTE network system. Furthermore, we detail on the functionality of the control and data planes of the proposed mobility support approach, to come off its implementation at the new LTE architecture. To the best of our knowledge, we are the first exploiting the MPTCP in the context of mobility management in a decentralized LTE architecture.

III. LTE NETWORK

This section gives a brief background regarding the current LTE system (§ III-A), its mobility management protocols for 3GPP access (§ III-B), and an approach to realize a decentralized LTE network deployment (§ III-C). We have described aforesaid subjects in detailed in [7,8], and concisely present in the following, as they are essential for understanding the problem statement being addressed in this paper. Next, we discuss the required modifications in the decentralized LTE architecture (§ III-D), for the our new proposed mobility support approach.

A. Current LTE Architecture

The existing LTE network architecture is hierarchical, consisting of EPC (*Evolved Packet Core*) and E-UTRAN (*Evolved Universal Terrestrial RAN*, and allows for the convergence of packet-based services for 3GPP and non-3GPP radio access, and fixed access. The EPC consists of four main elements (Fig. 2): The PGW (*Packet Data Network Gateway*) connects the EPC to external IP networks and handle centrally all the MN's data and control planes functions. The SGW (*Serving Gateway*) provides data paths between the eNodeBs and PGW and manage MN's mobility locally. The MME (*Mobility Management Entity*) controls the MNs in accessing to the LTE network. The PCRF (*Policy and Charging Rule Function*) determines QoS policies and charging rules to the PGW (if GTP is used) and SGWs (if PMIP is used) [24].

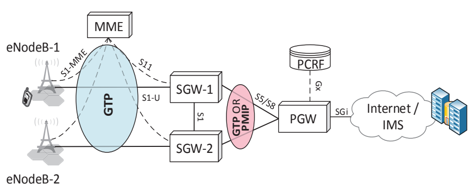


Fig. 2: Current LTE network architecture for the 3GPP access.

B. Mobility Management in the Current LTE Network

A mobility management mechanism consists a set of procedures, enabling seamless IP address and traffic session continuity for MNs within the network. In the current LTE for the 3GPP access, mobility management is based on either GTP (*GPRS Tunneling Protocol*) or PMIP (*Proxy Mobile*) protocols, where the PGW acts as a central mobility anchor point. When an MN connects to an eNodeB, its data traffic is anchored to the PGW and is encapsulated in a GTP tunnel between the eNodeB and SGW, and in another GTP (or PMIP) tunnel between the SGW and PGW. When the MN performs a handover between the source and target eNodeBs to keep the ongoing IP flow(s) active, the new S1-U and S5/S8 tunnels are established between the EPC and E-UTRAN entities (Fig. 2), depending on whether the target eNodeB is served or not by the same SGW. This procedure, shows that the existing data plane and mobility management procedure are highly hierarchical, demanding management of several tunnels between the PGW and the MNs. In a large LTE network, the PGW needs to maintain a considerable number of per-user tunneling data, which may cause scalability and performance issues.

C. Decentralized LTE Architecture

The hierarchy of the data and control planes in the current LTE system can be eliminated by co-locating the SGW and PGW functions into a single entity (*S/PGW* in Fig. 3). Accordingly, the S/PGWs are distributed closer to the edge network and can handle the MNs connection functions, data traffic, and mobility locally. This approach basically leads to a decentralized LTE architecture. In the current LTE, MNs rarely change their attached PGW. If this happens (*e.g.,* during inter-operator roaming) a *PDN Disconnection* procedure will be triggered by the network for the IP flow(s) initiated at the previous PGW. Following a decentralized architecture, relocation of the MNs' mobility anchors (S/PWG) will happen far more often. In this case, two layers of mobility management are needed in order to handle the MNs' IP-based traffic continuity: (i) within the EPC network (between the S/PGWs and eNodeBs); and (ii) outside the EPC network (between the S/PGWs and data networks), which hosts the MNs' corresponding services.

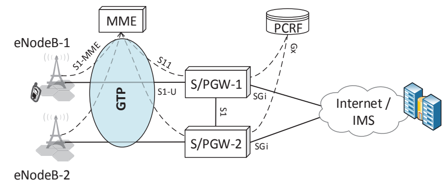


Fig. 3: Decentralization of the LTE core network architecture.

The following section describes the required modifications in the current EPC data and control planes for supporting the MNs' mobility within the EPC network of a decentralized LTE architecture. The mobility support outside the EPC network is based on our developed mechanisms, presented in § IV.

D. Mobility Management within the EPC of a Decentralized LTE Architecture

IP-based traffic session continuity, upon changing the attached PGW, is not supported in the current 3GPP's LTE standard. This is due to the fact that there is neither a signaling nor a data forwarding scheme available between two different PGWs. Following a decentralized architecture, the existing control messages and traffic forwarding mechanisms, used during a SGW relocation, can be revised for supporting the MN's mobility requirements, when it moves between the S/PGWs. The modifications should enable the following functions: (***I***) the target S/PGW must be informed to implement a new GTP bearer and allocate a new IP address for the moving MN. This bearer is used to establish the MN's new flow after a handover; (***II***) the source S/PGW must be informed when the bearer and the IP address, used by the moving MN, can be released.

During the *attach procedure* of an MN three types of bearers are set up to transport MN's traffic between the EPS entities; *Radio bearer*, *S1 bearer* and *S5/S8 bearer* (Fig. 4).

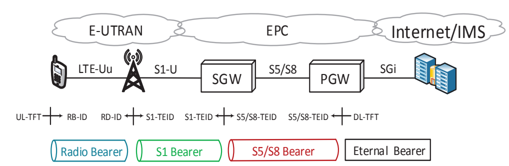


Fig. 4: EPS bearers in the current LTE network.

The S1 and S5/S8 bearers use the GTP protocol to identify the individual connections between two nodes. A Tunnel Endpoint Identifier (*TEID*)[1] is assigned to each GTP bearer,

[1]For the PMIP, a Generic Routing Encapsulation (*GRE*) key is used as a identifier.

allowing the nodes to determine to which specific bearer a particular packet belongs. Each EPS bearer is associated with one Traffic Flow Template (*TFT*), defining the filtering rules to differentiate data packets. In a decentralized EPC, the S5/S8 bearer is unnecessary and direct mapping from the External bearer to S1 bearer (DL-TFT→S1-TEID) can be done in the S/PGW.

The procedure for adjusting the S1-TEID during an MN's handover with a SGW relocation, using the X2-based handover technique[2] is specified in § 5.5.1.1.3 of [25]. We considered it as the baseline and defined a new handover mechanism to accomplish the above-mentioned issues as follows. For the first function (***I***)[3], we modified the *Create Session Request/Response* and *Modify Bearer Request/Response* messages, exchanged between the MME and the target S/PGW (which is replaced as PGW) to create a new DL-TFT in the target S/PGW. These changes realize the implementation of an EPS bearer and allocation of an IP address for the moving MN. Accordingly, this enable establishing a new data plane for the MN through the target S/PGW after its handover. For the second function (***II***)[3], we modified the *Delete Session Request/Response* messages, exchanged between the MME and the source S/PGW, in order to inquire for dropping the corresponding bearer context from the source S/PGW's list and accordingly to release the associated MN's IP address, see Fig. 7.

IV. Proposed Mobility Management Solution

This section describes the functional approach, architecture, and control messages of our proposed solution to handle the MN's mobility outside the EPC network, during a S/PGW relocation in a decentralized LTE architecture.

A. Functional Approach

During a handover procedure with a S/PGW relocation in a decentralized LTE network, the MN's traffic forwarding and session continuity between two eNodeBs can be managed by the modified X2-based handover mechanism, described in § III-D. In this section, we present the functional approach for a mechanism to handle the MN's traffic redirection outside the EPC network (in the transport network). The proposed scheme relies on the MPTCP, originally proposed to improve network throughput, resilience in the case of path failure, load balancing as well as to enable multiple network access for multi-homed (interfaces) hosts.

MPTCP is a set of TCP extensions [26], enabling transmission of a single TCP session through several subflows, with different IPs and ports. The packets belonging to various subflows can be transmitted via different available network paths towards the end point entities. The transport layer of MPTCP is splitted into *application-oriented semantic* layer (managing the application's end-to-end communication) and *network-oriented Flow/End point* layer (handling congestion control and endpoints identification), making it transparent to the both higher and lower layers [27], see Fig. 5.

MPTCP uses the standard TCP options [28] for signaling. In addition, the MPTCP options consist a set of sub-types, as listed in Table I. This enables supporting multiple connections between the end hots for a single TCP session. As the public IP address or port number of the end point entities may change in the network (*e.g.,* due to a NAT function), MPTCP assigns an 8-bits ID for each IP address. These IDs are not affected during the IP address modification and used by the end hosts to identify the subflows to be added to (or removed from) a particular connection.

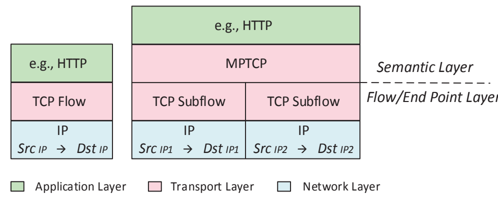


Fig. 5: TCP and MPTCP protocol stacks.

MPTCP supports three modes of operation [29]: (i) the *full-handover* mode, which refers to the regular MPTCP operation where all subflows are used simultaneously between two communicating hosts. The main objective of this mode is to obtain the best traffic rate for the users. This mode also provides the users a fast switching over different access technologies (*e.g.,* WiFi and cellular networks). (ii) the *backup* mode, in this mode MPTCP initiates subflows over all the existing interfaces but uses only a subset of those for data transmission. The subflows specified by *MP_PRIO* option define the backup subflows, and are only used when the others are inoperative (*e.g.,* due to a path failure or an invalid IP address). By this mode MPTCP can choose a premier path, for example it can use the available WiFi network and only switch to the cellular network when there is no WiFi connectivity (iii) the *single-path* mode, which is similar to the backup mode, except that at any moment a single subflow is established and used for each MPTCP connection. When the previously used interface (or path) goes down, a new subflow over the other interface (available path) is established and used for data forwarding. This mode benefits from the *break-before-make* capability of MPTCP, to maintain the ongoing session active and continue data transmission without disturbing the application.

TABLE I: MPTCP options subtypes.

Value	Symbol	Definition
0x0	*MP_CAPABLE*	Multipath Capable
0x1	*MP_JOIN*	Join Connection
0x2	*DSS*	Data Sequence Signal
0x3	*ADD_ADDR*	Add Address
0x4	*REMOVE_ADDR*	Remove Address
0x5	*MP_PRIO*	Change Subflow Priority
0x6	*MP_FAIL*	Fallback
0x7	*MP_FASTCLOSE*	Fast Close
0x8-0xe	Unassigned	---
0xf	Reserved for private use	---

In the following section, we describe the functional procedure of our proposed solution in detail, benefiting from the MPTCP's *single-path* operation mode to steer the MN's traffic on the transport network, during the MN's handover with S/PGW relocation in a decentralized LTE network architecture.

B. Architecture

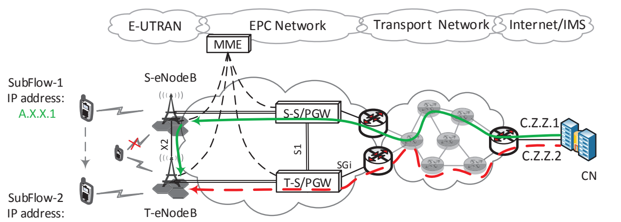
In a decentralized network architecture, the MN's IP address is anchored at the distributed anchor points. Hence, when an MN moves from one anchor point to another it receives a new IP address on the new network attachment. In this situation,

[2]The S1-based handover can also be used to handle MN's mobility during a SGW relocation (§ 5.5.1.2.2 of [25]). As in this paper we used the X2-based for development of our solutions, we ignore to present the required modifications for the S1-based approach.

[3]Due to space limitation, we only explain the messages modified for our purpose, and refer the readers to § 5.5.1.1.3, 5.10.3, 5.3.2.1 and of [25] for more information.

the MN's established IP based connection is not survived as it relies on constant endpoints IP addresses. Therefore, the MN's initial connection must be somehow protected upon obtaining a new IP address. For this, a supplementary mechanism must be applied. Through such a mechanism either the relevant endpoint updates the corresponding address or continue with the same addresses, using IP address translation or tunneling techniques. Generally, such mechanisms are implemented on the network layer and demand additional components as well as modifications in the network topology. Our developed approach handles the above-mentioned aspects without any IP address modification and impact on the network deployment. We benefit from the *multiple IP address supporting* and *break-before-make* features of the MPTCP to preserve the MN's ongoing traffic continuity during its movement as described above.

Fig. 6 and Fig. 7 show respectively, the architecture and the diagram of control messages and data traffic of the proposed approach to manage the MN's mobility in a decentralized LTE network. Given that the MN and CN are MPTCP-aware entities, in this architecture the subflows are added to/removed from the MN's active traffic via the available/inaccessible paths to support its seamless session continuity, upon handover between two eNodeBs (served by different S/PGWs).



	TCP Subflow-1	TCP Subflow-2
(a) Before Handover	C.Z.Z.1 → A.X.X.1	------------
(b) During Handover	C.Z.Z.1 → A.X.X.1	------------
(c) After Handover	~~C.Z.Z.1 → A.X.X.1~~	C.Z.Z.2 → B.Y.Y.1

Fig. 6: MPTCP to support the MN's traffic redirection and mobility.

Assume that a MPTCP-enabled MN performs an **Initial Attach Procedure** to the source eNodeB and gets an IP address allocated from the source S/PGW (*e.g.,* A.X.X.1), to inquire some data from the CN. In the discussed decentralized LTE network (§ III-C), the MN's attach procedure is similar to the one in the current LTE system (§ 5.3.2.1 of [25]), by co-locating the SGW and PGW as S/PGW. The MN establishes a connection to the CN by sending a *SYN* packet through the source S/PGW (Fig. 7). This packet includes the *MP_CAPABLE* option, specifying that the MN supports MPTCP protocol and wishes to initiate a MPTCP-based connection. If the CN is also a MPTCP capable entity, it responses to MN by sending a *SYN/ACK* packet, containing the *MP_CAPABLE* option as well. The *MP_CAPABLE* options, exchanged between the MN and CN, carry the unique identifiers for the MN and CN, entitled *MN Token* and *CN Token*, respectively. Next, the MN completes the three-way handshake with an *ACK* packet which also carries a *MP_CAPABLE* option (with *MN Token*+*CN Token*). These tokens are used by the endpoints to identify the connection and to authenticate each other, when a new subflow is added to the active connection. Upon a successful MPTCP connection establishment between the CN and MN, the MN's downlink data is transmitted through the transport network towards the source S/PGW, via subflow-1 using IP addresses: C.Z.Z.1→A.X.X.1 (the green solid-line between the CN and source S/PGW, Fig. 6).

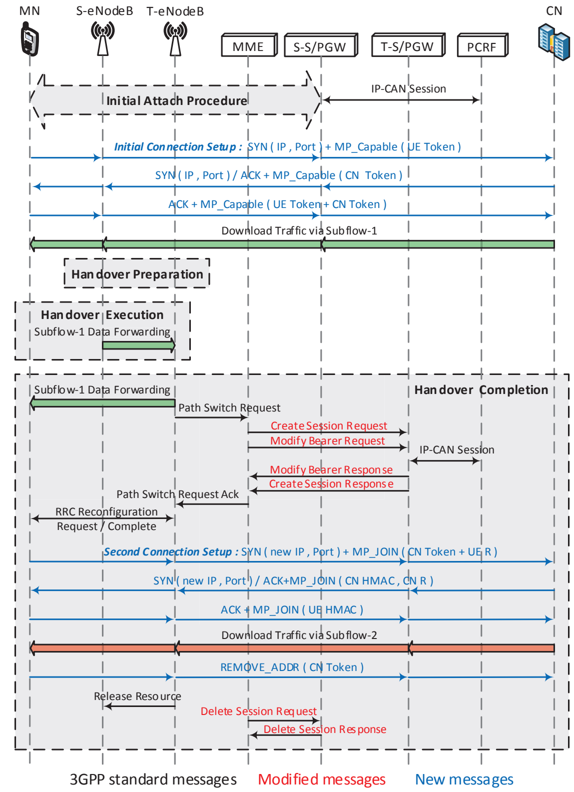


Fig. 7: MPTCP-based solution signaling and data traffic diagram.

In our solution we used the X2-based handover procedure, already specified in § 5.5.1.1.3 of [25] for the MN performing a handover with SGW relocation, as the baseline to define a new model of MN's handover between different S/PGWs. In the X2-based handover procedure, the MN's downlink traffic forwarding between two eNodeBs is fulfilled without the EPC network involvement and the related control messages on the **Handover Preparation** phase are directly exchanged between the source and target eNodeBs. Thereby, during the **Handover Executing** phase the source eNodeB forwards the MN's downlink data, received via subflow-1 from the source S/PGW, towards the target eNodeB (Fig. 7). Once the handover has been executed successfully (**Handover Completion** phase) and the MN is connected to the target eNodeB, the buffered data in the target eNodeB is delivered to the MN (the green solid-line between two eNodeBs, Fig. 6). In the **Handover Completion** phase, the target eNodeB sends a *Path Switch Request* message to inform the MME about the newly connected MN and its request for a new PDN connectivity. The MME exchanges the *Create Session Request/Response* and *Modify Bearer Request/Response* messages with the target S/PGW and asks to create a new bearer for the handed over MN. The target S/PGW creates a new entry in its EPS bearer table among other required information [30], allowing it to route MN's data between the E-UTRAN and the external network. After the bearer allocation, the target S/PGW allocates an IP address (*e.g.,* B.Y.Y.1) from its address pool to the MN. Next, the MME responds back to the target eNodeB with the *Path Switch Request Ack* message, including among other information (§

5.10.2 of [30]), IDs of the established EPS bearers, IP address of the target S/PGW and the newly assigned IP address of the MN. The target eNodeB exchanges the *RRC Reconfiguration Request/Complete* messages with the MN and informs it about establishment of the new PDN connection and the new IP address allocated by the target S/PGW. Afterwards, the MN uses its new IP address and initiates a new connection by sending a *SYN* packet through the target S/PGW to the CN. The new subflow may be opened either over a new IP address of the CN (*e.g.,* C.Z.Z.2) or the same IP address used for the previous subflow (C.Z.Z.1) with a different CN's port. The *SYN* packet carries also the *MP_JOIN* option, specifying that MN wishes to add a new subflow on a particular connection. The *CN Token*, consisted in the *MP_JOIN* option, is used as an identifier to determine to which MPTCP connection the new subflow must be associated. Thereupon, the CN and MN reply each other with the *SYN/ACK* and *ACK* packets, respectively, and complete the three-way handshake procedure for the new connection. During the connection establishment of the second subflow, the endpoints exchange also several authentication keys (*HMAC* and *R* [26]). This allows the CN to verify that the MN is the same as the one that set up the initial connection (for subflow-1). By setting up the new connection, the CN starts to transmit the MN's downlink data towards the target S/PGW, via the newly initiated subflow (subflow-2), using IP address: C.Z.Z.2→B.Y.Y.1 (the red dash-line between the CN and target S/PGW, Fig. 6).

In the both procedures (before and after handover), upon arrival of the data packets at the corresponding S/PGW, they are processed and encapsulated into a GTP tunnel and then forwarded to the relevant eNodeB to be delivered to the MN, via the LTE air interface.

After being detached from the source eNodeB, the MN uses the newly established connection through the target eNodeB and sends a *REMOVE_ADDR* option, to inform the CN about the lost connection. This option specifies the IP address (A.X.X.1) being unavailable for the MN after movement. Upon reception of this message, CN terminates the TCP subflow (subflow-1) associated with this IP address. Next, the target eNodeB asks the source eNodeB to release the resources by sending a *Release Resource* message. When the timer has expired (started by the MME upon receiving the *Path Switch Request* message [25]), the MME exchanges *Delete Session Request/Response* messages with the source S/PGW in order to release the bearers, used for the MN's first connection.

V. Set up of the Simulation Study

This section presents the evaluation scenario, implementation of the components, and parameters setting in the simulation environment. We set up the simulation environment as realistic as possible to get reliable results, as described below.

A. Evaluation Scenario

The logical network topology of the evaluation scenario is the one shown in Fig. 6. During 20 seconds simulation, 120 MNs attach to each eNodeB and generate the E-UTRAN traffic according to Table III. 30% of the MNs of the source eNodeB (MPTCP-enabled) run a TCP-based application (with 500 bytes packet size and 10 packets/second traffic stream) and move to the target eNodeB at different times between 10.74 and 12.51 seconds. The modified X2-based mechanism (within the EPC) and the MPTCP-based solution (within the transport network) handle forwarding of the MNs' downlink traffic towards the target position.

B. Implementation of the Scenario in the NS3-LENA

To evaluate the proposed solution, we extended the NS3-LENA environment to implement the following components, needed for the evaluation. The source codes to implement all of the modules can be found in [31].

1) Decentralized LTE Network: We modified the existing version of the NS3-LENA to: (i) instantiate multiple S/PGWs with different pool of IP addresses, serving separate eNodBs; and (ii) implement the X2-based handover procedure with S/PGW relocation, for realizing the implementation of a decentralized LTE network, described in § III-C and III-D.

2) Transport Network: The transport network topology is according to a small part of the EBONE (one of the European ISPs) network, covering The Netherlands, north-east of Belgium and north-west of Germany. To implement it we used a map provided by the Rocketfuel project [32].

3) MPTCP Functions: We extend the MPTCP module, already developed in [33,34], to integrate the MPTCP protocol functions in the discussed decentralized LTE network architecture. Our modifications are based on the requirements of the solution presented in this paper, see § IV-B.

4) The MNs Movement: In NS3-LENA, the MNs' handover times is based on a pre-defined schedule, set up in the simulation script. Therefore, no MNs' movement is needed to trigger the handover procedures. The MNs are only placed in positions at the same distances from the source and destination eNodeBs and a distribution of dwell times is used to trigger the handovers. We used the *Fluid-Flow* mobility model [35] to derive the average dwell time of the MNs in each S/PGW. As a S/PGW relocation most likely happens for the MNs that are on a highway, we used the *Free Speed Distributions* model [36] to compute the velocity of the MNs. We used a *Normal Distribution* with the *Mean = 32.1 m/s* and *Standard Deviation = 4.33* [36]. For the sake of simplicity we assumed that the MNs move in a straight road between the S/PGWs.

C. Simulation Parameters

This section presents the parameters setting in the simulation environment.

1) E-UTRAN Setting: Table II summarizes values of the configured parameters in the RAN. The values are based on the LTE release-8 specifications, implemented in the NS3-LENA.

TABLE II: The E-UTRAN parameters.

Parameters	Value
Uplink and Downlink bandwidth	5 MHz
Source eNodeB uplink / downlink	2535 / 2655 MHz
Target eNodeB uplink / downlink	2540 / 2650 MHz
Transmission mode	MIMO 2x2
MN / eNodeB transmission power & noise figure	26 / 49 dBm & 5 dB
Cell radius / distance	2 / 4 Km

2) E-UTRAN Traffic: We used the *traffic mix* model (Table III) specified in [37] to generate the RAN traffic.

TABLE III: The E-UTRAN traffic model.

Application	Traffic	Percentage of MN
TCP*/VoIP**	Interactive/Real-time	30%
FTP	Best effort	10%
HTTP	Interactive	20%
Video	Streaming	20%
Gaming	Interactive real-time	20%

∗) in the source eNodeB, ∗∗) in the target eNodeB

In the source eNodeB, a TCP-based traffic is selected as the one generated by the moving MPTCP-enabled MNs. Other types of traffic are used to generate the RAN background traffic by the stationary MNs attached to the both eNodeBs.

3) EPC and Transport Networks Setting: The values of parameters set up for the EPC and the transport networks are shown in Table IV. The size of buffer for entities in both the networks is set to ($\frac{C \times RTT}{10}$) [38]. Where C and RTT denote the link speed and the *Round Trip Time* ($= 250ms$ [38]) of the flows in the network, respectively.

TABLE IV: The EPC and transport network parameters.

Parameters	Value
Transmission technology	Ethernet
MTU size	1.5 KB
Transport / EPC network links data rate	10 / 1 Gbps
Queue scheme	Drop-tail
Transport / EPC network nodes buffer size	31.250 / 3.125 MB
Radius of a tracking area (TA) of each S/PGW	50 KM

4) The EPC and Transport Networks Traffic: We used the PPBP (*Poisson Pareto Burst Process*) model [39] to generate a realistic Internet traffic in the wired networks. 80% is chosen as the maximum level of link utilization for the both networks. The rest of available capacity is used to transfer the moving MNs' traffic and also to keep as the safety capacity.

VI. Performance Metrics, Results, and Discussion

In this section, using the simulations specified above, we evaluate the seamlessness of the proposed scheme. That is, if no significant delay and packet loss are introduced on the MN's downlink data traffic during a handover with S/PGW relocation. For this, we define the following performance measures.

A. Average Latency of Data Packet Delivery Before and After Handover

Fig. 8a presents, the average latency of the data packets received by the moving MNs, using subflow-1&2 via the alternative downlink paths (before and after handover). The graph clearly demonstrate that the proposed solution has no significant impact on the latency of the data packets, redirected to the MNs after a handover. Typically, the placement of the CN mainly depends on the topology (and size) of the transport network. The obtained simulation results show that after handover, the average latency of data packet delivery has a little variation ($\leq$ 8 ms), where the distance between the CN and the target S/PGW is changed from 1 to 7 hops. This accordingly may slightly affect the throughput of the proposed solution.

B. CDFs of Latency of the MNs' Downlink Data Packets

Fig. 8b presents the CDFs (*Cumulative Distribution Functions*) of latency of the first data packets delivered to the moving MNs before and after completion of a handover, using the alternative paths established by the MPTCP-based mobility support approach. We choose the first data packet of the MNs, as it is the most delayed packet during the delivery procedure. This is because its delivery time is directly influenced by the time required to build the connections (over the S/PGWs), and accordingly to set up the subflows and traffic forwarding paths in the transport network. The obtained results shows that the establishment time for the second connection (subflow-2, using *MP_JOIN* option) is slightly more than the one required to set up the initial connection (subflow-1, using *MP_CAP* option), but still is low enough to meet even the maximum acceptable latency for the real time application (*e.g.,* 250 ms for the VoIP [40]).

C. Packet Loss in the Redirected Downlink Data Traffic

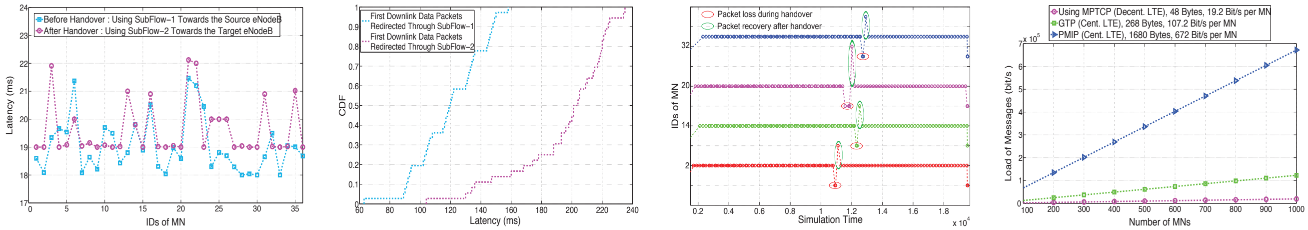
The X2 path is used to forward the MNs' downlink data during **Handover Execution**, and also **Handover Completion** when the second subflow of the MPTCP-based solution is not yet set up. In the latter case, provided that the MN's initial session has not been deleted yet by the MME [25], the downlink packets are still being received at the source eNodB are delivered to the target one via the X2 path. By using the X2 path (for $\leq$ 220 ms) after handover procedure, no loss is present in the MNs' data packets. It implies that 220 ms is sufficient time to establish the second subflow and to set up the new path in the transport network for forwarding the MN's active traffic to target position. Mobile network operators may not prefer to use the X2 path after the MN's handover. In this case, the moving MN's session is removed from the source S/PGW, once the S1 bearer is initiated at the target S/PGW. This may result in some packets loss for the MNs moving to a new S/PGW. However, it is observed that MPTCP is able to quickly detect the failures and retransmit the MN's unacknowledged data packets over the second active subflow, with only a small impact on the application delay (see the packets lost event and the packets recovery for the some of MNs, as examples in Fig. 8c).

D. Control Messaging Overhead of the Proposed Solution

In the proposed approach, several messages must be exchanged between the MN and CN entities (see Fig. 7) in order to set up the second subflow's traffic redirection path. Fig. 8d shows the overhead of the control messages of the proposed solution on the network in terms of the number of MNs, handed over to a new S/PGW. It is observed that the proposed approach has negligible impact on the network load. In this figure, we also compare our approach with the overhead impact of the existing centralized solutions (the GTP [24] and PMIP [25] protocols in the current LTE network), where a SGW relocation happens during the MNs handover.

VII. Conclusion

In this paper, we have developed a novel scheme to support seamless traffic continuity for an MN's active TCP session after a handover, with S/PGW relocation in a LTE decentralized architecture. Our solution is based on the MPTCP, which is also a promising protocol to be used in the 5G mobile networks. The proposed solution can handle the MN's mobility, *without demanding an additional network interface on the MN*, in a resource-efficient manner, with the cost of only four extra small size control messages (*3-way TCP handshake* & *Remove address*) upon a handover. As: (i) only the MNs and the corresponding CNs need to be MPTCP-aware entities; and (ii) the MPTCP is already available and operational for the most popular smart-devices OS platforms (*i.e.,* Android and iOS), implementation of the proposed solution is readily feasible for the network operators with a trivial overhead and complexity. Detailed simulations show that the proposed approach is fast enough in establishing the new connection after a handover and in setting up of the second traffic redirection path, considering the maximum allowed latency



(a) Average latency of the MNs' downlink data packets delivery before and after handover procedures.

(b) CDFs of latency of the MNs' first downlink data packets via subflow-1&2.

(c) The MNs' downlink data packets loss without using the X2 interface and MPTCP loss recovery after handover.

(d) Overhead of the control messages.

Fig. 8: Evaluation results of the MPTCP-based traffic redirection solution in a decentralized LTE network.

for a real-time application (*e.g.,* VoIP). Furthermore, it is also observed that the developed solution significantly outperforms the mobility support mechanisms of the current LTE in terms of the control plane overhead. Benefiting from the packet loss recovery mechanism of the MPTCP, the usage of X2-based data forwarding during the MN's handover can be avoided, with imposing only a small latency on the application.

ACKNOWLEDGEMENTS

This work has been partially funded by the EU FP7 Mobile Cloud Networking (FP7-ICT-318109) and EU FP7 Flamingo Network of Excellence (FP7-ICT-318488) projects.

REFERENCES

[1] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016−2021 White Paper, March 2017.
[2] Ericsson Mobility Report, November 2016.
[3] Dealing with Density: The Move to Small-Cell Architectures. In *RUCKUS Wireless White paper*, 2014.
[4] H. A. Chan, H. Yokota, J. Xie, P. Seite, and D. Liu. Distributed and Dynamic Mobility Management in Mobile Internet: Current Approaches and Issues. *Journal of JCM*, 2011.
[5] Z. F. L. Bokor and S. Imre. Flat architectures: Towards scalable future internet mobility. In *The Future Internet*, 2011.
[6] T. Taleb, K. Samdanis, and F. Filali. Towards supporting highly mobile nodes in decentralized mobile operator networks. In *IEEE ICC*, 2012.
[7] M. Karimzadeh, L. Valtulina, A. Pras, H. Van den Berg, R.O. Schmidt, M. Liebsch, and T. Taleb. Double-NAT Based Mobility Management for Future LTE Networks. In *IEEE WCNC*, 2017.
[8] M. Karimzadeh, L. Valtulina, , H. Van den Berg A. Pras, M. Liebsch, and T. Taleb. Software Defined Networking to Support IP Address Mobility in Future LTE Network. In *IFIP/IEEE WD*, 2017.
[9] M. Hoogesteger, R. Oliveira Schmidt, and A. Pras. ITSA: Internet Traffic Statistics Archive. In *IEEE/IFIP NOMS*, 2016.
[10] The Internet Traffic Statistics Archive. http://stats.simpleweb.org/statistics.php.
[11] A. Alheid, A. Doufexi, and D. Kaleshi. A study on MPTCP for tolerating packet reordering and path heterogeneity in wireless networks. In *WD*, 2016.
[12] S. Habib, J. Qadir, A. Ali, D. Habib, M. Li, and A. Sathiaseelan. The Past, Present, and Future of Transport-Layer Multipath. *Network and Computer Applications*, 2016.
[13] M. Agiwal, A. Roy, and N. Saxena. Next Generation 5G Wireless Networks: A Comprehensive Survey. *IEEE Communications Surveys and Tutorials*, 2016.
[14] https://support.apple.com/en-us/HT201373.
[15] https://multipath-tcp.org/pmwiki.php/Users/Android.
[16] H. Espeland, C.H. Lunde, H.K. Stensland, C. Griwodz, and P. Halvorsen. Transparent Protocol Translation for Streaming. In *ACMMM*, 2007.
[17] CableLabs. Content Delivery with Content-Centric Networking. Technical report, 2016.
[18] Y. Sun, Y. Cui, W. Wang, T. Ma, Y. Ismailov, and X. Zheng. Mobility support in multi-path TCP. In *IEEE ICCSN*, 2011.
[19] C. Paasch, G. Detal, F. Duchene, C.Raiciu, and O. Bonaventure. Exploring Mobile/WiFi Handover with Multipath TCP. In *ACM SIGCOMM workshop Cellnet*, 2012.
[20] G. Hampel, A. Rana, and T. Klein. Seamless TCP Mobility Using Lightweight MPTCP Proxy. In *ACM MOBIWAC*, 2013.
[21] 3GPP TS 23.829. Technical Specification Group Services and System Aspects; Local IP Access and Selected IP Traffic Offload (LIPA-SIPTO), Release 10. 2011.
[22] S. Eido, P. Mitharwal, A. Gravey, and C. Lohr. MPTCP Solution for Seamless Local SIPTO Mobility. In *HPSR*, 2015.
[23] M. Dawood, W. Fuhrmann, and B.V. Ghita. Assay of Multipath TCP for Session Continuity in Distributed Mobility Management. In *INC*, 2016.
[24] 3GPP.TS.23.402. Universal Mobile Telecommunications System (UMTS); LTE. 2015.
[25] 3GPP.TS.23.401. General Packet Radio Service (GPRS) Enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) Access, version 13.5.0 Release 13. 2016.
[26] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure. TCP Extensions for Multipath Operation with Multiple Addresses. In *RFC 6824, IETF*. https://tools.ietf.org/html/rfc6824.
[27] A. Ford, C. Raiciu, M. Handley, S. Barre, and J. Iyengar. Architectural Guidelines for Multipath TCP Development. In *RFC 6182, IETF*. http://www.potaroo.net/ietf/idref/rfc6182/.
[28] J. Postel. Transmission Control Protocol. In *RFC 793, IETF*. https://tools.ietf.org/html/rfc793.
[29] O. Bonaventure, C. Paasch, and G. Detal. Use Cases and Operational Experience with Multipath TCP. In *draft-ietf-mptcp-experience-07*.
[30] 3GPP.TS.32.251. Technical Specification Group Services and System Aspects, Telecommunication management, Charging management, Packet Switched (PS) domain charging. 2015.
[31] https://github.com/MortezaKarimzadeh/MPTCP_Mobility_LTE.
[32] Rocketfuel: An ISP Topology Mapping Engine. http://research.cs.washington.edu/networking/rocketfuel/.
[33] https://github.com/dgomezunican/multipath-ns3.13.
[34] D. Gomez, P. Garrid, C. Rabadan, R. Aguero, and L. Munoz. TCP Performance Enhancement over Wireless Mesh Networks by means of the Combination of Multi-RAT Devices and the MPTCP Protocol. *Journal of NPA*, 2014.
[35] Radhika Ranjan Roy. *Handbook of Mobile Ad Hoc Networks for Mobility Models*. Springer, 2010.
[36] S. P. Hoogendoorn. Vehicle-Type and Lane–Specific Free Speed Distributions on Motorways: A Novel Estimation Approach Using Censored Observations. *Journal of TRR*, 2006.
[37] R. Irmer. Next Generation Mobile Networks Radio Access Performance Evaluation Methodology. In *NGMN Alliance White paper*, 2008.
[38] Y. Ganjali and N. McKeown. Update on Buffer Sizing in Internet Routers. *ACM SIGCOMM*, 2006.
[39] D. Ammar, T. Begin, and I.G. Lassous. A New Tool for Generating Realistic Internet Traffic in NS-3. In *ICST (SIMUTools)*, 2011.
[40] http://www.cisco.com/c/en/us/td/docs/ios/solutions_docs/qos_solutions/QoSVoIP/QoSVoIP.pdf.