Distributed Mobility Management for Future 5G Networks: Overview and Analysis of Existing Approaches

Fabio Giust, Luca Cominardi, and Carlos J. Bernardos

Fabio Giust and Carlos J. Bernardos are with University Carlos III of Madrid.

Luca Cominardi is with University Carlos III of Madrid and IMDEA Networks Institute.

The research leading to these results has received funding from the European Community's Seventh Framework Program FP7/2007-2013 under grant agreement 317941 — project iJOIN. The European Union and its agencies are not liable or otherwise responsible for the contents of this document; its content reflects the view of its authors only.

ABSTRACT

The ever-increasing demand of mobile Internet traffic is pushing operators to look for solutions to increase the available bandwidth per user and per unit of area. At the same time, they need to reduce the load in the core network at a reasonable cost in their future 5G deployments. Today's trend points to the deployment of extremely dense networks in order to provide ubiquitous connectivity at high data rates. However, this is hard to couple with the current mobile networks' architecture, which is heavily centralized, posing difficult challenges when coping with the foreseen explosion of mobile data. Additionally, future 5G networks will exhibit disparate types of services, posing different connectivity requirements. Distributed mobility management is emerging as a valid framework to design future mobile network architectures, taking into account the requirements for large traffic in the core and the rise of extremely dense wireless access networks. In this article, we discuss the adoption of a distributed mobility management approach for mobile networks, and analyze the operation of the main existing solutions proposed so far, including a first practical evaluation based on experiments with real Linux-based prototype implementations.

INTRODUCTION

In the recent years, Internet data communications have experienced a paradigm shift from the traditional fixed cable access to the wireless and mobile world. The huge success of powerful handheld devices and the deployment of faster heterogeneous radio access technologies, like IEEE 802.11n and Long Term Evolution (LTE), have led to the familiar concept of being *connected anywhere*, *anytime*. Reports such as [1] show that mobile traffic growth will not decelerate; conversely, it will increase 11-fold from 2013 to the end of 2018.

Mobile operators, together with industry and research communities, are looking at cheap and effective solutions to cope with this tremendous growth. There are two main issues to tackle:

- How to provide enough capacity in the access
- How to handle all the traffic in the transport network

For the first issue, reducing the size of cells is the most feasible approach that can provide a significant bandwidth increase. Regarding the second issue, current architectures for mobile and cellular networks are highly centralized and hierarchical, forcing user traffic to traverse all the network parts up to the core, where key entities are deployed to function as border IP gateways and mobility anchors. Following this approach, the general packet radio service (GPRS) Tunneling Protocol (GTP) [2] and Proxy Mobile IPv6 (PMIPv6) [3] have been adopted as two possible choices to operate the Evolved Packet Core (EPC) of 4G networks. The advantage of the centralized approach resides in its simplicity, because the central anchor can follow user movements by simply rerouting the packets over tunnels created with the access router where the mobile node (MN) is currently connected. However, the mobility anchor represents a single point of failure, poses scalability issues (i.e., it is the cardinal point for the control and data plane for millions of users), and, in general, leads to suboptimal paths between MNs and their communication peers (also known as correspondent nodes, CNs) [4].

Therefore, future 5G mobile networks are expected to be more flexible, relaxing the constraint of binding user traffic to a central core entity and allowing Internet services to be located closer to the users. Extremely dense wireless deployments shall benefit from such features by reducing the congestion in the operator's core infrastructure and providing improved service to users. Another defining characteristic of future 5G networks is that the infrastructure is expected to simultaneously serve very different sets of users and applications. For example, 5G networks are foreseen to share resources to cope with both highly demanding video applications of a few mobile users and low-bit-rate traffic from a large bunch of sensors (the so-called Internet of Things, IoT). Along with these objectives, distributed mobility management (DMM) has recently emerged as a new paradigm to

0163-6804/15/$25.00 © 2015 IEEE

design a flat and flexible mobility architecture, allowing traffic to be broken out locally closer to the edge (i.e., offloading the network core) and exploiting the use of different gateways for traffic with different connectivity and mobility requirements.

In this article, we argue that DMM approaches are suitable candidates for mobility management in future 5G very dense deployments. Then we explore the DMM solution space by focusing on the main three families of solutions currently proposed:

- A protocol derived from a classical IP mobility management approach, PMIPv6
- A mechanism based on software defined Nnetworking (SDN)
- A routing-based solution

We describe in this article the main characteristics of each of these DMM approaches and then conduct a validation and performance assessment of each of them by implementing the three solutions in a real prototype. Finally, we derive some interesting conclusions from the comparison of the obtained results. In this work we focus on the comparison of DMM-only solutions, but readers interested in a centralized vs. distributed study might consider the analysis reported in [5].

Distributed Mobility Management

The deployment of extremely dense radio networks addresses the need to expand the network capacity, offering an increased bandwidth per user per unit of area. The cellular pico and femto cells, in conjunction with the new advances in the IEEE 802.11 family, like the .11n and .11ac amendments, are speeding up the development in this direction.

Within this context, the current mobile architecture's centralized model poses some scalability issues due to traffic and signaling handling. For instance, in the Evolved Packet System (EPS) architecture, traffic generated in the radio access network (RAN) is conveyed by intermediate nodes called serving gateways (S-GWs) to the packet data network gateway (P-GW) by means of tunneling. The P-GW hence aggregates the traffic from several edge networks and acts as a gateway between the operator's network and external IP networks. While the deployment of extremely dense wireless networks tackles the expected traffic growth in the access part, a solution is also necessary for the core. In this sense, a flatter mobile network is best suited, as it would permit traffic to be routed without traversing core links unless necessary. Moreover, future 5G networks will simultaneously serve traffic from multiple devices with disparate requirements, as, for example, the IoT is expected to increase its footprint in the coming years. This fact requires more flexible network architectures capable of coping with multiple flows with different requirements, and dynamically adapting to the current demands.

The Third Generation Partnership Project (3GPP)[1] has already started developing solutions for the EPS to avoid tying IP connections to core gateways, like the Local IP Access and Selected IP Traffic Offload (LIPA-SIPTO) techniques [6] and LIPA Mobility and SIPTO at the Local Network (LIMONET) [7]. Consequently, IP networks appear closer to user terminals, since the complex operator's backhaul and core infrastructure might be bypassed. Very dense wireless networks could take advantage of this scenario, as they can be deployed in campuses, malls, transportation systems, and so on, which can benefit from having a locally available connection to Internet services (called a local breakout point) so that traffic generated locally is not forced to pass through the core network. In addition, users should join and leave any of these networks without experiencing any service interruptions, enjoying transparent mobility support for those applications that require so.

The DMM paradigm embraces the concepts expressed above, aiming at designing a flat mobile architecture that enables enhanced access to IP services and built-in support for mobility and heterogeneous radio access technologies [4, 8, 9]. The DMM framework envisions an all-IP infrastructure where users' data flows are routed through the optimal path, exploiting multiple anchor points and deployment of IP services closer to the users. Note that this framework envisions supporting mobility across heterogeneous networks, without requiring complex dedicated support from MNs. In addition, a wise assignment of IP addresses to MNs according to the available services for each user provides a mobile operator with the flexibility to handle users' data traffic according to an extended set of policies, such as whether IP flows should be anchored locally (e.g., for short-term sessions) or to a centralized node (for long sessions). This feature — which is very attractive for future 5G deployments, as previously discussed — is known as *prefix coloring* [10], and consists of attributing some meta-data to the assigned prefixes so that they can be used to access particular services, differentiated for the geographical location or by other means depending on the operator's policy.

Description of the DMM Solutions

There have been several DMM approaches proposed so far, spanning from extensions of current standardized protocols to clean-slate solutions. In this section, we describe the operation of the three main families of DMM approaches. These are the most important types of solutions, approaching the elimination of a single mobility anchor from disparate perspectives. Since there are more than one possible solution fitting each of the families, we have selected one per category, and we argue that the obtained conclusions also apply to any other solution from the same family.

The first family of solutions is based on modifications of classical IP mobility protocols, in particular of the well known PMIPv6; thus, in the following we refer to the solution belonging to this family as the *PMIPv6-based DMM solution*; the second category follows an SDN paradigm, so we call the protocol from this cate-

Since there is more than one possible solution fitting each of the families, we have selected one per category, and we argue that the obtained conclusions also apply to any other solution from the same family.

[1] http://www.3gpp.org/

In our PMIPv6-based DMM solution, the MAG role is replaced by the DMM-Gateway. A DMM-GW evolves from a MAG as it is provided with links to the Internet that do not imply paths traversing the LMA. Hence, the DMM-GW acts as a plain access router to forward packets to and from the Internet.

gory an *SDN-based DMM solution*; and the third design leverages on IP routing protocols, hence the name *routing-based DMM solution* for the mechanism within this family. The three groups are made of network-based mobility management protocols, so no mobility client is required on the terminal. However, the first and third groups make extensive use of existing Internet Engineering Task Force (IETF)[2] standards, whereas the second is a clean-slate approach. All of them share the concept that the access router not only provides connectivity to the MNs (by being their default gateway), but are also enhanced with some specific DMM features. For this reason, throughout the text we refer to a DMM-enabled access router as a DMM-Gateway (DMM-GW).

PMIPv6-Based DMM Solution

In the next paragraphs we present a simplified description of the PMIPv6-based DMM solution described in more detail in [11], where the full protocol details can be found. Since this solution inherits many of its features from Proxy Mobile IPv6 (PMIPv6), we briefly describe this latter first. Proxy Mobile IPv6 is a centralized mobility management protocol where a core entity called the local mobility anchor (LMA) establishes bidirectional tunnels with mobility access gateways (MAGs) located in the access networks. Users' upstream data packets are collected by the corresponding MAG and sent through the tunnel to the LMA, which in turns forwards them to the Internet. Similarly, downstream packets are first received by the LMA, which then dispatches them through the tunnel terminating at the MAG to which the MN is currently attached. By using dedicated signaling messages, called Proxy Binding Update (PBU) and Proxy Binding Acknowledgement (PBA), between the MAG and the LMA, the PMIPv6 protocol coordinates the status of the network, letting the LMA know at which MAG an MN is connected to properly route its traffic. Indeed, since the LMA is traversed by users' data flows, it is straightforward for it to redirect the packets to the appropriate tunnel upon handover, based on the indications received from the MAGs. However, in this way, the data path may end up being suboptimal, and the LMA must be provisioned with high-speed and redundant links to the MAGs in order to convey the traffic for all the subscribers.

In our PMIPv6-based DMM solution, the MAG role is replaced by the DMM gateway. A DMM-GW evolves from a MAG as it is provided with links to the Internet that do not imply paths traversing the LMA. Hence, the DMM-GW acts as a plain access router (i.e., no tunneling) to forward packets to and from the Internet. Also, a DMM-GW features mobility anchoring functions, being able to forward without disruption the IP flows that an MN started while attached to it before moving to a new DMM-GW afterward. Moreover, PMIPv6's LMA is reduced to a control plane only entity, referred to as the control mobility database (CMD). The CMD stores, for every MN, all the prefixes advertised to the MN, which DMM-GW advertised each prefix, and to which DMM-GW the MN is currently connected. In addition, by means of extended PBU/PBA signaling, the CMD sends instructions to recover the MN's ongoing IP flows after a handover. As a result, this architecture's scalability is improved with respect to PMIPv6, as DMM-GWs are able of locally breaking out some traffic, thus avoiding the need to traverse the network core. This allows the overprovisioning typically performed when designing the aggregation links from the access to the network core to be reduced.

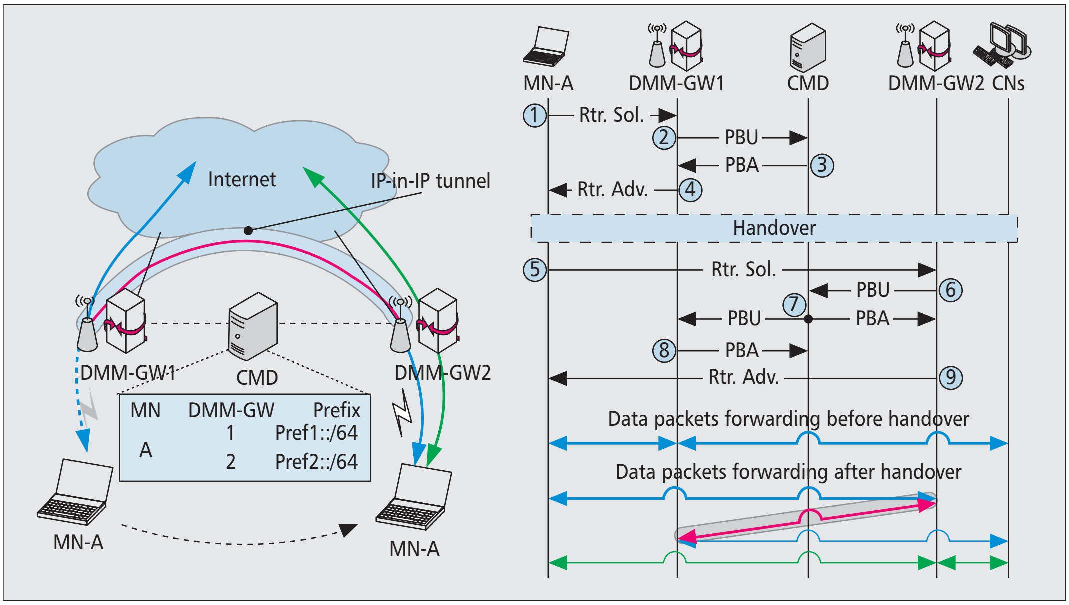
More details and the operations of the PMIPv6-based DMM solution are shown on the right side of Fig. 1, represented by circled numbers. A DMM-GW detects the MN attachment typically after receiving a Router Solicitation message [12] from the MN, or by means of a dedicated link detection mechanism, ①. Next, the DMM-GW notifies the CMD about the MN attachment by means of extended PBU/PBA signaling, ②③, which also contains the IPv6 prefix the DMM-GW is allocating for the MN. Since this is a fresh registration, the CMD creates a new entry for the MN, storing a pointer to the MN's current location (i.e., the DMM-GW that generated the signaling) and a field for the prefix assigned. The DMM-GW advertises the prefix to the MN in a Router Advertisement (RA) message ④ [12]. After a handover, ⑤, when the CMD receives the PBU from the new DMM-GW, ⑥, the database entry for the MN is updated, associating the MN's location with the new serving DMM-GWs. In addition, using the PBU/PBA signaling the CMD instructs the serving and old DMM-GWs to establish a tunnel between them, ⑦⑧. The tunnel is necessary to redirect ongoing IP flows anchored at the old DMM-GW to and from the new DMM-GW. However, the tunnel carries the packets only for those flows that were started before the MN handed over from the previous DMM-GW, whereas new communications are handled by the new DMM-GW as a plain router (i.e., without using any tunnels). This dynamic flow handling is achieved by assigning a new IPv6 prefix to the MN from each DMM-GW to which it connects. The prefix is announced by the DMM-GWs with an RA, ④⑨, which forces the MN to use the new prefix (advertised by the DMM-GW where the MN is currently connected) for new communications. Therefore, each DMM-GW is responsible for a pool of IPv6 prefixes from which it delegates one to each MN attached to its access links. Thereby, a DMM-GW handles users' packets selectively with or without encapsulation, depending on the IPv6 prefix they carry and where the MN is currently connected. Consequently, an MN configures several IPv6 address, one per each visited DMM-GW, and its flows might be anchored at different DMM-GWs.

According to the DMM terminology, this protocol falls within the *partially distributed* category. Indeed, the data plane is distributed among the DMM-GWs, and the control plane is kept centralized, tied to the role of the CMD.

SDN-Based DMM Solution

Software defined networking is a networking paradigm that separates the control and data forwarding planes. Such separation allows for quicker provisioning and configuration of net-

[2] http://www.ietf.org/



MN	DMM-GW	Prefix
A	1	Pref1::/64
	2	Pref2::/64

Figure 1. PMIPv6-based DMM: overview and operations.

With SDN, network administrators can program the behavior of both the traffic and the network in a centralized way, without requiring independent accessing and configuring each of the network's hardware devices.

work connections. With SDN, network administrators can program the behavior of both the traffic and the network in a centralized way, without requiring independent accessing and configuring each of the network's hardware devices. This approach decouples the system that makes decisions about where traffic is sent (i.e., control plane) from the underlying system that forwards traffic to the selected destination (i.e., data plane). Among other advantages, this simplifies networking as well as the deployment of new protocols and applications. In addition, by enabling programmability on the traffic and devices, an SDN network might be much more flexible and efficient than a traditional one.

In SDN environments, the network controller is the most important entity and is responsible for configuring the nodes in the network via a common application programming interface (API), the *Southbound API*. OpenFlow[3] is one API that can be used by an external software application to program the forwarding plane of network devices. The operations of the SDN-based DMM solution are shown on the right side of Fig. 2. In our solution, a core entity, called the network controller (NC), configures the forwarding rules on access routers (the DMM-GWs) using the OpenFlow 1.3 API. The DMM-GWs play the role of anchors. Upon the attachment of an MN to an access point, ①, the DMM-GW informs the NC, ②⑦, which assigns a network prefix to the MN, ④⑤⑨⑩. The network prefix is guaranteed to be unique by using a binding cache where the controller stores, similar to the PMIPv6-based solution, information about the MNs connected to the network. After attachment detection, the NC configures the OpenFlow rules in each DMM-GW visited by the MN, ③⑧.

Mobility is achieved by combining translation and forwarding rules on DMM-GWs. When a packet of an anchored flow reaches a visited DMM-GW, the anchor first rewrites the IP destination address with the last known MN's IP address and then redirects the traffic in the new MN's location. When the traffic reaches the last visited DMM-GW, the DMM-GW performs a reverse IP address translation first, restoring the old IP destination address, and then forwards the traffic to the MN. Note that this solution, unlike the PMIPv6-based one, does not involve any IP tunnels.

This solution, like the PMIPv6-based DMM, is *partially distributed*. While the data plane is distributed, the traffic does not pass through any centralized gateways, and the control plane is centralized at the network controller.

Routing-Based DMM Solution

The basic concept of this type of solutions is to remove any anchor from the architecture, letting all the network nodes re-establish a new routing map when terminals move by means of IP routing protocols. For the purposes of this analysis, we take the solution proposed in [13], which builds on top of the Border Gateway Protocol (BGP) [14] and the Domain Name System (DNS). This is achieved by enabling BGP on the access routers (the DMM-GWs) so that they propagate upward to their BGP peer routers the changes in the access links.

The operations of the routing-based DMM solution are shown in the right side of Fig. 3. Upon an MN attachment to a DMM-GW's access link, ①⑤, the access router learns the MN's DNS name after authentication, ②⑥. Next, the DMM-GW retrieves the IP address (and consequently the IPv6 prefix) associated with the MN's DNS record and announces itself as a next hop to reach the MN's prefix, ③⑦. By doing so, the DMM-GW triggers a BGP routing update in the rest of the network, ④⑧. When the BGP procedure converges, the MN is reachable at the new location using a new path within the network, as depicted in Fig. 3.

It is worth noting that this protocol is fully distributed, in the sense that both the data and control planes are not bound to a specific cen-

[3] https://www.opennetworking.org/

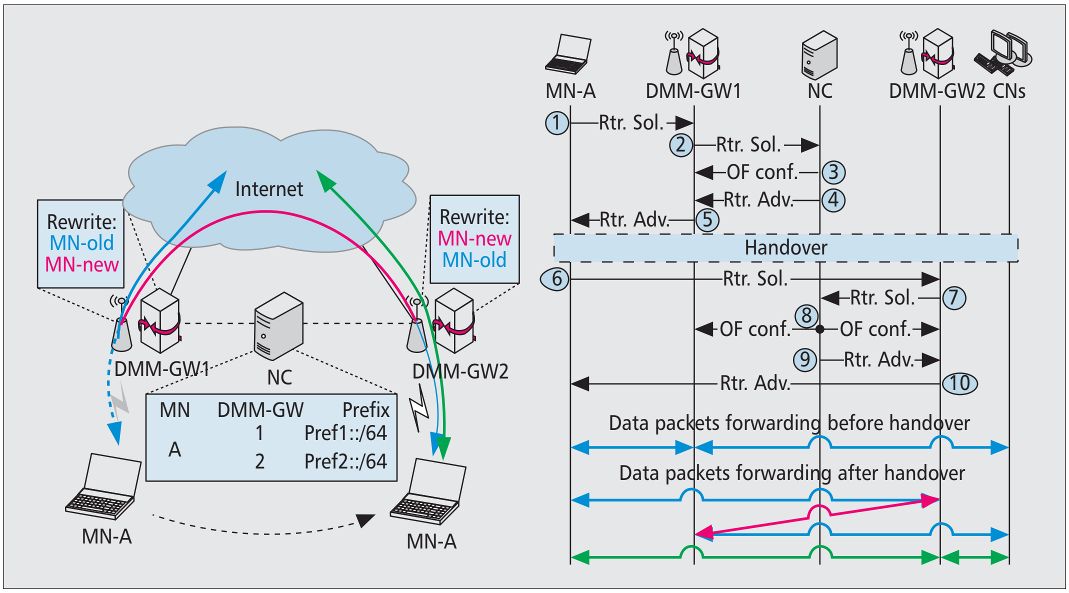


Figure 2. SDN-based DMM: overview and operations.

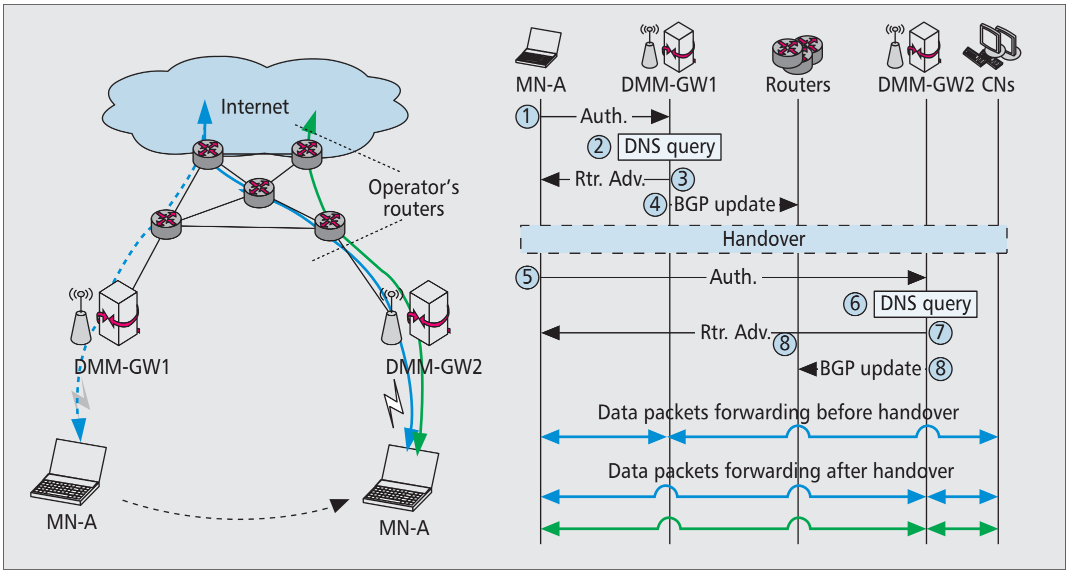


Figure 3. Routing-based DMM: overview and operations.

It should be noted that none of our DMM solutions requires any change on the MN. Indeed, the IP session continuity is provided without any intervention by the MN beyond the Neighbor Discovery operations, which are part of the standard IPv6 stack.

tralized node, but are instead handled by the routers in a distributed way.

Evaluation of the DMM Solutions

After describing how each of the three DMM solutions works, we now report on an evaluation aimed at identifying their advantages and disadvantages, as well as accomplishing an initial performance evaluation.

Table 1 presents a summary of the main characteristics of the DMM analyzed approaches, with a qualitative comparison between them. We next provide some proofs supporting the statements presented in the table.

We have implemented the three solutions described before in order to conduct an experimental performance assessment. The objective of this work is to carry out a proof of concept of these DMM approaches, showing their feasibility with real equipment. Each prototype has been implemented and assessed on the same common platform. The testbed has been realized as a set of three DMM-GWs providing WLAN access using IEEE 802.11b/g cards and a machine acting as CMD, NC, and DNS server. All the nodes run the GNU/Linux operating system. For the sake of clarity, employing 802.11 as the access technology does not affect the functional behavior of the prototypes because, as previously described, the considered DMM approaches are IP-based and layer 2 agnostic. Therefore, the use of other link layer protocols does not have any significant effect on the results.

The PMIPv6-based DMM prototype employs the Mobility Anchors Distribution for PMIPv6 (MAD-PMIPv6)[4] implementation, which runs in the DMM-GWs and the CMD. The code is writ-

[4] http://www.odmm.net/mad-pmipv6

	PMIPv6-based DMM	SDN-based DMM	Routing-based DMM
Type of DMM	Partially distributed (central mobility database)	Partially distributed (SDN controller)	Fully distributed
MN's multiple IP addresses	Mandatory	Mandatory	Supported
Mobility anchors	Multiple (depends on IP flows generation)	Multiple (depends on IP flows generation)	None
IPv6 in IPv6 tunneling	Yes	No	No
Route optimization	No support for anchored IP flows	No support for anchored IP flows	Yes for all IP flows
Handover latency	Low	Low	High
Signaling overhead	Low (depends on no. of active anchors)	Low (depends on no. of active anchors)	High (depends on no. of routers)

Table 1. Features of the three DMM solutions.

ten in ANSI C and provides all the features described earlier.

The SDN prototype employs Open vSwitch[5] as an OpenFlow implementation on DMM-GWs, and Ryu[6] as an OpenFlow-capable SDN framework on the NC. The SDN-based DMM solution is written in Python on top of Ryu's API and provides all the features described previously.

The BGP prototype extends the testbed with a "core" network formed by five routers. The DMM-GWs and "core" routers run the BGP protocol implemented within the Quagga project.[7] An additional piece of software, written in ANSI C, is deployed on the DMM-GWs. This software detects MN attachments and detachments, retrieves the MNs' names and addresses from the DNS server, and installs the local route. When a change in the routing table is detected, the BGP daemon propagates the information to all the other routers. The DNS service is provided by Bind[8] running on the DNS server.

Experimental Results

All the prototypes exhibit three DMM-GWs, each providing WLAN access via a co-located IEEE 802.11b/g access point (AP). The objective of the experiments is to observe how an MN reacts when a handover occurs, that is, what happens to the data traffic when the MN moves from one AP to the other. Therefore, an additional host is deployed, which role is to act as CN, generating *ping* traffic destined to the MN. *Ping* packets must traverse the prototypes once for the request to be delivered to the MN and another time for the reply sent by the MN to the CN. It should be noted that none of our DMM solutions requires any change on the MN. Indeed, the IP session continuity is provided without any intervention by the MN beyond the neighbor discovery operations, which are part of the standard IPv6 stack. We used a laptop as MN, with an out-of-the-box GNU/Linux system (Debian Wheezy OS) and the built-in WLAN card for the wireless access.

For each implementation, we have measured the time required to:

Perform a layer 2 switch, that is, the latency for the MN to change from one AP to another, measured as the time spent for the IEEE 802.11 operations to dissociate from the old AP and associate with the new one.

Perform a layer 3 handover, that is, the time spent since the dissociation from the old AP to the instant when a Router Advertisement is received by the MN, meaning that the MN's IP configuration is ready.[9]

Recover ping traffic, that is, the interval between the last ping packet received or sent by the MN before the handover and the first ping packet received or sent after the handover.

These measurements have been collected by capturing the traffic at the MN's WLAN interface for more than 200 handovers for each platform.

Table 2 showcases the values in milliseconds of the mean value and standard deviation for the three types of handover obtained on the different platforms. Figure 4 depicts in more detail the handover distribution for the ping traffic.

As expected, the layer 2 switch and layer 3 handover are low for all the solutions: this is because the operations performed by the network in order to re-assign the IP connectivity to the MN are very quick. Indeed, in all the schemes, upon detecting the MN attachment, the DMM-GW queries a database in order to retrieve the parameters for the MN's IP configuration. The database is either the CMD, the SDN controller, or the DNS server, respectively, for the PMIPv6-based, SDN-based, and routing-based solutions.

The main difference resides in the time required to recover ongoing data flows, as both the PMIPv6-based and SDN-based solutions are almost 100 times faster than the routing-based protocol. The reason is that the PMIPv6-based and SDN-based mechanisms operate in a conceptually similar way. Indeed, at the time the central database is queried (either the CMD or the NC), this latter sends instructions using the corresponding signaling to the DMM-GWs in order to immediately re-establish a routing path

[5] http://openvswitch.org/

[6] http://osrg.github.io/ryu/

[7] http://www.nongnu.org/quagga/

[8] http://www.bind9.net/

[9] For the sake of simplicity, we do not consider the duplicate address detection process that should be performed after configuring an IPv6 address on an interface.

	Handover type					
	Layer 2 switch		Layer 3 handover		Ping recovery	
	Mean (ms)	Std. sev. (ms)	Mean (ms)	Std. sev. (ms)	Mean (ms)	Std. sev. (ms)
PMIPv6-based DMM	14.0	4.3	26.4	6.7	38.2	10.8
SDN-based DMM	14.0	4.3	35.7	6.7	43.2	7.9
Routing-based DMM	14.0	4.3	59.0	5.9	4743.9	777.1

Table 2. Experimental handover latency results.

for ongoing data flows. In the case of the PMIPv6 solution, the new routing path is achieved with an IPv6-in-IPv6 tunnel between the old anchor and the new one, while in the SDN case new switching rules are installed in the old and new anchors. Hence, both the CMD and NC have an active role in the control plane. On the contrary, the routing-based system does not delegate any control role to the DNS server. Thus, when the access router receives the MN's IP address from the DNS server, the router installs a route for the MN's prefix and sends to the other routers a BGP update notifying itself as next-hop for the announced prefix. Therefore, in order to recover the data flow, all the routers involved in the old data path and those involved in the new target data path must be updated with the correct routing entry, leading to a few seconds latency to let the routing protocol converge. The impact that this might have when the solution runs on large domains is not negligible.

Final Remarks

After giving some figures on the handover latency produced by each solution, it is worth a brief analysis on the protocols' overhead by observing how the signaling messages proliferate during a location update.

In the PMIPv6-based solution, the CMD interacts with the new DMM-GW, and with each of the old DMM-GWs that is anchoring IP flows before the handover. Therefore, the signaling load varies with the traffic and mobility dynamics generated by the MN. In detail, the overhead introduced by the updated location grows with the number of IPv6 prefixes in use by the MN (i.e., the "active" prefixes) because each of them requires a signaling session with the corresponding DMM-GW that assigned the prefix. If the MN's mobility is low, or the IP sessions generated by the MN are short, the number of simultaneous active prefixes is low too, producing little signaling overhead. On the contrary, if the MN is visiting many access networks per unit of time, while keeping several long-lived applications that cannot survive an IP address change, the number of active prefixes is large and so the overhead. However, even if the number of active anchors is large, the latency introduced to recover a communication is impacted only by the distance of the furthest anchor.

The same reasoning applies to the SDN-based approach, leading to the same considerations as for the PMIPv6-based solution.

On the contrary, the number of messages sent by the routing-based DMM solution is determined by the size of the operator's network. Indeed, the DMM-GW needs to notify all of its BGP peers so that the amount of signaling messages sent is almost constant and determined by the number of BGP peers. In a large network, the number of BGP update messages increases dramatically unless adopting some expedients like BGP route reflectors. The time required to re-establish the data communication is affected by the number of BGP routers present in the new data path: as soon as this set of routers converges to the new routing state, the data traffic is recovered. With respect to the service differentiation a future mobile network should offer to the user, we note that the DMM solution proposed in this article may enforce this feature, exploiting a smart IPv6 prefixes assignment by the DMM-GWs. We have observed in the PMIPv6-based and SDN-based solutions that each DMM-GW visited by the MN assigns an IPv6 prefix to the MN used to configure an address at which Internet services can be accessed in a general way. DMM-GWs can assign additional prefixes to the MNs, specifically designated to access some services locally available at that DMM-GW, or addressing some other operator's policies. The routing-based solution is not excluded by this feature, as a DMM-GW can assign a specific prefix, in addition to the main one, to produce service differentiation. Sophisticated use of this technique can lead to a dynamic anchor assignment to the MN's IP flows. For instance, according to the operator's policies, an MN flow can be forced to use a determined prefix for specific IP flows so that the anchoring model (centralized or distributed) can be selected by the operator.

Conclusions

In this article we have focused on DMM as a suitable candidate framework for mobility management in future 5G networks. We have analyzed the DMM solution space by describing the three main solution families for distributing mobility management on a flat architecture for mobile networks.

These three solutions follow different approaches. The first is indeed an extension of a standard mobility protocol for the Evolved Packet System, called Proxy Mobile IPv6. The original protocol has been modified and extended to accommodate a new set of operations, so the outcome is distributed in nature. The second solution operates in a similar way to the previous one, but follows a software defined networking approach. The last mechanism employs the BGP routing protocol to perform the mobility functions required to deliver the packets to and from moving users.

These three types of proposals have been evaluated using real field experiments on Linux-based prototypes. Our findings confirm the intuition that the first two solutions react faster to the changes in the network, but they require dedicated signaling and specialized entities to perform the needed operations. The third mechanism relies on a well established routing protocol, inheriting the issues related to high convergence latency and signaling overhead when used on large network domains.

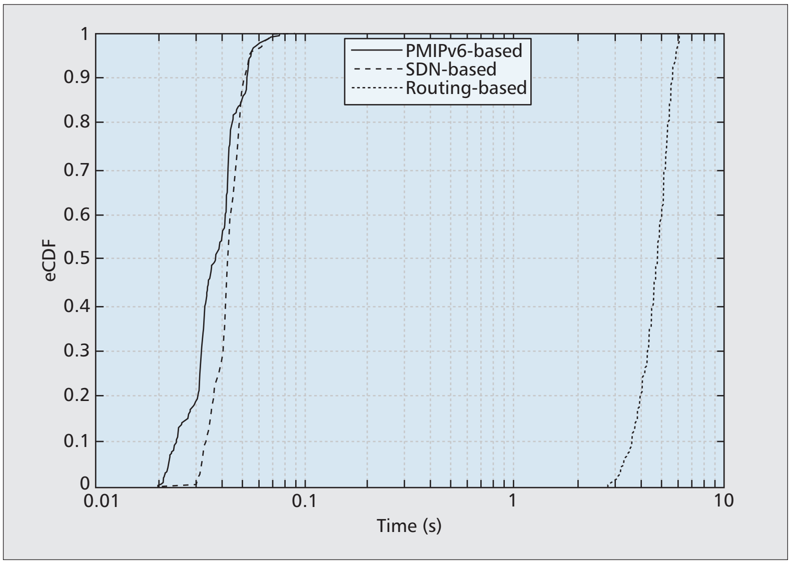


Figure 4. Empirical CDF of the handover measurements for ping traffic.

References

[1] Cisco, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2013-2018," White Paper, Feb. 2014.

[2] 3GPP TS 29.274, "Evolved General Packet Radio Service (GPRS) Tunnelling Protocol for Control plane (GTPv2-C)," Sept. 2011.

[3] S. Gundavelli *et al.*, "Proxy Mobile IPv6," IETF RFC 5213, Aug. 2008.

[4] H. Chan *et al.*, "Requirements for Distributed Mobility Management," IETF RFC 7333, Apr. 2014.

[5] F. Giust *et al.*, "Analytic Evaluation and Experimental Validation of a Network-based IPv6 Distributed Mobility Management Solution," *IEEE Trans. Mobile Computing*, vol. 13, no. 11, Nov. 2014, pp. 2484–97.

[6] 3GPP TR 23.829, "Local IP Access and Selected IP Traffic Offload (LIPA-SIPTO)," Oct. 2011.

[7] 3GPP TR 23.859, "LIPA Mobility and SIPTO at the Local Network," Apr. 2013.

[8] D. Liu *et al.*, "Distributed Mobility Management: Current practices and gap analysis," IETF Draft, draft-ietf-dmm-best-practices-gap-analysis-07, Sept. 2014.

[9] J. C. Zuniga *et al.*, "Distributed Mobility Management: A Standards Landscape," *IEEE Commun. Mag.*, vol. 51, no. 3, Mar. 2013, pp. 80–87.

[10] M. Le Pape, S. Bhandari, and I. Farrer, "IPv6 Prefix Meta-data and Usage," IETF Draft, draft-lepape-6man-prefix-metadata-00, July 2013.

[11] C. J. Bernardos, A. De La Oliva, and F. Giust, "A PMIPv6-Based Solution for Distributed Mobility Management," IETF Draft, draft-bernardos-dmm-pmip-03, Jan. 2014.

[12] T. Narten *et al.*, "Neighbor Discovery for IP version 6 (IPv6)," IETF RFC 4861, Sept. 2007.

[13] P. McCann, "Authentication and Mobility Management in a Flat Architecture," IETF Draft, draft-mccann-dmm-flatarch-00, Mar. 2012.

[14] Y. Rekhter, T. Li, S. Hares, "A Border Gateway Protocol 4 (BGP-4)," IETF RFC 4271, Jan. 2006.

Biographies

FABIO GIUST (fgiust@it.uc3m.es) received his Bachelor's and Master's degrees in telecommunications engineering at the University of Padova, Italy. After an internship at Alcatel-Lucent Bell Labs in France, he undertook a Master's in telematics engineering at University Carlos III of Madrid (UC3M), Spain. Currently he is working at UC3M, where he is also pursuing his Ph.D. His research interests cover IP mobility and wireless mobile networks, on which he has published several papers in international conferences and journals.

LUCA COMINARDI (luca.cominardi@imdea.org) received his Bachelor's and Master's degrees in computer science at the University of Brescia, Italy. He did an internship and undertook a Master's in telematics engineering at UC3M. Currently he is working at IMDEA Networks Institute and pursuing his Ph.D. at UC3M. His main research interests are SDN, NFV, and integration of the wireless medium into the two former.

CARLOS J. BERNARDOS (cjbc@it.uc3m.es) received a telecommunication engineering degree in 2003 and a Ph.D. in telematics in 2006, both from UC3M, where he worked as a research and teaching assistant from 2003 to 2008 and, since then, as an associate professor. His current work focuses on mobility in heterogeneous wireless networks. He has published over 50 scientific papers in international journals and conferences, and he is an active contributor to the IETF. He has served as Guest Editor of *IEEE Network*.