Edge caching with mobility prediction in virtualized LTE mobile networks

Andre S. Gomes, Bruno Sousa, David Palma, Vitor Fonseca, Zhongliang Zhao, Edmundo Monteiro, Torsten Braun, Paulo Simoes, Luis Cordeiro

PII: S0167-739X(16)30207-2
DOI: http://dx.doi.org/10.1016/j.future.2016.06.022
Reference: FUTURE 3088

To appear in: *Future Generation Computer Systems*

Received date: 30 January 2016
Revised date: 18 April 2016
Accepted date: 20 June 2016

Highlights (for review)

- A new model is introduced for optimized content migration in mobile networks.
- The architecture fully exploits new concepts such as Future Internet and NFV.
- Mobility prediction in LTE virtualized networks further improves the system.
- It achieves fivefold improvements in performance experienced by end users.
- Mobile network providers optimize resources and have significant savings.

Edge Caching with Mobility Prediction in Virtualized LTE Mobile Networks

Andre S. Gomes[a,b], Bruno Sousa[c,b], David Palma[d,c,b], Vitor Fonseca[c], Zhongliang Zhao[a], Edmundo Monteiro[b], Torsten Braun[a], Paulo Simoes[b], Luis Cordeiro[c]

[a]*Institute of Computer Science, University of Bern, Switzerland*
[b]*CISUC, University of Coimbra, Portugal*
[c]*OneSource, Consultoria Informática Lda. Coimbra, Portugal*
[d]*Department of Telematics, NTNU, Norway*

Abstract

Mobile Edge Computing enables the deployment of services, applications, content storage and processing in close proximity to mobile end users. This highly distributed computing environment can be used to provide ultra-low latency, precise positional awareness and agile applications, which could significantly improve user experience. In order to achieve this, it is necessary to consider next-generation paradigms such as Information-Centric Networking and Cloud Computing, integrated with the upcoming 5th Generation networking access. A cohesive end-to-end architecture is proposed, fully exploiting Information-Centric Networking together with the Mobile Follow-Me Cloud approach, for enhancing the migration of content-caches located at the edge of cloudified mobile networks. The chosen content-relocation algorithm attains content-availability improvements of up to 500% when a mobile user performs a request and compared against other existing solutions. The performed evaluation considers a realistic core-network, with functional and non-functional measurements, including the deployment of the entire system, computation and allocation/migration of resources. The achieved results reveal that the proposed architecture is beneficial not only from the users' perspective but also from the providers point-of-view, which may be able to optimize their resources and reach significant bandwidth savings.

Keywords:
Information-Centric Networking, Content Migration, Edge Caching, Mobility Prediction, LTE, Mobile Cloud, Follow-Me Cloud.

1. Introduction

Mobile Edge Computing (MEC) [1], as a key 5th Generation (5G) network enabling technique, enables a cloud-based Information Technology (IT) service environment at the edge of mobile networks. It provides benefits such as ultra-low latency, precise positional awareness and agile applications, being foreseen as an essential building block for the 5G mobile networks. Moreover, it also plays a key role in supporting new business solutions based on smart devices and machine-to-machine (M2M) communication. From services and applications to content, they can all be accelerated by taking advantage from increased responsiveness at the edge of the network. Therefore, end users' experiences will be enriched through efficient network and service operations, based on the radio and core network conditions. Considering these facts, we may conclude that the key characteristics of MEC are the following:

- *Proximity*: Being deployed close to the network end users, MEC is particularly useful to better serve and understand users' preferences on content. MEC may also have direct access to the devices, which can easily be leveraged by applications;
- *Low latency*: As edge services run close to end-devices, latency is considerably reduced. This can be utilized to react faster, to improve user experience, or to meet the requirements of delay-sensitive applications;
- *Location-awareness*: A locally-deployed service can leverage low-level signaling information to anonymously determine the location of each connected device. This enables various applications, such as Location-based Services and analytics solutions, among others;
- *Network context-awareness*: Real-time network data (such as network conditions, radio status and more) can be used by applications/services to offer context-related services that can differentiate the mobile broadband experience and be monetized. New applications can be deployed to connect mobile devices with local points-of-interest, events, among many other possibilities.

Using MEC as the starting point, in this paper we present and detail a model for content distribution optimization in mobile networks. At the same time, to explore new paradigms and gather additional benefits, we leverage emerging 5G concepts that are becoming standards in the industry.

Email addresses: gomes@inf.unibe.ch (Andre S. Gomes), bmsousa@onesource.pt (Bruno Sousa), palma@item.ntnu.no (David Palma), fonseca@onesource.pt (Vitor Fonseca), zhao@inf.unibe.ch (Zhongliang Zhao), edmundo@dei.uc.pt (Edmundo Monteiro), braun@inf.unibe.ch (Torsten Braun), psimoes@dei.uc.pt (Paulo Simoes), cordeiro@onesource.pt (Luis Cordeiro)

One of these concepts is Network Function Virtualization (NFV) [2], which is expected to improve dynamic adaptation of networks to different conditions and requirements, further increasing the impact of Mobile Cloud Computing (MCC) [3, 4] and MEC. Cloud Radio Access Networks (C-RAN) [5, 6] is a relevant trend in this direction, bringing the possibility of virtualizing the entire 3rd Generation Partnership Project (3GPP) Long Term Evolution (LTE) radio infrastructure, except for a part of the antenna hardware. Virtualized infrastructures extend the mobile cloud-computing concept to the Radio Access Network (RAN) and explore the modularity of the components, together with the usage of general-purpose hardware infrastructures to run evolved Node Bs (eNBs). Such fact transforms the C-RAN into an enabler for deployment of value-added services closer to the edge of the network (i.e. in close proximity to mobile users).

In parallel with these 5G concepts, Information-Centric Networking (ICN) [7] is becoming increasingly important, alongside other new Future Internet (FI) concepts, proposing a paradigm change in how content can be handled more efficiently and how users' experience can be improved. In fact, the fast-pace growth and evolution of Mobile Networks, driven mainly by smart-consumer devices, new contents, video streaming, Peer-to-Peer (P2P) applications, and induced by the usage of traffic-heavy applications by millions of people worldwide, results in a tremendous demand of bandwidth [8] and in several challenges that need to be tackled by new concepts and technologies.

With such challenges and requirements in mind, new proposals arise to explore the benefits of combining the new 5G and FI concepts to deliver a more efficient and performing end-to-end solution for mobile users. One example is the deployment of ICN co-located with 3GPP LTE mobile networks [9], using ICN together with C-RAN to deliver content efficiently very close to the edge. Despite the demonstrated improvements in terms of performance and reduction of traffic at the core network, such enhancement is not straightforward. Indeed, enhanced ICN caching mechanisms must efficiently populate caches and maintain content where it will yield the most benefit. Such benefit is intrinsically related with users' mobility patterns, as cached content should be available at users' new locations (e.g. associated to different LTE eNBs). This support requires preemptive actions resulting, for instance, from mobility prediction algorithms [10, 11, 12], which may be employed for triggering the migration of content. Nevertheless, determining the optimal set and subset of content to be migrated, for a given amount of available resources (e.g. cache sizes, number of routers) and user mobility patterns, falls into an NP-complete optimization problem.

This paper introduces an end-to-end architecture and the Mobile Follow-me Cloud (M-FMC) model to get an approximate solution for this problem with high accuracy and meeting real-time requirements. Therefore, the following technical contributions are part of this work:

- An end-to-end cloud architecture for the deployment and orchestration of ICN, its dependencies and M-FMC components is proposed.
- The usage of different modular software components is suggested in order to improve the efficiency of key services, such as monitoring services to enable auto-scaling depending on load.
- Multiple Attribute Decision Making (MADM) algorithms are proposed to get an approximate solution to the optimization problem of content migration, considering several criteria such as content popularity, content size and the capacity of the caches of different routers.
- Mobility prediction is analyzed, described and proposed as an enabler for the M-FMC model's content migration.
- The performance of the M-FMC model is assessed using different candidate MADM algorithms and validated against optimal solutions computed off-line. Moreover, multiple experiments are performed in realistic scenarios to evaluate benefits from both end users' and mobile network operators' perspectives.

Evaluation results, obtained in a realistic mobile core network testbed, demonstrate that the M-FMC model using the MeTHODICAL [13] algorithm performed better than the remaining competitors, returning its solutions in deterministic and polynomial time, thus not compromising its employment in ICN-based FI scenarios.

The remainder of this paper is organized as follows. An analysis of existing contributions and proposals to address content and service migration is presented in Section 2, followed by the presentation of an enhanced system for smart MEC, in Section 3, and the definition of the proposed model. Section 4 describes experimentation scenarios for the evaluation of this model, taking into account the users' location in a mobile cloud environment, which lead to the obtained results provided in Section 5. Finally, Section 6 discusses the main achievements of this work.

2. Related work

2.1. Information-Centric Networking

Nowadays, the prevailing paradigm for content requests is based on client/server principles: every time a user requests an object it queries a specific resource at a previously known server. However, according to the ICN approach the concepts of client and server no longer exist, and nodes may simultaneously play multiple roles. Moreover, requests are not directed to a particular node. When a user is interested in a certain content object, it sends an Interest message to the network, consisting of a provider and an object name (in most naming schemes). This message is then routed by other nodes (ICN routers), which have forwarding tables based on content naming prefixes. The requested content object will eventually be reached (assuming it is available in the network) and it will be forwarded to the original requester, using the same communication path traversed by the Interest message. ICN brings a number of advantages when compared to traditional approaches. First, performance has substantial gains due to caching, faster lookups and intelligent routing. Second, mobility support is greatly improved [14] due to a decoupled approach and the lack of content transfer sessions. Third, security is amplified by trust mechanisms and inherent avoidance of known vulnerabilities [15]. These improvements are particularly relevant for mobile networks, where mobile edge resources are scarcer and traffic loads become higher. Moreover, they also bring advantages regarding cost savings – less resources used and a lower investment required from mobile operators. In this line of thought, current works [16, 17] evaluate the feasibility of deploying ICN together with LTE mobile networks, leveraging the C-RAN concept and its role as an enabler for the deployment of additional services within these networks. Their findings include bandwidth savings at the core network and lower latency when retrieving content from ICN routers co-located with LTE eNBs. In addition, the impact of processing LTE frames is low and the performance of content caching in ICN is superior when compared to HyperText Transfer Protocol (HTTP) caching mechanisms. ICN provides distributed storage, caching and content relocation features that could be used to optimize the distribution of content and enhance caching strategies. For instance, taking into account mobility-prediction results, users' content could be stored and made available in future locations, which would reduce content access delays and unnecessary bandwidth spikes in the operators' networks.

2.2. Mobility Prediction

Predicting mobile users locations at any time moment in the future is essential for a wide range of mobile applications, including location-based services, mobile access control, mobile multimedia Quality of Service (QoS) provision, as well as the resource management for mobile communication and storage. In a cloudified LTE mobile network, different virtualized network services might need the end users' predicted location to optimize network performance. As an example, ICN could benefit from the mobility prediction results to have improved caching strategies and place the content closer to users' predicted locations in advance, thus improving the experience for users in terms of access delay.

A large number of different algorithms have been proposed in the literature for predicting the future positions of users in mobile networks. Generally speaking, proposed schemes carry out prediction based on mobility models that can be categorized into three main classes: *Temporal Dependency*, *Spatial Dependency*, and *Geographic Restriction* [18]. The mobility models represent the movement of mobile nodes, and how their location, velocity and acceleration change over time. Prediction schemes based on the *Temporal Dependency* mobility model assume that mobile node trajectories may be constrained by some physical characteristics such as acceleration, velocity, direction, and also affected by their movement history [19, 20]. In case of the latter, estimation is performed based on the assumption that mobile nodes incline to travel in a correlated manner and mobility of one node is affected by the mobility pattern of other neighboring nodes [19, 21].

The solutions relying on *Geographic Restriction* assume that node trajectories are subject to the environment and motion of mobile nodes is bound by geographic restrictions such as freeways or local streets in urban areas. Likewise, pedestrians may also be blocked by buildings and other obstacles [22, 23]. However, most of the works focus on providing an isolated prediction framework, which cannot be utilized by other network services. Few efforts have been made in providing mobility prediction as a supporting Virtual Network Function (VNF) for the virtualized LTE mobile networks. Previous work on Mobility Prediction as a Service (MOBaaS) [24] provides a fully cloudified mobility prediction service that supports the on-demand life-cycle management of the mobility prediction service instantiation and disposal on top of the cloud infrastructure. The mobility prediction algorithm is based on the Dynamic Bayesian Network (DBN) model, and the rational behind using DBN is that the next location visited by a user depends on (*i*) its current location, (*ii*) the movement time, and (*iii*) the day that user is in the movement.

2.3. Content Migration

Predicting mobile users' locations has big potential in various telecommunication applications, including mobile access control, resource management for mobility and storage, content migration, etc. However, there are few relevant and actually feasible proposed strategies for service or content placement/migration specifically taking into account users' mobility. Antonescu et al. take into account user mobility for placement and scaling of services [25], performing orchestration of distributed cloud services based on prediction of user mobility: more or less resources are allocated based on how the system predicts users will move to/from the area of each Data Center (DC). Nonetheless, migration of services from one location to another is not considered.

In this direction, Follow-Me Cloud (FMC) is an important concept towards service migration strategies. First proposed by Taleb et al. [26], FMC assumes that mobile networks are supported by a few large DCs at the core and many small DCs geographically distributed at the edge of the networks. At the same time, it proposes to deploy cloud services in the small DCs to improve proximity to the users. Hence, when users of a service move from one region to another, services (and content) shall follow the users, remaining with assured levels of quality and availability. The decision to migrate a specific cloud service is based on a set of models. Together with the random walk mobility model, the analytical model described by Taleb, Ksentini et al. [27, 28] and based on Markov Decision Processes, attempts to keep cloud services on the optimal DC according to users' mobility, interests and other network-related factors. This solution is already interesting, providing a preliminary perspective on the possibilities and raising important open issues. For instance, the system considers only a 1-dimensional mobility model, and a single destination has to be considered despite the fact that the user may be moving and passing by multiple destinations in the selected period of time. These open issues, as well as the decision about which services to migrate and whether group mobility is better than single user mobility, have so far not been subject to further study. Moreover, services are the focus of these works, where content specificities are disregarded.

Regarding strategies for content replication and migration, existing proposals look at the problem in Peer-to-Peer (P2P) networks from the providers' perspective [29], or rely on a Global Name Resolution Service to prefetch and cache relevant content [30]. Considering a typical hierarchical Content Delivery Network (CDN) architecture, the problem is how to distribute content among multiple network nodes in order to avoid network traffic at higher layers of the hierarchy and achieve latency improvements. Decisions to move objects to lower layer nodes (i.e. nodes with less resources) are usually based on the costs of migration together with the overall number of requests for the content object within a time interval. These decisions aim at maintaining an optimal usage of available storage space, together with the highest throughput possible, corresponding to an NP-complete problem. Such proposals typically yield a high benefit, but one may argue that such a system may not scale with complex hierarchies due to a large number of nodes and the need for global information about content. Moreover, decisions are made with low frequency and cannot keep up with the pace of very dynamic networks, where users often move and need content to be available beforehand. Also, these works do not explore the fact that popularity is typically local and cluster based [31, 32], and only account for overall popularity. In line with this, proposals that are simpler and exploit most of the benefits of hierarchical caching already exist [33] and can be considered as a basis for our own work.

However, mobility of users is still not taken into account by existing approaches [33, 34]. In this direction, Vasilakos et al. [35] suggest the use of proactive migration strategies for content: migration is triggered when it is predicted that the user will move to a neighbor location. Using proxies at most 1-hop away from the user, authors propose to pre-fetch subscribed content between proxies whenever it is predicted that the user will disconnect and move to the region of another proxy. To be able to do this pre-fetching, it is assumed that all destinations are known and that migration cost is minimized while attempting to maximize the benefits in terms of performance. Provided results show high gains regarding latency, but ignore some issues that can arise in larger scale networks: the number of criteria for proxy selection is low and not weighted, and replacement policies for caches without free storage space are not discussed at all. At the same time, mobility prediction is too simple and only for a single user, while the gain will only be meaningful if migrations are performed for multiple users (groups).

Considering all the described proposals and the issues they fail to address, in this paper we propose a system that relies on their positive findings and, at the same time, addresses the challenges not previously taken into consideration.

3. Mobility Prediction-Enhanced Smart Caching at Network Edges

The M-FMC proposal is described in this section, leveraging the benefits of the MEC paradigm (e.g. lower latency, reduced traffic at the core, more distributed systems, etc.) together with the cloud computing concept.

3.1. Base Architecture

One of the efforts that follows this approach is the Mobile Cloud Networking (MCN) project, an European Union (EU) FP7 Large-Scale research project [36], which integrates the use of cloud computing concepts in LTE mobile networks with the objective of increasing its performance by building a shared distributed LTE mobile network that can: (*i*) optimize the utilization of computation, storage and networking resources, (*ii*) minimize communication delays, (*iii*) reduce the required Core Network bandwidth, and (*iv*) enable multiple virtual mobile network operators, while using a common physical infrastructure. The MCN project extends the cloud computing concept beyond the typical (*macro*) data centers towards smaller (*micro*) data centers, which are distributed within the Evolved UMTS Terrestrial Radio Access Network (E-UTRAN). These data centers are able to deploy and run cloud-based E-UTRAN denoted as Radio Access Network as a Service (RANaaS), as well as parts of the Evolved Packet Core as a Service (EPCaaS) to be co-located with RANaaS for improved performance.

As far as ownership and operation of this base infrastructure is concerned, it can be distributed among several different entities, i.e. providers. For instance, a Mobile Network Operator (MNO) can own and control the entire infrastructure and therefore fully manage its components and technologies. In another example, a different service provider may act as a proxy and provide access to that infrastructure without owning or even operating any of its physical components. Such service provider would have to sign business agreements with the owner(s) of the physical infrastructure, which can for instance be MNOs that operate in the desired geographic areas or infrastructure providers that own/control DCs in strategic locations, either micro or macro DCs.

The innovation here is that a MCN provider exploits the MCN architecture to compose and operate virtual end-to-end infrastructures and platforms on top of several different and separated physical infrastructures belonging to different providers, thus providing an end-to-end service architecture that has no geographic boundaries and brings new potential benefits.

3.2. End-to-End M-FMC Architecture

With the proposed End-to-End architecture, M-FMC is able to support content migration with advanced strategies that consider user and group mobility predictions for optimal content migration. This way, M-FMC differs from related FMC approaches by allowing the migration of content at different cache levels according to the proximity of users, fulfilling the goals of MEC. M-FMC also includes support to a comprehensive set of metrics for decisions related to content migration, which go beyond content size or content access latency in hierarchical networks. Moreover, M-FMC is more granular and the associated optimization mechanism for content migration is not tied to the particularities of scenarios, e.g. number of routers or moving users.

One of the key benefits of the architecture defined in the previous subsection is the virtualization of the entire mobile network's infrastructure (except the radio antennas). With that virtualization and subsequent cloudification of functionalities as services, many enhancements and new features can be explored at the edge of the mobile networks.

For this work, we assume that ICN is a cloud service compliant with the MCN architecture – ICN as a Service (ICNaaS) – and can be integrated with other services to achieve better end-to-end performance for content delivery. Namely, with M-FMC features enabled by getting external input data from other services, it can cache the content objects where they will yield the greatest benefits for end users while saving network resources. As depicted in Fig. 1, this requires integration with other MCN services, such as RANaaS, EPCaaS and MOBaaS.

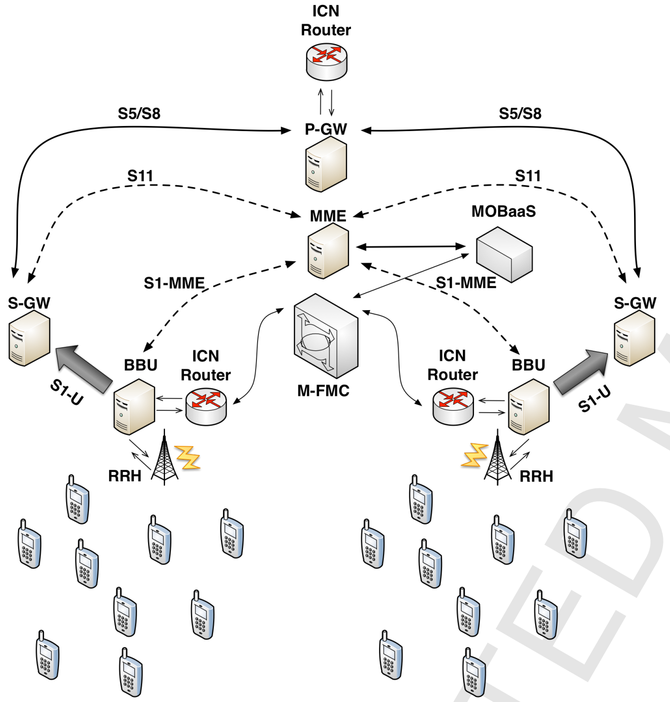

Figure 1: End-to-End M-FMC Architecture

With RANaaS, LTE 3GPP base stations (eNodeBs) can be split into two key components: Base Band Units (BBUs) and Remote Radio Heads (RRHs). While RRHs cannot be virtualized because they include the antennas, BBUs have that potential and therefore other services may be easily integrated with them at the same DC. One of those is ICNaaS [16]. ICNaaS, represented here as ICN Routers (cf. Section 3.3.1), gets filtered traffic that matches a certain number of rules (e.g. UDP port 9695) and can process it before it is encapsulated in a General Packet Radio Service (GPRS) Tunneling Protocol User plane (GTP-U) tunnel and sent to the Evolved Packet Core (EPC), here represented by its key components (S-GW, P-GW, MME) together with its interfaces (S1-U, S1-MME, S5, S8, S11). If the content is cached, the EPC has to be informed so that it will still charge the user for the data. Otherwise, the request is forwarded to the following ICN Router in the core network.

MOBaaS delivers user mobility detection and prediction by analyzing data coming from the EPC Mobility Management Entity (MME). With such information, smart decisions about content location can be made by the M-FMC components, as detailed in the following subsections.

3.3. M-FMC Model

The architecture of ICNaaS is twofold: its core ICN components and the M-FMC components. Therefore, the architecture of M-FMC is part of the architecture of ICNaaS, which is depicted in Fig. 2 and described in detail below.

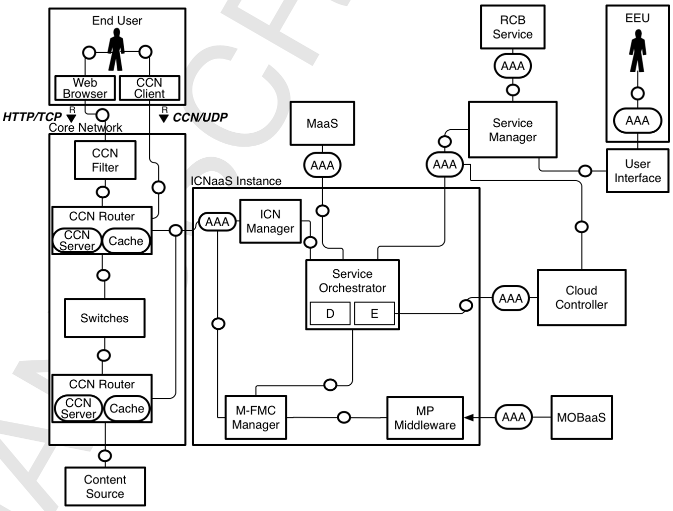

Figure 2: ICNaaS + M-FMC Architecture

3.3.1. Information-Centric Networking as a Service

ICNaaS is a service aimed at deploying an ICN infrastructure in a cloud environment, leveraging the advantages of cloud principles and pushing the boundaries of existing content delivery technologies. In this specific case, an ICN approach named Content-Centric Networking (CCN) [7] was selected due to its relevance in the research community and its open source code.

The core of ICNaaS consists of six main components: CCN Routers, CCN Filter, ICN Manager, Service Orchestrator, Service Manager and Management Agent. Initially, when an Enterprise End User (EEU) wants an instance of the service, it contacts the Service Manager (SM). This component has a catalog of ICN services that can be offered, and upon request it will deploy a Service Orchestrator (SO) by contacting the Cloud Controller (CC), which is the component that manages the cloud platform. Once the SO has been deployed, it will use its Execution (E) sub-component to deploy all the remaining components of the service instance in the following order:

1. The ICN Manager component to handle automated management of the entire ICN layer.
2. The CCN Filter, which converts HTTP Requests into ICN Interest messages and HTTP Responses into ICN Data messages.
3. The CCN Routers that implement a subset of ICN functionalities, in particular CCN.
4. The M-FMC components (described in the subsection below).

5. The Management Agent, which provides an interface for the EEU to have fine-tuned manual control over the service instance.

At the same time, the SM handles dependencies of the ICN service. Monitoring as a Service (MaaS) is used to monitor components and provide information to the Decision (D) subcomponent for scaling in and scaling out decisions that allow the service to have the exact number of resources to handle the load at a given time. The second dependency is MOBaaS, which as described in the previous section is used by the M-FMC components to get input related to user mobility, either predicted or detected. Another dependency is Authentication, Authorization, and Accounting (AAA), used to authenticate the requests between components. A fourth and last dependency is Rating, Charging and Billing as a Service (RCBaaS), used by the SM to charge and bill the EEU for the resources used by its service instances.

3.3.2. M-FMC Decision Mechanisms

The MADM module provides a quasi-optimal subset of content to migrate, providing an approximate solution for the NP-complete problem. The MADM algorithm provides a score of the content to be migrated based on the input criteria, such as content popularity and content size. Diverse MADM algorithms could be considered, such as MeTHODICAL [13], TOPSIS [37] and DiA [38]. Algorithm 1 details MeTHODICAL steps for B-benefits, K-costs and multiple criteria organized in a $M_{n,m}$ matrix for m-criteria and n alternatives. Step 2 allows the weighting of normalized $\widehat{\mathbf{B}}_{i,b} = \mathbf{b}_b \times \overline{\mathbf{B}_{i,b}}$ benefits and $\widehat{\mathbf{K}}_{i,c} = \mathbf{k}_c \times \overline{\mathbf{K}_{i,c}}$ costs, with $i = 1, 2, \cdots, n$, $b = 1, 2, \cdots, B$ and $c = 1, 2, \cdots, K$.

Algorithm 1 – MeTHODICAL optimization steps (as per [13])

```
Require: $\sum_j^B \mathbf{b}_j = 1$ #Benefits weights vector
Require: $\sum_j^K \mathbf{k}_j = 1$ #Costs weights vector
Require: $\sum_i^m \sum_j^B \mathbf{B}_{i,j} \geq 0$ #Benefits matrix
Require: $\sum_i^m \sum_j^K \mathbf{K}_{i,j} \geq 0$ #Costs matrix
Require: $\mathbf{s}_{i,(t-1)} = 0$ #Initialize Score vector for $(t)ime - 1$
1: $\overline{\mathbf{N}_{ij}} = \frac{\mathbf{M}_{i,j} - min(\mathbf{M}_{n,m})}{Max(\mathbf{M}_{n,m}) - min(\mathbf{M}_{n,m})}, i = 1, \cdots, n$ #Normalization
2: $\widehat{\mathbf{G}}_{i,j} = \mathbf{n}_j \times \overline{\mathbf{N}_{ij}}$ with $i = 1, 2, \cdots, n$ and $j = 1, 2, \cdots, m$
3: $I(\widehat{\mathbf{B}}_j) = max\{\widehat{\mathbf{B}}_{i,j} | i = 1, 2, \cdots, n\}$ #Ideal Benefits solution
4: $I(\widehat{\mathbf{K}}_j) = min\{\widehat{\mathbf{K}}_{i,j} | i = 1, 2, \cdots, n\}$ #Ideal Costs solution
5: $\Delta(\widehat{\mathbf{B}}_i) = \sum_{j=1}^{B} \left[ \frac{[I(\widehat{\mathbf{B}}_j) - \widehat{\mathbf{B}}_{i,j}]^2}{[I(\widehat{\mathbf{B}}_j) - A(\widehat{\mathbf{B}}_j)] + 0.01} \right] A(\widehat{\mathbf{B}}_j) = m(\widehat{\mathbf{B}}_j) + v(\widehat{\mathbf{B}}_j)$
6: $\Delta(\widehat{\mathbf{K}}_i) = \sum_{j=1}^{K} \left[ \frac{[I(\widehat{\mathbf{K}}_j) - \widehat{\mathbf{K}}_{i,j}]^2}{[I(\widehat{\mathbf{K}}_j) - A(\widehat{\mathbf{K}}_j)] + 0.01} \right] A(\widehat{\mathbf{K}}_j) = m(\widehat{\mathbf{K}}_j) - v(\widehat{\mathbf{K}}_j)$
7: $\mathbf{s}_i = \sqrt{\alpha \times \Delta(\widehat{\mathbf{B}}_i) + (1 - \alpha) \times \Delta(\widehat{\mathbf{K}}_i)},\ i = 1, 2, \cdots, n$
8: $\mathbf{s}_{i,t} = \mathbf{s}_i + v\big(\mathbf{s}_i, \mathbf{s}_{i,(t-1)}\big),\ i = 1, \cdots, n$ #Set current score
9: $\mathbf{r}_i = order\big(\mathbf{s}_{i,t}\big)$ #Vector in crescent order
```

The difference among the MADM algorithms is related mostly with the methods used to determine the optimal solution, which employ distance functions to determine the distance to ideal values, c.f. Algorithm 1 steps 5 and 6. TOPSIS employs the Euclidean distance, $D_i = \sqrt{Id_j - v_{i,j}}$, while DiA employs the Manhattan distance, $D_i = |Id_j - v_{i,j}|$. The employment of such distances leads to non-optimal results due to the missing correlation between the values of the different criteria [39]. The distance of MeTHODICAL, $A(\widehat{\mathbf{K}}_j) = m(\widehat{\mathbf{K}}_j) \pm v(\widehat{\mathbf{K}}_j)$, considers a range that is determined by the m-mean and v-variance functions, therefore supporting correlation.

Other optimization techniques could be used, such as Linear Programming, but these are commonly tied to the specificities of each scenario and therefore would require adaption whenever new criteria are considered or changes on the scenario are verified.

3.3.3. M-FMC components

M-FMC has two key components: the CCN Server, which runs at every router, and the FMC Manager, which is responsible for all the decision-making and control of the routers' cache management actions.

The CCN Server is responsible for sending all the monitoring data to the FMC Manager, so it can be stored at the centralized database. It is also responsible for migrating the contents when requested by the M-FMC Manager. Monitoring is achieved by a few minor changes in the code of CCN routers, when based on the popular CCNx framework [40].

The M-FMC Manager stores the monitoring data (i.e. name prefix information, size of caches, popularity of content) coming from the routers in a database and, when a migration is needed, it uses this data to determine the list of objects that the router should have in its cache to better serve its current and arriving users. After deciding on the content (and the respective subsets), the FMC Manager sends the subset list of objects to be migrated. The FMC manager is orchestrated (e.g. deployed, provisioned, disposed) by the service Orchestrator of ICNaaS. The interface with MOBaaS is employed to receive information regarding user mobility predictions. To support multiple criteria decision-making, the M-FMC Manager interfaces with the MADM module described in subsection 3.3.2.

The M-FMC model requires an accurate and efficient decision-making mechanism, no matter the scenario where it is operating. Therefore, the decision mechanism considers content popularity and content size metrics (since local caches of routers have size limitations). Content popularity, *popCont*, defined in Equation 1, is a composed metric that is formulated by considering three parameters, including the content popularity at the source routers (popSr/reqSr) and destination routers (popDs/reqDs); the number of users per source cell moving to the destination cell (movU); and the number of users at the destination router (dstU).

$$popCont_i = \frac{\sum_{k=1}^{N} \left(\frac{popSr_{i,k}}{reqSr_k} * movU_k \right) + \frac{popDs_i}{reqDs} * dstU}{N + 1} \quad (1)$$

Hence, Eq. 1 depicts the formula that determines the content popularity for the N source cells considered in each migration.

3.3.4. MOBaaS

As a cloud-based supporting service, MOBaaS should provide requested information on-demand. Accordingly, several cloud computing principles have been considered to its design, in which on-demand service management and prediction functions are the two most important features.

Its architecture includes several components: the mobility prediction algorithm, the history data retriever and data converter, a front end, a SO and a SM. SO and SM are responsible for management actions related to the service instances of MOBaaS, which include service initialization, disposal and scaling operations. Once the service instance is running, a front end will handle mobility prediction requests and algorithms. The mobility prediction algorithms block includes the prediction algorithm that is based on a Dynamic Bayesian Network (DBN). History data retriever and converter is in charge of collecting the user movement traces from system monitoring tools and converting them into the proper format. Due to the difficulties of collecting actual user traces, we used a dataset provided by Nokia as historical user movement traces [41].

The proposed mobility prediction algorithm [24] benefits from Dynamic Bayesian Networks (DBNs). The rationale behind using DBNs is that the next location (cell) visited by a user depends on: its current location, the current time, and the day of week on which the user is in movement. We model the future location distribution using both a location dependent distribution and a temporal dependent distribution. Hence, each of them can be modeled as a simple first order Markov Chain (MC), which encodes the probability of transitions between the cells.

4. Methodology

This section describes the evaluation methodology and scenarios we adopted to validate the proposed M-FMC model.

4.1. Evaluation Goals

The evaluation aims at validating the M-FMC model as an enabler of content migration mechanisms for ICN in the context of MEC environments. Moreover, a subsequent goal is to validate the MADM algorithm for the M-FMC model in terms of decision optimality, accuracy and efficiency.

The NP-complete problem of content migration can be solved by the Knapsack algorithm [42] by considering the best theoretical solution as the one that determines the correct content subset to be migrated to the cache of the destination routers. The correct content subset is the one that ensures lower latency based on the requests from users while, at the same time, maximizing the profit by filling the cache of routers. Indeed, the optimization of the M-FMC model follows the optimization pursued in the 0/1 Knapsack problem where, given the size of the destination router's cache, the content to fill the cache must be selected under the constraints of content popularity and size of content objects. As Knapsack is very intensive in terms of computation and takes a long time to be executed, the MADM algorithms have been considered because of their flexibility and efficiency when providing approximate optimal solutions irrespective of the scenario.

4.2. Evaluation Platform

In order to evaluate the proposed M-FMC model, two different platforms were considered: a small testbed to validate multiple MADM algorithms for decisions while selecting the most appropriate one and a larger testbed to evaluate the end-to-end integrated model from the users experience perspective in the context of the MCN project.

4.2.1. Platform A - MADM Algorithms Validation

The evaluation conducted in this platform had the goal of validating the algorithm to enable optimized migration of content as proposed in the M-FMC model. It includes two steps: a first step with a small testbed to fill databases and a second step at a Linux cluster to run large scale experiments in parallel.

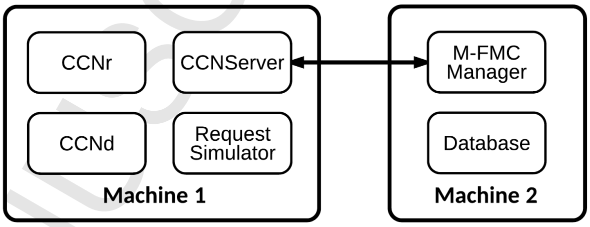

Figure 3: Evaluation testbed

In the first step, only the FMC Manager and the Request Simulator components are included, both implemented in the Python programming language, as depicted in Fig. 3. The Request Simulator component was configured to process the requests according to the configurations of each defined scenario. The CCN Server receives the commands to migrate content according to the decisions of optimization algorithms running in the FMC Manager. These components, as shown in Fig. 3, run in two Virtual Machines (VMs) inter-connected with 10 Gigabits per second (10 Gbps) links, each with 2 vCPUs, 4GB of RAM and 40GB of disk space. The MADM algorithms validation compares Knapsack [42], MeTHODICAL [13], TOPSIS [37] and DiA [38] algorithms. Knapsack, as an optimization technique providing optimal solutions for NP-complete problems, has associated a fully polynomial time approximation scheme, which leads us to the usage of the UBELIX cluster [43] for a timely determination of optimal solutions. Moreover, this validation did not consider mobility predictions. We assumed that users connected to one source cell move towards an arbitrary destination cell, as only the content subset migration is being analyzed (not location) when simulating file requests following the different distributions being evaluated.

To perform the first step of the validation, the small testbed depicted in Fig. 3 was run for several days to inject requests into the database of the FMC Manager for the files of each scenario, following the specified popularity distributions. With the database populated with the simulated requests of two different cells, the second step could start. Data was hence processed with multiple parallel jobs at the UBELIX cluster using Python and R [44] to apply the MADM and knapsack algorithms.

4.2.2. Platform B - End-to-End Users Experience Validation

To assess the users experience in an end-to-end fashion, a deployment of all the required services was done using the MCN architecture. Namely, ICNaaS, MaaS and MOBaaS were deployed as a service combination, i.e. simultaneously deployed and aware of each other. Due to existing constraints the EPCaaS data containing user mobility information was received from a mobility trace (described below in subsection 4.3), while RANaaS was left out to simplify the setup and allow a higher number of cells. This deployment's setup is shown in Fig. 4 at the MCN end-to-end orchestration framework's level, depicting AAA authentication where applicable.

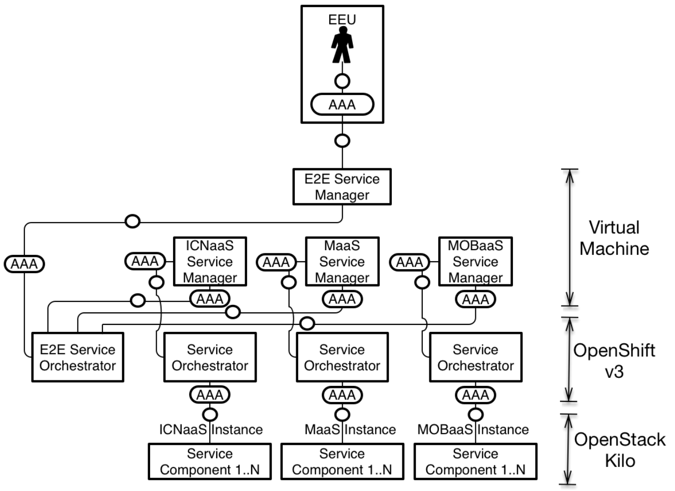

Figure 4: End-to-End Orchestration

The first step for deployment is the authenticated request from an Enterprise End User (EEU) to the End-to-End Service Manager (E2E SM). This component creates an End-to-End Service Orchestrator (E2E SO) to manage this deployment, using Red Hat OpenShift Origin version 3 [45] as a container platform. After the E2E SO has been created, it uses a manifest file to list the services to be deployed and their dependencies. Then, endpoints are queried from a catalog and services are requested by contacting their SMs, which have the descriptions of their services and information on how to create their SOs. Hence, these SMs create SOs at OpenShift, which will instantiate the required components at OpenStack Kilo [46]. When all the components have been deployed, endpoints of dependencies are provided to the dependent services, e.g. ICNaaS requires MOBaaS and needs its endpoint. That triggers the provisioning phase, which concludes the deployment and provides information to the EEU that the E2E deployment is ready to be used.

As for the components deployed by the orchestration framework at OpenStack Kilo for the required services, they are summarized in the following list (vCPU is a virtual CPU unit, mapped to a thread/core by the Kernel-based Virtual Machine (KVM) hypervisor [47]):

- One ICN Manager with 1 vCPU, 2GB of RAM and 20GB of disk space.
- Five CCN Routers, each with 2 vCPUs, 4GB of RAM and 40GB of disk space. Four are assigned to a given LTE cell ID, while the other is in a different layer (at EPC P-GW level). Hence, it is a 1-4 tree topology.
- One FMC Manager with 2 vCPUs, 4GB of RAM and 40GB of disk space.
- One MP Middleware with 1 vCPU, 2GB of RAM and 20GB of disk space.
- One MOBaaS Predictor with 2 vCPUs, 4GB of RAM and 40GB of disk space.
- One MaaS Zabbix with 1 vCPUs, 2GB of RAM and 20GB of disk space.

Regarding the ICN clients (end users), they were deployed manually as CCN Routers without repositories. After setting up content prefixes' routes, network connections at 100 Megabits per second (100 Mbps) to approximate LTE speeds and content request generators, they were ready to start the proposed experiments.

Finally, the OpenStack Kilo testbed consists of two nodes with distinct roles: a controller node with 8 physical CPU cores at 3.90 GHz, 16GB of RAM and 256GB of SSD disk for image storage; a compute node with 24 physical CPU cores (2 threads each) at 2.50 GHz, 192GB of RAM and 2.1TB of 15k RPM hard drives in RAID 5. Both nodes, together with the external iSCSI storage of the compute hard drives, are inter-connected at 10 Gbps with redundant links.

4.3. *Mobility prediction dataset*

In order to make location prediction, historical user traces should be provided. However, due to the fact that in a prototype implementation actual user traces are not available, we used a mobility data trace provided by Nokia for academic research. The dataset is collected during the Nokia Mobile Data Challenge (NMDC) [41], which is a large-scale research initiative aimed at generating innovations around smartphone-based research, as well as community-based evaluation of related mobile data analytics methodologies. This dataset includes rich context information running at the mobile phone for around 200 users for 2 years. It includes Global Positioning System (GPS) information, running applications, chat records, calling records, etc. However, for making location, we are only interested in the GPS location information. From this dataset, we picked data of 100 users ranging over 2-6 months. This is because the quality of the trace has significant impact on the prediction accuracy, and based on our previous findings [24], only around 100 users from the original dataset have good recordings of their location information. The rest of the users have their location data recorded discretely, which does not make it useful for the prediction inputs. For each user, we separated available data into two parts: the first part as the learning data set (L), and the rest as the testing data set (T). Learning data set is the first 70% of user's data and is used to derive the Markov Chain states and to calculate its transition probability matrix. Data set T contains 30% of the data trace, which is used to test and evaluate the accuracy of the proposed prediction algorithm. For instance if the length of a data trace is 2.5 months, which includes trace data for ten Mondays, we use the data traces of the first seven Mondays to make prediction and use the remaining three Mondays

Table 1: M-FMC Configuration Parameters

Parameter	Normal	YouTube	WebServer
Request Popularity	Zipf Distribution $\alpha = 1$	$\alpha = 2$	$\alpha = 1$
Number of Popularity Classes	10	20	20
Content Object sizes per class	Normal $\mu = 30.60$ $\sigma^2 = 15.72$ min 150Kb max 70Mb	Gamma $\alpha = 1.8$ $\beta = 500$ min 500Kb max 100Mb	Gamma $\alpha = 1.8$ $\beta = 1200$ min 50Kb max 50Mb
Content Object distribution per class	Zipf Distribution with reversed classes $\alpha = 2$	$\alpha = 1$	$\alpha = 1$
Total number of content objects	2000	2000	2000

Table 2: Configuration Criteria in MADM Algorithms

Item	Criteria description
Benefits:	Popularity contents as per Eq. 1
Costs:	Size of Files
Cache Size:	256, 512MB, 1, 2 and 4GB

for validation. The reason we divide the dataset deterministically is based on the non-stationary of the users' behavior [41].

4.4. Content and Requests

Table 1 summarizes the three content production and request scenarios that are defined according to three typical usage profiles: Normal, YouTube and WebServer.

Content popularity was defined according to the Zipf distribution [48], considering the characteristics of each scenario (e.g. number of files). The number of files present in each popularity class was determined by performing a reverse mapping of the Zipf distribution, as performed in the related work (e.g. [49] and [50]). Such reverse mapping is used because it has been demonstrated that the majority of the files are unpopular while only a few files are extremely popular. Finally, the total number of content objects and the content object size distribution per class follow the settings that are characteristic for each scenario. In the Normal scenario, the size of content objects follows a Normal Distribution, with mean 30.6MB and variance of 15.72MB. This distribution is based on a survey of current Internet statistics, such as the average size of a web page (2MB) [51] and the average size of 60 seconds YouTube videos with a resolution of 720p (40MB) [52]. The YouTube scenario follows a model already defined in a previous work [53], using a gamma distribution with $\alpha = 1.8$ and $\beta = 500$. For the WebServer scenario we defined a model based on the observed growth of the size of files available at file servers [54], following a gamma distribution with $\alpha = 1.8$ and $\beta = 1200$.

4.5. Configuration Parameters

As described in the M-FMC subsection 3.3, different parameters can be configured. MeTHODICAL, as the decision mechanism of M-FMC, enables weighting the distance of benefits criteria and the distance of costs criteria α in Algorithm 1 step 7. In this evaluation, we considered $\alpha = 0.5$ for a balanced importance. In addition, step 8 was not considered since it aims to avoid fluctuations in handover decisions. It should also be noticed that both TOPSIS and DiA do not have such configuration parameters.

As depicted in Table 2, the MADM algorithms have considered the popularity and the file size criteria metrics. In the initial steps of Algorithm 1, no weights have been considered, since the popularity is a composite criterion as it includes the relation between requests and popularity at source and destination cells. Since different cache sizes may produce different results, we consider 5 possible cache sizes: 256MB and 512MB, 1GB, 2GB and 4GB. The selection of these cache sizes is explained by the fact that we are considering the first level of caching (Content Store in CCNx), which is stored in Random Access Memory (RAM) or other similar high performance memory. While the usage of such type of caching significantly reduces access latencies, its size cannot be expanded arbitrarily to accommodate more files due to power and cost limitations [55]. Therefore, a second level of caching is typically considered (hard-drive based, as the CCNx repositories) [33] and our model can easily be extrapolated to both of these levels. However, as the selection of content stays unchanged by the type of memory in use, multiple levels of caching are out of scope in this evaluation.

4.6. Evaluation metrics

The evaluation of the mobility prediction includes the accuracy evaluation of the mobility prediction algorithms, and we will show that the predictions' accuracy heavily depends on the quality of the collected mobility traces. To support this claim, we also depict trace qualities for different users and periods of time.

The evaluation of MADM algorithms considers accuracy and efficiency. For assessing accuracy we consider (i) the number of files correctly selected for migration by the MADM mechanism, when compared with the optimal solution provided by knapsack [42] (the optimal set of files that can be moved given cache constraints); and (ii) the accumulated ratio of content requests corresponding to cache hits. 100% indicates that all content requests were for content available on cache and 0% indicates that all content requests were for content not available in the cache. Efficiency is assessed in terms of the processing time required to determine the content to be migrated. Regarding the evaluation of file selection correctness, we considered two distinct metrics. The first is the percentage of files that each algorithm correctly identifies as migration candidates, when compared to the reference results produced by knapsack (compared to the number of files). The second metric is the relative proportion of "correct" content included in the migration (e.g. a 65%

value means 35% of the migrated content was wrongly selected for migration and 65% correctly selected).

The End-to-End Users Experience validation attempts to assess the experience perceived by end users with and without M-FMC, thus depicting the benefits that can be obtained when using this model. It consists of two different metrics: 1.) average download time of a file within a certain group size (determined according to the settings in the YouTube and WebServer scenarios) and 2.) users' satisfaction. While the first metric is straightforward, the second requires further explanation. Users' satisfaction can be evaluated in multiple different ways and with numerous methods, and it is often very subjective depending on the users themselves. As we do not have real users in our evaluation or even content transfer types that can easily match common Quality of Experience (QoE) metrics, we propose that a Sigmoid function [56] is used to approximate the users' satisfaction with respect to perceived QoS [57, 58]. Therefore, the users' satisfaction can be given by:

$$U(x) = \frac{1}{1 + e^{-\alpha(x-\beta)}} \qquad (2)$$

where $U(X) \in [0, 1]$ is the satisfaction factor, x is the variable representing calculated download bandwidth in Mbps and α and β are constants that need to be defined to set the steepness and center of the curve.

For the definition of α and β, common sense from a user's perspective was the strategy. Starting with β, the center of the curve corresponds to a satisfaction factor of 0.5. We can assume that the user will have an average satisfaction if the bandwidth is more or less what it is expecting to be from its daily usage. Therefore, it makes sense that an average download bandwidth value is considered for β. From Akamai's latest report on the state of the Internet [59], a value of 5.6 Mbps is assumed to be the average download bandwidth in the Internet. Thus, $\beta = 5.6$. As for α, the steepness of the curve, we decided to use a value of $\alpha = 0.4$ because 0.9 satisfaction is achieved when download bandwidth is around twice the average and 0.99 satisfaction is achieved when download bandwidth is around triple the average.

The evaluation metrics presented so far, and summarized in Table 3, include mainly functional aspects. The non-functional aspects have also been evaluated and consider the service lifecycle performance in a cloud context with the deployment, provisioning and disposal phases. The deployment phase considers the time required to instantiate VMs (a.k.a. instances), while the provisioning phase includes the necessary time for configuring the deployed instances, for instance to set up the endpoints of MOBaaS in the FMC Manager component. The disposal phase considers the phase where all the deployed resources are deleted from the cloud infrastructure.

5. Results

This section presents the evaluation results in terms of non-functional results including the deployment of the complete M-FMC model and the functional results including the performance and accuracy results.

Table 3: Evaluation Metrics

Category	Metric	Meaning
Functional	Accuracy	% of cache hits % of files correctly selected % of content volume correctly selected
Functional	Efficiency	Processing time of algorithms Average download time
Functional	Experience	Average download time Users' Satisfaction
Non-Functional	Performance	Time of deployment, provisioning and disposal phases of M-FMC model

5.1. Non-functional results

This subsection discusses obtained results, in terms of deploying, provisioning and disposing the services included in the M-FMC architecture. In Fig. 5 we illustrate the times taken for these phases with a confidence level of 95%.

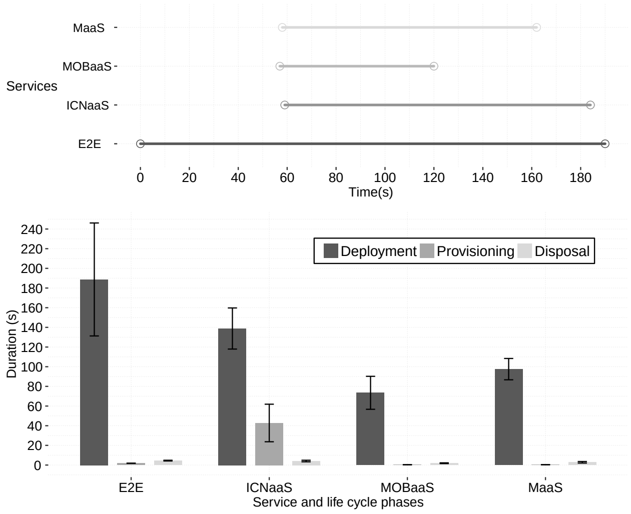

Figure 5: Non-Functional evaluation

The deployment phase, as expected, is the one that has a higher duration, since it includes the instantiation of VMs in the OpenStack testbed, according to the settings described in section 4.2.2. Is it also noticed that ICNaaS, since it has more instances to create (i.e. 5 CCN routers and FMC components) has an extended duration, in comparison to the remaining services, such as MOBaaS or MaaS. Indeed, these services only deploy a single instance. The E2E entity is used mainly to manage the service orchestration between the diverse services of the M-FMC. The deployment occurs in parallel, as illustrated in Fig. 5 for a specific run, where the deployment time is around 3 minutes. The provisioning phase is also longer in case of ICNaaS, since this service requires the configuration of the endpoints of MOBaaS, to allow MP Middleware to process the mobility predictions, and of MaaS for monitoring the performance of the several instances (e.g. number of interests received per minute). Both MOBaaS and MaaS do not have any dependency on other services. The disposal time takes a reduced time (below 2s) to delete all the resources at OpenStack and OpenShift.

5.2. *Functional results*

This section discusses obtained results, in terms of accuracy and efficiency, assessing M-FMC as an enabler for content migration in ICN and Mobile Cloud Computing environments.

5.2.1. *MOBaaS Accuracy*

In this subsection, we depict the accuracy evaluation results of the mobility prediction algorithms, which have significant impact on the usefulness of content migration and the performance of M-FMC evaluated below in Section 5.2.4. Our prediction mechanism is a DBN-based network model, which can be further presented as a simple first order MC that encodes the frequency of transitions between the cells. The number of valid states in the derived MC for each user highly depend on the quality of the data trace in each day, so the trace dataset plays a key role in the accuracy. In order to evaluate accuracy of the proposed algorithms we selected, for each user, 50 random states (*representing the random times and IDs of the cells that user has been there*) out of the MC states derived for each particular day of a week from the dataset of *L*. Afterwards, we performed the calculation of predictions to find the future possible cells for those users in the next 20 minutes. We repeated the predictions for the same random states in the data set of *T*. Afterwards, the Mean Absolute Error (*MAE*) for the corresponding test points, chosen from the learning and testing datasets, is computed in order to obtain accuracy of the predictions for each user in a particular day of a week.

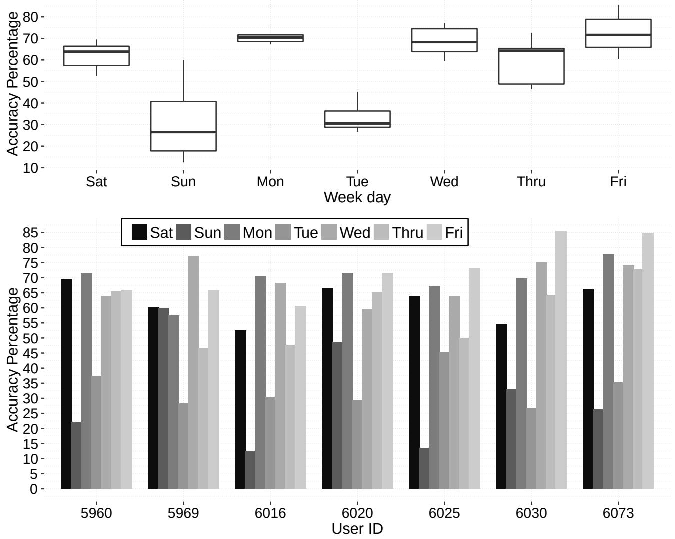

Figure 6: Accuracy of proposed algorithm for some users per day.

Fig. 7 displays the overall accuracy of mobility prediction for 100 users. For each user it represents the average accuracy calculated for each single day of a week.

Accuracy of predictions effectively pertains to the quality of mobility data traces used to derive the transition probability matrix. Fig. 8, as an example, demonstrates states of two users' in the mobility data trace, leading to low (*for user 6026*) and high (*for user 5960*) prediction accuracies, c.f. Fig. 7.

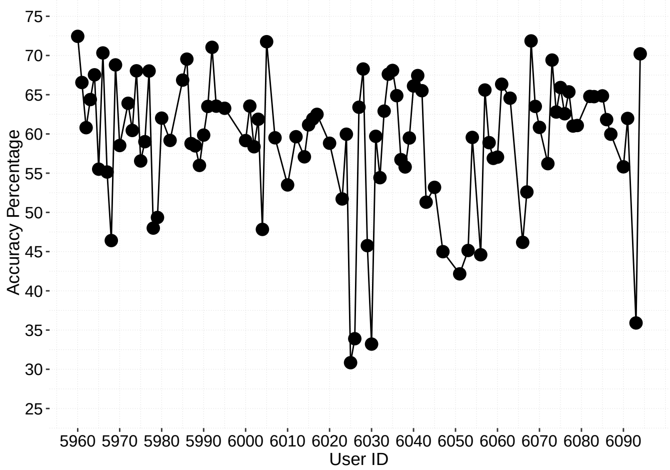

Figure 7: The overall accuracy of mobility prediction for 100 users.

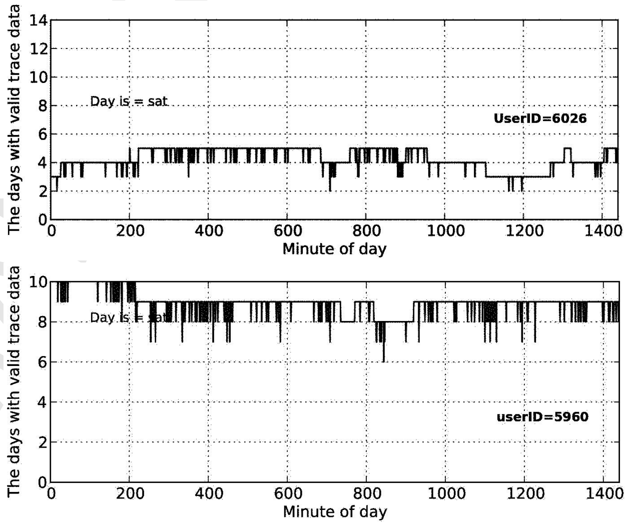

Figure 8: Quality of data trace for two different users.

5.2.2. *Content Migration Accuracy*

In this subsection we present and describe the accuracy results of the content migration algorithms. Table 4 presents the results obtained with the Knapsack algorithm for each scenario and for each cache size considered, depicting the optimal solution determined by Knapsack in terms of number of content objects and ratios of content requests corresponding to cache hits. These results highlight the impact of the popularity distribution in the different scenarios. For instance, 47 files in the YouTube scenario account for ≈71% of the requests received, while for an equivalent value in the WebServer scenario 366 files are required. Recall that the main difference between these scenarios relies on the size of the content objects per class.

It should be noticed that the Knapsack algorithm takes the cache size as a parameter, and determines a different set of files for each cache size. The MADM algorithms do not take the cache size as a parameter, and being deterministic, the solution for the different cache sizes is always the same. After obtaining the ordered set of files from the MADM algorithms, the files to have in cache are selected until the cache size limit is reached. By taking into consideration the results obtained, it is possible to see that the results from the MeTHODICAL algorithm are very close to the specific results of the Knapsack algorithm for each cache size.

Table 4: Knapsack Results per Scenario

Cache Size	Normal		YouTube		WebServer	
	Content obj.	Cache Hits (%)	Content obj.	Cache Hits (%)	Content obj.	Cache Hits (%)
256MB	15	29.71	47	70.69	233	63.06
512MB	29	43.10	83	82.07	366	75.74
1GB	47	56.31	160	90.50	648	87.75
2GB	78	69.67	300	95.28	1202	96.32
4GB	156	80.60	558	98.12	1980	99.95

For the results that are not deterministic, the confidence interval is presented over the average and shows what results can be expected from further repetitions. It was determined with a confidence level of 95% and allow us to not only show the consistency of the good results while using MeTHODICAL, but also that the performance of both DiA and TOPSIS is poor across the Normal and YouTube scenarios, and highly variable in the WebServer scenario.

Fig. 9 shows the ratio of files correctly selected by the different MADM algorithms, compared to the optimal solution determined by the Knapsack algorithm for each scenario. MeTHODICAL, for instance, selected around 90% of the files identified as the optimal solution for the Normal scenario (i.e. determined by knapsack), while DiA and TOPSIS only selected around 5% and 10%, respectively.

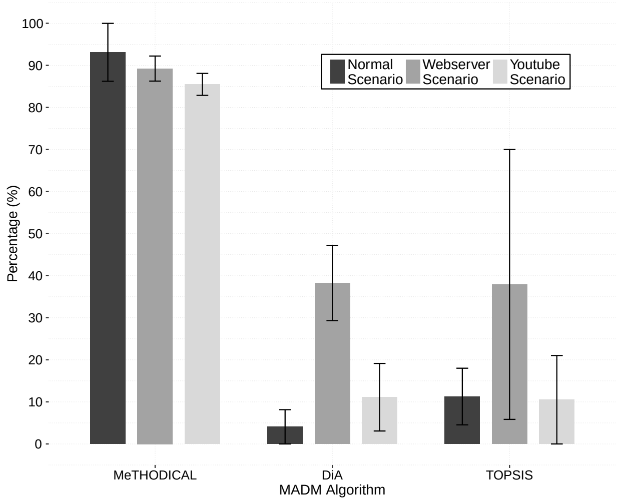

Figure 9: Number of Files Correctly Selected for Migration

Fig. 10 illustrates the relative volume of content correctly selected for migration, compared to the total volume of content actually migrated (including files which were wrongly selected for migration). The higher these values are, the more accurate the results provided by the MADM algorithm will be.

Considering the depicted results shown in Fig. 9 and Fig. 10, it is clear that MeTHODICAL is the best performing algorithm independently of the cache size. On the other hand, the performance of DiA and TOPSIS is not consistent and is impacted with the cache size of routers, as they present a high standard deviation. The low performance of DiA and TOPSIS is due to the fact that both of these techniques do not correlate values to determine optimal solutions. MeTHODICAL supports correlation of values through the distance function, by correlating the values through the mean and variance functions, as depicted in Algorithm 1 in steps 5 and 6.

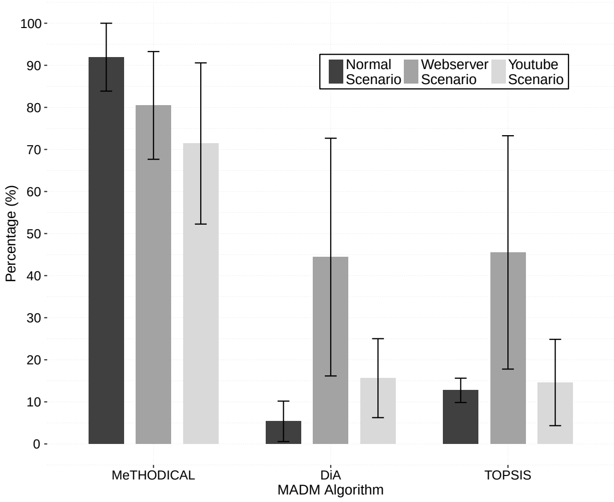

Figure 10: Volume of Content Correctly Selected for Migration

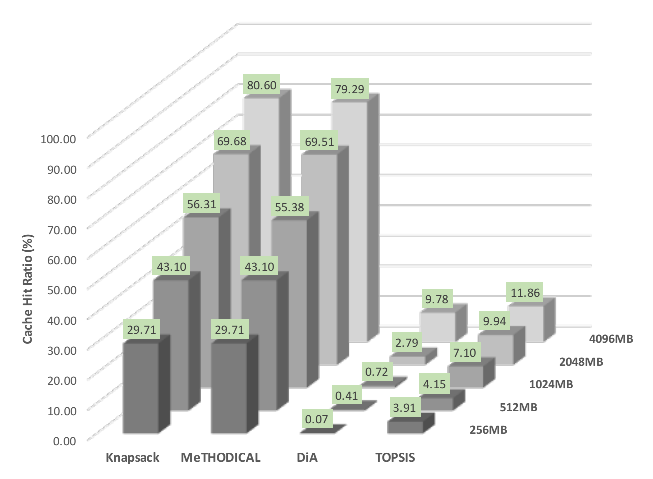

Figure 11: Percentage of cache-hits in the Normal Scenario

Next, we look at the cache hits in the form of content requests percentage that is served from the cache of routers. Cache hits are considered for the various algorithms, scenarios and cache sizes. In the Normal scenario (c.f. Table 1), MeTHODICAL produces results almost as good as knapsack, as it can be observed in Fig. 11. As the cache size increases, the percentage of cache-hits also increases, since the cache can accommodate more files. Both MeTHODICAL and Knapsack achieve cache hit ratios of around 80% with 4GB caches. On the other hand, both DiA and TOPSIS perform unsatisfactorily, with ratios below 10%. In the Normal scenario the number of files to migrate is lower and files are larger (c.f. Tables 1 and 4).

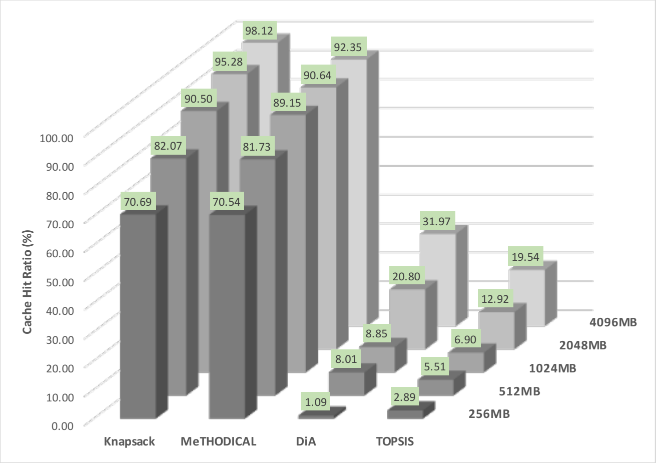

Figure 12: Percentage of cache-hits in the YouTube Scenario

Results observed for the YouTube scenario follow the same trend (c.f. Fig. 12), with Knapsack and MeTHODICAL achieving top cache hit ratios around 90%. The increased performance is explained by the characteristics of the files in the YouTube scenario, which are smaller than in the Normal scenario. It should be noted that DiA and TOPSIS also improve their performance, mainly due to the increased number of cached files.

In the WebServer scenario many more files can be cached with a cache size of 4 GB, as depicted in Fig. 13. This fact explains the cache hit ratios around 99% for Knapsack and MeTHODICAL and 96% for DiA and TOPSIS. Indeed, with 2 GB cache size the cache hit ratio for DiA and TOPSIS also increases to values of around 50%. The size of files in this scenario is the lowest leading to the case where the cache can accommodate almost all the files.

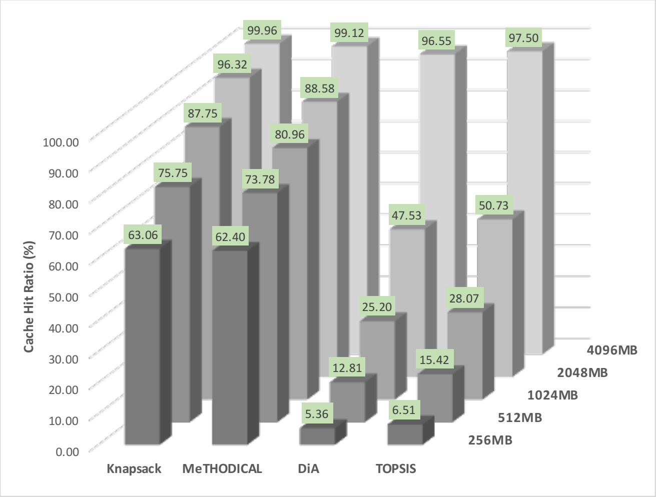

Figure 13: Percentage of cache-hits in the WebServer Scenario

Experiments for assessing accuracy suggest the selection of Knapsack and MeTHODICAL algorithms to solve the NP-complete problem of content migration in the M-FMC model, as these algorithms consistently produce the best results for all scenarios, also being less affected by cache size, file size or number of files. However, as discussed next, it is still necessary to determine the computational costs of both options to assess their practical viability.

Table 5: Processing Time (s) of M-FMC Decision Algorithms in the Normal Scenario

Algorithm / Cache Size	256MB	512MB	1GB	2GB	4GB
Knapsack	612.8	1398.9	2845.6	5705.5	13094.2
MeTHOD.	0.003	0.003	0.003	0.003	0.003
DiA	0.102	0.102	0.102	0.102	0.102
TOPSIS	0.104	0.104	0.104	0.104	0.104

Table 6: Processing Time (s) of M-FMC Decision Algorithms in the YouTube Scenario

Algorithm / Cache Size	256MB	512MB	1GB	2GB	4GB
Knapsack	518.4	1054.6	2124.9	4262.1	8547.5
MeTHOD.	0.023	0.023	0.023	0.023	0.023
DiA	0.075	0.075	0.075	0.075	0.075
TOPSIS	0.075	0.075	0.075	0.075	0.075

5.2.3. Efficiency

Table 5 presents the processing time for the different algorithms in the Normal scenario. The processing time of the Knapsack algorithm increases with the size of the cache, which is not the case of MeTHODICAL (MeTH) and related algorithms. The same behavior can be observed in the YouTube and WebServer scenarios regarding the processing time, as depicted in Table 6 and Table 7, respectively.

The reason for the fact that MADM algorithms do not present different processing times for each cache size is that they determine the same optimal solution for each cache size. The difference only relies on the number of files that can be selected for migration. Hence, these algorithms are deterministic. As long as the input is the same, they always output the same result. In this case, they output an ordered list of content objects to be moved, being the first element the most important to be moved, i.e. the one that should be migrated first. Later on, the list is iterated to add the files that can be moved until the cache size limit is reached. As stated before, the M-FMC model, which operates in the same fashion as a real-time application, requires high-efficiency in terms of performance (not using unnecessary resources) and processing time. The results depicted so far demonstrate that MeTHODICAL is the most accurate and efficient MADM algorithm, fulfilling the performance requirements of the M-FMC model.

It is also demonstrated that in-network caching with cache sizes as small as 512MB can provide improvements for users and operators, since it becomes possible to achieve cache-hit ratios around 50%, which meets the requirements of high performance caching hardware implementing first-level of caching mechanisms.

Table 7: Processing Time (s) of M-FMC Decision Algorithms in the WebServer Scenario

Algorithm / Cache Size	256MB	512MB	1GB	2GB	4GB
Knapsack	645.2	1293.8	2590.4	6371.2	12004.5
MeTHOD.	0.031	0.031	0.031	0.031	0.031
DiA	0.096	0.096	0.096	0.096	0.096
TOPSIS	0.094	0.094	0.094	0.094	0.094

5.2.4. User perspective

The End-to-End user experience evaluation includes download times and users' satisfaction in the WebServer and YouTube scenarios, as depicted in Fig. 14 and Fig. 15. As presented in Table 1 the number of files and their respective size depends on the distribution associated with the specific scenario. To facilitate the analysis of achieved results, files are grouped according to their size, where 1MB includes files with a size below 1.5MB, while 2MB includes files with size in the range [1.5, 2.5[, and so on. In addition, download time is measured for each file, with and without M-FMC. This value is also used to calculate bandwidth in Mbps and thus the satisfaction factor. The aim of such evaluation is to demonstrate the performance benefits of the M-FMC model and associated services perceived by end users.

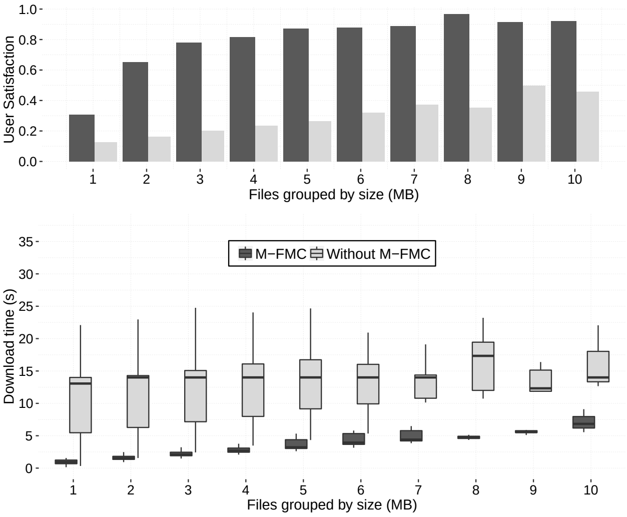

Figure 14: Download Time in seconds for the WebServer Scenario

Independently of the size of files, download times are higher when M-FMC is not employed. Indeed, M-FMC is able to reduce the download times by a factor of 2-5 (as perceived by end users). It is also noticed that M-FMC is able to provide consistent results for files with approximately the same size. The download time increases in a linear fashion as the size of grouped files increases. This is an expected result, as bigger files required more time to download. The variation in the case without M-FMC leads to dissatisfaction of users, who may experience inconsistent download times. For instance, for a file with a size below 1MB, it can take a minimum of 6 and a maximum of 13 seconds. This is further validated by the users' satisfaction analysis. Even with small file sizes that have greater overhead in ICN transfers, satisfaction with M-FMC is much higher and in fact a size of 5MB is already enough to obtain a satisfaction factor close to 0.9. It also shows that users' satisfaction is much more consistent with M-FMC, and hence that users will have a greater experience overall.

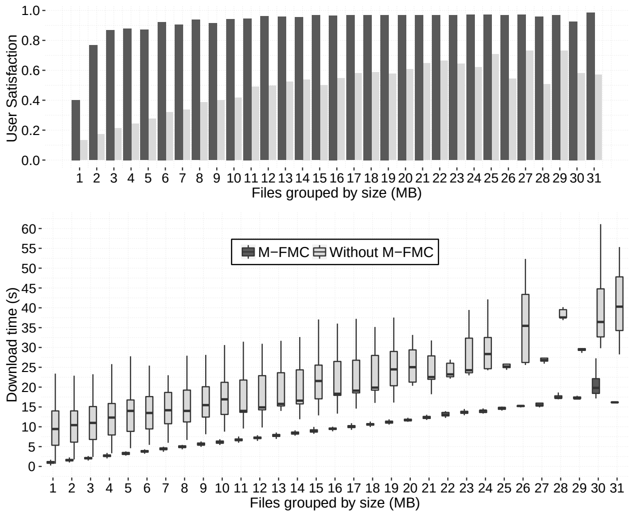

Figure 15: Download Time in seconds for the YouTube Scenario

The YouTube scenario is characterized by files with bigger sizes, i.e. sizes above 30MB. In the plot of Fig. 15 they are grouped as 31MB, which considers file sizes in the range $[30.5, +\infty[$. In this scenario the difference between the file size and the performance achieved with FMC is more evident, in comparison to the WebServer scenario. The download time increases linearly, for the cases with and without FMC, but has high variations when FMC is not employed. FMC is able to support stable download times irrespective of the file sizes. By the contrary, higher variations and inconsistencies are observed with bigger file sizes in the case where FMC is not employed. As in the WebServer case, this behavior leads to the dissatisfaction of users as the downloads vary a lot in time, without considering the impact that content requests may have in the core network. Moreover, we can observe again that users' satisfaction with M-FMC is very consistent and much higher, and that with a small size of 3MB the satisfaction is already achieving values close to 0.9. In fact, even when file sizes are much higher (30MB) the satisfaction with M-FMC is about 50% higher than without it.

Considering the scenarios for which the M-FMC model is intended, the perceived experience for users that follow popularity trends is improved, decreasing latency when obtaining popular files while not decreasing the perceived experience of other users. In addition, the M-FMC model also enhances core networks by introducing bandwidth savings and allowing a better usage of resources, i.e. cache of edge routers.

6. Conclusions

As users move to different locations, they still want to access content on which they are interested with low latency and without delays or breaks, especially if dealing with multimedia content. From the network perspective, this can only be granted if caches exist at the edge of mobile networks and the content kept in those caches (with limited resources) is the right content, i.e. popular content that local users will consume.

A number of related proposals already exist, but cannot be applied to content (only to services) or have other limitations such as assuming a very specific scenario or scope.Therefore, a broader approach has been proposed – M-FMC –, able to deal with content migration, handling multiple criteria decisions and considering multiple factors that will trigger content migration.

In this paper, M-FMC has been introduced and validated as an enabler for content migration in MEC networks. The performed assessments took into account realistic requirements for next-generation mobile networks and revealed up to fivefold improvements in terms of reducing content download time, increasing cache hit ratios and providing accurate results.

The M-FMC model was evaluated in multiple scenarios in terms of the percentage of files correctly identified for migration and cache-hits enabled by those migrations. Results show that the selected algorithm is efficient while providing an accuracy always above 80% when compared to the optimal solutions determined by Knapsack. This results in a smooth operation with real-time applications with the M-FMC model being able to deliver content with lower latency to end-users while, simultaneously, allowing savings of network bandwidth and enabling FI concepts such as ICN.

Acknowledgments

The research leading to these results has been partially funded by the European Union's FP7 Programme Mobile Cloud Networking project (FP7-ICT-318109).

The authors would also like to acknowledge the CTIT Research Center of UTwente due to its involvement in the Mobile Cloud Networking project's team that carried out the development of MOBaaS.

References

[1] A. Ahmed, E. Ahmed, A survey on mobile edge computing, in: IEEE International Conference on Intelligent Systems and Control, 2016.

[2] P. Demestichas, A. Georgakopoulos, D. Karvounas, K. Tsagkaris, V. Stavroulaki, J. Lu, C. Xiong, J. Yao, 5g on the horizon: Key challenges for the radio-access network, Vehicular Technology Magazine, IEEE 8 (3) (2013) 47–53. doi:10.1109/MVT.2013.2269187.

[3] E. Ahmed, A. Gani, M. K. Khan, R. Buyya, S. U. Khan, Seamless application execution in mobile cloud computing: Motivation, taxonomy, and open challenges, Journal of Network and Computer Applications 52 (2015) 154 – 172. doi:http://dx.doi.org/10.1016/j.jnca.2015.03.001.
URL http://www.sciencedirect.com/science/article/pii/S1084804515000545

[4] E. Ahmed, A. Gani, M. Sookhak, S. H. A. Hamid, F. Xia, Application optimization in mobile cloud computing: Motivation, taxonomies, and open challenges, Journal of Network and Computer Applications 52 (2015) 52 – 68. doi:http://dx.doi.org/10.1016/j.jnca.2015.02.003.
URL http://www.sciencedirect.com/science/article/pii/S1084804515000417

[5] Suggestions on Potential Solutions to C-RAN by NGMN Alliance, Tech. rep., The Next Generation Mobile Networks (NGMN) Alliance (Jan. 2013).
URL http://www.ngmn.org/uploads/media/NGMN_CRAN_Suggestions_on_Potential_Solutions_to_CRAN.pdf

[6] B. Haberland, F. Derakhshan, H. Grob-Lipski, R. Klotsche, W. Rehm, P. Schefczik, M. Soellner, Radio Base Stations in the Cloud, Bell Labs Technical Journal 18 (1) (2013) 129–152. doi:10.1002/bltj.21596.
URL http://dx.doi.org/10.1002/bltj.21596

[7] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, R. L. Braynard, Networking named content, in: Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies, CoNEXT '09, ACM, New York, NY, USA, 2009, pp. 1–12. doi:10.1145/1658939.1658941.
URL http://doi.acm.org/10.1145/1658939.1658941

[8] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2014¢?2019, http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.pdf (Feb 2015).

[9] A. Gomes, T. Braun, Load balancing in lte mobile networks with information-centric networking, in: 2015 IEEE International Conference on Communication Workshop (ICCW), 2015, pp. 1845–1851. doi:10.1109/ICCW.2015.7247449.

[10] H. Li, G. Ascheid, Mobility prediction based on graphical model learning, in: Vehicular Technology Conference (VTC Fall), 2012 IEEE, 2012, pp. 1–5. doi:10.1109/VTCFall.2012.6398888.

[11] S. Rajagopal, N. Srinivasan, R. Narayan, X. Petit, Gps based predictive resource allocation in cellular networks, in: Networks, 2002. ICON 2002. 10th IEEE International Conference on, 2002, pp. 229–234. doi:10.1109/ICON.2002.1033316.

[12] Y. Chon, H. Shin, E. Talipov, H. Cha, Evaluating mobility models for temporal prediction with high-granularity mobility data, in: Pervasive Computing and Communications (PerCom), 2012 IEEE International Conference on, 2012, pp. 206–212. doi:10.1109/PerCom.2012.6199868.

[13] B. Sousa, K. Pentikousis, M. Curado, Methodical: Towards the next generation of multihomed applications, Computer Networks 65 (2014) 21–40.

[14] D.-h. Kim, J.-h. Kim, Y.-s. Kim, H.-s. Yoon, I. Yeom, Mobility Support in Content Centric Networks, in: Proceedings of the Second Edition of the ICN Workshop on Information-centric Networking, ICN '12, ACM, New York, NY, USA, 2012, pp. 13–18. doi:10.1145/2342488.2342492.
URL http://doi.acm.org/10.1145/2342488.2342492

[15] D. Smetters, V. Jacobson, Securing network content, Tech. rep., PARC (Oct. 2009).
URL https://www.parc.com/content/attachments/TR-2009-01.pdf

[16] A. Gomes, T. Braun, Feasibility of Information-Centric Networking Integration into LTE Mobile Networks, in: Proceedings of the 30th Annual ACM Symposium on Applied Computing, SAC '15, ACM, 2015, pp. 628–634. doi:10.1145/2695664.2695790.

[17] S. K. Fayazbakhsh, Y. Lin, A. Tootoonchian, A. Ghodsi, T. Koponen, B. Maggs, K. Ng, V. Sekar, S. Shenker, Less pain, most of the gain: Incrementally deployable icn, in: Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM, SIGCOMM '13, ACM, New York, NY, USA, 2013, pp. 147–158. doi:10.1145/2486001.2486023.
URL http://doi.acm.org/10.1145/2486001.2486023

[18] F. Bai, A. Helmy, A Survey of Mobility Modeling and Analysis in Wireless Adhoc Networks, in: Wireless Ad Hoc and Sensor Networks, Springer Berlin Heidelberg, 2004.

[19] R. R. Roy, Autoregressive individual mobility, in: Handbook of Mobile Ad Hoc Networks for Mobility Models, Springer, 2011, pp. 775–789.

[20] B. Liang, Z. J. Haas, Predictive distance-based mobility management for pcs networks, in: INFOCOM'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, Vol. 3, IEEE, 1999, pp. 1377–1384.

[21] X. Hong, M. Gerla, G. Pei, C.-C. Chiang, A group mobility model for ad hoc wireless networks, in: Proceedings of the 2nd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems, ACM, 1999, pp. 53–60.

[22] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, M. Degermark, Scenario-based performance analysis of routing protocols for mobile ad-hoc networks, in: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking, ACM, 1999, pp. 195–206.

[23] J. Tian, J. Hahner, C. Becker, I. Stepanov, K. Rothermel, Graph-based mobility model for mobile ad hoc network simulation, in: Simulation Symposium, 2002. Proceedings. 35th Annual, IEEE, 2002, pp. 337–344.

[24] M. Karimzadeh, Z. Zhao, L. Hendriks, R. de O Schmidt, S. la Fleur, H. van den Berg, A. Pras, T. Braun, M. Corici, Mobility and bandwidth prediction as a service in virtualized lte systems, in: Cloud Networking (CloudNet), 2015 IEEE 4th International Conference on, 2015, pp. 132–138. doi:10.1109/CloudNet.2015.7335295.

[25] A.-F. Antonescu, A. Gomes, P. Robinson, T. Braun, Sla-driven predictive orchestration for distributed cloud-based mobile services, in: Communications Workshops (ICC), 2013 IEEE International Conference on, 2013, pp. 738–743. doi:10.1109/ICCW.2013.6649331.

[26] T. Taleb, A. Ksentini, Follow me cloud: interworking federated clouds and distributed mobile networks, Network, IEEE 27 (5) (2013) 12–19. doi:10.1109/MNET.2013.6616110.

[27] T. Taleb, A. Ksentini, An analytical model for follow me cloud, in: Global Communications Conference (GLOBECOM), 2013 IEEE, 2013, pp. 1291–1296. doi:10.1109/GLOCOM.2013.6831252.

[28] A. Ksentini, T. Taleb, M. Chen, A markov decision process-based service migration procedure for follow me cloud, in: Communications (ICC), 2014 IEEE International Conference on, 2014, pp. 1350–1354. doi:10.1109/ICC.2014.6883509.

[29] H. Liu, Y. Sun, M. S. Kim, Provider-level content migration strategies in p2p-based media distribution networks, in: Consumer Communications and Networking Conference (CCNC), 2011 IEEE, 2011, pp. 337–341. doi:10.1109/CCNC.2011.5766485.

[30] F. Zhang, C. Xu, Y. Zhang, K. K. Ramakrishnan, S. Mukherjee, R. Yates, T. Nguyen, Edgebuffer: Caching and prefetching content at the edge in the mobilityfirst future internet architecture, in: World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2015 IEEE 16th International Symposium on a, 2015, pp. 1–9. doi:10.1109/WoWMoM.2015.7158137.

[31] M. P. Wittie, V. Pejovic, L. Deek, K. C. Almeroth, B. Y. Zhao, Exploiting locality of interest in online social networks, in: Proceedings of the 6th International COnference, Co-NEXT '10, ACM, New York, NY, USA, 2010, pp. 25:1–25:12. doi:10.1145/1921168.1921201.
URL `http://doi.acm.org/10.1145/1921168.1921201`

[32] M. D. Choudhury, H. Sundaram, A. John, D. D. Seligmann, A. Kelliher, "birds of a feather": Does user homophily impact information diffusion in social media?, CoRR abs/1006.1702.

[33] C. Anastasiades, A. S. Gomes, R. Gadow, T. I. Braun, Persistent caching in Information-Centric networks, in: 40th Annual IEEE Conference on Local Computer Networks (LCN 2015), Clearwater Beach, USA, 2015, pp. 64–72.

[34] D. Stynes, K. N. Brown, C. J. Sreenan, Using opportunistic caching to improve the efficiency of handover in lte with a pon access network backhaul, in: Local Metropolitan Area Networks (LANMAN), 2014 IEEE 20th International Workshop on, 2014, pp. 1–6. doi:10.1109/LANMAN.2014.7028621.

[35] X. Vasilakos, V. A. Siris, G. C. Polyzos, M. Pomonis, Proactive selective neighbor caching for enhancing mobility support in information-centric networks, in: Proceedings of the Second Edition of the ICN Workshop on Information-centric Networking, ICN '12, ACM, New York, NY, USA, 2012, pp. 61–66. doi:10.1145/2342488.2342502.
URL `http://doi.acm.org/10.1145/2342488.2342502`

[36] EC FP7 Mobile Cloud Networking project, `https://www.mobile-cloud-networking.eu/` (May 2015).

[37] C.-L. Hwang, Y.-J. Lai, T.-Y. Liu, A new approach for multiple objective decision making, Computers & Operations Research 20 (8) (1993) 889–899.

[38] P. N. Tran, N. Boukhatem, The distance to the ideal alternative (dia) algorithm for interface selection in heterogeneous wireless networs, in: Proc. MobiWac '08, 2008, pp. 61–68.

[39] M. Lahby, L. Cherkaoui, A. Adib, Article: New Optimized Network Selection Decision in Heterogeneous Wireless Networks, International Journal of Computer Applications 54 (16) (2012) 1–7, published by Foundation of Computer Science.

[40] Project CCNx (Sep. 2015).
URL `http://www.ccnx.org/`

[41] J. K. Laurila, D. Gatica-Perez, I. Aad, B. J., O. Bornet, T.-M.-T. Do, O. Dousse, J. Eberle, M. Miettinen, The Mobile Data Challenge: Big Data for Mobile Computing Research, in: Pervasive Computing, 2012.

[42] M. R. Garey, D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman, 1979.

[43] Ubelix - university of bern linux cluster, `http://cmslive3.unibe.ch/unibe/verwaltungsdirektion/informatikdienste/content/services/ubelix/overview/index_ger.html` (Sep. 2015).

[44] R Foundation for Statistical Computing, Vienna, Austria, R: A Language and Environment for Statistical Computing (2014).
URL `http://www.R-project.org/`

[45] OpenShift Origin (Sep. 2015).
URL `http://www.openshift.org`

[46] OpenStack Kilo (Sep. 2015).
URL `https://www.openstack.org/software/kilo`

[47] Kernel-based virtual machine, `http://www.linux-kvm.org` (Sep. 2015).

[48] D. Rossi, G. Rossini, Caching performance of content centric networks under multi-path routing (and more) (2011).

[49] D. Frommer, K. Angelova, Chart of the day: Half of youtube videos get fewer than 500 views, `http://www.businessinsider.com/chart-of-the-day-youtube-videos-by-views-2009-5?IR=T` (2009).

[50] T. Yu, J. Chen, W. Liu, Measurements and analysis of an unconstrained user generated content system, IEEE Internation Conference in Communications (ICC) (2011) 1–5.

[51] HTTP Archive, `http://httparchive.org/interesting.php`.

[52] Recommended upload encoding settings, `https://support.google.com/youtube/answer/1722171?hl=en`.

[53] A. Abhari, M. Soraya, Workload generation for youtube, Multimedia Tools and Applications 46 (1) (2010) 91–118.

[54] A. Williams, M. Arlitt, C. Williamson, K. Barker, Web workload characterization: Ten years later, Web Content Delivery 2 (1) (2005) 3–21.

[55] D. Perino, M. Varvello, A reality check for content centric networking, in: Proceedings of the ACM SIGCOMM Workshop on Information-centric Networking, ICN '11, ACM, New York, NY, USA, 2011, pp. 44–49. doi:10.1145/2018584.2018596.
URL `http://doi.acm.org/10.1145/2018584.2018596`

[56] D. H. v. Seggern, CRC Standard Curves and Surfaces with Mathematica, Second Edition (Chapman & Hall/Crc Applied Mathematics and Nonlinear Science), Chapman & Hall/CRC, 2006.

[57] G. D. Stamoulis, D. Kalopsikakis, A. Kyrikoglou, Efficient agent-based negotiation for telecommunications services, in: Global Telecommunications Conference, 1999. GLOBECOM '99, Vol. 3, 1999, pp. 1989–1996 vol.3. doi:10.1109/GLOCOM.1999.832520.

[58] S. Pal, S. K. Das, M. Chatterjee, User-satisfaction based differentiated services for wireless data networks, in: IEEE International Conference on Communications, 2005. ICC 2005. 2005, Vol. 2, 2005, pp. 1174–1178 Vol. 2. doi:10.1109/ICC.2005.1494532.

[59] D. Belson, Akamai State of the Internet Report, Q4 2015, Tech. Rep. 4, Akamai Technologies (Mar. 2016).
URL `https://content.akamai.com/PG5641-Q4-2015-SOTI-Connectivity-Report.html`

Andre S. Gomes is a PhD student and researcher at both the University of Bern, Switzerland and the University of Coimbra, Portugal. He has received his BSc in 2010 and his MSc in 2012, both from the University of Coimbra, Portugal. His past activities include several national and international research projects in the area of Wireless Sensor Networks, with special highlight to FP7 GINSENG project. As main research topics, he is currently focused on Information-Centric Networking and Mobile Cloud Computing, on which he is developing active research in the context of the FP7 Mobile Cloud Networking project.

Bruno Sousa, Project Manager at OneSource. He owns a PhD in Information Science and Technology. His research interests include Multi-Homing, Wireless Networks, Mobility, Resilience and Multiple Attribute Decision Mechanisms. He has several publications in journals, book chapters and conferences in these areas. He has participated in European R&D projects such as FP6 WEIRD, FP7 CityFlow, FP7 MCN and FP7 SALUS.

David Palma is a Post-Doctoral fellow at the Department of Telematics from NTNU and has worked in the past as a Researcher and Project Manager at OneSource, as well as an invited Assistant Professor at the University of Coimbra. He holds a PhD in Information Science and Technology received from the University of Coimbra. His current research interests are on Routing, IoT, Cloud-Computing and Software-Defined Networks, subjects on which he as authored and co-authored multiple papers in refereed conferences and journals. He has participated in several TPCs, national and international research projects, including European Projects (FP6/FP7/H2020) and in the preparation of successful research proposals.

Zhongliang Zhao is currently a post-doctoral senior researcher at University of Bern. In June 2014, he got the PhD degree from the same university, advised by Prof. Torsten Braun from the Communication and Distributed Systems group. He got the BS degree from Southeast University, China, and then the Master degree from Politecnico di Torino, Italy. His current research interests include mobile ad-hoc and sensor networks, mobile cloud computing, SDN/NFV, and urban computing.

Vitor Fonseca is a Master's student at University of Coimbra, Portugal. He received his Bachelor degree in Informatics Engineering from University of Coimbra in 2014. Since 2012 he has been with the Communications and Telematics group of Centre for Informatics and Systems of the University of Coimbra (CISUC), researching topics related with wireless networks performance assessment, namely quality of service and quality of experience.

Edmundo Monteiro is Full Professor at the Department of Informatics Engineering (DEI) of the University of Coimbra (UC), Portugal. He is also a Senior Member of the research Centre for Informatics and Systems of the University of Coimbra (CISUC). He graduated in Electrical Engineering (Informatics Specialty) from the University of Coimbra in 1984, and received his PhD in Informatics Engineering (Computer Communications) and the Habilitation in Informatics Engineering from the same university in 1996 and 2007 respectively. He has near 30 years of research and industry experience in the field of Computer Communications, Wireless Technologies, Quality of Service and Experience, Network and Service Management, and Computer Security. He participated in many Portuguese and European research projects and initiatives. His publication list includes 6 books (authored and edited) and over 200 publications in journals, book chapters, and international refereed conferences. He is also co-

author of 9 international patents. He is member of the Editorial Board of Elsevier Computer Communication and Springer Wireless Networks journals, and involved in the organization of many national and international conferences and workshops.
Edmundo Monteiro is Member of Ordem dos Engenheiros (the Portuguese Engineering Association), and Senior Member of IEEE Communication Society, and Senior Member of ACM Special Interest Group on Communications. He is also the Portuguese representative in IFIP TC6 (Communication Systems).

Torsten Braun got his diploma and Ph.D. degrees from the University of Karlsruhe, Germany, in 1990 and 1993, respectively. From 1994 to 1995 he was a guest scientist with INRIA Sophia Antipolis. From 1995 to 1997 he worked as a project leader and senior consultant at the IBM European Networking Center, Heidelberg, Germany. Since 1998 he has been a full professor of computer science at the Institute of Computer Science and Applied Mathematics (IAM) of the University of Bern (Switzerland), heading the Computer Networks and Distributed Systems research group. He has been a board member of SWITCH (Swiss Education and Research network) since 2000. He has been director of IAM since 2007. His research interests include Quality of Experience, wireless networks, mobility and multimedia.

Paulo Simoes is a Tenured Assistant Professor at the Department of Informatics Engineering of the University of Coimbra, Portugal, from where he obtained his doctoral degree in 2002. He is also a senior researcher at the Centre for Informatics and Systems of the University of Coimbra. He has been involved in several European research projects, with technical and managerial duties, and he regularly leads industry-funded technology transfer projects for companies such as telecommunications operators and energy utilities. He was also founding partner of two technological spin-off companies. His research interests include Future Internet, Network and Infrastructure Management, Security, Critical Infrastructure Protection and Virtualization of Networking and Computing Resources. He has over 120 publications in refereed journals and conferences, and he regularly serves on program committees of international conferences of these areas. He is also member of the IEEE Communications Society.

Luis Cordeiro, CTO of OneSource, received an MSc in Communications and Telematics and has been actively involved in European research projects such as FP6 EUQOS, FP6 WEIRD, FP7 LiveCity, FP7 CityFlow, FP7 SALUS and FP7 Mobile Cloud Networks. He has several publications in journals, book chapters, conferences and Internet Drafts in the areas of Signaling, end-to-end QoS and Network Management.