

An evolutionary path for the Evolved Packet System

Marc Portoles-Comeras^{1§}, Josep Mangués-Bafalluy¹, Andrey Krendzel^{1†}, Manuel Requena-Esteso¹,
Albert Cabellos-Aparicio²

Abstract — New architectural requirements appear with the evolution of mobile networks, such as the provisioning of multihoming or offloading. In general, this requires the design of ad hoc schemes on top of the Evolved Packet System (EPS), which may be seen as an indicator of the need for a back-to-basics architectural analysis. This paper analyzes the basic architectural principles of the EPS and compares them with those of SDN and LISP. We then describe an evolutionary path for the EPS, by showing how, with slight modifications inspired in SDN and LISP, future mobile carrier networks could natively fulfill some of their flexibility and scalability requirements. The key design principles for that are: 1) a generalized use of traffic flow templates (i.e., 5-tuple flows) for more flexible IP flow handling, 2) a full decoupling of control and user plane for flexibility, and 3) an on-demand (or pull-based) state setting at network nodes for scalability. Some examples are given to illustrate the thesis of this paper.

Index Terms — mobile network architecture, LTE/EPS, SDN, LISP, control plane, IP flow handling, multihoming, offloading, scalability

I. INTRODUCTION

As future networks are developed, they will increasingly become more heterogeneous and more complex. This poses new requirements, such as the need for offloading at various points of the network (e.g., mobile node or femtocell), or the need for multihoming (of mobile nodes or of entire sites). In any case, scalability will be increasingly relevant as the number of network nodes increases.

As far as mobile networks are concerned, the Evolved Packet System (EPS) [1] has been selected as the carrier network for LTE and LTE-A, and it is being widely adopted. However, when trying to fulfill all the above requirements, EPS systematically needs to design ad hoc schemes for each of the above problems (e.g., LIPA, SIPTO, IFOM, MAPCON, or S1-flex [2], [3], [4]), which may eventually be integrated in 3GPP specifications (releases 10 to 12), but additional issues appear again in subsequent releases. For instance, the NB-IFOM work item is included in release 13 with similar objectives ([5], [16]). This patchwork may be taken as the symptom that there is some design issue. We have identified three main causes of such problems, namely cumbersome IP flow handling, incomplete decoupling of user and control planes, and its push-oriented state information handling.

This paper discusses the design principles of EPS revolving around the above three concepts and proposes slight modifications to solve the limitations of EPS when facing such new requirements. In so doing, we evolve the EPS to embed design principles of Software-defined networking (SDN) [6] and Locator Identity Separation Protocol (LISP) [7] (section II). In fact, LISP has been easily included in SDN frameworks, such as OpenDaylight [8], as its design principles perfectly fit in those of SDN. Therefore, it may be sometimes seen as an instantiation of SDN.

Despite the hype behind SDN, only recently the Open Networking Foundation has created study groups dealing with mobile and wireless networks. There have also been some previous attempts in this direction ([9], [10]). However, these approaches were more disruptive in the sense that the set of protocols used was substantially modified. On the other hand, this paper proposes an alternative by which most 3GPP procedures are kept, and by means of slight modifications, such novel architectural principles are introduced in the EPS. Additionally, some of the initial proponents of SDN have identified the need to move towards an SDNv2 that better considers the needs of carriers, particularly the heterogeneity of equipment (incl. legacy), hence the need for a smooth migration path [11]. Bearing this in mind, this paper leverages and enhances existing technologies with concepts and principles found in more disruptive approaches. As for LISP, it shares some common design principles with EPS (e.g., separation of IP address space). However, while EPS requires ad hoc mechanisms to solve some current and future needs, such as multihoming, offloading, and scalability, LISP solves them natively.

In brief, our discussion on the architectural design principles can be classified into three main groups: 1) generalized use of traffic flow templates (TFTs¹) for a more flexible IP flow handling (section III), 2) full decoupling of control and user plane for flexibility (section IV), and 3) on-demand (or pull-based) state set up in network nodes for scalability (section V). To illustrate this, we present some examples in which we show how by embracing these principles, future mobile carrier networks can be simplified when fulfilling the above requirements. Moreover, adding such design principles to EPS requires significantly fewer modifications than one could imagine, making the evolutionary path towards a more future-proof mobile network easier to achieve.

¹ Centre Tecnològic de Telecomunicacions de Catalunya (CTTC), Castelldefels (Barcelona), Spain

² Universitat Politècnica de Catalunya (UPC), Barcelona, Spain

[§] Cisco Systems, Inc., San José, CA, US (Work done while at CTTC)

[†] Huawei Technologies Oy, Finland (Work done while at CTTC)

¹ TFT is a packet filter used to identify a flow and it is typically based on the 5-tuple (source and destination addresses, source and destination port and protocol).

II. ON THE ARCHITECTURAL DESIGN OF EPS, SDN, AND LISP

A. IP address space separation

A first important observation that stems from the comparison of the EPS and LISP architectures is that both solutions offer a similar structure of the user plane. In particular, both architectures introduce one layer of indirection and divide the network into two address spaces. This allows the implementation and optimization of certain functions, like mobility.

Figure 1 (upper part) illustrates the EPS architecture for user plane connectivity to an IP network called a Packet Data Network (PDN) in 3GPP terms, where IP addressing is separated into two different spaces, the internal and external addressing spaces.

There are network entities (Packet Data Network Gateways, or PDN-GWs, and base stations, or eNBs in LTE terminology) that lay at the border of both address spaces and are able to encapsulate flows from the external IP address space to be routed through the internal address space. Furthermore, there is an internal user plane entity (Serving Gateway, or S-GW) that is able to re-encapsulate flows between user plane nodes, and can be used to optimize certain functions.

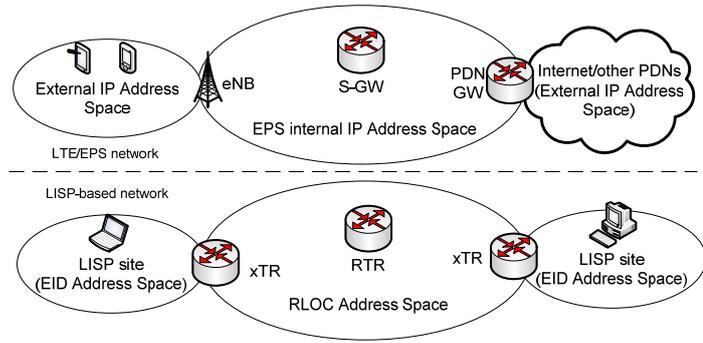


Figure 1. IP addressing architecture of LTE/EPS and LISP networks

When deploying LISP to operate intra-domain we can establish a clear parallelism between the data plane structure of the network and the user plane in the EPS architecture. In particular, as Figure 1 (lower part) illustrates, the addressing space is separated in two, creating one level of indirection between Endpoint Identifiers (EIDs) space and Routing Locators (RLOCs) space. Tunneling Routers (xTRs) are the ones in charge of encapsulating (and decapsulating) flows from EID space that are routed through RLOC space. Finally, there exist special elements called Re-encapsulation routers (RTR) that are used to implement and optimize certain functions that benefit from implementing middle boxes, much à la S-GW in EPS.

However, despite these clear similarities in the conception of the data/user plane, the two architectures present clear differences in the way the interaction is implemented between the two address spaces. The next subsection clarifies this.

B. Embracing the IP paradigm: IP flows, identifiers, and routing functions

EPS utilizes an all-IP network that provides better handling of end-user IP addresses and support for the whole system core (i.e., control plane and user plane) functions on top of IP-based

communications. However, the EPS architecture still maintains some legacy architectural structures that limit the full adoption of the IP paradigm (i.e., the handling of flows based on TCP/IP header fields and identifiers/addresses, and the use of IP routing functions). This is especially evident in user plane elements.

First, the attach procedure [13] creates logical/virtual connections (called PDN connections) between user equipment (UE) and the PDN-GW that are only IP-aware at both ends, but not inside the network. Upon attachment, the system creates state in all the elements that form the user plane path towards the PDN-GW to ensure appropriate data forwarding. These virtual connections are defined as *bearers*, which are then used to apply appropriate QoS and security requirements to flows. In the uplink, the mapping between an IP flow and a bearer is done at the UE, and user's (external) IP flow information is never used again to make decisions until packets are decapsulated at the PDN-GW. The same happens with downlink flows that are only interpreted at the PDN-GW, when encapsulated, and at the UE, when decapsulated.

Second, and related to the previous observation, the eNB and the S-GW, in some deployment options, do not consider at all end-user IP information when performing encapsulation. Data forwarding and flow mapping to bearers, in these two entities, is done using TEID (Tunnel Endpoint Identifier) fields and Radio Bearers. Upon attachment, they store enough context information to be able to encapsulate and decapsulate end user's data without the need to process external IP header information. The operation of these two entities can be understood as a natural evolution of the 3GPP architecture from the circuit switching paradigm. However, in practice, PDNs are IP-based networks [1] and IP packets generally contain enough information to be used to identify and map flows, a proof of that being the TFT filters used in the PDN-GW to map traffic and bearers, which are equivalent to the packet matching rules of SDN forwarding nodes.

Third, it is interesting to notice that, during the attachment procedure, each node of the user plane path checks, verifies, and stores multiple identifiers in relation to different network functions. At each node forming the user plane path, TEIDs are used for appropriately forwarding GTP (GPRS Tunneling Protocol) encapsulated traffic between entities. At the edges, the UE IP address assigned by the PDN-GW (along with other fields of the header of the packet) is used to differentiate incoming flows, whilst a number of identifiers are used to identify the UE in different scenarios (e.g., S-TMSI is used for paging of UEs and service request and S-RNTI is used for radio identification between the UE and the eNB [13]).

All these aspects are simplified in SDN and LISP. At the conceptual level, the former would allow flexible data plane processing by using multiple header fields (including the IP header). For instance, OpenFlow, one popular element of current SDN architectures allows forwarding rules at nodes based on IP header fields [12]. The LISP architecture also natively handles IP-level information [7] and, in this sense, it is IP-centric. This is why the LISP vs. EPS discussion is relevant here. In fact, all xTRs use end-user IP layer information to encapsulate and forward packets through RLOC space and to implement functions that require identifiers. As a consequence of that, xTRs feature full layer-3 routing functionality and are able to differentiate flows based on end user's IP layer data, without the

need for additional information. As we show below, the possibility to differentiate IP flows in elements of the architecture highly facilitates the implementation of offloading strategies (see section III)

C. Decoupling of the Control and User/Data Planes

The EPS architecture aims at introducing a clear separation between control plane and user plane operation. Control plane elements, which are mostly decoupled from the user plane path, handle Authentication, Privacy, QoS and Mobility functions [1].

However, this decoupling between the two planes has not been developed to its full extent. In particular, when observing the control plane structure of the EPS network, one can notice that the architecture is built around the couple MME/S-GW (Mobility Management Entity and Serving Gateway, respectively) that is located at the center of the whole system. While the MME only has control plane functionality, the S-GW shares both control and user plane responsibilities. In particular, the S-GW acts as a transit point for the signaling exchange between MME and PDN-GW. Interestingly, while most control plane decisions have been decoupled from user plane elements, the responsibility to disseminate them relies on specific interfaces between the PDN-GW and S-GW.

One of the founding tenets of SDN is the decoupling between control plane and data plane. Open interfaces between its building blocks are expected to bring network-awareness to the applications (and vice versa) and higher flexibility through network programmability. This flexibility inherent to such full decoupling is precisely what EPS is lacking. Furthermore, what is relevant for the requirements under consideration (i.e., offloading, multihoming, scalability) is that such flexibility is also present in nodes at the edge of the network, in which traffic diversion actually happens. This is done in SDN by applying the appropriate forwarding rules in such nodes. In a similar way, LISP xTRs (at the edge of the network) inherently provide such flexibility by potentially tunneling each flow through a different path, according to the rules configured in the Mapping System (control plane).

In this way, the complexity and patchwork introduced by IFOM, MAPCON, S1-flex, or more recently, NB-IFOM ([2],[3],[4],[5],[16]) would be removed by design. The paper reviews this specific case in section IV.

D. Approach to architecture scalability: push vs. pull strategies and their consequences

Finally, there is also a key difference between architectures in the strategy adopted to handle interfaces between control plane and user/data plane elements.

The EPS signaling model used to form user plane paths is based on a push-model. That is, upon attachment of a mobile node, path information is *pushed* to all user plane network entities that store this information as context information. In general, a key advantage of push-based signaling models is that path changes are rapidly spread throughout the network. The downside is that network resources are (potentially) wasted sending messages for setting up and storing state at nodes that is used during a very low proportion of time.

On the contrary, under the LISP paradigm, the interaction between data plane and control plane elements is pull-based, i.e., path information is requested and cached only when needed, generally. Equivalently, in certain SDN implementations (i.e., reactive flow setup), the first packet of a new flow triggers the

set up of state at forwarding nodes by the control plane. This model requires the definition of some extra mechanisms to control changes in path configurations, but it scales better, as nodes only set up and store state information that they need when they need it.

Additionally, in order to ensure scalability, the EPS defines a hierarchical structure of the user plane, where the forwarding path is structured as a tree between the PDN-GW and the eNBs, acting as leaves. And so, it inherits the rigidity of the circuit-switched world. Using this strategy, access nodes do not need to maintain much state in relation to their users, as they are assigned a default S-GW and PDN-GW to send data to and receive data from. This solution to provide scalability completely removes the flexibility required to offer route optimization, as opposed to what happens with a pull-based strategy (see section V for details on this scenario).

E. Summary of observations: main findings of EPS design analysis

Table 1 summarizes the main findings of EPS design analysis in comparison with SDN- and LISP-based architectures in terms of the basic design principles under discussion.

| Feature | EPS | SDN | LISP |
|---|--|--|--|
| Embracement of the IP paradigm | Limited adoption: - per PDN-connection basis, - intermediate nodes do not use end-user IP info for packet, - multiple identifiers | Traffic processing at data plane nodes defined generically through packet matching rules, hence also including IP header fields. | Full adoption: - per IP flow basis (5-tuple), - all nodes use end-user IP info, - unique identifier |
| Decoupling Control Plane and User Plane | Not full decoupling of user and control plane operation (S-GW shares user and control plane operation) | Main design principle of SDN | Separation of data and control plane functionality |
| Scalability | Push-based state and hierarchical user plane topology | Combination of pull- and push-based with arbitrary topologies | Pull-based state with arbitrary user (data) plane structure |

Table 1. Summary of EPS design analysis in comparison with SDN- and LISP-based architectures

It can be seen that the EPS presents a limited adoption of these key design principles. As a consequence, EPS faces a series of problems when dealing with more flexible data networking requirements, which are hence handled through various patches (e.g., to implement offloading or multihoming). In the following three sections, we discuss about current challenges caused by these limitations. We then propose an evolutionary path for EPS in the sense that with small modifications to 3GPP procedures, the EPS can embrace some of the key SDN and LISP design principles, hence solving by design what otherwise is solved through ad hoc patches.

III. FLEXIBLE FLOW HANDLING: THE CHALLENGE OF OFFLOADING TRAFFIC

A. The challenge of offloading traffic

Mobile data offloading has become one of the key strategies adopted to face the exponential growth of mobile data in cellular

networks. 3GPP approached the problem with proposals like LIPA and SIPTO [14].

LIPA is an offloading technique by means of which a UE through a base station is able to exchange data with IP capable entities within its local network. SIPTO is an offloading technique by means of which the mobile operator is able to offload certain types of traffic through a network node close to the UE's point of attachment.

This subsection builds on the observation that LIPA and SIPTO are, in fact, solutions that introduce *external* IP address awareness in elements of the user plane path that do not normally use this information (e.g., eNB). However, as they are currently defined in [14], LIPA and SIPTO developed mechanisms to circumvent the requirement to use external IP addresses with support of multiple PDN connections. In a scenario with full embracement of the IP paradigm in all elements of the architecture, LIPA and SIPTO would reduce to configuring routing tables and rules.

B. Proposed solution for the challenge

The EPS architecture uses TFTs whenever an element needs to interpret external IP layer information to classify flows. TFTs are distributed during the attachment procedure with the generation of dedicated bearers, but only those elements that are going to use them (the PDN-GW, the UE and, in some scenarios, the S-GW) store TFTs as context information. In fact, such TFTs could be seen as equivalent to the packet matching rules one may find in data plane nodes of SDN.

Interestingly enough, transitioning from the current EPS architecture to a full embracement of the IP paradigm requires minimal changes to the EPS architecture. Indeed, the proposal here is to extend the use of TFT filters to all the elements in the user plane path. This includes storing TFT information in eNBs and S-GWs in addition to the PDN-GW, which was already using this information. Even more, an analysis of the attachment procedure reveals that during this process enough TFT-related information is carried through all interfaces, and so, it can be exploited and stored locally at each of the user plane elements. As a result, eNBs and S-GWs just need to store this information that was previously ignored.

In order to prove the previous statement, we have analyzed the possibility to use TFT filters in all nodes of the user plane path to map flows to bearers in [15]. The study provides a detailed analysis revealing how none of the interfaces of the EPS architecture needs to be neither modified nor extended to support the dissemination of TFT information to all user plane entities. The flow diagram illustrating the TFT dissemination process specified by 3GPP is shown in Figure 2. The interested reader will find a detailed description of all the necessary steps in [15].

C. Offloading with a TFT-based architecture

With an EPS system embracing the IP paradigm (or generically, an arbitrary header field matching rule), as described above, the problem of offloading reduces to a question of distributing appropriate TFTs and routing information to affected nodes from the centralized control we discuss about in the following section.

In particular, under the proposed scheme, the process of offloading traffic in a SIPTO scenario would not differ much from the process of establishing a dedicated bearer, where the eNB receives a TFT filter associated to the traffic that needs to be offloaded, as well as the alternative (offloading) path to use. It

can be noticed here that, once using TFTs, the base station is able to differentiate particular flows and has the possibility to implement SIPTO functionality without requiring that the UE supports multi-PDN connectivity, hence making it transparent to the UE.

In the particular case of LIPA, we need to ensure that TFT rules are applied before routing rules [14] to ensure that traffic that is not meant to be offloaded (i.e., the one that must traverse the core network) is not routed locally before being tunneled to the core.

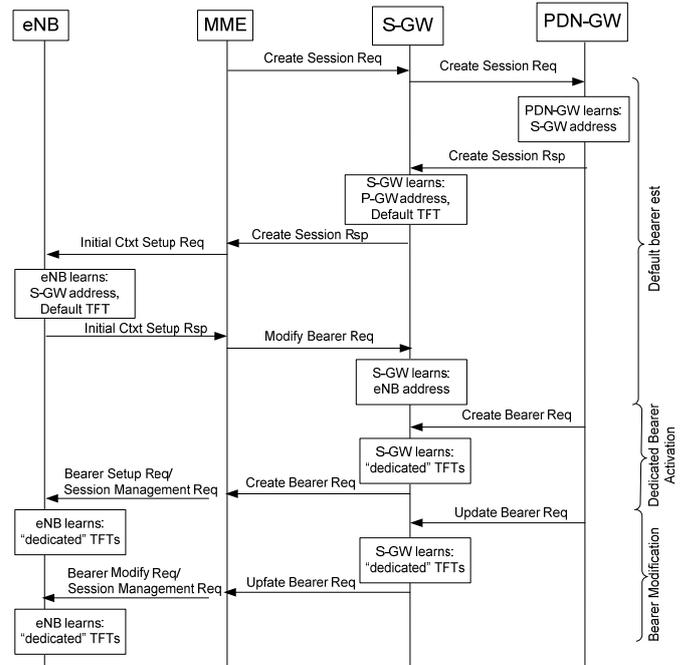


Figure 2. Dissemination of TFTs between all user plane entities

IV. DECOUPLING CONTROL AND USER/DATA PLANE FUNCTIONALITY: THE CHALLENGE OF MULTIHOMING

A. The challenge of multihoming

The multihoming problem appears repeatedly in the EPS architecture with various flavors and applied at different network entities. In the *IFOM* (and *NB-IFOM* in 3GPP release 13) scenario, multihoming is required so that the end-user is able to use more than one access network to send data and so that the PDN-GW can also send flows through different paths through a single PDN connection. In the *MAPCON* scenario, multihoming is required so that the UE can use more than one PDN connection through 3GPP and non-3GPP access technologies. Another initiative that may require the configuration of multihomed entities is the introduction of the *SI-flex*, where the system introduces MME (and its associated S-GW) redundancy. The *SI-flex* proposal only considers the possibility of having a setup of the type *active/multiple-passive* S-GW. With this setup, only one S-GW can be active in a given moment of time, and this is the only one that the eNB can use to forward data towards the PDN-GW.

In all these cases, multihomed elements of the network core face the challenge of also having to multihome their control plane procedures. As a consequence, duplicating user plane functions leads to having to solve a problem in the coordination of control plane functions.

The limited decoupling between control plane and user plane in EPS entails additional challenges in multihoming scenarios. However, with fully decoupled control and data planes, as in SDN, multihoming user plane elements (e.g., for load balancing) is not an issue.

B. Proposed alternative for the architecture: full decoupling of control and user plane

Since the MME is the central control entity of the EPS, we propose to move all control plane responsibilities of the S-GW to the MME, given the limitations that having control plane responsibilities in user plane elements introduces (e.g., for multihoming). As a consequence, the MME becomes the core responsible for the establishment and maintenance of the user plane path during the attach process and during mobility events.

Therefore, one additional interface must be established between the MME and the PDN-GW, which plays the role of S5/S8 interface for control plane information (see Figure 3). Additionally, the S11 interface (between the MME and the S-GW) needs to be extended to accommodate those signaling messages that are directly exchanged between the PDN-GW and the S-GW.

In [15], we have analyzed the feasibility to move the messages exchanged between the PDN-GW and the MME in the current EPS architecture to the new proposed interface. The study reveals that both the attachment and the mobility procedures can be supported without modification using the new proposed interfaces and with some extension of the functionality provided through the S11 interface. For instance, in the considered procedures [15], out of 10 signaling messages currently exchanged through a combination of S11 and S5 interfaces, 8 messages are sent through the new interface and 2 additional messages are sent through S11. As a result, the S5 interface is released from control plane operations.

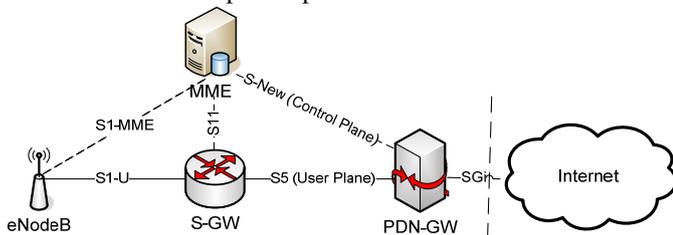


Figure 3. MME with interface to every node

Notice that this paper focuses on connectivity and mobility management, given its focus on fundamental design principles underlying the EPS. QoS and Security should be devoted adequate treatment in future refinements, given their importance.

C. Multihoming with full decoupling of User Plane and Control Plane functions.

Assuming an SDN-like full decoupling of control and user/data planes as the one proposed above, implementing multihoming in the user plane path reduces to informing the different elements of the multiple options that they have to forward data.

An example of this would be the implementation of an advanced *S1-flex* scenario where the eNB receives a list of alternative S-GWs to use simultaneously. It could be done in two ways. First, the MME can send to the eNB a signaling message (e.g., Initial Context Setup Request/Attach Accept [13]) including addresses of several candidate S-GWs for user plane instead of one, as currently done. Or second, the MME can send

to the eNB the same signaling message, but several times, with a different S-GW address in each. Similarly, the PDN-GW can receive the same information about candidate S-GWs, directly from the MME responsible for this particular user plane path.

The same approach can be followed to implement NB-IFOM and MAPCON scenarios [16], where the MME informs the corresponding nodes about the multiple forwarding options to send and receive information.

V. APPROACH TO NETWORK SCALABILITY: THE CHALLENGE OF ROUTE OPTIMIZATION

A. Network architecture and route optimization

The logical architecture of the user plane of an EPS system is highly hierarchical, with the PDN-GW being the root of the system and the eNBs located at the leaves. The advantage of this architecture is that it scales when the number of users and eNBs grow without increasing the complexity and requirements of eNBs. The reason is that they just need routing information to reach their corresponding S-GW and PDN-GW, and are completely unaware of other S-GWs and eNBs. However, for this same reason, it is hard to offer the flexibility required by novel scenarios.

More specifically, route optimization (Figure 4) in this architecture is challenging. For example, when two UEs belonging to the same network are communicating, the flow must traverse the complete hierarchy (up to the PDN-GW) and back to the correspondent UE. Providing direct communications (i.e., route optimization) between eNBs in the current push-based EPS would require that all elements receive and store information about all the rest of elements and this would not scale.

Alternatively, in a pull strategy, nodes just request the information they need and do not cache information on inactive communications. In this way, flexibility and scalability can be simultaneously achieved.

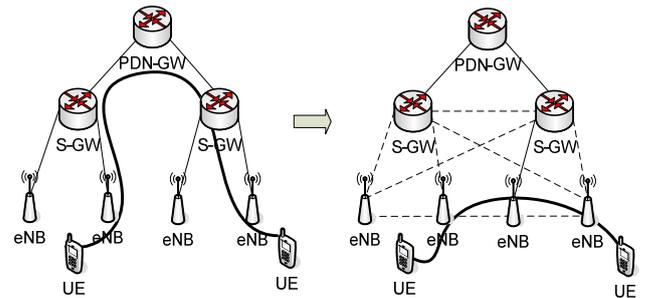


Figure 4. Example of route optimization in an EPS network

B. Proposed alternative for the architecture: from push to pull

Following the reasoning above, our proposal is to move from a push-based scheme to disseminate information to a pull-based scheme, where eNBs request information about flows when they require it.

More specifically, the attach procedure would be used to establish the default bearer following regular 3GPP procedures. After that, when the UE generates the first uplink packet of a new flow, the eNB would request (i.e., pull-based) whether a dedicated bearer must be established to support the flow. Such dedicated bearers would not be constrained by the hierarchical architecture and could be established between any two elements

of the system (e.g., two eNBs) thanks to the data plane information provided by the control plane. We have analyzed the requirements to use a pull-based strategy to support path formation in EPS systems in [15]. Slight modifications are needed in some EPS procedures, but the basic structure of the procedure can be maintained. For instance, the Initial Context Setup Request/Attach Accept message [13] must contain the destination eNB address.

C. Scalability and route optimization with a pull scheme

With the scheme proposed, the problem of route optimization is moved to the control plane, which decides, on a per-request basis, whether a flow must follow the hierarchy up to the PDN-GW, or it can be routed directly between nodes of the topology.

As an example, during a handover, X2-based data forwarding would be simplified. When the source eNB received data for a node that was no longer under its control, it would request the establishment of a dedicated bearer to forward the data and it would receive information about the destination eNB. As a result, data would be directly encapsulated between the two eNBs without the need for complex logical link setups.

We focused our above discussion on the route optimization problem among two nodes of the same network. However, the same pull-based approach could be used for LIPA and SIPTO towards local or external nodes, hence benefitting from the same advantages. In this case, offloading rules would not need to be pre-configured in advance, but when the UE actually starts sending data of a given flow, which allows the control plane to dynamically adapt to network conditions in a scalable way.

| Challenge | 3GPP initiatives | EPS limitations | Proposed solution |
|--------------------|--------------------------------|--|--|
| Traffic offloading | LIPA, SIPTO | External IP address unawareness in some of the nodes of the user plane; UE needs to support multi-PDN connectivity | Distributing TFTs and routing information to all nodes of the user plane to differentiate IP flows |
| Multi-homing | IFOM, MAPCON, S1-flex, NB-IFOM | Duplicating user plane operation in coordination with control plane functionality is an issue because of limited decoupling between planes | Direct interface between MME and PDN-GW for full decoupling of user and control operation |
| Route optimization | X2, LIPA, SIPTO | Push-based approach for path formation and hierarchical user plane topology (forwarding always through the PDN-GW) | Pull-based approach on a per-request basis and arbitrary topology for data path formation |

Table 2. Network challenges, related 3GPP concepts, EPS limitations, and proposed solutions.

VI. SUMMARY

This paper presents an evolutionary path for EPS towards a more flexible and future-proof architecture to fulfill current and future requirements (e.g., multihoming, offloading, route optimization). Such architectural reasoning exploits design principles that have shown their potential in the SDN and LISP

contexts. Our proposal is evolutionary in the sense that the minimum possible subset of modifications to current 3GPP procedures is applied to include the new design principles in EPS.

This paper claims that the flexibility and scalability required by novel scenarios can be provided: 1) by using TFTs at each data plane node for a more flexible IP flow handling (similar to SDN packet matching rules or LISP packet handling at xTRs), 2) by introducing a new interface between the MME and the PDN-GW for full decoupling of user and control planes à la SDN, and 3) by applying a pull-based model for on-demand state set up at network nodes for efficient path handling. With the proposed modifications, the above requirements could be natively solved by EPS without the need for ad hoc schemes.

The current data networking challenges, 3GPP initiatives, EPS limitations, and proposed solutions to overcome them are summarized in Table 2.

ACKNOWLEDGMENTS

This work has been partially funded by the Spanish Ministry of Science and Innovation under grant TEC2011-29700-C02-01 and TEC2011-29700-C02-02. The authors would also like to acknowledge the valuable comments of Scott Mansfield and the help of Huub van Helvoort.

REFERENCES

- [1] M. Olsson, et al., "SAE and the Evolved Packet Core: Driving the mobile broadband revolution", Academic Press, 2009.
- [2] F. Rebecchi et al., "Data Offloading Techniques in Cellular Networks: A Survey." Accepted for publication in IEEE Communications Surveys & Tutorials. To be published in 2015.
- [3] Small Cell Forum, "Integrated Femto-WiFi (IFW) Networks", SCF Release 4, Doc. 033.04.01, December 2013.
- [4] S. Sesia, et al., "LTE – The UMTS Long Term Evolution: From Theory to Practice", John Wiley&Sons, 2009.
- [5] 3GPP. "Release 13 Work Item: IP Flow Mobility support for S2a and S2b Interfaces (NB-IFOM)", Available at: <http://www.3gpp.org/DynaReport/WiSpec--640047.htm>
- [6] Open Networking Foundation, "SDN architecture v1.0." ONF technical reference document, June 2014.
- [7] D. Farinacci et al., "Locator/ID Separation Protocol (LISP)," IETF RFC 6830, January 2013.
- [8] OpenDaylight project. Available at: <http://www.opendaylight.org/>
- [9] K. Pentikousis, Y. Wang, W. Hu. "Mobileflow: Toward software-defined mobile networks." IEEE Communications Magazine, vol. 51, no. 7, July 2013.
- [10] A. Basta, W. Kellerer, M. Hoffmann, K. Hoffmann, and E. D. Schmidt. "A Virtual SDN-enabled LTE EPC Architecture: a case study for S-/P-Gateways functions." Proc. of the IEEE Software Defined Networks for Future Networks and Services, November 2013.
- [11] C. Matsumoto. "Time for an SDN Sequel? Scott Shenker Preaches SDN Version 2." SDNcentral, October 29, 2014.
- [12] A. Lara, A. Kolasani, and B. Ramamurthy. "Network Innovation using OpenFlow: A Survey." IEEE Communications Surveys & Tutorials, vol. 16, no. 1, first quarter 2014.
- [13] 3GPP TS 23.401, "General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access v13.0.0." September 2014.
- [14] 3GPP TR 23.829, "Local IP Access and Selected IP Traffic Offload (LIPA-SIPTO) v10.0.1," October 2011.
- [15] A. Krendzel, M. Portoles, J. Mangues. "Fundamental analysis of the EPS architecture facing future data networking needs", CTTC technical report, November 2012. Available at: http://networks.cttc.cat/wp-content/uploads/sites/2/2013/05/future_eps.pdf
- [16] 3GPP TR 23.861 "Network based IP flow mobility V1.11.0," November 2014.

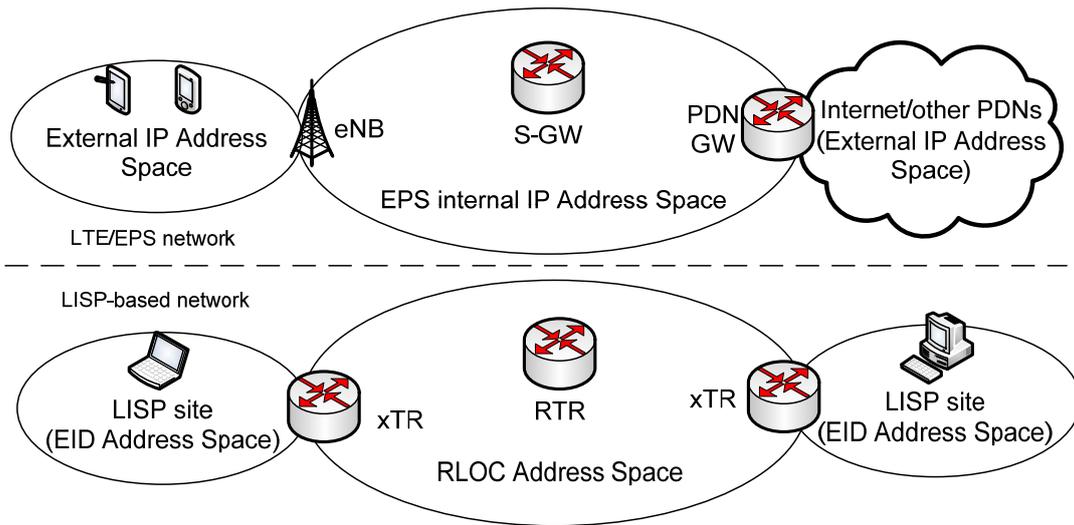


Figure 1. IP addressing architecture of LTE/EPS and LISP networks

| Feature | EPS | SDN | LISP |
|---|---|--|--|
| Embracement of the IP paradigm | Limited adoption: - per PDN-connection basis, - intermediate nodes do not use end-user IP info for packet , - multiple identifiers | Traffic processing at data plane nodes defined generically through packet matching rules, hence also including IP header fields. | Full adoption: - per IP flow basis (5-tuple), - all nodes use end-user IP info, - unique identifier |
| Decoupling Control Plane and User Plane | Not full decoupling of user and control plane operation (S-GW shares user and control plane operation) | Main design principle of SDN | Separation of data and control plane functionality |
| Scalability | Push-based state and hierarchical user plane topology | Combination of pull- and push-based with arbitrary topologies | Pull-based state with arbitrary user (data) plane structure |

Table 1. Summary of EPS design analysis in comparison with SDN- and LISP-based architectures

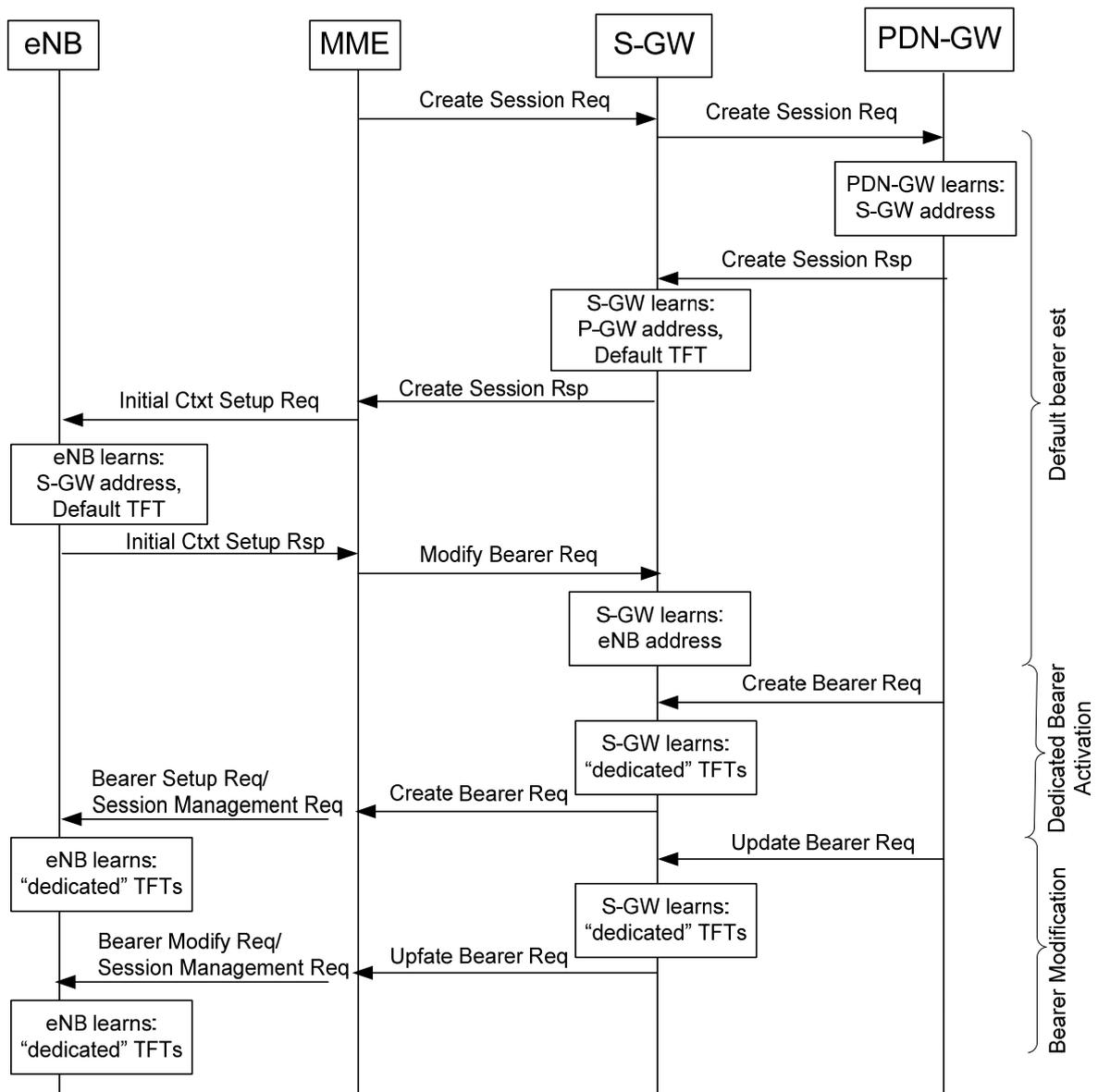


Figure 2. Dissemination of TFTs between all user plane entities

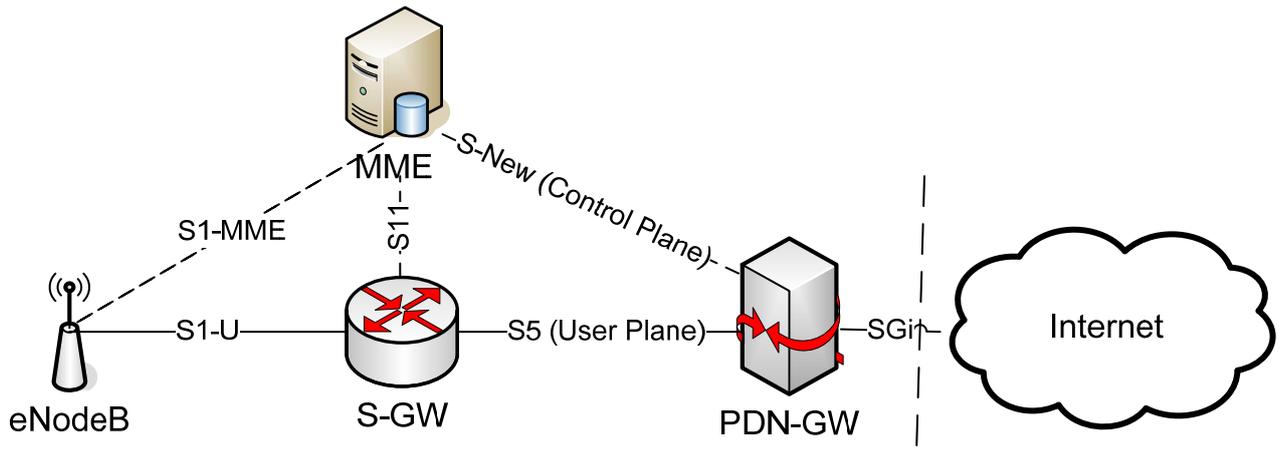


Figure 3. MME with interface to every node

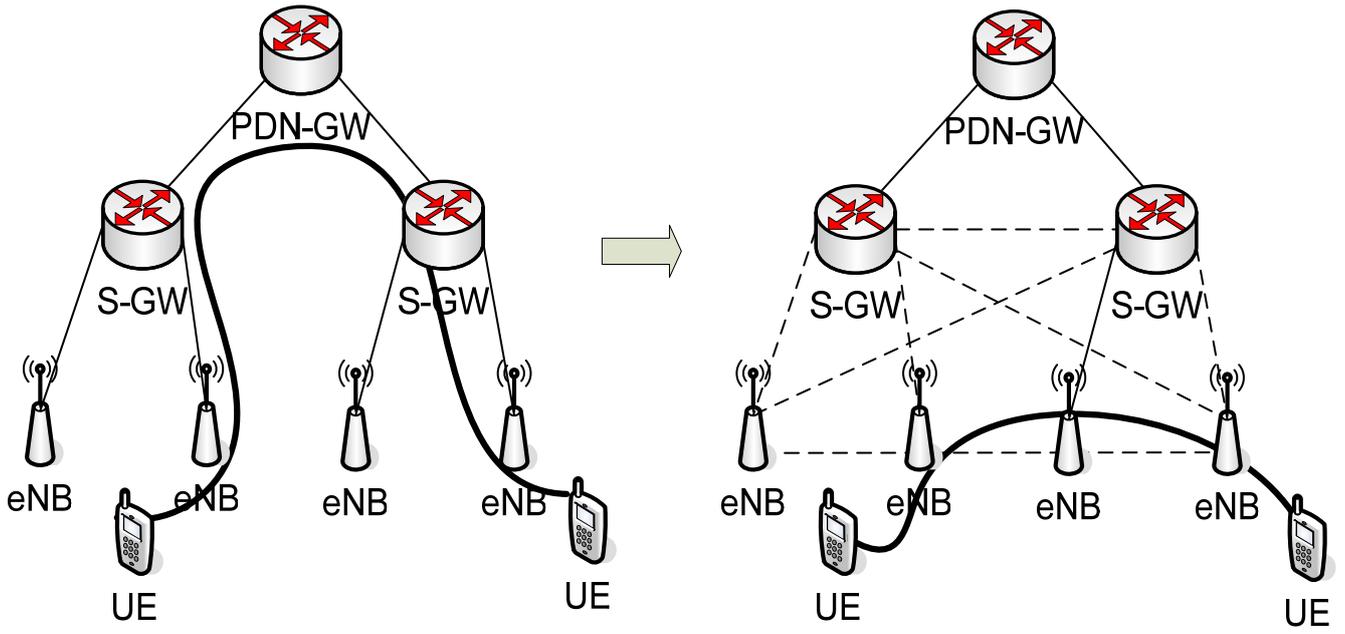


Figure 4. Example of route optimization in an EPS network

| Challenge | 3GPP initiatives | EPS limitations | Proposed solution |
|--------------------|--------------------------------|--|--|
| Traffic offloading | LIPA, SIPTO | External IP address unawareness in some of the nodes of the user plane; UE needs to support multi-PDN connectivity | Distributing TFTs and routing information to all nodes of the user plane to differentiate IP flows |
| Multi-homing | IFOM, MAPCON, S1-flex, NB-IFOM | Duplicating user plane operation in coordination with control plane functionality is an issue because of limited decoupling between planes | Direct interface between MME and PDN-GW for full decoupling of user and control operation |
| Route optimization | X2, LIPA, SIPTO | Push-based approach for path formation and hierarchical user plane topology (forwarding always through the PDN-GW) | Pull-based approach on a per-request basis and arbitrary topology for data path formation |

Table 2. Network challenges, related 3GPP concepts, EPS limitations, and proposed solutions.

Marc Portoles Comeras received his Degree in Telecommunications Engineering from the Technical University of Catalonia (UPC) and is currently working as a Software Engineer at Cisco Systems Inc. participating in the development of the LISP protocol architecture. Before joining Cisco he was a Research Engineer at the Centre Tecnològic de Telecomunicacions de Catalunya (CTTC) where he participated in multiple R&D projects funded by the EU and the Spanish Government, such as IST-WIP, BeFEMTO and SYMBIOSIS. His current research interests are on SDN and network virtualization solutions.

Josep Mangués-Bafalluy received the MSc (1996) and PhD (2003) in Telecommunications from the Technical University of Catalonia (UPC), including an Erasmus grant in ENSEEIHT, Toulouse, France. He is senior researcher and head of the communication networks division (networks.cttc.cat) of the Centre Tecnològic de Telecomunicacions de Catalunya (CTTC) and coordinated its IP Technologies Area from 2003 to 2012. He regularly participates in EU (FP6, FP7, H2020), industrial (e.g., Cisco, Orange, Aviat Networks), Spanish, and Catalan research projects, of which six of them were led by him. His research interests include software-defined networks (SDN) and Network Functions Virtualization (NFV) applied to self-organized mobile networks.

Andrey Krendzel is a Senior Researcher at Huawei Technologies Oy (Finland), 5G Networks, since Oct 2013. From 1995 to 2013 he was involved in research activity related to xG networks ($x=2,3,4$) at the Centre Tecnològic de Telecomunicacions de Catalunya (CTTC), Tampere University of Technology (TUT), and Leningrad R&D telecommunication centre (LONIIS). He received the PhD degree from TUT in 2005 and the licentiate degree (Candidate of Science) from St.-Petersburg University of Telecommunications (SUT) in 2001.

Manuel Requena Estes (1974) is M.Sc. in Computer Science from the Technical University of Valencia (1998). He realized his master thesis in the ENSEA (France). He worked as Software Engineer developing Telecommunication Software Solutions (Atos Origin Integration, 1998 - 2003). Currently, he is the coordinator of the EXTREME Testbed in the Mobile Networks Department (CTTC). Since 2003, he has participated in industrial and public-funded projects in wireless, mobile and optical networking.

Albert Cabellos received a BSc (2001), MSc (2005) and PhD (2008) degree in Computer Science Engineering from the Technical University of Catalonia (www.upc.edu). In September 2005 he became an assistant professor of the Computer Architecture Department. He is an editor of the Elsevier Journal on Nano Computer Network and founder of the ACM NANOCOM conference, the IEEE MONACOM workshop and the N3Summit. He has also founded the LISPmob open-source initiative (<http://lispmob.org>) along with Cisco. He has been a visiting researcher at Cisco Systems and Agilent Technologies and a visiting professor at the Royal Institute of Technology (KTH) and the Massachusetts Institute of Technology (MIT). He has participated in several national (Cicyt), EU (FP7), USA (NSF) and industrial projects (Samsung and Cisco). He has given more than 10 invited talks (MIT, Cisco, INTEL, MIET, Northeastern Univ. etc.) and co-authored more than 15 journal and 40 conference papers. His main research interests are future architectures for the Internet and nano-scale communications.