Offloading Service Provisioning on Mobile Devices in Mobile Cloud Computing Environments

Marco Conti, Davide Mascitti(✉), and Andrea Passarella

IIT-CNR, Via G. Moruzzi 1, 56124 Pisa, Italy
{marco.conti,davide.mascitti,andrea.passarella}@iit.cnr.it

Abstract. Mobile cloud computing is one of the facets of cloud based systems, whereby mobile nodes obtain services from a global remote cloud platform in a more efficient way with respect to local service execution. Unfortunately, recent forecasts on cellular bandwidth (that is the key enabler for this paradigm) pose significant challenges to the practical applicability of this approach. In this paper, we explore a complementary mobile cloud computing solution, where mobile nodes can also rely on other nodes in the vicinity that could provide the sought service. These nodes are contacted via direct communication based on WiFi or Bluetooth, which therefore offloads traffic from the cellular network. In the proposed system, mobile nodes decide dynamically whether to access global or local cloud services based on the availability of the latter in their vicinity, and the load on the cellular network. Simulation results show that this solution provides lower average service provision times with respect to an alternative based exclusively on a remote cloud. As a side effect, such a system avoids cellular congestion and possible saturation, even in case of significant load.

1 Introduction

Mobile cloud computing is considered a very promising area in the cloud computing domain [5]. A popular approach to mobile cloud computing consists in moving the execution of services from mobile users' devices to the cloud. This approach is motivated by the fact that executing services on the cloud, instead of locally on users' devices, saves mobile devices resources, and service execution times can be shortened thanks to the inherent scalability of cloud service provisioning platforms. The core assumption at the basis of this approach is that mobile devices are constantly connected to the Internet through an extremely high capacity wireless network, such that it is easy to move data back and forth between the mobile devices and the remote cloud platform. In this view, the capacity leap expected from 4G cellular networks (LTE-A) [3] is supposed to fully support this mobile cloud computing paradigm.

Unfortunately, recent forecasts challenge the practical applicability of this approach. While 4G cellular networks will definitely provide much higher capacity compared to 3G, it is also expected that the data traffic generated by mobile

S. Hunold et al. (Eds.): Euro-Par 2015 Workshops, LNCS 9523, pp. 299–310, 2015.
DOI: 10.1007/978-3-319-27308-2_25

users will increase much faster. For example, CISCO [3] foresees that mobile traffic demand will increase by at least ten times between 2014 and 2019, while cellular capacity will grow only by a factor of 1.4 in the same time frame. This challenges the possibility to support very frequent and possibly large data transfers required by this type of mobile cloud computing solutions. In cases where the cellular network is congested, it would be too slow (or even impossible) to reach remote cloud services, thus making this approach technically unfeasible. In addition, this might also result in significant economic losses for cloud service providers, as it has been recently shown that there is a direct impact on the provider revenues of even small additional delays (in the order of hundreds of milliseconds) in accessing remote cloud services [9]. Another possible scheme for mobile cloud computing proposed in the literature consists in providing services directly at the edges of the infrastructure, i.e. on cellular base stations (eNodeB in the LTE terminology) [2]. This would not solve the aforementioned problem, as typically the bandwidth bottleneck would be in the cellular access network, and therefore even data transfer between mobile devices and eNodeBs might be problematic.

To counteract the mismatch between mobile data traffic demand and cellular capacity, a promising approach is traffic offloading [13]. In one of the typical offloading scenarios, nodes receive data through direct device-to-device (D2D) communications with other mobile nodes, instead of through the cellular network. Opportunistic networking solutions are typically used [12], whereby mobile nodes exploit direct data transfer opportunities enabled by various wireless technologies (such as WiFi or Bluetooth in ad hoc mode) when they come close enough to be in each other's direct transmission range.

In this paper, we exploit a conceptually similar approach to offload traffic related to service provisioning to mobile users. Specifically, we explore another concept for mobile cloud computing, applicable when services can also be provided locally between mobile devices, by exchanging the related data between them during opportunistic contacts. Service provisioning between mobile devices through opportunistic contacts has been investigated in the literature as the opportunistic computing paradigm [10]. In opportunistic computing, mobile nodes form *mobile clouds at the edges of the global Internet infrastructure*, through which local service provisioning is supported. While exploiting opportunistic computing, our solutions goes one step beyond. In our solution, nodes requiring a service (hereafter referred to as *seekers*) evaluate whether it is more efficient to execute the service on a remote cloud, or on a locally available mobile node (not necessarily in contact with the seeker when the service request is generated). This approach is able to exploit both remote cloud platforms, when the cellular network is not congested, and local service provisioning, otherwise. As such, it takes the best of the conventional mobile cloud computing approach and pure opportunistic computing paradigms. This approach is appealing also because of the resources already available on modern mobile personal devices. For example, high computational capability, ample storage and sensors, can be exposed to other users as services that can be accessed by other devices through direct contacts [5]. While it is clearly unreasonable to assume that any cloud service could also be provided locally, it is sensible to assume that a reduced set of services might be provided by other mobile nodes in local proximity. Note that in some cases, this might indeed even preferable. For example, when services consist in elaboration of data locally available on mobile users, it might be more appropriate for privacy reasons that data stay on the device of their owners.

Together with the specification of the algorithms to realise this mobile cloud computing approach, in this paper we also present simulation results showing that our solution is capable of offering better service provisioning time than a system where only the remote cloud is used. We show that the proposed system is able to autonomously adapt to the level of congestion of the cellular network, avoiding to contribute to its saturation, and still preserving low service provisioning times to the users, even in cases where the cellular network is highly congested.

This paper is organized as follows. Section 2 describes the main approaches in mobile cloud computing and for service provisioning through opportunistic computing. The structure and behaviour of the proposed system is presented in Sect. 3. Performance evaluation results are presented and discussed in Sect. 4. Finally, concluding remarks are reported in Sect. 5.

2 Related Work

The field of mobile cloud computing has seen multiple contributions aiming at building solutions for service provisioning with very different applications and objective [5]. Depending on the application and objectives, mobile cloud computing solutions may differ in their architectures and in how the service behaves in case of service requests. Four main types of system architecture may be individuated: remote cloud solutions, local mobile clouds, cloudlets and hybrid solutions [1].

Systems for remote cloud computing offload functionalities (computation, storage, coordination) on the remote cloud. [8] describes numerous proposals for transferring computation functionalities from mobile devices to the cloud to improve performances or with the objective of saving energy.

Local mobile clouds are systems where mobile devices collaborate in an area in order to provide functionalities to other participants, without using the infrastructure. MobiCloud [6] is a cloud framework in which mobile devices in a MANET are virtualized to service nodes or service broker, linked through a MANET routing protocol. In [10], instead, opportunistic computing is used to enable mobile users to access services on other mobile devices, with the possibility to create sequential compositions of services to extend the functionality available at individual nodes.

In cloudlets, services and resources are located dynamically on static devices connected to the wireless infrastructure in the vicinity of the mobile devices. For example [15] describes how to use cloudlets to dynamically instantiate Virtual Machines for mobile users that can be accessed through wireless LAN networks.

Hybrid solutions unite remote, local clouds and cloudlets to create systems where functionalities can be provided on different sites. Some initial proposals going into this direction have been proposed recently. For example SAMI [14] and MOCHA [16] are two examples of systems where computation activities needed by mobile devices are divided and distributed to sites of different nature. SAMI has the objective of minimizing the energy and monetary cost of computation when deciding to execute code on other mobile devices, a local cloudlet or on the remote cloud, while MOCHA uses information on latency and response times for all available remote cloud sites and the local cloudlet to decide where to execute code. However, none of these solutions exploit collaborative service provisioning among mobile devices, which is the key element of our approach.

3 Hybrid Mobile Cloud Computing Solution for Service Provisioning

In this section we present the characteristics of our solution that enables the establishment of a local mobile cloud to support the execution of services available both on the cloud and on mobile devices in the area. The main components of the system can be seen in Fig. 1: the local mobile cloud, which is made up of mobile devices that can communicate with each other through wireless interfaces and that can request and provide services (pictured in the figure as S_1, S_2, S_3, S_4) to the other nodes; the eNodeB, which grants connectivity to the infrastructure to the local mobile cloud; the remote cloud, which hosts services the mobile nodes can access through the eNodeB.

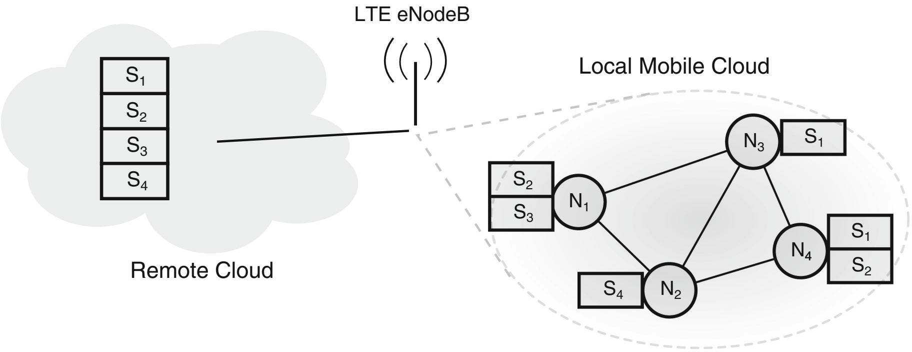

Fig. 1. Actors of the systems

At the high level, when a request for a service is generated at a mobile node (seeker), our algorithm decides whether this request should be served by the global cloud platform, or by some other mobile node nearby. We explain the details of the algorithm in the following subsections. Specifically we describe the system structure and behaviour by analysing the decision process involved in deciding how to solve a service request (Subsect. 3.1), the data that must be collected in order to take the decision (Subsect. 3.2) and the model used to determine how to resolve a request (Subsect. 3.3).

3.1 Resolution Process

The resolution process is shown in Fig. 2 and starts with the *service request generation*, when a mobile node (*seeker*) runs an application that generates a request for a service. The seeker sends a message (*eNodeB inquiry*) to the eNodeB asking for information about the state of the LTE available data rates in upload and download, and an estimate of the time needed to execute the service on the remote cloud.[1] The eNodeB, at the reception of the message, observes the bandwidth occupation and sends this data as a response (*eNodeB response*) to the seeker, including the estimate on the service execution time on the remote cloud (*remote knowledge collection*).

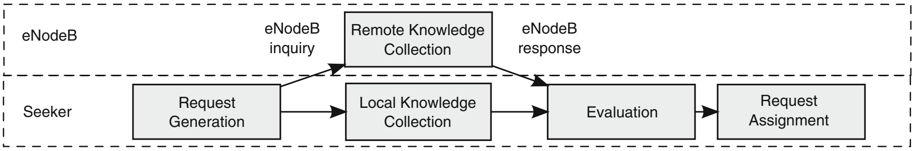

Fig. 2. Request resolution process

At the reception of the response, the seeker estimates the total service provisioning time of the request using the remote cloud service. The seeker also uses a local knowledge base containing previously collected data (*local knowledge collection*) on the other providers in the mobile cloud, like statistics on the mobility of the providers, the state of their computation queue and the offered services. The information in the local knowledge base is refreshed whenever two modes are in direct contact.

Thanks to the knowledge base, the seeker can evaluate the expected service provisioning time for all the known mobile providers that can be used to solve the request. These expected times are compared to the estimated service provisioning time of using the remote cloud service (*evaluation*).

If the seeker selects the remote cloud solution, it immediately starts sending the service request using the LTE infrastructure. Instead, if the selected provider is in the local mobile cloud and the seeker is currently not in contact with it, it waits the next contact with the selected provider in order to start sending the request. In this period of time further contacts between the seeker and other mobile providers may happen, triggering new information exchanges, a possible re-evaluation of the most suitable provider, and therefore a change in the service execution plan.

3.2 Data Collection

The information, required to decide how to serve a request, consists in the upload and download data rates in using the LTE infrastructure and the aver-

[1] Note that the size of this traffic is minimal, and therefore can be considered negligible from the cellular network congestion standpoint.

age execution time of the service that is requested. This data is obtained by the seeker through the *eNodeB response message* that is created by the eNodeB. The eNodeB collects the average execution times of the services requested by the nodes and stores them in a database. It estimates the upload and download data rates based on the current traffic generated by mobile users in the cell.

As will be more clear from the following section, the information required about the other mobile provider is: (i) the average duration of contact and intercontact times with the seeker, (ii) the average data rate in the communications with the seeker, (iii) the service list of the provider, (iv) the provider queue statistics, like the average load, the average request arrival rate and the average service time, (v) the average queue of data to transfer from the provider to the seeker. This information is collected by each node by monitoring contacts with other nodes (for what concerns contact, intercontact times and average data rate), and by exchanging the other statistics during direct contacts.

3.3 Evaluation of Service Provisioning Alternatives

The seeker uses two models to evaluate respectively the expected service provisioning time for each provider in the local mobile cloud that can solve the request and the expected service provisioning time using the remote cloud.

The first model is based on the model for opportunistic computing described in [10]. For a given provider, this model gives a closed form expression for the expected value of the random variable representing the service provisioning time R_{mobile}, characterizing it as the sum of five successive periods of time that can be also formulated as random variables:

1. *Contact of the service provider* (W)*.* The time needed by the seeker to encounter the provider after the point in time when the evaluation is performed. If the seeker is already in contact with the provider, the value is zero, otherwise it is the expected duration of the intercontact period.
2. *Data transfer* (Input time B, Output time θ)*.* The time needed to transfer the input parameters from the seeker to the provider and the output parameters from the provider to the seeker (after the execution time is complete). These values include possible additional delays due to disconnection periods when the transfer is suspended as well as delays due to the presence of data from previous requests that need to be transferred to (or from) the same provider. The value for B is calculated as the time needed to transfer the data to the provider without disconnection, plus the expected duration of all the intercontact phases occurring before the end of the transfer. The expected value of θ is analogous to B, but it must consider the state of the connection seeker-provider at the end of the service execution: if θ starts during in intercontact period, it must consider an added delay to begin the transfer, if it starts during a contact it considers whether there could have been disconnections before the phases to estimate its residual duration.
3. *Queue waiting time* (DQ)*.* Once onto the provider, actual execution may be delayed due to previous pending requests. To calculate the expected time

of the phase, the model regards the provider as a $M^{[X]}/M/1$ queue and calculates the value using knowledge on the average load, service time and request arrival rate.

4. *Service execution time* (DS). The time to execute the service on the provider. It is calculated as the average previous executions on the provider of the requested service.

The formulation of the expected service provisioning time using a given mobile node becomes:

$$E[R_{mobile}] = E[W + B + DQ + DS + \theta]$$

For the remote cloud alternative, we can estimate of the service provisioning time t_{remote} using the information provided by the eNodeB in the eNodeB response and data locally available to the seeker. The service provisioning time can be estimated as the sum of the estimate of three delays: the time needed to upload data to the eNodeB t_{upl}, the time needed for the eNodeB to send data to the remote service provider and wait for the result of the computation t_{exec}, and the time needed for the seeker to download the output data of the service t_{down}. These estimates can be formulated as:

1. *LTE upload Time t_{upl}.* The time needed to transfer the service input data of size k_{input} and possibly queued data of size $k_{lte\ queue}$ from the seeker to the eNodeB, using the upload link that has a data rate of V_{upl}. k_{input} is a property of the service request generated and consequently known by the seeker. $k_{lte\ queue}$ is a value directly observable by the seeker at the moment of the evaluation. With these values, the total estimated LTE upload time can be formulated as:
$$t_{upl} = \frac{k_{input} + k_{lte\ queue}}{V_{upl}}$$
2. *Remote cloud latency and service execution time t_{exec}.* The time needed to transfer the input data from the eNodeB to the remote cloud provider, the time needed to execute the service, and the time needed to transfer the output data back to the eNodeB. Given that the amount of time spent transferring the data and executing the service is dependent on many factors that are out of the control of the system, like the actual provider location, the bandwidth available on the path to the provider, and the amount of resources dedicated to service executions, we can estimate t_{exec} using the average of previous actual values of the remote cloud latencies and service execution times for the same requested service.
3. *LTE download time t_{down}.* Similarly to the upload time, it represents the time needed to transfer the service output data, of size k_{output} which value is a property of the request, from the eNodeB back to the seeker, using the download link of data rate V_{down}, whose value is provided in the eNodeB response. t_{down} can be expressed as:
$$t_{down} = \frac{k_{output}}{V_{down}}$$

4 System Evaluation

In this section we compare the performance of the hybrid solution explained in Sect. 3 with one that only uses a global cloud platform. We show a comparison of the average service provisioning times for both approaches in a range of scenarios that differ for amount of data that are transferred as service input and output for each request, and also for the amount of requests that are generated by the devices. We also detail the behaviour of the hybrid approach by analysing the fraction of requests that are solved using mobile providers in each scenario.

Table 1. Default simulation parameters

Simulation runs per scenario	10
Number of mobile nodes	30
Simulation space	500 m × 500 m
Total simulation time	400000 s
Mobility warm-up period	10000 s
Statistics warm-up period	10000 s
Request generation phase duration	360000 s
Wi-fi connectivity range	90 m
LTE download transmission speed	300 Mbps
LTE upload transmission speed	75 Mbps
Wi-fi transmission speed	54 Mbps
Density of each service	25 %
Number of different services	15
Average mobile service execution time	10 s
Average remote cloud service execution time	5 s

Simulation were developed using TheOne, which is a reference simulation environment for opportunistic networking and computing [7]. The basic simulation parameters used in this paper are listed in Table 1. In these simulations, the mobile devices move following RandomWayPoint mobility traces as specified in [11]. We assume that mobility of nodes is confined withing a single LTE cell, served by a unique eNodeB. Each simulation run lasts 400000 s. For each request, a target service is randomly chosen and also a device is randomly chosen to act as a seeker for the request. The service and the seeker are chosen according to uniform distributions. The services that can be provided and requests are 15 in total, with each of them available on the remote cloud and on 25 % of the mobile nodes, chosen randomly following an uniform distribution. Simulated LTE data rates are 300 Mbps for download and 75 Mbps for upload based on current estimates of the maximum 4G capacity [4], opportunistic transfers are supposed to occur at the maximum capacity of 802.11g technology of 54 Mbps.

We assume that a variable number of additional mobile devices generate traffic in the same LTE cell. The number of additional devices is generated according to a standard birth/death process. The total LTE capacity is shared between the active devices (i.e., the seekers and providers, plus these additional ones), such that the bandwidth available to the seekers and the providers changes over time based on the number of other active mobile devices in the cell. The number of additional nodes can vary between 0 and 40, and the transition rate to a new state is 0.01 per second both for birth events and death events. We replicate each simulated scenario 10 times. In all runs, the events related to the transition of the process defining the additional nodes activity are exactly the same. This guarantees that the congestion on the LTE network due to the additional nodes is the same when we vary the other simulation parameters.

The tests are repeated varying the rate of request generation by the system. In "10–15" scenarios a new request is generated after a time interval in the range [10,15]s after the previous one. This value is changed in the other scenarios to "15–20" and "20–30". For each of these values the tests are repeated changing the amount of data that has to be transferred as input and output of the services, from 40 MB to 80 MB and 160 MB. In each simulation, the input and output data sizes are the same for all services and requests. All the results shown are the average results of the 10 independent simulation runs executed for each scenario, with 95 % confidence intervals.

4.1 Service Provisioning Time Comparison

Figure 3 compares the average service provisioning times for the hybrid approach and for the pure LTE approach. The x axis marks the different tested scenario, from the one generating maximum traffic (on the left) to the one generating minimum traffic (on the right).

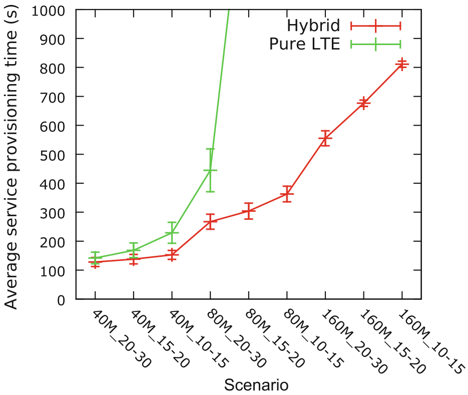

Fig. 3. Service provisioning times

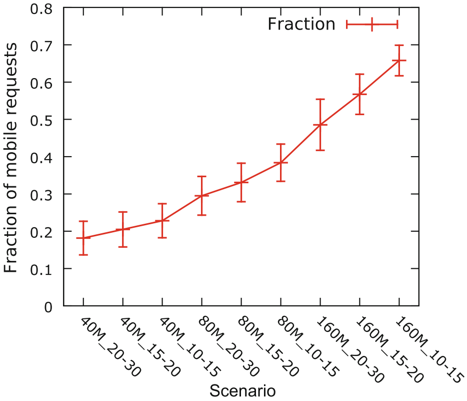

Fig. 4. Fraction of requests

We can see that the average service provisioning times are faster for the hybrid approach in all scenarios, even when the request load is at its lowest (40 MB 20–30), the hybrid approach achieves an average that is about 10 % lower than the pure LTE approach. This difference in results grows as scenarios get heavier in load, with the 40 MB 15–20 and 40 MB 10–15 scenarios having differences respectively of about 18 % and 32 %.

This difference continues to grow as the traffic in the scenario grows, with the pure LTE approach that is unable to avoid saturation from the 80 MB scenarios and is unable to complete the service requests for any seeker. The hybrid approach, instead is able to keep service provisioning consistent without overloading the infrastructure in all the analysed scenarios.

4.2 Split of Service Executions in the Hybrid Approach

Figure 4 shows the fraction of requests served locally by mobile nodes in the different scenarios. We can see first of all that the shape of the graph resembles the one seen in Fig. 3, indicating a correlation. It is also notable that for the scenario with the lowest load the hybrid approach still assigns about 20 % of requests to the local cloud. This indicates that local service provisioning might be useful even in cases when the LTE network is not particularly congested (this is the case, for example, when the seeker and provider are already in contact when the service request is generated, and the size of the input/output parameters is not that large). In the highest load scenario the ratio rises to an average of 65 %, This indicates that our solution avoids cellular saturation, and is still able to exploit remote cloud execution when appropriate.

To further explore the behaviour of the system, we analysed the variation of the fraction of requests solved through the mobile cloud during specific simulation runs. To better understand this index, we plot it together with the fraction of additional nodes generated by the birth/death process (the fraction being computed over the maximum number of nodes, i.e. 40). In Fig. 5 we can see the results for run number 2 of the 10 total simulation runs for each scenario.

The graphs show a correlation between fraction of additional nodes generating traffic and the fraction of the requests assigned to mobile providers. Scenarios with the 40 MB requests (blue lines), corresponding to a light transfer size due to service provisioning, have long periods of time where all requests are assigned to the remote cloud, until the added traffic is heavy enough. Instead the scenario with 160 MB requests (red lines) rarely has periods with no requests assigned to mobile providers, and at the highest request generation rate ("10–15") the ratio never goes below 20 %. This last result indicate that the system consistently assign requests to mobile providers even during periods when the added traffic is negligible.

Based on the above results, we can conclude that the hybrid approach provides significant advantages in achieving better average service provisioning times. This is achieved also thanks to a dynamic detection of the status of the LTE network, that allows the proposed solution to correctly estimate whether remote or local service provisioning is more appropriate. This solution is thus able to avoid to saturate the LTE network, and to guarantee service provisioning also when the LTE network becomes congested.

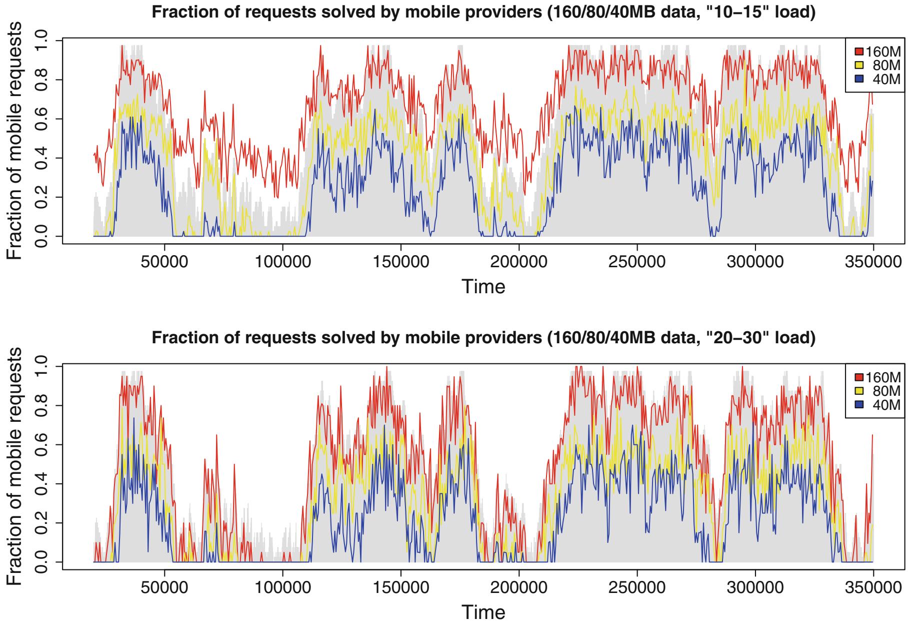

Fig. 5. Share of requests

5 Conclusions

In this paper we presented a mobile cloud computing solution that enables the creation of local mobile cloud networks to offload service provisioning from the remote cloud. We defined the behaviour of the system when a decision is to be taken whether a service should be provided from the remote platform or through some nearby mobile node, taking into account the state of the LTE network and of the surrounding devices. We presented sets of simulations to show the advantages in using this approach instead of relying exclusively on remote cloud services by showing that seekers experience better average service provisioning times and that the system is able to avoid congestion of the LTE network.

References

1. Abolfazli, S., Sanaei, Z., Ahmed, E., Gani, A., Buyya, R.: Cloud-based augmentation for mobile devices: motivation, taxonomies, and open challenges. IEEE Commun. Surv. Tutor. **16**(1), 337–368 (2014)

2. Barbarossa, S., Sardellitti, S., Di Lorenzo, P.: Communicating while computing: distributed mobile cloud computing over 5g heterogeneous networks. IEEE Sign. Process. Mag. **31**(6), 45–55 (2014)
3. Cisco: Cisco visual networking index: Global mobile data traffic forecast update, 2014–2019, February 2015
4. Dahlman, E., Parkvall, S., Sköld, J. (eds.): 4G LTE/LTE-Advanced for Mobile Broadband. Academic Press, Oxford (2011)
5. Fernando, N., Loke, S.W., Rahayu, W.: Mobile cloud computing: a survey. Future Gener. Comput. Syst. **29**(1), 84–106 (2013)
6. Huang, D., Zhang, X., Kang, M., Luo, J.: Mobicloud: building secure cloud framework for mobile computing and communication. In: 2010 Fifth IEEE International Symposium on Service Oriented System Engineering (SOSE), pp. 27–34, June 2010
7. Keränen, A., Ott, J., Kärkkäinen, T.: The one simulator for DTN protocol evaluation. In: Proceedings of the 2nd International Conference on Simulation Tools and Techniques, ICST Simutools 2009, Brussels, Belgium, pp. 55:1–55:10 (2009)
8. Kumar, K., Liu, J., Lu, Y.H., Bhargava, B.: A survey of computation offloading for mobile systems. Mob. Netw. Appl. **18**(1), 129–140 (2013)
9. Linden, G.: Marissa mayer at web 2.0. http://glinden.blogspot.it/2006/11/marissa-mayer-at-web-20.html
10. Mascitti, D., Conti, M., Passarella, A., Ricci, L.: Service provisioning through opportunistic computing in mobile clouds. Procedia Comput. Sci. **40**, 143–150 (2014). Fourth International Conference on Selected Topics in Mobile and Wireless Networking (MoWNet 2014)
11. Navidi, W., Camp, T.: Stationary distributions for the random waypoint mobility model. IEEE Trans. Mobile Comput. **3**(1), 99–108 (2004)
12. Pelusi, L., Passarella, A., Conti, M.: Opportunistic networking: data forwarding in disconnected mobile ad hoc networks. IEEE Commun. Mag. **44**(11), 134–141 (2006)
13. Rebecchi, F., de Amorim, D.M., Conan, V., Passarella, A., Bruno, R., Conti, M.: Data offloading techniques in cellular networks: a survey. IEEE Commun. Surv. Tutor. **17**(2), 580–603 (2015). Secondquarter 2015
14. Sanaei, Z., Abolfazli, S., Gani, A., Shiraz, M.: Sami: Service-based arbitrated multi-tier infrastructure for mobile cloud computing. In: 1st IEEE International Conference on Communications in China Workshops (ICCC 2012), pp. 14–19, August 2012
15. Satyanarayanan, M., Bahl, P., Caceres, R., Davies, N.: The case for VM-based cloudlets in mobile computing. IEEE Pervasive Comput. **8**(4), 14–23 (2009)
16. Soyata, T., Muraleedharan, R., Funai, C., Kwon, M., Heinzelman, W.: Cloud-vision: real-time face recognition using a mobile-cloudlet-cloud acceleration architecture. In: IEEE Symposium on Computers and Communications (ISCC 2012), pp. 59–66, July 2012