This is a postprint version of the following published document:


DOI: https://doi.org/10.1002/ett.3762

© 2019 John Wiley & Sons, Ltd.
SPECIAL ISSUE ARTICLE

A Flexible IPv6 Mobility Management Architecture for SDN-based 5G Mobile Networks

Muhammad Mohtasim Sajjad* | Dhammika Jayalath | Carlos J. Bernardos

1School of Electrical Engineering and Computer Science, Queensland University of Technology, Brisbane, Queensland, Australia
2Universidad Carlos III de Madrid. Avda de la Universidad, 30 E-28911 Leganés (Madrid), Spain

Correspondence
*Muhammad Mohtasim Sajjad, 2 George Street, Gardens Point, Queensland University of Technology, Brisbane, Queensland, 4001, Australia. Email: m.sajjad@qut.edu.au

Abstract
With the advent of increasingly diverse services, applications and use cases, the 5G mobility environment is expected to be highly heterogeneous. The Distributed Mobility Management (DMM) is emerging as a promising approach for 5G, which provides decentralized handover and traffic management at the edge of the network. However, in order to meet the disparate mobility needs, the DMM process requires flexible design considerations. This paper presents a novel Mobile Node (MN) centred, Software-Defined Networking (SDN) based flexible mobility management architecture, named Adaptive Multimode Mobility Management (A3M), which can adaptively operate in multiple modes. The proposed A3M architecture incorporates a novel Handover Mode Selection phase among the traditional handover phases during which a suitable mode of handover operation is evaluated. This makes the A3M handover process adaptable to the varying mobility requirements of the MN, and enables it to provide differentiated handover management for MNs with different mobility profiles. The performance analysis of A3M shows that it offers optimal handover performance in terms of primary handover performance metrics such as session disruption delay, packet losses and signalling costs, compared to the popular network-based DMM approaches.

KEYWORDS: 5G, Adaptive Mobility Management, Distributed Mobility Management, Software Defined Networking

1 INTRODUCTION

The desire to offer wide-ranging communication services with ultra-high data rates and imperceptible latencies, is pushing for the deployment of various complementary technologies in 5G networks. The 5G environment is thus envisioned to be a heterogeneous ecosystem of several technologies such as device-to-device (D2D) communications, 3D wireless, indoor and outdoor ultra dense networks (UDNs), vehicular networks, small cells and multi-tier networks. This gives rise to a plethora of
mobility management scenarios, which exhibit highly heterogeneous characteristics. Following are some example scenarios, a combination of which, as illustrated in Figure 1, characterizes the diversity of 5G mobility use cases.

- The user can be stationary, or roaming at different speeds as a pedestrian, cyclist, or at vehicular or high-speed railway services.

- The mobile user may undergo handover in a traditional macro cell deployment, ultra-dense small cell, crowded environment supported by 3D wireless, multi-tier network, or while engaged in a vehicular or D2D communications scenario.

- The user may not have any active ongoing session, or an active session classified as Non-Real-time (e.g. web browsing, email), Conversational (e.g. a VoIP call or video conferencing), or non-Conversational multimedia session (e.g. an adaptive streaming application).

- The session requirements of the ongoing sessions may also differ highly, and may demand, for instance, ultra-high data rates (e.g. for augmented reality/virtual reality services), ultra-high reliability (e.g. Vehicle-to-Vehicle (V2V) communications), and ultra-low latency communications (e.g. tactile internet).
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3D Wireless</td>
<td>Three Dimensional Wireless</td>
</tr>
<tr>
<td>3GPP</td>
<td>3rd Generation Partnership Project</td>
</tr>
<tr>
<td>5G</td>
<td>5th Generation of Mobile Networks</td>
</tr>
<tr>
<td>A3M</td>
<td>Adaptive Multimode Mobility Management</td>
</tr>
<tr>
<td>A3M-P</td>
<td>A3M-Predictive</td>
</tr>
<tr>
<td>A3M-R</td>
<td>A3M-Reactive</td>
</tr>
<tr>
<td>AMF</td>
<td>Access and Mobility Management Function</td>
</tr>
<tr>
<td>AP/AR</td>
<td>Access Point/Access Router</td>
</tr>
<tr>
<td>BCE</td>
<td>Binding Cache Entry</td>
</tr>
<tr>
<td>BU</td>
<td>Binding Update</td>
</tr>
<tr>
<td>CDN</td>
<td>Content Delivery Network</td>
</tr>
<tr>
<td>CMD</td>
<td>Central Mobility Database</td>
</tr>
<tr>
<td>CN</td>
<td>Correspondent Node</td>
</tr>
<tr>
<td>CTR</td>
<td>(SDN) Controller</td>
</tr>
<tr>
<td>CUPS</td>
<td>Control and User Plane Split</td>
</tr>
<tr>
<td>D2D</td>
<td>Device-to-Device</td>
</tr>
<tr>
<td>DASH</td>
<td>Dynamic Adaptive Streaming over HTTP</td>
</tr>
<tr>
<td>DBC</td>
<td>Database Update Cost</td>
</tr>
<tr>
<td>DMM</td>
<td>Distributed Mobility Management</td>
</tr>
<tr>
<td>FE</td>
<td>Forwarding Entity</td>
</tr>
<tr>
<td>FHMIPv6</td>
<td>Fast Hierarchical Mobile IPv6</td>
</tr>
<tr>
<td>FMIPv6</td>
<td>Fast Handovers for Mobile IPv6</td>
</tr>
<tr>
<td>FPMIPv6</td>
<td>Fast Handovers for Proxy Mobile IPv6</td>
</tr>
<tr>
<td>HA</td>
<td>Home Agent</td>
</tr>
<tr>
<td>HMIPv6</td>
<td>Hierarchical Mobile IPv6</td>
</tr>
<tr>
<td>HTTP</td>
<td>Hyper Text Transfer Protocol</td>
</tr>
<tr>
<td>IETF</td>
<td>Internet Engineering Task Force</td>
</tr>
<tr>
<td>L2</td>
<td>Layer 2/Link-layer</td>
</tr>
<tr>
<td>LMA</td>
<td>Local Mobility Anchor</td>
</tr>
<tr>
<td>MAAR</td>
<td>Mobility Anchor and Access Router</td>
</tr>
<tr>
<td>MADM</td>
<td>Multiple Attributes Decision Making</td>
</tr>
<tr>
<td>MAG</td>
<td>Media Access Gateway</td>
</tr>
<tr>
<td>MAP</td>
<td>Mobility Anchor Point</td>
</tr>
<tr>
<td>MIPv6</td>
<td>Mobile IPv6</td>
</tr>
<tr>
<td>MME</td>
<td>Mobility Management Entity</td>
</tr>
<tr>
<td>MN</td>
<td>Mobile Node</td>
</tr>
<tr>
<td>MR</td>
<td>Mobile Router</td>
</tr>
<tr>
<td>NEMO</td>
<td>Network Mobility</td>
</tr>
<tr>
<td>n-FE</td>
<td>new Forwarding Entity</td>
</tr>
<tr>
<td>n-MAAR</td>
<td>new MAAR</td>
</tr>
<tr>
<td>N-PMIPv6</td>
<td>NEMO-enabled PMIPv6</td>
</tr>
<tr>
<td>PBA</td>
<td>Proxy Binding Acknowledgement</td>
</tr>
<tr>
<td>PBU</td>
<td>Proxy Binding Update</td>
</tr>
<tr>
<td>p-FE</td>
<td>previous FE</td>
</tr>
<tr>
<td>p-MAAR</td>
<td>previous MAAR</td>
</tr>
<tr>
<td>PMIPv6</td>
<td>Proxy Mobile IPv6</td>
</tr>
<tr>
<td>PMIPv6-DMM</td>
<td>Proxy Mobile IPv6 based DMM</td>
</tr>
<tr>
<td>PO</td>
<td>Path Optimization</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>ROH</td>
<td>Route Optimized Handover</td>
</tr>
<tr>
<td>SAW</td>
<td>Simple Additive Weighting</td>
</tr>
<tr>
<td>SBA</td>
<td>Service-based Architecture</td>
</tr>
<tr>
<td>SDN</td>
<td>Software-Defined Networking</td>
</tr>
<tr>
<td>TD</td>
<td>Tunnelling Delay</td>
</tr>
<tr>
<td>UDN</td>
<td>Ultra Dense Network</td>
</tr>
<tr>
<td>URLLC</td>
<td>Ultra-Reliable Low Latency Communications</td>
</tr>
<tr>
<td>V2V</td>
<td>Vehicle-to-Vehicle</td>
</tr>
<tr>
<td>VANET</td>
<td>Vehicular Ad Hoc Networks</td>
</tr>
<tr>
<td>VMA</td>
<td>Virtual Mobility Anchor</td>
</tr>
<tr>
<td>VoIP</td>
<td>Voice over IP</td>
</tr>
</tbody>
</table>
It has been recognized that, for such diverse and complex communication paradigm, a uniform network deployment will not be able to fit all scenarios and use cases. Hence, the 5G networks require flexibility, not only in the network topology and architecture, but also in the management protocols to ensure efficient provisioning of desired services. In particular, flexibility in the mobility management protocols is crucial since they would be required to handle mobility for end-users with constantly changing roaming patterns, application requirements and preferences.

The software defined networking (SDN) technology is a key enabler for flexibility in 5G networks. The recent 3GPP system specifications, which feature the Control and Data plane split architectures, support the adoption of SDN for 5G. These specifications include the Control and User Plane Split (CUPS) architecture (Release 14) and a most-recent Service-based Architecture (SBA) (in Release 15). The softwarization of 5G systems through SDN introduces new prospects, not only for flexible network deployment, but also for flexible management and operation of mobile networks.

The aforementioned technological advancements and the anticipated capabilities of 5G networks demand new mobility management solutions. To this end, the Distributed Mobility Management (DMM) has emerged as a prominent approach in recent years, which preserves the existing IPv6 mobility management principles and considers Control/Data plane split and flatter network architectures to meet 5G requirements. It is thus able to provide low-latency handover and traffic management at the edge of the network. Based on the envisaged softwarization of 5G networks, some of the DMM solutions have also evolved towards SDN. These solutions, although offer several benefits, have not been able to address the diverse mobility requirements of the 5G environment. In order to effectively deal with constantly varying mobility conditions, these solutions require flexibility features for their adaptive operation.

The need for the adaptive operation of IPv6 mobility management protocols has been identified in the past, and several solutions have been proposed. These solutions aim to execute the handover protocol in a particular mode of operation which best suits the current mobility needs of the MN. However, such solutions only offer among the alternate options for the protocol’s execution and are thus not suitable enough for rather diverse mobility requirements in a 5G mobile environment.

Contribution

This paper presents a MN-centred, SDN-based flexible mobility management architecture, named Adaptive Multimode Mobility Management (A3M) which can adaptively operate in multiple modes. The main contribution of the proposed A3M solution is the introduction of a novel concept of Handover Mode Selection phase, which co-occurs during a handover process among the traditional handover phases. During the Handover Mode Selection phase, a suitable mode of handover operation is proactively evaluated for the imminent handover event. The comparative analysis of the proposed protocol shows that it offers optimal handover performance compared to the PMIPv6-based DMM (PMIPv6-DMM) solutions, which are currently being standardized by the IETF.
The proposed concept of Handover Mode Selection phase, as we discuss later, can be further explored for a variety of emerging mobile environments to achieve adaptive mobility. However, in this paper, we focus on a generic DMM network topology, as considered at the IETF standardization.

Paper Organization

The rest of this paper is organized as follows. Section II provides an overview of the existing work on DMM, as well as the adaptive mobility management solutions for IPv6-based mobility protocols. Section III describes the principles of the proposed architecture along with a detailed description of its operation. Section IV presents the analytical models for performance evaluation for both PMIPv6-based DMM protocol as well as the proposed A3M solution. Section V presents a comparative numerical analysis of both protocols, while Section VI concludes the paper, and provides some potential research directions to further explore the proposed concept of Handover Mode Selection phase in different scenarios.

2 | RELATED WORK

The mobility management solutions in 5G, in general, are required to preserve the standard IPv6 mobility principles to ensure a seamless evolution of existing networks and protocols towards 5G[9]. In this vein, the IPv6-based DMM solutions and their enhancements through SDN have received a lot of attention. The popular DMM solutions have evolved from the existing IPv6-based mobility management mechanisms such as Proxy Mobile IPv6 (PMIPv6)[10,11]. In fact the IPv6 mobility management protocols, since their inception, have constantly undergone an evolution process, which coincides with the advancement of the mobile network capabilities across various generations of mobile networks. As illustrated in Figure 2, the adaptive mobility management solutions have also emerged and evolved correspondingly. In general, the primary motivation for the evolution of IPv6-based mobility management solutions has been to meet specific mobility requirements of a particular generation of mobile networks. The respective adaptive mobility management approaches for these solutions evolved mostly to address challenges, enhance and optimize their handover performances through adaptive operation.

With the outset of 5G era, the IPv6-based DMM solutions and their SDN enhancements have focused on the emerging architecture principles of decentralization and softwarization respectively. However, as we discuss later in Section 2.3 these solutions also require to operate adaptively, in order to meet the highly diverse mobility requirements in the heterogeneous 5G environment.

In the following, we first provide an overview of the existing PMIPv6-based and SDN-based DMM solutions, which preserve the IPv6 mobility management principles. Subsequently, the efforts made for adaptive IPv6 mobility management solutions
FIGURE 2 Evolution of Baseline IPv6 Mobility Management Protocols and their enhancements for Adaptive/Flexible operation are reviewed. Finally, we discuss the motivation for SDN-based adaptive mobility management solutions, which is the primary focus of this paper.

2.1 Distributed Mobility Management

The DMM can be realized through PMIPv6, SDN or routing-based approaches. The routing-based approach inherently suffers from high convergence latency which results in high handover delays and signalling costs, and has thus not been actively utilized in the protocol designs. In the following subsections, an overview of proposed solutions based on PMIPv6 and SDN principles is provided along with an analysis of their design considerations.

PMIPv6-based DMM

The DMM process considers the evolution of the existing centralized networks towards more decentralized and flatter network architectures. The PMIPv6-DMM specification defines a Mobility Anchor and Access Router (MAAR) entity which enriches the access router (AR) with anchoring function. The anchoring in the legacy MIPv6-based solutions is performed by a centralized
node (e.g. through the Local Mobility Anchor (LMA) in PMIPv6). In DMM, the centralized node is reduced to function only as a database entity and is termed as Central Mobility Database (CMD). A generic representation of a PMIPv6-DMM is given in Figure 3(a).

At MN’s handover, the signalling exchange among the DMM entities is shown in Figure 3(b). It mainly comprises of Proxy Binding Update (PBU) and Proxy Binding Acknowledgement (PBA) messages, which are exchanged among the previous MAAR (p-MAAR), the new MAAR (n-MAAR) and CMD. This solution, however, has been shown to have limited applicability, as it incurs very high latencies and signalling overheads in high mobility and session activity scenarios.

SDN-based DMM

The mobility management process in DMM can be simplified through SDN as it helps in removing the complexity and workload of the involved functions such as routing and tunnelling. In particular, it can help in offloading the underlying entities like ARs from functions such as maintenance of the binding cache, detection of the MN’s handoff, traffic resumption for MN at its new location etc., and thus makes them merely the traffic forwarding entities.

The major benefit of SDN is the deployment of network control at a centralized controller, by decoupling it from the data plane. The SDN Controller (CTR), with a central view of the mobility domain, has thus been utilized for various mobility-related functions in several SDN-based DMM solutions. An SDN-based DMM solution in suggests the binding information for a MN to be stored at CTR in the form of flow bindings. These bindings are installed in the underlying forwarding entities, and are updated as the MN performs a handoff. The CTR also computes new path at MN’s handover and installs the respective...
flow entries in the underlying switches. In such an approach, the path reconfiguration might incur high handover delays as it first involves computation and then installation of new rules in the forwarding entities. In a similar approach, a Border Controller is responsible for maintaining the mobility-related information for MNs, and for updating the flow tables on the data plane entities after the MN’s handover. In a similar solution proposed in the CTR, on learning the MN’s handover, coordinates with the CTR at the Correspondent Node’s (CN) network, and requests it to redirect the MN’s traffic to its new attachment point. However, such approaches can also incur high handover delays, especially under high session activity scenarios when a MN with several active sessions undergoes frequent handovers.

The inefficient handover performance owing to the multiple ongoing flows of MN are partially addressed in wherein the virtual mobility anchor (VMA) functions are proposed. These functions also work on CTR and evaluate a suitable anchor point for each flow. This anchor selection process, however, takes place only after the CTR learns about the new attachment point of the MN. Thus, the communication interruption at handover can be significantly higher since the installation of the traffic forwarding rules at the new anchor node represents an additional delay factor in this case.

2.2 Adaptive Mobility Management

The majority of the existing DMM solutions do not represent any flexible design considerations in their respective operations. In order to meet the disparate mobility needs in 5G, these solutions are required to provide adaptive mobility support to MNs in accordance to their mobility profile. An adaptive mobility management protocol in this context is defined as the protocol which is capable to change its execution principles and/or their order, to be adaptable for varying conditions or requirements.

The current MIPv6 and its extensions such as PMIPv6, Hierarchical Mobile IPv6 (HMIPv6) and Network Mobility (NEMO) also provide same mobility support regardless of the MN’s requirements. The fast handovers for MIPv6 (FMIPv6) and its enhancement for PMIPv6 (FPMIPv6) and HMIPv6 (FHMIPv6) however, offer distinction from these standards since these can alternately operate in reactive mode in case of the predictive mode failure, which is their primary mode of operation.

In general, the enhancements to these protocols focus on improving their performance based on a specific performance metric. For example, a solution aimed at reducing the handover delays normally results in high signalling and tunnelling costs and vice versa. Thus adaptive mobility management approaches came to prominence which aimed at balancing these generally conflicting performance optimizations. The evaluation of the best mode of operation in these approaches is mostly based on a Session-to-Mobility Ratio (SMR) parameter which is the ratio of the MN’s session arrival rate to its handoff rate. The SMR estimation provides a simplistic yet pragmatic approach, as it encompasses both the ongoing session and mobility activity of the MN. The SMR-based solutions normally evaluate certain threshold values for SMR. As the current estimated SMR crosses that threshold, a particular mode is executed for handover operation.
The adaptive mobility management solutions, based on the prevailing requirements, either operate an alternate mode of the same protocol, or select a different protocol all together as their alternate mode. A proposal in\(^{28}\) executes FPMIPv6 protocol which incurs high packet delivery costs for lower SMR, while a Route Optimized Handover (ROH)\(^{29}\) for PMIPv6 protocol for higher SMR. This is because the ROH-PMIPv6 causes high handover latencies since it needs to carry out the route optimization signalling between access node and the CN’s access entity. Hence, by adaptively operating FPMIPv6 and ROH according to the current SMR, an overall handover performance improvement, both in terms of handover latency and the packet delivery costs, is achieved. On similar principles, in\(^{30}\), HMIPv6 is only executed if the estimated packet delivery cost and bandwidth consumption are lower. In case of higher values for these parameters, MIPv6 is executed. Similarly, in\(^{31}\), HMIPv6 is executed for only lower traffic load, and high handover frequency for MN, while MIPv6 is operated otherwise.

The process of binding update (BU) in MIPv6-based protocols is carried out with a local or a global anchor for optimal traffic routing after the MN’s handover, and may involve registration of a local or a global CoA of MN. This process also incurs high costs, and has thus also been optimized through SMR-based adaptive solutions. In\(^{32}\), the MN registers its regional CoA with the current anchor only when the current SMR value is lower than threshold. For higher SMR values, the local CoA is registered with the Anchor. The BU process can also be optimized by re-locating and selecting a suitable anchor node on MN’s handover. In this regard, some solutions also focus on adaptive selection of anchor nodes to achieve flexibility in their operation. These schemes choose an anchor which would ensure minimal mobility costs. Examples include\(^{32}\) for HMIPv6 and\(^{33}\) for MIPv6.

Recently, some adaptive mobility management solutions for DMM have also been proposed. The dynamic DMM protocol in\(^{35}\) provides mobility support to MNs which are currently mobile and not to the non-mobile nodes, thus achieving reduced overhead. The tunnelling process is a key factor for performance degradation in DMM, since the DMM supports simultaneous flows managed by multiple anchors. Interestingly, such issues do not exist in the traditional centralized mobility management solutions. Accordingly, some works have focused on providing hybrid centralized-distributed mobility management solutions. An SMR-based dynamic tunnelling solution for DMM in\(^{36}\) suggests tunnel establishment between the access nodes if the estimated SMR is below threshold. For higher SMR, the tunnel is instead established between LMA and the access node, thus essentially providing the centralized mobility support. A similar hybrid centralized-distributed proposal in\(^{37}\) suggests that the DMM operation is executed by an AR based on its routing costs towards its neighbouring ARs. DMM is executed at an AR only if the routing costs to its neighbouring ARs are lower than a pre-defined threshold. Alternately, the AR operates the baseline PMIPv6 protocol, and thus optimizing the tunnelling overheads.

2.3 | Outlook and Motivation

Most of the current DMM solutions represent inflexible protocol design and involve a number of handover subprocesses which include movement detection, buffering, route optimization, tunnelling, packet forwarding and binding registration. In general,
for a particular DMM solution, all these subprocesses have to take place for each mobile node at every handover event regardless of its ongoing session activity and frequency of handovers. Accordingly, the majority of these solutions are suitable only for a particular mobility environment, and due to lack of flexibility, are unable to provide efficient mobility support in varying conditions.

Certain attempts have been made in past to achieve flexibility in the IPv6-based mobility management schemes. However, as the 5G networks are redefining the existing network architectures with decentralization, control and data plane separation and softwarization, integrating such approaches to 5G is unlikely to bring efficiency to the mobility management process. In addition, the diverse 5G services are also resulting in a highly heterogeneous mobility environment which incurs several bottlenecks that aggravate due to smaller cell sizes and constant handovers. In this context, intelligent and flexible, yet non-complex mobility management protocols are required which are suitable not only for distinct mobility scenarios, but are also adaptable should the mobility requirements of the MN change over time.

3 | PROPOSED ADAPTIVE MULTIMODE MOBILITY MANAGEMENT ARCHITECTURE

The proposed A3M architecture aims to provide a flexible mobility management solution based on IPv6 mobility and SDN principles. In order to achieve a flexible protocol operation, it considers decoupling of the handover subprocesses which are derived from MIPv6-based protocols and their enhancements. These subprocesses include handover prediction, MN’s attachment detection, registration or binding update, packet forwarding, buffering, bicasting and path optimization.

Some of these handover subprocesses are mandatory to accomplish the handover process while others play a supportive role during handover. Accordingly, in the A3M architecture, these are classified as the primary handover subprocesses and the supportive handover subprocesses. As shown in Figure 4, the MN’s handover prediction, its attachment detection, binding update or binding registration and packet forwarding are classified as primary handover subprocesses, while buffering, bicasting and path optimization are categorized as supportive handover subprocesses.

In terms of execution, certain handover subprocesses among these get into effect as the MN hands over from one cell to another. Some of these subprocesses such as attachment detection and registration can not be controlled by the network nodes\(^1\). Others, although still get executed at MN’s handover, can be controlled by network nodes. Implementation of such handover subprocesses as "standalone modules" on CTR is described as a practical approach in\(^2\). This approach also ensures seamless evolution of the IPv6 mobility principles towards SDN. The A3M architecture also follows the same approach.

---

\(^1\) The path optimization in SDN domain corresponds to the route optimization process in MIPv6.

\(^2\) The MN’s handover prediction at the network side has been considered in several proposals in the literature. However, it is more effective if it is carried out by the MN itself\(^3\). Therefore, in A3M, the handover prediction is considered to be done at the MN.
Network-based Handover Subprocesses

Primary Handover Subprocesses
- MN’s Handover Prediction*
- Attachment Detection
- Registration
- Packet Forwarding

Supportive Handover Subprocesses
- Path Optimization
- Buffering
- Bicasting

* Can be controlled by CTR
** Cannot be controlled by CTR
* Applicable to Predictive Handovers only.

**FIGURE 4** Classification of the Handover Subprocesses in the A3M Architecture

In A3M, during the Handover Mode Selection phase, the CTR evaluates the enforcement of network controllable supportive handover subprocesses which best suit the current mobility requirements of the MN. The Handover Mode Selection phase, as represented through the Figure 5, precedes the traditional Handover Decision and Handover Execution phases and comes after the Handover Information Gathering phase. The Handover Mode Selection and the Handover Information Gathering phases essentially represent the Handover Preparation process. Both these phases are repeated at regular intervals defined by a Handover Preparation (HP) timer. The HP timer is also reset at the completion of a handover event, which triggers the execution of the Handover Preparation again.

3.1 The Handover Information Gathering phase:

During this phase, the context information of MN is updated at the CTR. The context information includes the information related to the MN’s active sessions as well as its handover frequency (\(\gamma\)). The active number of different session types which include conversational, non-conversational multimedia or non-real-time sessions are represented by \(N_c\), \(N_{nc}\) and \(N_{nR}\) respectively. The CTR updates the current packet arrival rate (\(\lambda_p\)) for each session as well. In addition, through simple computations in Algorithm 1, the CTR adjusts the suitable handover frequency thresholds (\(\gamma_{th1}\) and \(\gamma_{th2}\)) for MN, as follows.

For evaluation of \(\gamma_{th1}\) and \(\gamma_{th2}\), a default time interval \(T_{def}\) is assumed over which a default number of handovers \(N_{def}\) take place. The default handover frequency (\(\gamma_{def}\)) is given by \(\frac{N_{def}}{T_{def}}\). The \(T_{def}\), \(N_{def}\) and the resultant \(\gamma_{def}\) are defined as reference variables, whose values are set at the network operator’s discretion. For example, an operator deploying a network of small cells
Algorithm 1 Handover Information Gathering phase

Input: $T_s$ and $\gamma_s$

1. \textbf{if} $T_s < T_{def}$ \textbf{then}
2. \hspace{1em} $\gamma_{th_1} \leftarrow \gamma_{def}$, $\gamma_{th_2} \leftarrow \gamma_s$; $\tau = 2 \cdot T_s$; $\delta \leftarrow \left\lceil \frac{|T_s - T_{def}|}{T_{def}} \right\rceil$;
3. \textbf{else if} $T_s > T_{def}$ \textbf{then}
4. \hspace{1em} \textbf{if} $|T_s - T_{def}| < \epsilon$ \textbf{then}
5. \hspace{2em} $\gamma_{th_1} \leftarrow \gamma_{def}$; $\gamma_{th_2} \leftarrow \gamma_{def}$;
6. \hspace{1em} \textbf{else}
7. \hspace{2em} $\gamma_{th_1} \leftarrow \gamma_{def}$, $\gamma_{th_2} \leftarrow \gamma_s$;
8. \hspace{1em} $\tau = 2 \cdot T_{def}$;
9. \hspace{2em} $\delta \leftarrow \left\lceil \frac{|T_s - T_{def}|}{T_{def}} \right\rceil$;
10. \textbf{return} $\gamma_{th_1}, \gamma_{th_2}, \tau, \delta$

\textbf{FIGURE 5} Handover Mode Selection phase among the traditional handover phases
where frequent handovers would be expected, would like to adjust higher values of $N_{def}$ over a comparatively shorter value of $T_{def}$.

According to the current mobility scenario, as the time period $T_s$ over which the $N_{def}$ handovers are actually taking place, varies, the $\gamma_{th1}$ and $\gamma_{th2}$ are accordingly adjusted through Algorithm 1. The variable $\gamma_{th1}$ represents a moderate handover frequency above which enabling support for real-time sessions through buffering and bicasting services is desirable, and the variable $\gamma_{th2}$ represents high handover frequency, above which these handover support services for the critical flows (e.g. conversational sessions) must be enabled.

The CTR also evaluates the current handover frequency ($\gamma_{curr}$) and a handover delay sensitivity factor ($\delta$), which are also used during the subsequent Handover Mode Selection phase. $\gamma_{curr}$ is calculated over another time variable $\tau$, which is based on the lower value among $T_s$ and $T_{def}$. $\tau$ is defined in order to circumvent the effects of sudden decrease of $T_s$ due to abrupt handovers e.g. due to ping-pong effect, especially if a lower value for $N_{def}$ is adjusted. Correspondingly, $N_{\tau}$ represents the number of handovers occurring over the interval $\tau$, and hence $\gamma_{curr}$ is given as $\frac{N_{\tau}}{\tau}$. Other factors include $\delta$, which is $\frac{|T_s-T_{def}|}{T_{def}}$ and $\epsilon = 10^{-2}$ which represents the error offset factor for $T_s$ as it approaches $T_{def}$, during the handover frequency threshold adjustments.

### 3.2 The Handover Mode Selection phase:

During the Handover Mode Selection phase, the CTR evaluates the handover support services through Algorithm 2 (path optimization service) and Algorithm 3 (buffering, bicasting, and packet forwarding). In order to focus on the DMM paradigm in this work, and in particular the high mobility and high session activity scenarios, the algorithms constituting the Handover Mode Selection phase, evaluate these services based on parameters such as handover frequency, number of active sessions, duration of ongoing sessions, anchor load, packet arrival rates, etc.

The enforcement of path optimization, buffering, bicasting and packet forwarding services occurs through installation of flow entries at the data plane FEs. For predictive handover, the CTR pre-installs the necessary flow entries on p-FE and the current Anchor (c-Anchor) during the Handover Mode Selection phase. Their priorities in the flow tables are however kept low so that they are not executed until the handover actually takes place. In reactive case, these flow entries are installed during the Handover Execution phase.

### Evaluation for Path Optimization service (Algorithm 2):

The Algorithm 2 consists of three stages. In the first stage, the CTR determines $\mu_{th}$, which represents the maximum allowable flows with Path Optimization for the prospective handover event. In the second stage, the CTR determines the flows which should be provided with the path optimization service based on the value of $\mu_{th}$. The path optimization service essentially requires new
anchor for the new optimized path. Hence, in the third stage, the CTR evaluates suitable anchor(s) for each of selected flows for Path Optimization.

In the first stage, \( \mu_{th} \), which which represents the maximum allowable flows with path optimization for the prospective handover event, is evaluated. The \( \mu_{th} \) corresponds to the value of \( \delta \) which is rounded off in Algorithm 1 such that it restricts the execution of path optimization for \( T_s < T_{def} \), and supports it otherwise. The path optimization service, in general, is provided only for handover frequencies below \( \gamma_{th2} \) to ensure its successful and timely completion.

Once the \( \mu_{th} \) is evaluated, the CTR chooses flows for path optimization during the second stage. This stage essentially executes in two main phases: flow prioritization and flow selection. In the flow prioritization phase (lines 8-14), different flow types are prioritized, with highest priority given to conversational sessions, followed by the non-conversational multimedia and Non-Real-time sessions. These sessions are represented through arrays of their flow IDs, which given respectively as \( \alpha_c[N_c] \), \( \alpha_{nc}[N_{nc}] \) and \( \alpha_{nr}[N_{nr}] \). Based on the prioritization, the ongoing flows with higher priority are essentially short-listed for path optimization service provisioning. The flow IDs of such flows are stored in the vector array, \( candFlows[] \).

After flow prioritization the algorithm next performs flow selection among the candidate/short-listed flows. This phase has two main parts. The first part (lines 16-20) executes if the initial number of candidate flows is less than \( \mu_{th} \). In this case, the algorithm enables the path optimization service to all flows in the \( candFlows[] \), which are accordingly moved to another vector array \( selectedFlows[] \). The total number of selected flows for path optimization is represented by \( N_S \). The second part (lines 21-31) takes place when the CTR is unable to provide the PathOptimization support to all the initially shortlisted flows. Hence, the algorithm sequentially evaluates \( \pi \) for each flow in the \( candFlows[] \). After iterating the loop for \( N_S \) times, the algorithm selects the Flow ID with highest \( \pi \) for the PathOptimization service. Finally, if \( \mu_{th} > 0 \), the outer loop continues its iteration, executing the flow prioritization and flow selection phases sequentially until \( \mu_{th} \) becomes 0.

Finally, in Stage 3, the Algorithm 2 performs the anchor selection among candidate anchors represented through vector array \( candAnchors[] \) for each \( selectedFlow[] \) during the third stage. The anchor node in A3M acts as a reference entity which manages an ongoing flow, and with a new optimized path for a flow at handover. The CTR performs the anchor selection process to ensure a suitable anchor for each ongoing flow. The Simple Additive Weighting (SAW) technique of the Multiple Attributes Decision Making (MADM) is used for anchor selection in A3M. The attributes of SAW include (a) the active users density \( (\Psi) \), (b) the current load of the candidate anchor \( (\xi) \), (c) number of hops between n-FE and the candidate anchor \( (\eta) \), and (d) the H-factor \( (H) \). The CTR with view of the entire domain inherently maintains \( \Psi \) and \( \xi \), while \( \eta \) can be obtained as \( \eta = h + 1 \). Here, \( h \) is the number of hops between the p-FE and the candidate anchor node. The H-factor is the ratio of the total number of flows of MN to be anchored by the candidate anchor after handover to \( \mu_{th} \). It is based on Heat factor in [10] and indicates that the candidate anchor handling maximum flows for MN is given higher preference. The \( \phi_i \) represents the value function for any
Algorithm 2 Evaluating Path Optimization Service during Handover Mode Selection phase

Input: $y_{cur}$, $N_c, N_{nc}, N_{nR}$, $a_c[N_c], a_{nc}[N_{nc}], a_{nR}[N_{nR}]$

Initialization: $i, j, k, \pi_{max}, N_x, candFlows[], selectedFlows[] \leftarrow 0$

Stage 1: Evaluation of $\mu_{th}$:

1. if $y_{cur} > y_{th_2}$ then
   2. $\mu_{th} \leftarrow 0$
   3. $PathOptimization \leftarrow \text{FALSE}$
   4. else if $y_{th_1} \leq y_{cur} \leq y_{th_2}$ then
      5. $\mu_{th} \leftarrow \delta$
   6. else
      7. $\mu_{th} \leftarrow \delta + 1$

Stage 2: Flow Selection:

8. for $i = 0; \mu_{th} \neq 0; i + +$ do
   9. if $N_c > 0$ then
      10. $candFlows[] \leftarrow a_c[N_c]; N_x \leftarrow N_c$; goto FlowSelection;
   11. else if $N_{nc} > 0$ then
      12. $candFlows[] \leftarrow a_{nc}[N_{nc}]; N_x \leftarrow N_{nc}$; goto FlowSelection;
   13. else if $N_{nR} > 0$ then
      14. $candFlows[] \leftarrow a_{nR}[N_{nR}]; N_x \leftarrow N_{nR}$; goto FlowSelection;
   15. FlowSelection:
      16. if $N_x \leq \mu_{th}$ then
         17. $selectedFlows[] \leftarrow candFlows[]$
         18. $selectedFlows[] \leftarrow PathOptimization$
         19. $\mu_{th} = \mu_{th} - N_x$
         20. $N_x = N_x + \mu_{th}$
      21. else if $N_x > \mu_{th}$ then
         22. for $j = 0; j < N_x; j + +$ do
            23. Evaluate $\pi_j$ for the $candFlows[j]$
            24. if $\pi_j > \pi_{max}$ then
               25. $\pi_{max} \leftarrow \pi_j$
               26. $varFlow \leftarrow candFlow[j]$
            27. if $j == N_x$ then
               28. $selectedFlows[] \leftarrow varFlow$
               29. $selectedFlows[] \leftarrow PathOptimization$
               30. $N_x = N_x - 1; N_x = N_x + 1$
               31. $\mu_{th} = \mu_{th} - 1$
      32. Stage 3: Anchor Selection for selected flows:
      33. $\forall selectedFlows[]$
         34. Evaluate value function for a candidate anchor $l$ for $k-th$ flow.
         35. for $l = 0; l \leq candAnchors[]; l + +$ do
            36. $o_l = o_{a1} \cdot \Psi_k + o_{a2} \cdot \Psi_k + o_{a3} \cdot \eta_k + o_{a4} \cdot H_k$
            37. if $o_l \geq o_{max}$ then
               38. $o_l \leftarrow o_{max}$
            39. if $l == N_l$ then
               40. $selectedAnchors[] \leftarrow candAnchors[l]$
         41. return
candidate anchor $l$, while $\omega_k$ is the weight assigned to the $k$–th attribute, and is given by the rank sum weights technique as,

$$\omega_k = \frac{N - r_k + 1}{\sum_{i=1}^{N} (N - r_i + 1)} \quad (1)$$

Buffering, Packet Forwarding and Bicasting (Algorithm 3)

The Algorithm 3 evaluates the buffering, bicasting and packet forwarding services both for predictive and reactive handovers for A3M, represented as A3M-P and A3M-R respectively. Some steps in this algorithm represented through symbol (*) are applicable only to predictive handovers (A3M-P), while others represented through the symbol (‡) are applicable to reactive handovers (A3M-R) only. The algorithm initializes the handover support services to FALSE (i.e. disabled by default), while the local variables $\lambda_{\text{max}}$, $\text{selectedFlows}[]$ to 0. The information such as $N_c$, $N_{nc}$, $N_{nR}$, $a_c[N_c]$, $a_{nc}[N_{nc}]$, $a_{nR}[N_{nR}]$, and $\gamma_{curr}$ is provided as input variables.

The algorithm executes in three main parts. The first part (lines 2-12) relates to the handover support services for the conversational sessions, which are given higher priority for both A3M-P and A3M-R. The bicasting service, among the handover support services, is an effective technique to minimize the handover interruption. However, it is only applicable to A3M-P, since A3M-R executes after the MN has joined the new network. In predictive operation however, when the $\gamma_{curr}$ is very high (i.e. $\gamma_{curr} > \gamma_{th2}$), the bicasting service for conversational session is activated both at p-FE and c-Anchor. The buffering service, on the other hand, is only useful if $\gamma_{curr}$ is lower (i.e. $\gamma_{curr} < \gamma_{th1}$). Hence it is disabled otherwise, as it might result in voice duplication and packet out-of-order problems for conversational sessions.

In the second part (lines 13-30), all handover support services are enabled for a non-conversational multimedia flow with maximum packet arrival rate $\lambda_{\text{max}}$, when $\gamma_{curr} < \gamma_{th1}$. However, bicasting is not enforced for high $\gamma_{curr}$ values since it results in high network resource consumption as the MN undergoes rapid handovers, and the multimedia sessions normally tend to remain active for a longer period of time. In the third part, only the packet forwarding service is enabled for non-real time sessions regardless of the $\gamma_{curr}$ in both A3M-P and A3M-R (lines 31-36).

The buffering, bicasting and packet forwarding services remain enabled only for a specific time period, which is managed by the CTR by sending messages to the respective FEs. The buffering and bicasting services are enabled through the message ofp_ho_support_cmd (Θ) and disabled through ofp_path_update_cmd (Θ), while the packet forwarding service is enabled through ofp_path_update_cmd (Θ) and disabled through ofp_path_rem_cmd (Θ). As shown in Table[1] the symbolic notations $[x]$ and $\{x\}$ for any message $x$, represent the messages involved in the Handover Mode Selection and the Handover Execution phases respectively.
Algorithm 3 Evaluating Buffering, Bicasting and PacketForwarding services during Handover Mode Selection phase

Input: $y_{\text{curr}}, N_c, N_{nc}, N_{nR}, \alpha_c[N_c], \alpha_{nc}[N_{nc}], \alpha_{nR}[N_{nR}]$

Initialization: Buffering, Bicasting, PacketForwarding $\leftarrow$ FALSE; $\lambda_{\text{max}}, \text{selected\ Flows[]} \leftarrow 0$

Evaluating Buffering, Bicasting and PacketForwarding:

1. if $N_c > 0$ then
   2. \hspace{1em} selectedFlows[] $\leftarrow \alpha_c[N_c]$;
   3. \hspace{1em} for ($i = 0; i \leq N_c; i + +$) do
      4. \hspace{2em} if ($y_{\text{curr}} < y_{t_1}$) then
         5. \hspace{3em} $\forall$ selectedFlows[]:
            6. \hspace{4em} Buffering $\leftarrow$ TRUE at p-FE;
            7. \hspace{4em} Buffering $\leftarrow$ TRUE at c-Anchor;
            8. \hspace{4em} Bicasting $\leftarrow$ TRUE from c-Anchor;
            9. \hspace{4em} Bicasting $\leftarrow$ TRUE from p-FE;
            10. \hspace{4em} PacketForwarding $\leftarrow$ TRUE;
         11. \hspace{2em} else
            12. \hspace{3em} Buffering $\leftarrow$ FALSE;
            13. \hspace{3em} Bicasting $\leftarrow$ TRUE from c-Anchor;
            14. \hspace{3em} PacketForwarding $\leftarrow$ TRUE;
   15. \hspace{1em} else if $N_{nc} > 0$ then
      16. \hspace{2em} CTR evaluates the Non-Conversational Multimedia session with $\lambda_{\text{max}}$
      17. \hspace{2em} for ($j = 0; j \leq N_{nc}; j + +$) do
         18. \hspace{3em} Evaluate $\lambda_j$ for $\alpha_{nc}[j]$;
         19. \hspace{3em} if $\lambda_j > \lambda_{\text{max}}$ then
            20. \hspace{4em} $\lambda_{\text{max}} \leftarrow \lambda_j$;
            21. \hspace{4em} candFlows[] $\leftarrow \alpha_{nc}[j]$;
            22. \hspace{3em} if $j == N_{nc}$ then
               23. \hspace{4em} selectedFlows[] $\leftarrow$ candFlows[j];
            24. \hspace{3em} \hspace{1em} if ($y_{\text{curr}} < y_{t_1}$) then
               25. \hspace{4em} Buffering $\leftarrow$ TRUE at c-Anchor;
               26. \hspace{4em} Bicasting $\leftarrow$ TRUE from c-Anchor;
               27. \hspace{4em} PacketForwarding $\leftarrow$ TRUE;
            28. \hspace{3em} else
               29. \hspace{4em} Buffering $\leftarrow$ TRUE at c-Anchor;
               30. \hspace{4em} Buffering $\leftarrow$ FALSE;
               31. \hspace{4em} Bicasting $\leftarrow$ FALSE;
               32. \hspace{4em} PacketForwarding $\leftarrow$ TRUE;
         33. \hspace{3em} else
            34. \hspace{4em} selectedFlows[] $\leftarrow \alpha_{nR}[N_{nR}]$;
            35. \hspace{4em} $\forall$ selectedFlows[]:
               36. \hspace{5em} Buffering $\leftarrow$ FALSE;
               37. \hspace{5em} Bicasting $\leftarrow$ FALSE;
               38. \hspace{5em} PacketForwarding $\leftarrow$ TRUE;
      39. return
<table>
<thead>
<tr>
<th>Proposed Message</th>
<th>OpenFlow Message Type</th>
<th>Represented by</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>ofp_ho_adv_buffer_support</td>
<td>flow-mod (ADD)</td>
<td>①</td>
<td>Proactive installation of new flow entries for Buffering.</td>
</tr>
<tr>
<td>ofp_ho_adv_bicast_support</td>
<td>flow-mod (ADD)</td>
<td>②</td>
<td>Proactive installation of new flow entries for bicasting.</td>
</tr>
<tr>
<td>ofp_ho_indicate_info</td>
<td>Packet-In</td>
<td>③</td>
<td>Transferring the HO_Ind to CTR.</td>
</tr>
<tr>
<td>ofp_ho_support_cmd</td>
<td>flow-mod (MODIFY)</td>
<td>④</td>
<td>Modification of flow entries for buffering and bicasting.</td>
</tr>
<tr>
<td>ofp_ho_update_cmd</td>
<td>flow-mod (ADD)</td>
<td>⑤</td>
<td>Installation of new flow entries at n-FE.</td>
</tr>
<tr>
<td>ofp_ho_attach_info</td>
<td>Packet-In</td>
<td>⑥</td>
<td>Transferring Attachment Announcement to CTR.</td>
</tr>
<tr>
<td>ofp_path_update_cmd</td>
<td>flow-mod (MODIFY)</td>
<td>⑦</td>
<td>Modification of flow entries for Traffic Redirection</td>
</tr>
<tr>
<td>ofp_path_rem_cmd</td>
<td>flow-mod (REMOVE)</td>
<td>⑧</td>
<td>Removal of flow entries.</td>
</tr>
<tr>
<td>ofp_anch_init</td>
<td>flow-mod (ADD)</td>
<td>⑨</td>
<td>Installation of new flow entries at new Anchor</td>
</tr>
<tr>
<td>ofp_anch_rem</td>
<td>flow-mod (REMOVE)</td>
<td>⑩</td>
<td>Removal of flow entries from an Anchor</td>
</tr>
</tbody>
</table>

**TABLE 1** The Proposed Signalling Messages for the A3M Architecture

### 3.3 The Handover Decision phase

The **Handover Decision** process, in general, involves the criteria as well as the timing for the handover decision. Several handover decision criterion have been proposed in the literature, however, in the A3M architecture, we assume the generic RSS criteria. The handover can be triggered as soon as the current RSS falls below an acceptable threshold, and the MN can accordingly proceed with initiating the actual handover process.

### 3.4 The Handover Decision and Handover Execution phases:

The **Handover Execution** phase succeeds the **Handover Decision** phase. The MN can make handover decision proactively, or reactively, if it is unable to carry out the predictive handover. Assuming that the CTR enables all handover support services for all ongoing flows during **Handover Mode Selection** phase, both predictive and reactive handover operations are described as follows.

**Predictive Handover:**

The MN sends the handover indication \( \text{HO}_\text{Ind} \) to p-FE if it initiates predictive handover e.g. on receiving better RSS from a neighbouring access point (AP). The p-FE in turn sends the \( \text{HO}_\text{Ind} \) to the CTR in the \( \text{oofp}_\text{h}_\text{o}_\text{i}_\text{n}_\text{d}_\text{i}_\text{c}_\text{a}_\text{t}_\text{e}_\text{i}_\text{n}_\text{f}_\text{o}_\text{f} \) (①) message. If path optimization is to be carried out, the CTR, according to the prospective n-FE, also evaluates the suitable Anchor(s) for the ongoing flows. It also installs new flow entries at n-FE so that it can handle the ongoing flows of the MN. If the buffering and bicasting services were enabled during the **Handover Mode Selection** phase, the CTR sends the \( \text{oofp}_\text{h}_\text{o}_\text{s}_\text{u}_\text{p}_\text{p}_\text{o}_\text{r}_\text{t}_\text{e}_\text{c}_\text{m}_\text{d} \) (②) message to p-FE or c-Anchor. The p-FE or c-Anchor accordingly starts buffering as well as bicasting for the respective flows. The CTR, meanwhile, sends the \( \text{oofp}_\text{h}_\text{o}_\text{u}_\text{p}_\text{d}_\text{a}_\text{t}_\text{e}_\text{c}_\text{m}_\text{d} \) (③) to n-FE to install new flow entries for MN’s ongoing sessions on it.
On the other hand, as the n-FE receives Attachment Announcement from MN, it forwards it to the CTR through ofp_ho_attach_info (1) message. The CTR in turn stops the buffering and bicasting services and enforces packet forwarding at p-FE towards n-FE through ofp_path_update_cmd (2). The CTR also sends the ofp_path_update_cmd (3) to c-Anchor(s) in order to reconfigure the path from c-Anchor(s) towards n-FE.

On successfully updating the flow entries at current Anchors, the CTR removes flow entries from p-FE through ofp_path_rem_cmd (4) message. With traffic resumption from current Anchor(s) towards n-FE, the CTR now starts the path optimization process, if enabled during the Handover Mode Selection phase. Having evaluated the new Anchor(s) beforehand in Step (b), the CTR sends the ofp_anch_init (5) message to install new flow entries on them. The new Anchor(s) install the entries, and on their successful installation, the CTR proceeds to remove the flow entries from previous anchors through ofp_anch_rem (6) message.

FIGURE 6 Signalling Sequence for (a) Predictive and (b) Reactive A3M Handover
Reactive Handover:

The MN sends an Attachment Announcement message to n-FE to initiate the reactive handover. The n-FE sends this message to the CTR in ofp_ho_attach_info (3). The CTR in turn installs new flow entries in n-FE through ofp_ho_update_cmd (3). These flow entries are pre-computed during the Handover Mode Selection phase. The CTR also simultaneously sends the ofp_ho_support_cmd (3) to p-FE to enforce buffering if enabled during the Handover Mode Selection phase.

The CTR, after successfully installing the flow entries at n-FE in Step (b), now proceeds to enforce packet forwarding from p-FE to n-FE through ofp_path_update_cmd (3) message. At this point, it also stops buffering at p-FE. Next, the CTR updates flow entries on c-Anchor(s) to divert MN’s flows from p-FE towards n-FE, through ofp_path_update_cmd (3) message. On successful installation of these flow entries, the CTR removes flow entries from p-FE through ofp_path_rem_cmd (3) message. This effectively stops the ongoing packet forwarding process at p-FE.

The CTR finally starts path optimization if enabled during the Handover Mode Selection phase, by sending the ofp_anch_init (9) message to the already evaluated new Anchor(s) in Step (a). New flow entries to handle MN’s ongoing flows are installed through this message. On successful installation of rules on new Anchor(s), the CTR sends the ofp_anch_rem (9) to current Anchor(s), to remove flow entries for the respective flows of the MN.

4 | COMPLEXITY ANALYSIS

The Handover Mode Selection phase algorithms are the main add-ons of the proposed A3M architecture which are supported through Algorithm 1 of the Handover Information Gathering phase. We evaluate complexity of these algorithms both in time and space, in the worst case scenarios to evaluate their impact on the overall system performance.

The Algorithm 1 performs a sequential processing without any iterative loops, which implies that it has constant time space and time complexity, given as $O(1)$. Likewise, it deals with the same set of variables due to which the space complexity is also given as $O(1)$.

The Algorithm 2 has different complexity for each stage. The Stage 1 has constant time space and time complexity since it executes same set of procedures regardless of input variables. In Stage 2, the worst case scenario occurs when (a) $N_x$ is equivalent to $N_T$ (i.e., all active flows are to be evaluated during Stage 2), which implies that $N_T > \mu_{th}$ (line 21) and (b) $N_T - \mu_{th} = 1$ (i.e., both $N_T$ and $\mu_{th}$ have a minimal difference of 1 flow between them). The outer loop in this case iterates for $\mu_{th}$ times. Each iteration of the outer loop decrements $N_x$ by 1. Hence, the inner loop, at each iteration of the outer loop, iterates for $N_x - 1$ times from the previous loop. The total iterations for the inner loop occur in the worst case scenario are given as $N_x + (N_x - 1) + (N_x - 2) + (N_x - 3) + \ldots + 3 + 2$, which is equivalent to $\frac{N_x(N_x+1)-2}{2}$. This implies that the time complexity for the Stage 2 of Algorithm 2 is $O\left(\frac{N_x^2}{2}\right)$. 
In the Stage 3 of Algorithm 2, the worst case scenario occurs when maximum flows out of $N_T$ are selected for path optimization and $N_T - \mu_{th} = 1$, where $N_T > \mu_{th}$. Moreover, the attributes of all candidate anchors such as $H$ and $L$ etc. make them suitable for each selected flow for path optimization. If the number of candidate anchors is represented through $N_A$, the time complexity for Stage 3 is given as, $O(N_T \cdot N_A)$. Hence, the overall time complexity of Algorithm 2 is given as $O(N_T^2 + N_T \cdot N_A)$.

For space complexity of Algorithm 2, the worst case scenario occurs when all the flows are chosen for path optimization (i.e. $N_T \leq \mu_{th}$) and hence all candidate flows (which are also equivalent to selected flows) will undergo the anchor selection process. During the anchor selection process, each flow is evaluated for every single anchor. Hence, the space complexity is given as, $O(N_T \times N_A)$.

For Algorithm 3, in the worst case, all services would be enabled for all ongoing sessions. The time complexity in this case depends on $N_c$ and $N_{nc}$ because the first loop runs for all conversational sessions, and $\pi_j$ is evaluated for each non-conversational multimedia session. Hence the time complexity for Algorithm 3 is $O(N_c + N_{nc})$. The space complexity of Algorithm 3 is also defined by the worst case scenario when all ongoing flows are provided all the handover support services. In this case, the vector array $selectedFlows[]$ will hold $N_T$ number of flows, which implies that the space complexity is $O(N_T)$.

5 | ANALYTICAL MODELLING

In this section, the network, traffic and mobility models under consideration for performance comparison of PMIPv6-DMM and A3M are first described. The mathematical expressions for handover performance metrics both for PMIPv6-DMM and the A3M are subsequently formulated. For the handover performance comparison of both solutions, we have two major points of interest.

1. To evaluate the impact of flexibility features of A3M on the handover performance.
2. To evaluate the handover performance of A3M in high mobility and high session activity scenarios where PMIPv6-DMM solutions have been shown to incur high latencies and overheads.

5.1 | Network Model

A PMIPv6-DMM and an SDN domain have different traffic forwarding mechanisms. In order to evaluate their performance on a similar scale, a generic representation for both domains is provided in Figure 7.

In order to focus on the performance evaluation of flexible handover operation of the A3M architecture, the hop count between the MN’s attachment point and the CMD/CTR is considered constant. The bandwidth and link delay for all links connecting the attachment points to CMD/CTR are also considered constant. Moreover, it is assumed that the MN has reachability to $n$ different CNs across the domain.
5.2 Traffic and Mobility Model

To focus on A3M flexibility and DMM limitations, we assume the worst case mobility and session activity scenarios. For high mobility, we assume that a different AR/FE controls each cell visited by the MN. The MN thus visits different MAARs/FEs in the domain, and performs handover above Layer 2 (L2) each time. Likewise, we assume that a new session with a different CN is established, which remains active while the MN remains inside the current CMD/CTR’s domain. For each session, the packet arrival rate ($\lambda_p$) is also considered equivalent.

5.3 Performance Metrics for PMIPv6-DMM

In this section, the mathematical expressions for primary handover performance metrics for PMIPv6-DMM are formulated, which include session disruption delay, packet loss and signalling costs. Table 2 provides description of several parameters used in these expressions.

5.3.1 PMIPv6-DMM - Session Disruption Delay

For delay analysis, we consider session disruption delay parameter instead of the generally used handover delay metric. The session disruption delay represents the delay associated to all ongoing sessions of MN. Considering there are $n$ active sessions, and the $m$-th session among them takes maximum time to resume, the session disruption delay is defined as the interval between the last packet of the $m$-th session received at the previous network to its first packet received at the new network. The session disruption delay also comprises of the delay factors identified in [42] which include (a) link delay between any two
<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_{L2}$</td>
<td>L2 handover delay</td>
<td>$TD$</td>
<td>Tunnelling delay</td>
</tr>
<tr>
<td>$\Omega_{maar/fe}$</td>
<td>Queuing delay</td>
<td>$DBC$</td>
<td>Database update cost</td>
</tr>
<tr>
<td>$P$</td>
<td>Average per-byte processing delay</td>
<td>$\lambda_p$</td>
<td>Average packet arrival rate per session</td>
</tr>
<tr>
<td>$U$</td>
<td>Average per-byte transmission delay</td>
<td>$n$</td>
<td>Number of active session of MN</td>
</tr>
<tr>
<td>$s_p$</td>
<td>Average size of a PMIPv6 message</td>
<td>$k$</td>
<td>Sessions with handover support services</td>
</tr>
<tr>
<td>$s_{af}$</td>
<td>Average size of an OpenFlow message</td>
<td>$\beta$</td>
<td>Unit processing cost per byte</td>
</tr>
<tr>
<td>$s_d$</td>
<td>Average size of a data packet</td>
<td>$u$</td>
<td>Unit transmission cost per byte</td>
</tr>
<tr>
<td>$s_o$</td>
<td>Average size of tunnel header</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**TABLE 2** Parameter notations with description for mathematical expressions for PMIPv6-DMM and A3M

Nodes $x$ and $y$ ($T_{x,y}$), (b) L2 handover delay ($t_{L2}$), (c) Queuing delay at MAAR/FE ($\Omega$), and (d) Processing delay at any node $x$ ($PD_x$). Accordingly, the session disruption delay is given as,

$$D_{PMIPv6-DMM} = t_{L2} + 2 \cdot PD_{maar} + PD_{cmd} + 2 \cdot T_{maar,cmd} + (m + 1) \cdot (\Omega + TD + PD_{maar,cmd}) + m \cdot T_{maar,maar} + T_{mn,maar}$$  \hspace{1cm} (2)

The transmission and processing delays as a function of packet size can be given as $T_{x,y} = s_p \cdot U$ and $PD_x = s_p \cdot P$ respectively.

$$D_{PMIPv6-DMM} = t_{L2} + s_p \cdot P + (m + 1) \cdot (\Omega + TD + s_d \cdot P) + m \cdot P(s_d + s_o) + s_d \cdot U$$  \hspace{1cm} (3)

### 5.3.2 PMIPv6-DMM - Packet Loss

The packet loss for PMIPv6-DMM is proportional to $\lambda_p$ for each active session$^{[23]}$ and is given as,

$$P_{loss}^{PMIPv6-DMM} = \sum_{i=1}^{n} \lambda_p \cdot D^i$$  \hspace{1cm} (4)

Here, $D^i$ represents the delay incurred to resume any session anchored by an $i-th$ anchor. It is given as,

$$D^i = t_{L2} + s_p \cdot P + (i + 1) \cdot (\Omega + TD + s_d \cdot P) + i \cdot P(s_d + s_o) + s_d \cdot U$$  \hspace{1cm} (5)

### 5.3.3 PMIPv6-DMM - Signalling Cost

The signalling cost relates to the cost associated to the transmission and processing of a control signal, as well as the database update as a result of its processing$^{[23]}$. It is given as,

$$C_{PMIPv6-DMM} = (n + 2) \cdot DBC_{maar} + (n + 1)(2 \cdot s_p \cdot \beta + 2 \cdot s_p \cdot u + DBC_{cmd})$$  \hspace{1cm} (6)
5.4 | Performance Metrics for A3M

Since the A3M protocol can operate in multiple modes of operation, the provision of the handover support services correspondingly determines the respective expressions for each metric. The expressions are stated keeping in view that \( k \) flows out of \( n \) have the handover support services enabled, where \( 0 \leq k \leq n \).

5.4.1 | A3M - Session Disruption Delay

The session disruption delay for A3M scheme’s predictive operation is impacted due to bicasting. If the bicasting is enabled only for \( k \) flows, the session disruption delay, by definition, will still be dependent upon the resumption of all \( n \) flows. Thus, in order to study the impact of bicasting on the session disruption delay, we formulate expressions considering that bicasting is either enabled for \( n \) flows, or is disabled for all flows. The latter expression thus, also relates to the case when bicasting is enabled for \( k \) flows.

Assuming that MN starts L2 handover right after sending the \( \text{HO}_\text{Ind} \) to p-FE as shown in Figure 6, the session disruption delay with bicasting service enabled for \( n \) flows can be expressed as follows:

\[
D^{\text{Bicast}(n)}_{\text{A3M-P}} = \max \left\{ t_{L2}, (n + 1) \cdot (\Omega_{fe} + s_{of} \cdot U) + (n + 2) \cdot (s_{of} \cdot P) + s_d \cdot U + T_{mn,fe} \right\}
\] (7)

If bicasting is not enabled, or is enabled only for \( k \) sessions, the session disruption delay for A3M-P is given as,

\[
D^{\text{Bicast}(0/k)}_{\text{A3M-P}} = 2 \cdot T_{mn,fe} + 4 \cdot s_{of} \cdot U + (2n + 2)\Omega_{fe} + 6 \cdot s_{of} \cdot P + s_d \cdot U
\] (8)

Unlike the A3M-P, bicasting is not applicable to the A3M-R. Moreover, the reactive operation starts at the L2 handover. The expression for session disruption delay is thus given as,

\[
D_{\text{A3M-R}} = t_{L2} + 2 \cdot T_{mn,fe} + (n + 1) \cdot \Omega_{fe} + 2n(s_{of} \cdot P) + (2n + 1)(s_{of} \cdot U) + s_d \cdot U
\] (9)
5.4.2 A3M - Packet Loss

In A3M, the adaptive buffering controls the packet loss. However, packet loss is still expected for a short interval even if buffering is enabled for some flows i.e. until the CTR updates flow entries after learning about the MN’s prospective handover. Considering that buffering is enabled for \( k \) ongoing flows, the packet loss can be expressed as follows:

\[
P_{loss}^{A3M-P} = \sum_{i=0}^{k} \lambda_p \cdot D_{A3M-P}^{*i} + \sum_{j=0}^{n-k} \lambda_p \cdot D_{A3M-P}^{*j}
\]

(10)

Here, \( i \) and \( j \) are mutually exclusive sets, with \( i \in \{0, ..., k\} \) i.e. the set of flows with buffering service enabled, and \( j \in \{0, ..., n-k\} \) i.e. the set of flows without active buffering service. \( D_{A3M-P}^{*i} \) represents the interval before buffering is enabled for the \( i \)-th session among the \( k \) sessions, and is given as,

\[
D_{A3M-P}^{*i} = (i + 1) \cdot (\Omega_{fe} + s_{of} \cdot U) + (i + 2) \cdot s_{of} \cdot P
\]

(11)

Similar to the A3M-P, the packet loss in A3M-R can be formulated as,

\[
P_{loss}^{A3M-R} = \sum_{i=0}^{k} \lambda_p \cdot D_{A3M-R}^{*i} + \sum_{j=0}^{n-k} \lambda_p \cdot D_{A3M-R}^{*j}
\]

(12)

Here, \( D_{A3M-R}^{*i} \) is given as,

\[
D_{A3M-R}^{*i} = t_{L2} + T_{mn,fe} + (i + 1) \cdot (\Omega_{fe} + s_{of} \cdot U) + (i + 2) \cdot (s_{of} \cdot P)
\]

(13)

5.4.3 A3M - Signalling Cost

In A3M, in order to adaptively provide the handover support services on a per-flow basis, it is necessary that different flow entries for different flows are installed at FE. Therefore, if a particular service is enabled for \( k \) ongoing flows, the CTR is required to send \( k \) flow-mod messages to the respective FEs. However, the packet forwarding service can be enabled efficiently if it is effectuated through a single flow entry which would match and forward the incoming traffic on destination address basis.

The signalling cost functions for both A3M cases are given in Table 3 and include (i) all handover support services enabled for \( k \) flows, \( C_{A3M-P}^{all}, C_{A3M-R}^{all} \), (ii) no handover support services enabled, \( C_{A3M-P}^{none}, C_{A3M-R}^{none} \), (iii) only path optimization enabled for \( k \) flows, \( C_{A3M-P}^{po(k)}, C_{A3M-R}^{po(k)} \), and (iv). Only buffering and bicasting are enabled for \( k \) flows, \( C_{A3M-P}^{bb(k)}, C_{A3M-R}^{bb(k)} \).
6 | NUMERICAL ANALYSIS

For numerical analysis, we consider a high mobility and session activity scenario where an MN undergoes \( n \) handovers with \( n \) active sessions, each anchored by a different subnet the MN has visited in the CMD/CTR domain. Thus, \( n \) being a common parameter which represents both mobility and session activity is varied to study the handover performance in different scenarios. In addition to \( n \), we also study the impact of other key factors which can significantly influence these handover performance metrics, such as link delay for session disruption delay, and packet arrival rate (\( \lambda_p \)) for packet loss. The parameter notations along with their default values are given in Table 4.

### 6.1 | Session Disruption Delay Comparison

The A3M-P ensures the least session disruption delay compared to A3M-R and PMIPv6-DMM. This is because the MN indicates its handover initiation from its previous subnet which allows packet forwarding from p-FE to n-FE which may start even before the MN attaches to the n-FE. However, no such indication is provided during A3M-R, as a result of which packet forwarding can start only after the CTR receives the notification of MN’s attachment to n-FE, and subsequently updates flow entries at p-FE.
FIGURE 8 The impact of (a) increasing number of ongoing sessions \( n \), and (b) per-byte transmission delay \( U \), on Session Disruption Delay and n-FE. Consequently, the A3M-R incurs up to twice the session disruption delay compared to A3M-P, as indicated in Figure 8a and 8b.

On the other hand, in PMIPv6-DMM which inherently operates in reactive mode, a session resumes at n-MAAR only after the CMD updates its current anchor. Moreover, unlike A3M, the packet forwarding in PMIPv6-DMM takes place only through tunnelling which results in higher delays if the anchor is several hops away from n-MAAR. This is because, each intermediate MAAR the MN visited after starting that flow, would require tunnelling and de-tunnelling of traffic. As shown in Figure 8a and 8b, this results in 47\% to up to 100\% higher delays as the number of handovers increase, compared to the A3M-R.

6.2 Packet Loss Comparison

The A3M-P incurs least packet loss since buffering (if set to \textit{TRUE} during the Handover Mode Selection phase) is promptly enabled by the CTR as soon as it learns about the MN’s handover. As shown in Figure 9a, packet loss can be significantly reduced if the increasing number of ongoing flows are provided with buffering service. Nonetheless, packet loss is inevitable in A3M, unlike, for instance, FMIPv6 protocol, since this process is controlled by the CTR and not the underlying FE entity itself. Moreover, Figure 9b shows that with increasing \( \lambda_p \), significant packet loss is incurred in PMIPv6-DMM compared to A3M. For the sake of consistency with Figure 9a, the default value of \( k \) in this case is increased to 8. Figure 9b also indicates that if buffering support is provided to a higher number of flows, the packet loss even in A3M-R can be significantly reduced which is at par with A3M-P.
6.3 Signalling Costs Comparison

The signalling costs for PMIPv6-DMM and A3M protocols vary considerably due to a number of factors. It is mainly impacted by the size of the signalling packet. The control packet sizes for PMIPv6 are much larger than OpenFlow flow-mod or Packet-In signals. This is because a PMIPv6 control packet carries several mandatory options. A PMIPv6 control packet is of 144 bytes on average\(^\text{10}\), while an OpenFlow packet is considered to be of 56 bytes\(^\text{11}\).

In PMIPv6-DMM, the DBC also has a significant impact on its overall signalling cost. Each MAAR (or c-Anchor) in PMIPv6-DMM requires to update its binding update list at every handover of MN. Consequently, for \(n\) ongoing flows, \(n\) PBU* messages are sent to the respective MAARs, each of which in turn update their binding update lists. In contrast, in A3M, the database is maintained at the CTR only, and no such updates are carried out at FEs.

In A3M, the signalling costs of A3M-R in some cases are lower compared to A3M-P. With an advanced handover initiation, the A3M-P requires more signalling and database updates than A3M-R to pre-empt the handover support services e.g. packet forwarding. Figure 10a shows a fundamental signalling cost comparison between PMIPv6-DMM and A3M when no handover support services are enabled for any flows. Figure 10b shows a similar trend when all handover support services are enabled.

It is to be noted that in A3M, a dedicated flow entry is required for each flow to ensure the provision of per-flow handover support services. This provisioning takes place through flow-mod messages from the CTR. Consequently, the signalling costs in A3M significantly increase if higher number of flows are provided the handover support services. As shown in Figure 10b, signalling costs in A3M are higher even compared to PMIPv6-DMM as \(k\) increases. However, for lower values of \(k\) (e.g. \(k = 2, 3\)), the A3M incurs comparatively lower signalling costs. Likewise, enabling buffering and bicasting services for an increased

---

\(^\text{10}\) According to OpenFlow specification\(^\text{22}\), a flow-mod message is of 56 bytes on average, while a Packet-In message has 32 bytes. However since Packet-In carries some received data, its actual size is normally larger. Hence, in performance evaluation its size is also considered to be 56 bytes.
FIGURE 10 The impact of $k$ on comparative Signalling Costs with (a) None handover support services enabled, (b) all handover support services enabled, (c) only buffering and bicasting services enabled, and (d) only path optimization service enabled.

number of flows, also incur very high signalling costs in A3M, as shown in Figure 10c. However, in case only path optimization service is enabled for $k$ flows, the A3M-P can ensure lower signalling costs compared to the A3M-R and PMIPv6-DMM as shown in Figure 10d.

7 | CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS:

The mobility environment in 5G networks requires intelligent handover management protocols, which not only achieve optimal handover performance but are also flexible enough to adaptively provide mobility services that best suit the mobile user needs. In this regard, we have proposed an SDN-based adaptive mobility management architecture, named A3M, which is capable of operating in multiple modes. It introduces a novel concept of Handover Mode Selection phase during which a suitable mode of handover operation is evaluated. The proposed mobility management architecture provides a realistically flexible protocol operation based on the standard IPv6 mobility and SDN principles, and incurs minimal operational complexity. With its flexibility
features, it ensures optimized handover performance in terms of session disruption delay and packet loss, even under extreme mobility conditions characterized by high mobility and high session activity. Moreover, in normal conditions, the A3M also promises reduced signalling costs compared to the PMIPv6-DMM protocol.

7.1 Future Research Directions:

The proposed concept of Handover Mode Selection phase can be applied to optimize a broad range of 5G mobility scenarios, in addition to the SDN-based DMM process considered in this paper. Following are some examples of complex 5G mobility management scenarios which can be potentially optimized through the proposed concept of Handover Mode Selection phase.

- The Handover Mode Selection concept can be employed in the control plane of the 3GPP CUPS and SBA architectures at MME (Mobility Management Entity) and AMF (Access and Mobility Management Function) respectively. The mobile network operators may deploy sophisticated policies for Handover Mode Selection based on big data analytics and machine learning techniques. Such policies might incorporate dynamic adjustments as the network/user conditions vary over time.

- The evaluations for enforcing handovers support services might incorporate policies specific to mobility scenarios, network conditions or user requirements. The operators, at their discretion, may also implement such policies that are centred around, for instance, cost savings, profit-making, resource saving, improved resource utility or ensuring high QoS for users. In some scenarios, the operators may wish to activate such services only for selected users (e.g. their premium user base).

- In the 5G systems and beyond, it is anticipated that novel application session types, especially for multimedia applications, will continually emerge. Such session types are expected to have their own specific set of mobility service requirements. An example technique aimed at enhancing the video streaming experience is the Dynamic Adaptive Streaming over HTTP (DASH). The DMM handover with DASH is studied in [44], in which the MAAR entities are considered to co-locate the mobile CDN (Content Delivery Networks). The DASH video chunks, in this case, can be transferred to the MN, from the currently serving MAAR or from other local caches/MAARs, if available. Otherwise, the content (video chunks) has to be fetched from a another local cache/MAARs as indicated by the CDN Controller, or the central video server. At MN’s handover in such a scenario, the Handover Mode Selection phase can consider the CDN cache’s location, as well as the available chunks information in all local caches as Handover Mode Selection parameters (e.g. for new Anchors Selection). It can also allow the pre-emptive coordination with the central CDN controller, for early retrieval of the desired content.

- The CDN Caching can be explored as a handover support service which can be evaluated through Handover Mode Selection phase and thus enabled adaptively for handover process optimization, especially for critical sessions such as the emerging Ultra-Reliable Low Latency Communications (URLLC).
UDNs is another popular technology, which is aimed at increasing the capacity of 5G networks. However, the deployment of UDNs results in several limitations, especially from the mobility management perspective, which include high interference, signalling overhead, high energy consumptions and operational costs. These limitations although can be somewhat addressed through the SDN technology, the concept of Handover Mode Selection, can significantly enhance the mobility experience in UDNs. For instance, the Handover Mode Selection phase, may consider the cell sizes, the interference level, the density of base stations (per square km), power adjustments (deciding to switch certain small cell base stations on and off, according to the current mobility patterns), and session preferences.

The Handover Mode Selection can also be used to decide which tier in the multi-tier Ultra-Dense Network (i.e. whether a small cell tier or a macro cell tier) is suitable for the MN to move to.

References


6. 3GPP. Architecture enhancements for control and user plane separation of EPC nodes; Stage 2 (Release 14) (TS 23.214). 2018.


