

Enabling a Mobility Prediction-aware Follow-Me Cloud Model

Bruno Sousa^{1,5}, Zhongliang Zhao², Morteza Karimzadeh³, David Palma^{1,5,6}, Vitor Fonseca⁵, Paulo Simoes¹, Torsten Braun², Hans van den Berg^{3,4}, Aiko Pras³, Luis Cordeiro⁵

¹University of Coimbra, Portugal

²University of Bern, Switzerland

³University of Twente, Netherlands

⁴Organization for Applied Scientific Research (TNO), Netherlands

⁵OneSource, Consultoria Informatica Lda. Coimbra, Portugal

⁶NTNU, Norwegian University of Science and Technology, Norway

Abstract—The location of data centres is crucial when mobile network operators are moving towards cloudified mobile networks to optimize resource utilization and to improve performance of services. Quality of Experience (QoE) can be enhanced in terms of content access latency, by placing user content at locations where they will be present in the future.

The Follow-Me Cloud (FMC) concept aims at optimising operations of moving Mobile Network Operators Services towards cloudified environments, where Information Centric Networking (ICN) and the appropriate content migration policies are of paramount importance. However, several factors need to be considered, including user movements and mobility prediction (MP), content popularity, and migration. This paper addresses all these aspects by implementing a fully integrated multi-criteria FMC and mobility prediction mechanisms (MP-FMC) on a cloud infrastructure. Experimental evaluation shows that MP-FMC can be orchestrated on-demand within a reasonable time frame, and it could deliver $\approx 33\%$ improvement of content retrieval time.

Keywords: Follow-Me Cloud, Content Migration, Mobility Prediction as a Service, Information Centric Networking, Mobile Cloud Networking.

I. INTRODUCTION

The huge appreciation received by cloud computing technologies in recent years encouraged mobile network operators to plan the adoption of virtualization in their future networks. Applying the cloud computing model to realize a shared distributed Long-Term Evolution (LTE) mobile network has a lot of advantages [1], such as: (i) optimize the utilization of networking resources, (ii) minimize communication latency, (iii) mitigate bottlenecks, and (iv) enable emerging multiple mobile virtual networks, using a common physical infrastructure. To leverage on such benefits, cloud computing data centres should be located closer to 3GPP Evolved Universal Terrestrial Radio Access (E-UTRAN) and the Evolved Packet Core (EPC), leading to cloud-based E-UTRAN (denoted as RANaaS) and EPC (denoted as EPCaaS). The most important objectives of this approach are the support of on-demand deployment, provision, disposal of virtualized LTE system components, and the elasticity support to enable dynamic resource allocation, according to the load faced by the system.

With today's cloud infrastructure technology such as Open-Stack [2], time consumption of virtual machine deployment and live migration procedures is critical for an acceptable delay in the LTE network [3]. Therefore, an appropriate prediction mechanism is needed to estimate the system state changes in the future to give enough time for the system to adapt. Prediction information could be used to trigger the decision making process within a proper time.

Prediction of mobile users' locations is essential for a wide range of mobile applications. In cloudified LTE networks, different virtualized services might need the location prediction of the subscribers to optimize the network performance. Radio Access Network (RAN) might benefit from this information for efficient deployment, upgrading, and scaling of the resources [4]. In particular, seamless mobility and service continuity [5] can be supported with optimized traffic forwarding mechanisms that consider the location of users.

As another future Internet technology, Information Centric Networking (ICN) [6] changes how content can be efficiently disseminated. With the exponential increase of mobile traffic, such as video streaming and gaming, bandwidth requirements become an extremely severe challenge that needs to be efficiently tackled by telecommunication network operators.

The Follow-Me Cloud (FMC) paradigm has mainly focused on resource placement in data centres, which can be located in distinct geographical locations [7]. Such placement focused mainly on virtual machine resources and not on services or content as opposed to our previous work [8], which validates a model for FMC to enable optimized content migration for Information Centric Networking. However, most of the existing content migration schemes are reactive, which means they move the content after user movement to let the content "follow" the user. Few of them are proactive such that the content can be migrated before user movement thanks to the prediction of user location in near future.

This paper extends our previous works to the consolidation of the FMC concept as an enabler for bandwidth savings at the core network and lower latency when retrieving content from ICN routers co-located with the LTE evolved Node B (eNB).

In particular, this paper details the components of the FMC model that enables the integration of FMC with Mobility and Bandwidth Prediction as a Service (MOBaaS) to leverage the support of mobility predictions for optimal content migration. Our contributions are threefold: (i) FMC is integrated with MOBaaS to benefit from users' future location predictions; (ii) applying a set of metrics (e.g., the number of users in a particular cell and content popularity) to optimal content migration using the Multiple Attribute Decision Mechanism (MADM) approach; (iii) a solid implementation and evaluation of the proposed model in a cloud testbed infrastructure. To evaluate FMC we consider a scenario where users are moving constantly and require the availability of files at their final destination. These data files can include installation packages of applications, or maps files with high resolution. In this scenario the mobility predictions, assured by MOBaaS, perform a prediction based on set of real traces of mobile users collected in Nokia Mobile Data Challenge (NMDC) [9]. FMC performs optimal content migration considering mobility predictions and multiple metrics based on the content popularity.

The structure of the paper is as follows: an analysis of the proposed mechanisms for content and services migration is presented in Section II. Section III details the FMC architecture and describes the integration scenario with MOBaaS. Section IV presents the evaluation scenarios to validate the proposed integration model (integration of FMC with MOBaaS). The achieved results are detailed in Section V. Finally, Section VI concludes the paper and discusses the main achievements of this research.

II. RELATED WORK

The FMC paradigm is often associated with the placement and migration of virtual machines in Cloud Data Centres (CDC) [10]. For instance, the D-URP proposal [11] aims to optimize the resource placement given the demands in different cloud regions. An algorithm based on a discrete function enables the optimization of efficient resource placement. In this scope, the authors consider resources as virtual machines that need to be placed in distinct geographical locations with time-varying bandwidth demands.

Within a demand and user mobility perspective, Rahul et al. [7] consider the optimization of service placement with a Markov Decision Problem approach, which does not require statistical knowledge of the system. The evaluation of this work relied on real user mobility traces. Ksentini et al. [12,13] also consider a similar Markov Decision approach, but formulate optimization by considering the trade-off between cost of service migration among Data Centres (DC) and the user perceived quality. Indeed, it proposes to deploy cloud services in the small DCs to increase proximity with the users. Hence, when users of a service move from one region to another, services (and content) shall follow the users, with assured levels of quality and availability.

Mobility prediction is essential for a wide range of applications. A larger number of prediction algorithms have been proposed in the literature. Most often, they have been developed

to address a prediction problem in a specific scenario based on location, movement history, movement pattern, velocity, etc [14]. However, most of the works focus on providing an isolated prediction framework, which can not be utilized by other network services. Few efforts have been made to provide mobility prediction as a virtualized network function (VNF) for the virtualized LTE network. Our previous works [15] proposed an initial architecture for MOBaaS in virtualized LTE networks, and proved that the system can be easily instantiated, deployed and disposed in cloud computing infrastructures. The works also show that cloudified MOBaaS can be integrated with any other virtualized LTE components to provide the mobility and link bandwidth prediction information. Detailed evaluation of the proposed prediction algorithms, as well as the MOBaaS service life-cycle management implementation have also been demonstrated [16].

Content Distribution Networks (CDN) are known to leverage the multimedia content distribution according to the geographical location of users [17]. With a Content-Centric Networking (CCN) approach, Michele et al. [6,18] aim to maximize the theoretical gains that are possible with CDN by reducing the total amount of traffic exchanged through the network. Such reduction is based on the popularity of content (e.g. Youtube videos with more views). The performed evaluations demonstrate that by using advanced cache replacement algorithms the overall traffic exchanged between network nodes is highly reduced.

Optimization mechanisms to enhance content migration in the FMC approach, have already been studied in the literature. Multiple criteria such as content popularity, size of the routers cache, can be considered taking into account the restrictions of an optimization solution [8]. In this work, we derive benefits from the mobility prediction information provided by MOBaaS to further improve the FMC functionality, namely in terms of triggering the migration of content before handover.

III. MOBILITY PREDICTION-AWARE FOLLOW-ME CLOUD

This section details the proposed mobility prediction-aware follow-me cloud model to enable efficient content migration. The description of such model focuses mainly on the MOBaaS and the FMC components.

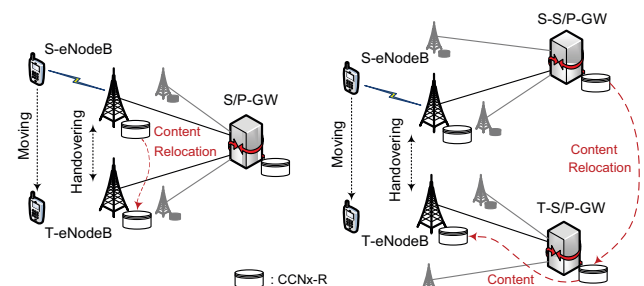


Fig. 1: Content relocation based on users mobility prediction data.

Based on the network design and implementation [19,20], the content may be relocated on the CCNx routers/repositories that are placed on the eNodeBs and/or the Serving/PDN-Gateway (S/P-GWs). When a user moves from one cell (source

eNodeB) to another cell (target eNodeB), which is served by the same S/P-GWs, the content may be relocated between the CCNx routers/repositories implemented in the eNodeBs. If the source and target eNodeBs are not served by the same S/P-GWs, the content may be moved between the CCNx routers/repositories, which are placed in the source S/P-GW (S-S/P-GW), and target S/P-GW (T-S/P-GW), as shown in Figure 1.

A. System Architecture

The overall system architecture is depicted in Figure 2, including functional and logical entities. Functional entities include MOBaaS and FMC components, such as: MP Middleware, FMC Manager, and CCNx Server. Logical entities are components that are related with service deployment on cloud infrastructures (such as OpenStack, CloudStack). They mainly include the Service Orchestrator (SO) to enable the orchestration (i.e. management of scaling) of the different components, the Service Manager (SM) to deploy the SO, and the Cloud Controller (CC), which is a component that enables the abstraction of the cloud infrastructure. These logical entities have been designed and developed in the context of the Mobile Cloud Networking (MCN) project [21].

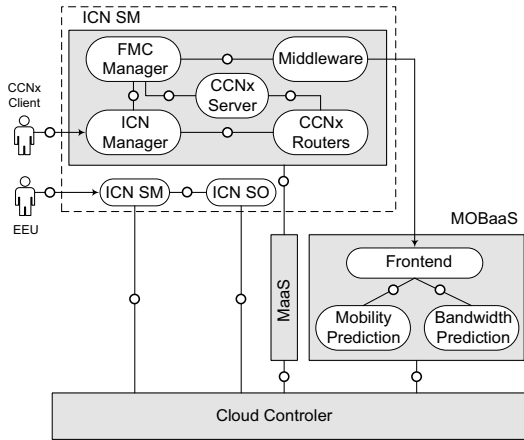


Fig. 2: Mobility Prediction-aware Follow-Me Cloud Architecture

The interconnection of FMC Manager and MOBaaS is achieved by the MP Middleware, which integrates the mobility prediction results with FMC Manager decision logics. In addition, the FMC model depends on some monitoring functions, supported by the Monitoring as a Service (MaaS) sub-system. MaaS monitors several components of FMC (e.g. CPU usage ratio, memory usage, etc), and it will trigger scaling operations based on some pre-defined key performance indicators. The Service Orchestrator of ICN pulls the key performance indicators and triggers the appropriate scaling-in and scaling-out operations according to the configured thresholds (e.g. CPU load > 70%). The FMC model has a dependency on a monitoring service, represented here as MaaS, to assure that the several components are monitored and in order to support scaling-in and scaling-out operations

triggered by MaaS, according to certain thresholds on pre-defined key performance indicators, such as CPU usage ratio.

B. MOBaaS Components

MOBaaS supports two types of notification mechanisms to provide the prediction results: request-based and trigger-based. In the request-based mode, the prediction results will be provided after a prediction request has been received. In the trigger-based mode, the prediction results will only be sent back when pre-defined metrics meet certain thresholds [15]. The FMC model supports both approaches. In the request-based approach, the FMC Manager receives the mobility prediction results based on the requests that are issued towards the MOBaaS Frontend component. In particular, the MOBaaS Frontend component, which is implemented as a web-based application, constantly waits for the prediction requests from the FMC Manager. Whenever a FMC Manager requires a mobility prediction, it sends the request information to MOBaaS using a JSON (Java Script Object Notation) message. The request message includes the User ID (for single user), the current time and date, the current Cell ID, and the time period of the required prediction. Given this information, MOBaaS using the mobility prediction algorithm predicts in which cell the specified user(s) will be located at a certain future time. Next, MOBaaS Frontend returns the prediction results in the JSON message to the consumers, including multiple pairs of <Cell ID, Probability>. Further details regarding probabilities per Cell ID can be found in [15]. Within the trigger-based approach, the components need to register their interest in terms of prediction results on the MOBaaS's Frontend. For each of the registrations, a "callback address endpoint" needs to be provided, so that when predictions results are available they can be provided to the registered components.

C. MP Middleware

The Mobility-Prediction Middleware (MP Middleware) acts as convergence/interoperability layer due to the fact that different approaches can be followed to implement MOBaaS and diverse integration schemes can be pursued.

The MP Middleware works as an interface between FMC Manager and MOBaaS to receive mobility prediction results and to forward the mobility prediction request. Therefore, the MP Middleware is located between MOBaaS and FMC Manager, and it aims to pre-process the data provided by MOBaaS and prepare it according to the format that the FMC Manager is waiting for.

MOBaaS sends back prediction results in the form of a list of users, including the current cell they are located at, and a list of cells with probabilities to where they will be at a future time. The FMC Manager, however, expects a list of source cells with the number of users in each cell, the destination cells of those users, and the number of users already located in those cells. So, instead of sending the mobility prediction data directly to the FMC Manager, MOBaaS will send it to the MP Middleware, which implements the "callback address endpoint" functionality in the trigger-based approach. This

module analyses all the data, processes them and sends them to the FMC Manager with the correct format. To achieve this, it combines all the users that have the same destination by considering the destination cell that has the highest probability or with a probability value higher than 50%, and grouping those with the same originating cell. Afterwards, the data is sent to the FMC Manager.

It is possible that multiple unrelated migrations may happen at the same time when multiple future destination cells have identical probabilities. In this case, the MP Middleware detects that there are different destination cells with identical probabilities and sends this information in separated messages to the FMC Manager, which then performs multiple content migrations simultaneously.

D. FMC Manager

The FMC Manager is the main component of the FMC model and it is responsible for the communication among CCNx Routers, receiving and processing the monitored data, handling prediction results, making content migration decisions and transmitting the actions to the related CCNx Routers.

The FMC Manager determines the list of objects that the router should have in its cache to better serve its current and arriving users. To do so, it stores the monitoring data coming from the routers in a database, and when a migration is needed, it uses this data to support the necessary decision process. After deciding on which objects should be migrated, and the respective subsets, the FMC Manager sends the subset list of content objects to the CCNx Routers at the destination, so that they can be migrated.

Due to the fact the FMC Manager makes the content migration decision based on various types of information, a multiple criteria decision-making scheme is required to support the content migration decision. To achieve this, the FMC Manager interfaces with the Multiple Attribute Decision Mechanism (MADM) module, which provides the optimal subset of content objects to be migrated, solving the NP-Hard optimisation problem. The MADM algorithm provides a score of the content objects to be migrated based on input criteria such as content popularity and content size. The MADM algorithm considered in the FMC model relies on MeTHODICAL [22], according to the accuracy and performance results of previous work [8].

TABLE I: Metrics

Metric	Description
$popCo$	Content Popularity
$popSr$	Popularity of content at source routers
$reqSr$	Number of request at source routers
$popDs$	Popularity of content at destination routers
$reqDs$	Number of request at destination routers
$movU$	Number of users moving to destination cell
$dstU$	Number of users at destination router

The FMC model requires an accurate and efficient decision-making mechanism. Therefore, the decision mechanism takes into consideration content popularity and content size metrics (since local caches of routers have size limitations), as summarised in Table I. Content popularity 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 moving to the destination cell per source cell and the number of users at the destination router.

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

Eq. 1, providing $reqSr > 0$ and $reqDs > 0$, expresses the the content popularity for the N source cells considered in each migration. The values of the $movU$ criterion rely on the mobility prediction information provided by MOBaaS. This module is implemented in Python, and has an interface to R-project [23] to run the MeTHODICAL algorithm, and is available as open source [24].

E. CCN Server

The CCN Server component is responsible for sending all the monitoring data to the FMC Manager, such that it can be stored at the centralized database. In addition, the CCN Server is also responsible for migrating the content objects when requested by the FMC Manager. Monitoring at each router is achieved by a few minor changes in the code of CCNx routers, which is based on the CCNx framework [25]. The monitoring data is temporarily stored at each router and is sent periodically by the CCN Server to the FMC Manager. If the communication fails, the data is kept by the CCN Server so that it can retry to send data successfully in the next communication exchange.

When the CCN Server receives a command from the FMC Manager to migrate content, it will process the list of content objects it needs to have in order to better serve the current and new arriving users. The CCN Server will interact with the CCNx framework in order to generate requests for the list of content objects it received from the FMC Manager.

IV. EVALUATION

This section describes the evaluation methodology and the scenarios we adopted to validate the proposed MP-FMC model. The evaluation aims at validating the integration of the FMC model components with the mobility predictions from MOBaaS in an ICN use case that can be deployed in a cloudified mobile network, and to show the benefits of the integration between MOBaaS and FMC.

A. Evaluation Scenario

The evaluation scenario targets an ICN use case, where a user is able to retrieve content according to its location at a given time instant and according to its movement behavioural pattern. The scenario includes mobility of a user that moves

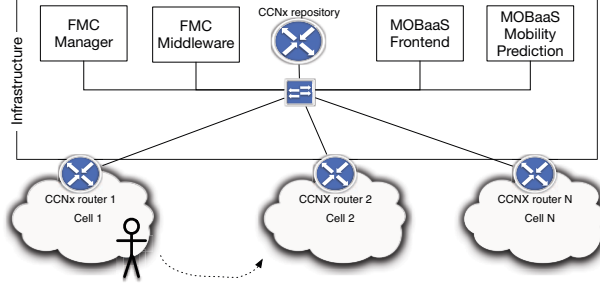


Fig. 3: Evaluation Scenario

from one cell to another, where content must be placed in advance for an improved user experience. Such movement pattern is employed by the Mobility Prediction algorithm based on real-world traces of mobile users trajectories. Figure 3 depicts the evaluation scenario with CCNx Routers co-located with different cells of a cloudified LTE architecture. The goals associated with this scenario focus mainly an accurate usage of mobility predictions in the FMC architecture. Scalability concerns are not considered.

The FMC components and the MOBaaS related components are deployed in a mobile cloud infrastructure and platform, which is based on OpenStack and OpenShift. The database of the MOBaaS is configured with the necessary data to predict the mobility of the moving user.

B. Experimentation Platform

An OpenStack Kilo cloud infrastructure has been used to validate the proposed system. It consists of two nodes with different functions: 1 controller node with 8 physical CPU cores at 3.90 GHz, 16GB RAM, and a 256GB SSD disk for image storage; a compute node with 24 physical CPU cores at 2.50 GHz, 192GB RAM, and a 2.1TB 15k RPM RAID 5 hard disk. Both of these nodes, together with the external iSCSI storage of the compute hard drivers, are inter-connected at 10Gbps with redundant links.

The settings of all the components involved in the experiments are listed in below:

- One ICN Manager instance with 1 vCPU, 2GB of RAM and 20GB of disk space.
- Two CCN Routers (one as source and another one as destination) instances, each with 2 vCPUs, 4GB of RAM and 40GB of disk space.
- One FMC Manager instance with 2 vCPUs, 4GB of RAM and 40GB of disk space.
- One MP Middleware instance with 1 vCPU, 2GB of RAM and 20GB of disk space.
- One MOBaaS instance with 2 vCPUs, 4GB of RAM and 40GB of disk space.
- One MaaS Zabbix instance with 1 vCPUs, 2GB of RAM and 20GB of disk space.

C. Methodology

The evaluation of the MP-aware FMC model includes a non-functional and a functional part. The first one is related with

the deployment, provisioning and disposal of the component instances in a cloud infrastructure. We implemented the FMC model with ICN as a Service (ICNaaS) to enable content migration according to user preferences, and location prediction. Thus, the components of the FMC model are integrated with ICNaaS entities, such as CCNx Routers, the SM and the SO. In addition, MOBaaS includes frontend, mobility prediction and bandwidth prediction components. Monitoring as a Service (MaaS) is employed to gather performance and status information from the different components. This service relies on a Zabbix server. The non-functional metrics, summarized in Table III, are retrieved from the monitoring service and for the logging service that was configured in the MCN project [26] thanks to a graylog cluster [27]. The non-functional metrics were parametrized in the Service Manager and Service Orchestrator entities of the involved services.

To assess the benefits of the FMC model we run a functional evaluation where the mobility of the user has been emulated by disconnecting the user from the origin cell (CCNx Router 1) and then attaching it to the destination cell (CCNx Router 2). Before doing such disconnection at a given time, the MOBaaS estimates the prediction of user location at a certain instant (e.g., 17:00 CET). Such a prediction is then processed by the MP Middleware, which in conjunction with the FMC Manager triggers the content migration of the user to the destination cell. The content migration operations are conducted through specific CCNx commands supported by the CCN Server. FMC is parametrized with the settings providing better performance, namely using the MeTHODICAL [22] mechanism to optimize content migration given multiple criteria [8]. To better describe the message sequences, Figure 4 depicts the information flow in the experiments. Table II summarizes the evaluation criteria and their values that are used in the experiments. The content that is migrated is based data files (e.g., packages to install applications, maps with high resolution).

TABLE II: Tests setup according to evaluation criteria

Evaluation Criteria	Values
MADM in FMC	MeTHODICAL
Employ FMC	FMC and noFMC, with and without FMC model.
File Size	10, 20, 50, 100Mb
MOBaaS users	1 (employed in FMC), 30, 100

To clearly show the benefits of integrating FMC with MP, two main scenarios are established:

- **FMC scenario:** The FMC benefits from the mobility prediction information for optimal content migration. In this case, based on a user's future location (cell) information, the FMC starts the content migration procedure to the target location before the user performs the handover (proactive approach).
- **noFMC scenario:** With noFMC, the content is migrated to the target location after performing the user handover procedure (reactive approach).

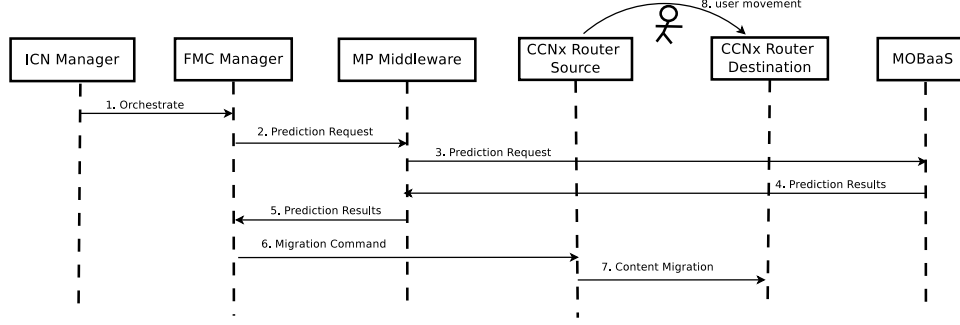


Fig. 4: Experiment Message Flow Chart

The evaluation considered five runs per each test. The results in the non-functional evaluation are entirely dependent on hardware and the main goal is to understand the impact of each service in each phase (i.e., deployment, provision and disposal). The results in the functional evaluation were stable enough to not include more runs per test.

D. Mobility Prediction Evaluation Dataset

In order to make mobility prediction, historical user traces are required. However, due to the fact that the actual user traces are not available in a prototype implementation, we used a mobility data trace collected during the Nokia Mobile Data Challenge (NMDC) [9], which is a large-scale research initiative aimed at generating innovations around smart phone-based research. This dataset includes rich context information running at the mobile phone for around 200 users over 2 years. It includes Global Positioning System (GPS) information, running applications, chat records, calling records, etc. However, for mobility predictions, we are only interested in the GPS location information. From this dataset, we picked data of 100 users over 2-6 months. This is because the quality of the trace has significant impact on the prediction accuracy, and based on our findings in [16], only around 100 users from the original dataset have continuous recordings of their location information. The remaining users have their location data recorded discretely, which is useless to perform prediction of users location. For each user, we separated available data into two parts: the first part is the learning data set (L) and the rest is the testing data set (T). The learning data set includes the first 70% of user's data. It is used to derive the Markov Chain states and to determine its transition probability matrix. The testing data set T contains the last 30% of data traces, which is used to test and evaluate the accuracy of the proposed prediction algorithm.

E. Performance Metrics

This subsection discusses the Key Performance Indicators (KPIs) that are collected to assess the performance of the MP-FMC model in two main areas, as listed in Table III: the *non-functional* and the *functional* KPIs. The *non-functional* KPIs describe the metrics that are related to the deployment, configuration/provisioning and disposal of services in an OpenStack infrastructure and OpenShift v3 platform. The *non-functional*

KPIs inspect performance of the MP-FMC model in terms of content retrieval and migration time, as well as, the mobility prediction time and accuracy.

TABLE III: FMC Evaluation KPIs

Evaluation	KPIs
Non-Functional	Deployment Time (s)
	Provisioning Time (s)
	Disposal Time (s)
Functional	Content migration time (s)
	Content Retrieval time (s)
	Mobility prediction time (s)
	Mobility prediction accuracy (probability %)

V. EVALUATION RESULTS

A. Non-Functional Evaluation

The *Non-functional* evaluation includes the time measurements of deployment, provisioning and disposal of the FMC model and mobility prediction service instances in an OpenStack cloud infrastructure. These results are illustrated in Figure 5. All the services are deployed in the same data centre using the same OpenStack infrastructure and OpenShift platform.

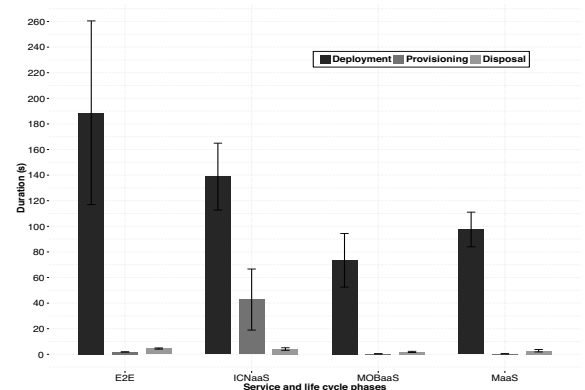


Fig. 5: Non-Functional results

The deployment phase includes the creation of virtual machines (VMs), and allocation of related resources to the VMs (cf. section IV-B). We can observe that the deployment phase takes more time than provisioning and disposal phases.

This is because during the deployment phase, it will create virtual machines, assign associated resources, allocate floating IP addresses to VMs, among others. The “end to end (E2E)” scenario is employed to manage the service composition dependencies, namely FMC model, MOBaaS and MaaS. It should be noticed that the deployment includes also the installation of the SO of each service as a docker container in the OpenShift platform. Considering the achieved time results, we argue that the deployment and configuration of the MP-FMC model is in the order of seconds, allowing it to be employed with other services such as ICN for Content Delivery Networks (CDNs).

The provisioning phase means to properly configure all the VMs/components that have been deployed in the deployment phase. In our implementation, the FMC components are deployed in ICNaaS, which means the evaluation use case includes several CCNx routers, CCNx repositories and the FMC components integrated with MOBaaS. Therefore, all the CCNx and FMC components need to be provisioned, which means to have the endpoints (IP address+ port) of the monitoring service configured, as well as the endpoint of MOBaaS in the MP Middleware component. Such number of configuration operations for multiple components leads to an increased provisioning time, when compared to MOBaaS and MaaS alone.

The disposal phase includes the deletion of resources that are no longer required, virtual machines, allocated IP addresses and others. Since the platform supports parallel operations, the time to perform the disposal is low (around 5s) and is practically similar to all the services.

B. Functional Evaluation

The *functional* evaluation results include the measurements of the following metrics:

- content migration time of files with different sizes towards the destination CCNx router;
- content retrieval time experienced by the end user when it reaches the destination cell with and without applying the FMC mechanism;
- mobility prediction latency according to the number of users that have the prediction requests;
- mobility prediction accuracy of users with different trace qualities.

Figure 6 depicts the content migration time observed in FMC for files with different sizes. This time is measured from the moment when the initial trigger of migration is released until the instant when the final chunk of a file is placed at the cache of the destination router (migration is completed). The FMC manager instructs the CCNx Router(s) and CCNx repositories at the destination cell to populate/fill their caches with the files that are more popular. In our experiments, the destination cell is chosen according to the user mobility prediction results, which means that we configure the destination cell ID as the cell ID of the predicted future cell with the highest probability (derived from the mobility prediction results). As expected, the content migration time increases when the file

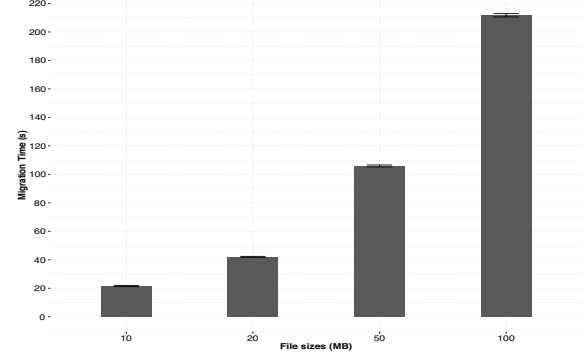


Fig. 6: Migration time of FMC

size becomes large. Indeed, there is a linear growth. For instance, results with 100 Mb size has 200ms, while 50Mb size has 100ms. The obtained results demonstrate the relevance of performing the content migration with the aid of the user mobility prediction information to enhance the retrieval latency, particularly for files with larger size. One may argue that the data transfer time is slow, $\approx 0.47MB/s$. Nonetheless the migration is performed to distinct nodes, namely to the cache of CCNx routers and the CCNx repositories placed at the destination cell. Such performance values apply for a single user and are also restricted with the base protocol used for migration of data, that relies on TCP. Performance gains can be verified with optimized configuration of CCNx routers, for instance, setting UDP as the base transport protocol and size of receiving and transmitting buffers in CCNx routers.

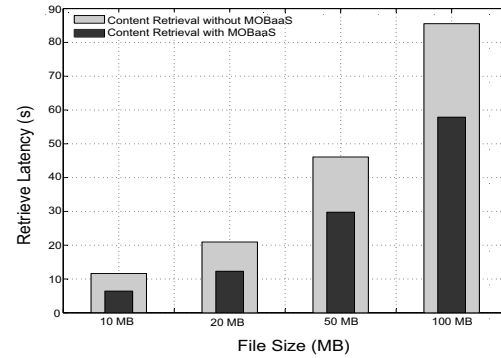


Fig. 7: Content Retrieval Time

Figure 7 shows the content retrieval time by a user when it reaches the destination cell for both the case with and without using MP-FMC. The benefit of using MP-FMC is clear from a user’s perspective, since the content retrieval time is $\approx 33\%$ lower for all file sizes. This is due the fact that the FMC handles the content to be readily available at the next location before user movement thanks to the mobility prediction. In scenarios without using FMC, after completing the handover procedure (user moves to the destination CCNx router), the content requests need to be forwarded to the CCNx repository (through source CCNx router 1), as depicted in Figure 3. Next, the requested content can be retrieved directly from the CCNx repository, which takes more time compared to the scenario using FMC with mobility prediction in advance.

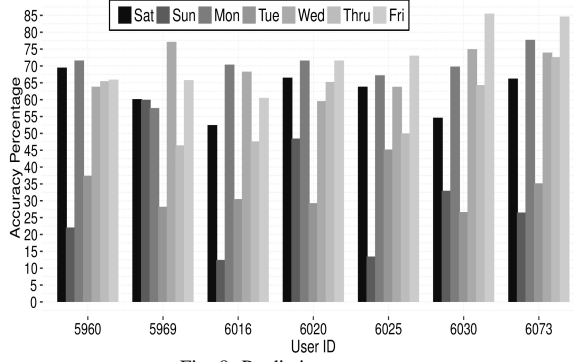


Fig. 8: Prediction accuracy

As we can see before, mobility prediction results in significant benefits to reduce the content retrieval time. However, this is under the assumption that the mobility prediction algorithm could deliver accurate prediction results. Below, we present some results about the prediction accuracy and its determining factors. Figure 8 shows the prediction accuracy for different users on different weekdays. As we can see, for some users, the algorithm can provide 75% as a maximum accuracy for predictions, while for some other users the accuracy is only 15%. The big accuracy variety is due to the difference of users' trace qualities. Some users have continuous recordings of their location information, which leads to accurate prediction outputs. However, some users have their location data recorded discretely, which makes the prediction algorithm difficult to generate accurate results. For instance, we can observe that user 5960 has a much higher prediction accuracy than user 6025. This is because user 5960 has a high quality trace available as the input for the prediction algorithm, which is shown in Figure 9 that representing the number of valid trace dates for user 5960 and 6025. As we can see, user 5960 has much more valid trace data recorded than user 6025, which leads to a much higher prediction accuracy.

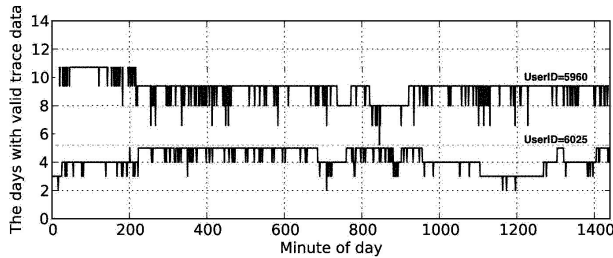


Fig. 9: Trace qualities of different users

Even though accurate prediction could help to improve the content retrieval performance, the time it takes to produce the prediction results also matters. Therefore, Figure 10 shows the time latency of the prediction algorithm as a function of the number of users that have prediction requests. Figure 10 illustrates the average prediction time for different numbers of users that have the prediction requests. As we can see, with a high number of current user predictions (100 users), the mobility prediction algorithm takes around 65 seconds to

generate the prediction locations for the 100 users, while it takes only 2 seconds to make the prediction for a single user.

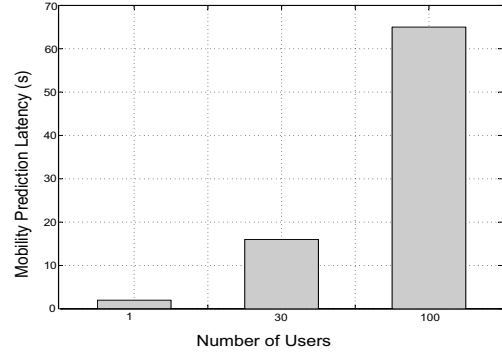


Fig. 10: Prediction latency

The MP middleware component in the FMC model proves its usefulness in making the interface with MOBaaS, given the fact that it mitigates the prediction latency and only provides to the FMC manager the information of mobility predictions with higher probability (greater than 50%), or greater than the probability of staying at the current cell.

The results prove that the proposed MP-FMC model can be efficient from both the user and operator perspectives. This is because it assists to retrieve the user content from the nearest location of the request issued, which results in lower content retrieval latency, higher user quality of experience, as well as optimal bandwidth consumption.

VI. CONCLUSIONS

In this work, the FMC model is fully integrated with a mobility prediction service to exploit the benefits of proactive content migration mechanisms. More specifically, mobility predictions (i.e. number of users at a certain cell) and restrictions of CCNx routers in terms of cache sizes are considered in order to optimize content migration in ICN. Mobility predictions enable content migration in a timely fashion to locations where the content has a high probability of being requested.

The evaluation results demonstrate an improvement of $\approx 33\%$ for the content retrieval time when applying FMC enhanced with mobility prediction information, compared against scenarios without proactive content migration. These results motivate the introduction of additional optimization schemes in order to further reduce the prediction latency, making it suitable for large scale networks. Indeed, the current 4G networks can benefit from FMC model by enhancing distributed mobility management. DMM can proactively make a redirection path in the network according to the next location of users' information. In addition, the MME can balance traffic according the predicted load at each cell.

ACKNOWLEDGEMENTS

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

REFERENCES

- [1] A. Basta, W. Kellerer, M. Hoffmann, H. J. Morper, and K. Hoffmann, "Applying NFV and SDN to LTE Mobile Core Gateways, the Functions Placement Problem," in *Proceedings of the 4th Workshop on All Things Cellular: Operations, Applications & Challenges*. ACM, 2014, pp. 33–38.
- [2] "Openstack cloud infrastructure management platform," August 2016. [Online]. Available: <https://www.openstack.org/>
- [3] M. Mao and M. Humphrey, "A Performance Study on the VM Startup Time in the Cloud," in *Proceedings of the 2012 IEEE Fifth International Conference on Cloud Computing*. IEEE Computer Society, 2012, pp. 423–430.
- [4] L. S. Ferreira, D. Pichon, A. Hatefi, A. Gomes, D. Dimitrova, T. Braun, G. Karagiannis, M. Karimzadeh, M. Branco, and L. M. Correia, "An architecture to offer cloud-based radio access network as a service," in *Networks and Communications (EuCNC), 2014 European Conference on*. IEEE, 2014, pp. 1–5.
- [5] L. Valtulina, M. Karimzadeh, G. Karagiannis, G. Heijenk, and A. Pras, "Performance evaluation of a SDN/OpenFlow-based Distributed Mobility Management (DMM) approach in virtualized LTE systems," in *Globecom Workshops (GC Wkshps), 2014*. IEEE, 2014, pp. 18–23.
- [6] M. Mangili, F. Martignon, and A. Capone, "Performance analysis of content-centric and content-delivery networks with evolving object popularity," *Computer Networks*, vol. 94, pp. 80 – 98, 2016.
- [7] R. Urgaonkar, S. Wang, T. He, M. Zafer, K. Chan, and K. K. Leung, "Dynamic service migration and workload scheduling in edge-clouds," *Performance Evaluation*, vol. 91, pp. 205 – 228, 2015, special Issue: Performance 2015.
- [8] A. S. Gomes and V. Fonseca and B. Sousa and D. Palma and P. Simoes and E. Monteiro and L. Cordeiro, "A Mobile Follow-me Cloud Content Caching Model," in *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*, April 2016, pp. 763–766.
- [9] J. K. Laurila, D. Gatica-Perez, I. Aad, B. J., O. Bornet, T.-M.-T. Do, O. Dousse, J. Eberle, and M. Miettinen, "The mobile data challenge: Big data for mobile computing research," in *Pervasive Computing*, 2012.
- [10] R. W. Ahmad, A. Gani, S. H. A. Hamid, M. Shiraz, A. Yousafzai, and F. Xia, "A survey on virtual machine migration and server consolidation frameworks for cloud data centers," *Journal of Network and Computer Applications*, vol. 52, pp. 11 – 25, 2015.
- [11] W. Wei, Y. Zhang, and Y. Liu, "A time-efficient solution to the general resource placement problem in cloud," *Mathematical Problems in Engineering*, vol. 2014, p. 10, 2014.
- [12] A. Ksentini, T. Taleb, and M. Chen, "A markov decision process-based service migration procedure for follow me cloud," in *Communications, 2014 IEEE International Conference on*, June 2014, pp. 1350–1354.
- [13] T. Taleb and A. Ksentini, "Follow me cloud: interworking federated clouds and distributed mobile networks," *Network, IEEE*, vol. 27, no. 5, pp. 12–19, September 2013.
- [14] B. Liang and 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, Mar 1999, pp. 1377–1384 vol.3.
- [15] M. Karimzadeh, Z. Zhao, L. Hendriks, R. de O Schmidt, S. la Fleur, H. van den Berg, A. Pras, T. Braun, and M. Corici, "Mobility and bandwidth prediction as a service in virtualized lte systems," in *Cloud Networking (CloudNet), 2015 IEEE 4th International Conference on*. IEEE, 2015, pp. 132–138.
- [16] Z. Zhao, M. Karimzadeh, T. Braun, A. Pras, and H. van den Berg, "A demonstration of mobility prediction as a service in cloudified lte networks," in *Cloud Networking (CloudNet), 2015 IEEE 4th International Conference on*. IEEE, 2015, pp. 78–80.
- [17] T. Plagemann, V. Goebel, A. Mauthe, L. Mathy, T. Turlatti, and G. Urvoy-Keller, "From content distribution networks to content networks issues and challenges," *Computer Communications*, vol. 29, no. 5, pp. 551 – 562, 2006, networks of Excellence.
- [18] M. Mangili, F. Martignon, and A. Capone, "Optimal design of information centric networks," *Computer Networks*, vol. 91, p. 638, 2015.
- [19] M. Karimzadeh, T. Satria, and G. Karagiannis, "Utilizing icn/ccn for service and vm migration support in virtualized lte systems," in *4th International Conference on Cloud Computing and Services Science, CLOSER 2014*. ScitePress, 2014, p. 84.
- [20] T. A. Satria, M. Karimzadeh, and G. Karagiannis, "Performance evaluation of icn/ccn based service migration approach in virtualized lte systems," in *Cloud Networking (CloudNet), 2014 IEEE 3rd International Conference on*, 2014, pp. 461–467.
- [21] F. Dudouet and et al., "A case for CDN-as-a-service in the cloud: A Mobile Cloud Networking argument," in *IEEE ICACCI, 2014*, 2014.
- [22] Bruno Sousa and Kostas Pentikousis and Marilia Curado, "MeTHODICAL: Towards the Next Generation of Multihomed Applications," *Computer Networks*, vol. 65, pp. 21–40, June 2014.
- [23] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2015. [Online]. Available: <https://www.R-project.org/>
- [24] OneSource, "Follow Me Cloud," August 2016. [Online]. Available: <https://github.com/OneSourceConsult/FMC>
- [25] Palo Alto Research Center, "Project CCNx," August 2016. [Online]. Available: <http://www.ccnx.org/>
- [26] M. Consortium, "EC FP7 Mobile Cloud Networking project," May 2015. [Online]. Available: <https://www.mobile-cloud-networking.eu/>
- [27] Graylog, Inc, "Open source log management platform," August 2016. [Online]. Available: <https://www.graylog.org/>