EURASIP Journal on Wireless Communications and Networking
a SpringerOpen Journal

RESEARCH Open Access

# A lightweight framework for prediction-based resource management in future wireless networks

Eleni Patouni*, Damianos Kypriadis and Nancy Alonistioti

**Abstract**

The vast proliferation and widespread use of a variety of mobile devices in the heterogeneous networking environment necessitates the introduction of lightweight management mechanisms to ease the administration complexity and optimise the overall system performance. To this end, one key research problem is the design of novel functionalities in network nodes to enable their self-adaptation to varying operational conditions, e.g. their own resources saturation–and to the status of other neighbouring nodes, to assure stability and optimality in the resource management. In these terms, the introduction of advanced techniques for the load balancing of users' requests in order to avoid the resources saturation is a fundamental objective. The latter addresses both the local node level as well as the cluster level of neighbouring nodes. In this article, an appropriate model for the management of computational system resources is proposed, enhanced with prediction schemes. An algorithmic framework is introduced for the proactive load balancing of user decision-making requests, assuming reconfigurable and autonomous mobile devices. The latter is based on the proposed metric of user satisfaction; such metric is a function of the network response time for serving the decision-making requests. An analytical model has been proposed to compute the predicted values of the user satisfaction, extending the prediction models by Andreolini. Acting on top of the typical load-balancing actions for handling the current resources saturations, the goal of this framework is to avoid the full utilisation of system resources in the near future. Afterwards, the introduced prediction-based load-balancing framework has initially been evaluated in a test-single node system and then applied in a case study system. The obtained results show the gains of the presented framework in terms of the number of dropped user requests. The introduction of prediction schemes enables to minimise the number of dropped user requests for both classes of mobile devices. It should be noted that the prediction framework optimises the failure rates for the autonomous mobile devices. This outcome indicates that the introduction of intelligence in the mobile devices eases their proactive management.

**Keywords:** resource management, load balancing, user satisfaction, prediction, decision-making, autonomous

## 1. Introduction

The vast proliferation in the number and type of mobile devices along with their widespread use has been the emerging trend that dominated next-generation mobile communication systems. Such an environment raises an unprecedented demand for the dynamic, lightweight management of the multitude of mobile devices, to ease the complexity of their administration and optimise the overall system performance [1,2]. Focusing on the overall network management aspects, a key issue is the efficient management of the system resources, spanning from the physical layer, to the protocol stacks and up to the application and services layer.

To this end, the notions of reconfigurability and autonomic networking provide a solution to this problem, fostering the introduction of intelligence in mobile devices and network nodes [3]. The intelligence is translated in awareness and adaptation capabilities as well as distribution of the management overhead in the system

* Correspondence: elenip@di.uoa.gr
Departments of Informatics and Telecommunications University of Athens, Athens, Greece

entities, thereby enabling the flexible and efficient system administration. In such an environment, the following technical challenge is raised:

- How to optimise the network resource management in terms of handling user requests, taking into account the overall system performance and the devices capabilities?

This article investigates this challenge, by addressing the dynamic resource management of the network nodes that serve the control plane signalling coming from the mobile devices. A key question is what happens with the huge amount of control plane data that are generated to the network side. Inevitably, there is a need to handle the generated control plane signalling/decision-making requests efficiently, alleviating the network management burden and optimising the utilisation of the respective system resources. In this work, two types of mobile devices are considered in our system: reconfigurable and autonomous ones. The differentiation among them lies in their decision-making capabilities. Reconfigurable mobile devices simply produce decision-making requests to the network side, whereas autonomous mobile devices are able to specify a first set of alternatives that are then communicated to the network side for validation. To this end, a lightweight management mechanism of the decision-making requests is proposed, which allows for their specialised administration according to the type of originating mobile device.

In this article, we introduce a lightweight management framework for the nodes' resources management, when they are close to their full utilisation (system saturation). Such framework enables the resources reallocation, through the load balancing of the user requests to neighbouring nodes. At first, the system capacity in terms of managing user requests is investigated; next the management of the requests exceeding the system capacity is examined, targeting the minimisation of the requests that cannot be served and therefore will have to be dropped. To enable the optimisation of the system performance, one major issue is the introduction of proactivity in the resources administration. This is achieved by introducing advanced prediction functionality that will enable us to forecast the saturation of system resources and thereby proactively schedule actions to avoid such situation. To this end, we propose an innovative prediction-based load-balancing scheme which allows for the proactive management of the resource saturation, in terms of handling the decision-making requests. The possibility of predicting future loads is very important–it will enable to proactively manage the system resources, by triggering load-balancing mechanisms on time. Enhancing the load-balancing mechanisms, the proposed prediction framework enables (a) to forecast the number of requests that the system will not be able to handle and (b) schedule their relocation to neighbouring nodes that are not saturated. This work is based on the user satisfaction metric: a metric of the network response time, in terms of serving the decision-making requests, defined in our previous work in [3].

The proposed prediction framework enables the prediction of future values of the network response time, employing load-prediction techniques based on the exponential moving average approach presented in [4,5]. More specifically, we consider the load-prediction models applied in Web-based systems [5,6]. The latter are not based directly on resource measures but on the representation of the load behaviour of system resources (load trackers). Such models ensure that not only a limited view of the resources is provided, but also a view of the behavioural trend. The prediction model enables us to forecast future values of the network response time and consequently the user satisfaction metric. Specifically, a two-step approach is employed, as in [4]. Initially, the representation of the resource load conditions is computed based on the measured raw data of the network response time–this is realised by the load-tracker module. Next, the future values of the network response time are forecasted based on a set of load tracker values for the network response time. This is realised by the load-prediction module [4].

The key part of this work is that the predicted values of the user satisfaction are used to proactively trigger the load balancing of the decision-making requests. The fine-tuning and integration of the prediction mechanisms in our load-balancing framework fosters the automation of the load-balancing procedure, in terms of allowing the better utilisation of the system resources and proactively handling to-the-edge saturation cases. The low computational complexity of the prediction schemes yields a lightweight framework for load balancing which enables the optimal proactive management of the decision-making requests.

This article is organised as follows. First, related work in the area of resource management and load balancing in future wireless networks is presented in Section 2. Next, Section 3 analyses the algorithmic framework for prediction-based requests management. The analytical and prediction models that were used for the computation of the user satisfaction are introduced in Section 4. The results of this article are presented in Section 5. Initially, the evaluation and fine-tuning/parameterisation of the prediction model is realised using a single network-node test system. Next, the parameterised prediction model is integrated in the load-balancing framework and applied in a case study system. Finally, conclusion remarks are drawn in Section 6.

## 2. Related study

The dynamic resource allocation as well as the end-users' load-balancing impose significant challenges, which have been the objective of several research activities in the literature, even in the context of mobile networks. The common basis for such activities was that the huge amount of generated user traffic represents a significant stress for every domain of a mobile operator network [7-10]. The proposed solutions have addressed different parts of the network: (a) service domain, including adaptation actions in the application layer (e.g. codec adaptation), (b) backhaul/core network domain, including traffic rerouting through the reconfiguration of Label Switched Paths or Virtual LANs [10] and load distribution by introducing content caches [11] as well as (c) access domain, focusing on actions for the efficient and fair scheduling of the wireless link to the set of mobile user devices. At this point, it should be noted that there is already early standardisation on load balancing in the access network as regards the user traffic (e.g. scheduling, cell load balancing) [7]. In the presence of many users and high traffic demand, this should be applied in collaboration with other base stations. On the access front, means of load balancing of user traffic are network-initiated handover (either within the same RAT or between different RATs and antenna tilt adaptation). Besides, base stations normally also perform admission control and traffic shaping for uplink traffic [12,13]. It should be noted that related work in the area of resource management/load balancing in the access domain has mainly focused on the user plane data and fails to address control signalling.

The proposed system also leverages on the prediction models in [4,6], to realise the prediction of future values of the network response time and consequently the user satisfaction metric. Specifically, Casolari and Andreolini propose an analytical prediction model to forecast future load values of system resources under real-time constraints; such model has been applied for the users' service demands in a web-based system. Several linear and nonlinear prediction models have been considered and tested for their accuracy and precision. For example, the Exponential Weighted Moving Average (EMA) prediction method, which forms a good candidate for modelling the prediction of the system resources in this work, has been discussed in detail in [14].

## 3. Algorithmic framework for prediction-based requests management

In this work, we consider the decision-making requests originating from mobile devices to network nodes, which include dynamic adaptation alternatives (namely handover and protocol reconfiguration) [15]. This work is based on the system model enabling the dynamic adaptation of mobile devices in cognitive radio networks analysed in [1] by introducing prediction schemes for the load balancing of user requests.

To this end, as analysed in [1], two types of physical entities are considered in our model: mobile devices and network nodes. Mobile devices include both reconfigurable and autonomous mobile devices, as analysed in Section 1. On the network side, we focus on the nodes that receive the decision-making requests (e.g. eNB): the latter incorporate decision-making functionality for the requests handling. This work proposes a lightweight algorithmic framework for the efficient management of decision-making requests in a system, enhanced with prediction schemes. The key concept addressed is the dynamic computation of the system capacity in terms of computational resources and the management of the requests that exceed the system capacity. In addition, future values of the system capacity are predicted. To this end, the presented resource management framework allows for the proactive administration of future resources saturation.

Based on our previous work [1], the system capacity is defined as the number of simultaneous decision-making requests that can be handled by the network. Moreover, the system capacity is affected by three parameters: (a) the class of mobile devices (due to different response time requirements per class [1]), (b) the number of reconfiguration decision-making requests and (c) the frequency of reconfiguration decision-making requests. Following the definition in our previous work [1], there is a need for a common metric that will encapsulate the abovementioned requirements and will be used to guide the requests management. To this end, the notion of the user satisfaction metric is used, as a metric of the satisfaction of the user based on the network response time for serving the decision-making requests. Low values of the user satisfaction imply that the system resources are close to saturation and therefore slow serving rate of the decision-making requests; high values of the user satisfaction denote adequate system capacity for handling additional requests [1].

At this point, the basic steps of the algorithm dealing with prediction-based requests management are presented (Figure 1). At first, the user satisfaction degree is dynamically computed during real-time based on network response time measurements, per class of mobile device. Next, the current value of the user satisfaction is compared with the satisfaction threshold: the lowest possible value of the user satisfaction. If the user satisfaction is found to be lower than this threshold, then the requests reallocation procedure is triggered.

In addition, future values of the network response time are forecasted using the prediction framework. At

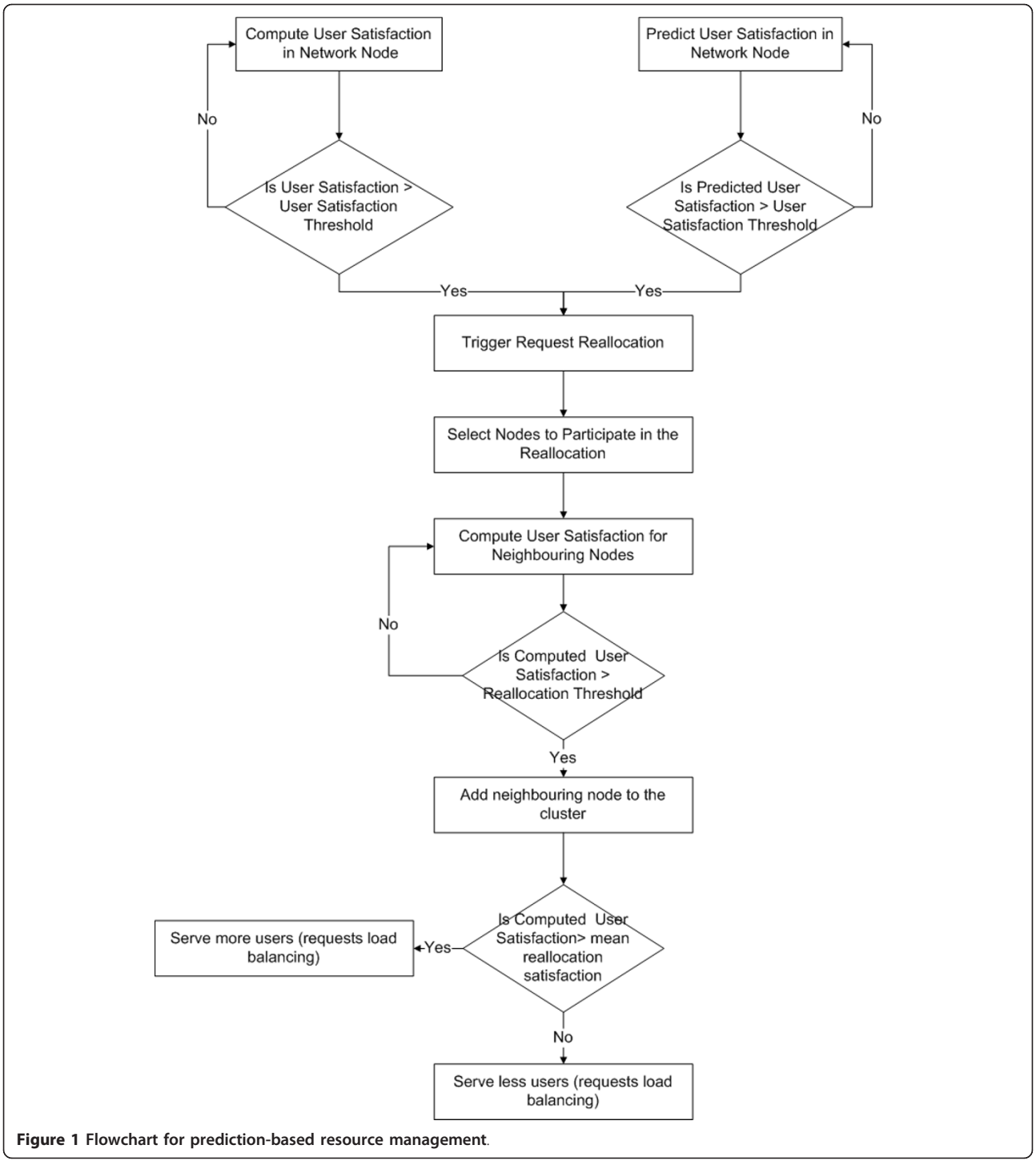


**Figure 1 Flowchart for prediction-based resource management**.

this point, it should be noted that the innovation of this work is the possibility for proactive load balancing based on the predicted value of the user satisfaction. More specifically, if the predicted value of the user satisfaction is found to be lower than the user satisfaction threshold, the requests reallocation is triggered to avoid future saturation of the system computational resources. Next, the requests reallocation phase is initiated. This phase includes two steps:

- The selection of the nodes that will participate in the reallocation procedure.
- The application of the reallocation procedure through realising the requests offloading.

The first step includes a negotiation procedure for the selection of the appropriate neighbouring nodes to participate in the reallocation procedure. The latter is applied using a threshold-based approach. We follow the concept of the reallocation threshold in [1]: the minimum acceptable value of the node's user satisfaction that allows its participation in the reallocation procedure. We consider that the neighbouring nodes with the highest values of user satisfaction that fulfil this policy will participate to the requests reallocation. If no suitable nodes are identified, then this means that certain decision-making requests will have to be dropped.

If a set of suitable neighbouring nodes are identified to participate in the nodes', next the reallocation of the requests will take place. This procedure is realised using a threshold-based approach, based on the mean reallocation satisfaction metric: this is defined as the mean value of the satisfaction degree of the participating nodes in the reallocation procedure [1]. The nodes with lower value than the mean reallocation satisfaction should allocate a percentage of the serving mobile devices to the nodes with higher user satisfaction than the mean reallocation satisfaction. The actual percentage can be either fixed (static) or can be computed in a dynamic manner targeting the fair request reallocation according to the user satisfaction of the nodes.

At this point, we should point out that the proposed scheme enables the management of decision-making requests coming from both reconfigurable and autonomous mobile devices. This lightweight algorithmic framework is to be integrated within the network nodes handling the decision-making requests and is fully autonomous. The network operator should manage the scheduling and the realisation of the resource measurements as well as the definition of the required thresholds for the implementation of the decision-making procedure (e.g. user satisfaction threshold, reallocation threshold).

## 4. Analytical and prediction models for user satisfaction

Based on the proposed algorithmic framework, we need to define the user satisfaction metric as a function of the network response time. As analysed in the previous section, user satisfaction is the key metric for load balancing: if the computed or the predicted value of the user satisfaction is above the user satisfaction threshold, then the load-balancing procedure is triggered. First of all, based on our previous work [1], we define as network response time the response time experienced by a mobile device making a decision-making request to the network side. We differentiate the network response time per class of mobile devices. Therefore, we define the response time of class $c$ $R_c$ as the response time experienced by a class $c$ mobile device making a decision-making request. We also define as user satisfaction $SA_c$, the normalised distance of the network response time $R_c$ from the maximum value of the response time $R_c^{\max}$ to the interval of the maximum response time minus the minimum response time $R_c^{\min}$. Therefore, user satisfaction is analysed as follows, based on the work [1]:

$$SA_c = \frac{R_c^{\max} - R_c}{R_c^{\max} - R_c^{\min}} \tag{1}$$

At this point, it should be noted that the maximum and minimum values of the response time are dependent on the number of the decision-making requests and the average time between requests. Their values can be computed only with iterative techniques which are characterised by high computational complexity. To this end, targeting a lightweight mechanisms for load balancing we compute the bounds of the maximum and minimum response time. Therefore, the approximate value of user satisfaction is given below:

$$\overline{SA_c} = \frac{R_c^{Up} - R_c^{Ms}}{R_c^{Up} - R_c^{L}} \tag{2}$$

In this direction, the computation of user satisfaction requires: (a) to also compute the upper and lower bounds of the network response time and (b) to measure the network response time. As regards the bounds computation, this is realised using the methodology and respective analytically model proposed in our previous studies [1,16,17]. The latter models our system as a closed queuing network considering multiple resources, different classes of devices based on their service request patterns and multiple workload mixes [18].

In a similar manner, we introduce the notion of the predicted value of the user satisfaction, $LSA_c$, which is defined as the normalised distance of the predicted network response time $LR_c$ from the maximum value of the response time $R_c^{\max}$ to the interval of the maximum response time minus the minimum response time $R_c^{\min}$. Therefore, the predicted user satisfaction is analysed as follows:

$$LSA_c = \frac{R_c^{\max} - LR_c}{R_c^{\max} - R_c^{\min}} \tag{3}$$

In addition, the approximate value of predicted user satisfaction is given below:

$$\overline{LSA_c} = \frac{R_c^{Up} - LR_c^{Ms}}{R_c^{Up} - R_c^{L}} \tag{4}$$

From now on in this article, when we refer to the user satisfaction/predicted user satisfaction, the approximate values will be considered.

### 4.1. Prediction model

The computation of the predicted value of the user satisfaction $LSA_c$ requires forecasting the values of future response time, based on existing measurements. To this end, we consider the methodology presented in [4] for load prediction, in order to forecast the future network response time.

More specifically, as analysed in the literature [6], the application of load prediction directly to raw resource measurements fails to capture the behavioural trend of the resource metrics. To this purpose, the application of the prediction to a set of "filtered" data leads to a more representative view of the load conditions, smoothing the resources measurements. This is achieved by using the load tracker functions, which allow filtering the noise and excluding outlier values that correspond to numerically distant observations from the rest of the resource measurements. Next, the prediction is applied using the load tracker values.

As regards the load tracker, both linear and nonlinear load trackers have been investigated in the literature [5,6]. Linear load trackers are based on moving averages. Such trackers are characterised by low computational complexity while achieving a smooth filtering and approximation of the resource measurements. Their disadvantage lies in the introduction of delay in the representation of the load trend, and oscillations when limited set of data are used for resource measurements. This limitation is effectively handled by the nonlinear load trackers; such spline-based approaches allow for a better approximation of the resource load variations. On the other hand, their computational complexity is higher.

Based on the analysis in [4,6], we selected the Exponential Moving Average (EMA) load tracker, to be used for load prediction in this work. The EMA load tracker leads to higher accuracy in the final load prediction compared to other load trackers, while its low computational complexity allows for its incorporation in the presented lightweight framework for resource management.

At this point, the model used for load representation and prediction is analysed; it should be noted that the resource metric that is employed for this model is the network response time. We consider the samples of the network response time $R_{c,i}$ at time $t_i$, and a set of collected $n$ measures denoted as $\overrightarrow{R_{c,n}}(t_i) = (R_{c,i-1}, .., R_{c,i})$. In addition, the load tracker is denoted as a function $LT(\overrightarrow{R_{c,n}}(t_i)) : \Re^n \rightarrow \Re$ that takes as inputs $\overrightarrow{R_{c,n}}(t_i)$ and gives a representation of the resource load conditions $l_i$ at time $t_i$ [4].

In this work, we employ the EMA load tracker, which is the weighted mean of the $n$ resource measures of the set $\overrightarrow{R_{c,n}}(t_i)$, where the weights decrease exponentially. The EMA-based load tracker $LT(\overrightarrow{R_{c,n}}(t_i))$, for each time $t_i$ where $i > n$, is equal to

$$\mathrm{EMA}(\overrightarrow{R_{c,n}}(t_i)) = \alpha * R_{c,i} + (1-\alpha) * \mathrm{EMA}(\overrightarrow{R_{c,n}}(t_{i-1})) \quad (5)$$

where the constant $\alpha = \frac{2}{n+1}$ is the smoothing factor.

In addition, we denote the load prediction as a function $LR_{c,k}(\overrightarrow{L_q}(t_i)) : \Re^q \rightarrow \Re$ which takes as input the set of q values $\overrightarrow{L_q}(t_i) = (l_{i-q}, .., l_i)$ and returns the predicted value at time $t_{i+\kappa}$, where $k > 0$. In this analysis, we consider the load predictor employed in [4], which is based on the linear regression of two available load tracker values. Each predictor is characterised by the following set of values:

- The predicted window $k$ which represents the size of the prediction interval.
- The past time window $q$, where $q$ is the distance between the first $l_{i-q}$ and the last $l_i$ load tracker value.

The load predictor of the load tracker is the line that intersects the two points $(t_{i-q}, l_{i-q})$ and $(t_i, l_i)$, and returns $\hat{l}_{i+\kappa}$ that is the predicted value of the load tracker $l_{i+\kappa}$ at time $t_{i+K}$, which is given by [4]

$$LR_{c,k}(\overrightarrow{L_q}(t_i)) = m * (t_{i+k}) + \alpha \quad (6)$$

where $m = \frac{l_i - l_{i-q}}{q}$ and $a = l_{i-q}\text{-}m * t_{i-q}$.

In addition, the input values $\overrightarrow{L_q}(t_i)$ are given by the output of the EMA load tracker.

Therefore,

$$\overrightarrow{L_q}(t_i) = \mathrm{EMA}(\overrightarrow{R_{c,n}}(t_i)) = \alpha * R_{c,i} + (1-\alpha) * \mathrm{EMA}(\overrightarrow{R_{c,n}}(t_{i-1})) \quad (7)$$

Using (5), Equation (7) is analysed as follows:

$$LR_{c,k}(\alpha * R_{c,i} + (1-\alpha) * \mathrm{EMA}(\overrightarrow{R_{c,n}}(t_{i-1}))) = m * (t_{i+k}) + \alpha \quad (8)$$

After computing the predicted value of the network response time using Equation (8), the predicted value of the user satisfaction can be computed using Equation (4).

## 5. Results

### 5.1. Evaluation of the prediction model

To evaluate the prediction model, we consider its application in a simplified single-node test system which comprises both reconfigurable and autonomous mobile devices and a network node that handles the decision-making requests; such node also incorporates the enhanced functionality for prediction. The simulations were realised using MATLAB Simulink tool. At the starting point of the simulation, the network node serves 600 reconfigurable and another 600 autonomous devices. In order to be in accordance with a real system as regards the dynamic alteration of the network response time, we consider that it follows a gamma distribution. Next, using the presented model we apply the load tracker function to obtain a smoother and more representative view of the load conditions. Finally, we compute the predicted value of the network response time. In addition, during the simulation we are able to compute the delta between the actual value of the network response time and the predicted one.

The scope of the evaluation is threefold:

1. To investigate if the load tracker approaches well the behaviour of the system load in terms of network response time, for the given networking case study.
2. To examine if the predicted values of the response time approach adequately the future values of the load trend.
3. To realise the parameterisation and fine tuning of the presented model, towards the optimum combination of the values for the three following parameters: the number of past values that are considered in the load tracker, the predicted window $k$ and the past time window $q$.

In order to examine the first goal, we need to evaluate the number of past values considered in our system versus the predicted window $k$. Therefore, we evaluate the autocorrelation function (ACF) of the load tracker, which illustrates the correlation between the response time at a given time instance and its predicted value $k$ steps latter. The goal is to find which load tracker is considered predictable; to this end, we evaluate the AC function for the EMA load tracker taking into account multiple numbers of past measures. It is important to mention that a dataset is considered predictable with an adequate accuracy for a prediction window $k$, if the value of its ACF is greater than 0.3 [19].

Figure 2 presents the values of the ACF of the EMA load tracker, for various values of the considered past measures $n$. As shown in the figure, the upper threshold for the acceptable values of the predicted window is $k$ = 20, since for higher values of $k$, the ACF decreases abruptly. Given this threshold and based on numerical values of the ACFs for the different load trackers considered in Figure 2, the results show that the $EMA_{30}$ load tracker ($n$ = 30) is always more predictable than the others–this is shown by its higher ACF value. In addition, the best results are obtain for $k$ = 5 and 10. The same results were obtained for the class of autonomous mobile devices.

As regards the second goal, we evaluate the ability of the prediction model to forecast future values of the network response time using the prediction error metric, as in [4]:

$$\varepsilon_{i+k} = \left| \frac{l_{i+k} - \hat{l}_{i+k}}{l_{i+k}} \right| \tag{9}$$

where $l_{i+k}$ is the actual value of the load tracker at time $t_{i+k}$ and $\hat{l}_{i+k}$ is the predicted value. Next, we evaluate the mean value $\Delta_p$ of the prediction error for the entire observation period:

$$\Delta_P = \frac{\sum_i \varepsilon_{i+k}}{M} \tag{10}$$

where $T$ is the length of the observation period [4].

To this end, we consider that $n$ = 30, and we compute the mean value $\Delta_p$ of the prediction error for $k$ = 10 and 15 since these are among the best values for $EMA_{30}$. In addition, since $k/2 \leq q \leq k$, we consider the following values of $q$, namely $q$ = 5,8,10,15. The results show that the lowest value of the prediction error is obtained for $q$ = 10 and $k$ = 10, when considering the requests coming from reconfigurable mobile devices (Table 1).

Similar results were obtained for the autonomous mobile devices class; again the lowest value of the prediction error was obtained for $q$ = 10 and $k$ = 10.

#### 5.1.1. Case study for prediction-based load balancing

Following the evaluation and fine tuning of the prediction model, we consider its integration in the load-balancing framework and its application in a case study system. The target application environment considered for the simulations is an LTE environment. The results were derived through MATLAB simulations using the MATLAB Simulink toolbox. The reason for choosing MATLAB for the simulation of the algorithm is that MATLAB offers a generic simulation environment, allowing for greater freedom degrees compared to the restricted configurations that another platform, such as OPNET, NS-2, would impose, e.g. in terms of simulating-specific architecture. Using MATLAB Simulink, we

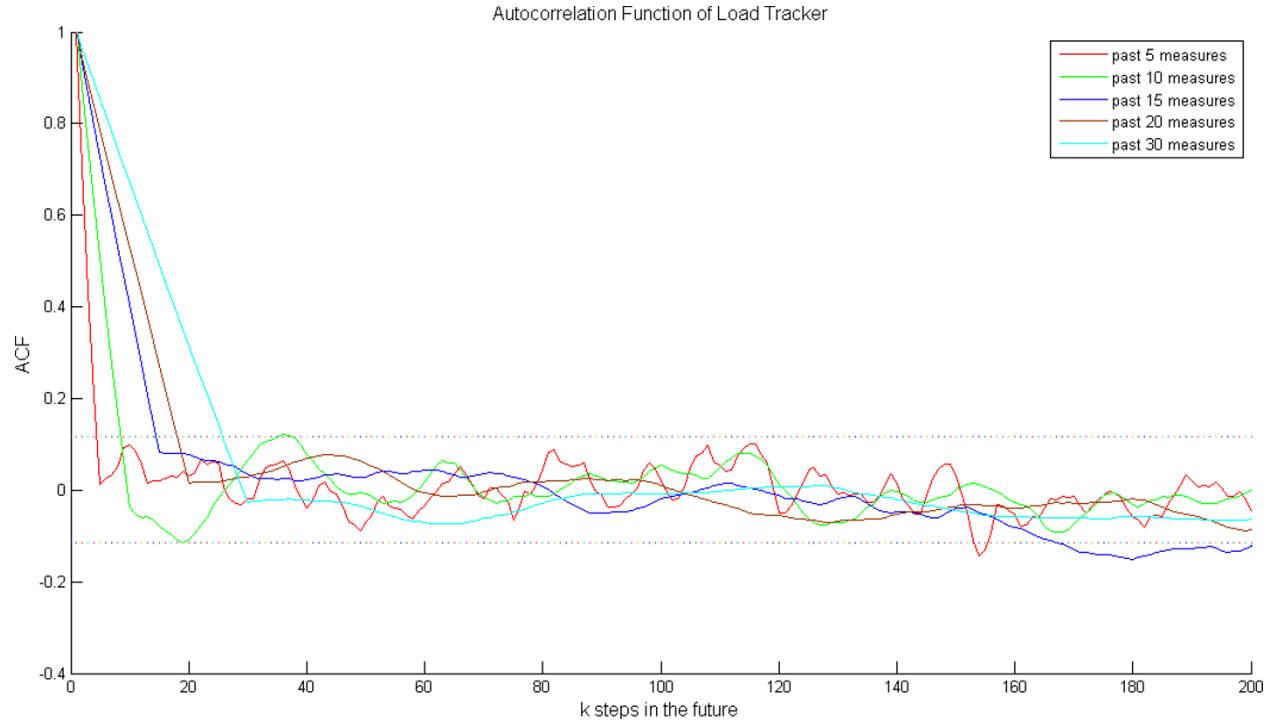


**Figure 2** ACF of the EMA load tracker for various values of *n*.

can ignore the delay arising from the cross-layer communication. Thus, the measurement of the network response time consists only of the time required for processing the mobile device request, thereby considering out-of-band control traffic. The cross-layer communication quantity is more or less constant for each network architecture and it can easily be incorporated when evaluating the performance of a real system.

The target application system consists of four network nodes, simulating the eNodeBs in an LTE architecture that manages the decision-making requests originating from mobile devices. The mobile devices consist of both reconfigurable and autonomous devices. Four separate load-balancing systems take place in the procedure. The first two load-balancing systems handle the decision-making request per class, while the other two systems handle the decision-making requests arising from the prediction modules residing in the network nodes.

**Table 1 Mean value of the prediction error for different combinations of *n*, *k*, and *q***

| *n* | *k* | *q* | $\Delta_p$ |
|---|---|---|---|
| 30 | 10 | 5 | 0.065 |
| 30 | 10 | 8 | 0.055 |
| 30 | 10 | 10 | 0.048 |
| 30 | 15 | 8 | 0.086 |
| 30 | 15 | 10 | 0.076 |
| 30 | 15 | 15 | 0.062 |

At this stage, the results of this work focus on the application and the evaluation of the load-prediction mechanisms in the load-balancing framework. At this point, we would like to clarify the assumptions on the simulation system setup, networking and other environmental factors. First of all, in order to approach the behaviour of a real system as regards the dynamic alteration of the network response time, we consider that it follows a gamma distribution. In addition, the upper and lower values of the network response time depend upon the time interval between two handover or reconfiguration decision requests, defined as the think time *z*. More details on the computation of these parameters are available in [1].

In this work, we considered three different simulation scenarios. At the first simulation scenario, we assume that the network nodes serve 600 reconfigurable and another 600 autonomous devices. Simulations were also executed with a population mix of 1,000 reconfigurable and 200 autonomous devices (second simulation scenario). Finally, the third scenario included a single-class population of 1,200 reconfigurable devices. The reason why we conducted additional experiments increasing the number of reconfigurable devices and decreasing the one of autonomous is the following: reconfigurable devices tend to impose more load to the network (as proved in [1]), so it

is critical to investigate what happens at the edge of the considered population mix, when only reconfigurable devices exist in the system.

Following the algorithm presented in Section 3, the user satisfaction degree is dynamically computed per class of mobile devices based on the actual network response time for serving the decision-making request.[a] Two values of the user satisfaction are evaluated: the actual value and the predicted value. The latter is computed based on the predicted value of the network response time. At first, the load tracker functions have been applied, using the $EMA_{30}$ load tracker; next, the load prediction functions take place based on the outputs of the load tracker. Such analysis was realised for both types of mobile devices.

It should be noted that we consider the optimal values for $n$, $k$ and $q$ that were derived from the evaluation of the prediction model ($n$ = 30, $k$ = 10, $q$ = 10) in the single-node test system. Next, for the presented multi-node case study system, we have computed the mean value of the prediction error using Equation (7). The results show that $\Delta_p$ = 0.0048, as in the single-node test system (third row of Table 1). Such outcome shows that the prediction of the response time approaches very well the respective load tracker value. Therefore, the introduction of the prediction functionality in the load balancing of the decision-making requests is expected to enable the proactive management of such requests, improving the network management procedure.

Next, after computing the predicted value of the network response time, we derived the actual and future values of the user satisfaction for each class of mobile devices. If one of these values is found to be less than the user satisfaction threshold, then the reallocation procedure takes place. For this case study, we consider that the user satisfaction threshold is equal to 0.075. In addition, we assume that the reallocation threshold is equal to 0.3. Such threshold indicates that nodes with user satisfaction lower than this value cannot participate in the load-balancing procedure. During such procedure, some devices could not be reallocated, so they will be dropped. At this point, we would like to point out that the assumption of efficient network access coverage of our system: there exists no energy gaps and the reallocation can be applied in the considered nodes. Figures 3 and 4 show the variation of the total number of mobile devices, both with and without the prediction module.

As seen in Figures 3 and 4 depict the simulation results for the reconfigurable and autonomous devices, respectively; the application of the prediction system in the load-balancing framework results to dropping less mobile devices requests than that of the initial load-balancing framework. As regards the reconfigurable mobile devices, for the first two experiments the percentage of dropped devices is almost the same. According to the first simulation scenario for the prediction-based load-balancing system, 1.8% of the reallocation requests are dropped, corresponding to 11 devices from the initial 600 (Figure 3a), while at the second one 1.9% are dropped, corresponding to 19 from the overall 1,000 devices (Figure 3b). It should be noted that the load-balancing framework without the prediction mechanism results to dropping more devices: 3% for the first scenario, corresponding to 18 of the 600 devices and 3.1% for the second, corresponding to 31 dropped devices from the 1000. For the third simulation scenario where the population mix is composed only of reconfigurable devices, the prediction model results in better improvement compared to the previous two scenarios (Figure 3c). Specifically, only 15 from the 1,200 devices could not be handled (1.25%), while without the prediction model 39 devices were dropped (3.25%). In this case, the absence of autonomous devices alleviated the system from additional control signalling and thus it performed better, resulting to an increased improvement of the overall performance.

For the case of autonomous mobile devices, the respective failure rate is 9.3% of the mobile devices (56/600) requests for the first simulation and 8.5% (17/200 devices are dropped) for the second simulation. The corresponding failure rates for the load-balancing framework without the prediction mechanism are 11.1% (67/600) and 10.5% (21/200), respectively, for the two simulation scenarios (Figure 4).

To this end, we come to the conclusion that the prediction-based framework optimises the failure rates for the autonomous mobile devices, since the improvement in the respective number of dropped requests is slightly better than the one for the reconfigurable mobile devices. Such conclusion reveals that the introduction of intelligence and proactivity in the mobile devices eases the network management of the decision-making requests. This is due to the fact that the prediction and proactive management operations are more effective for autonomous mobile devices rather than for the reconfigurable ones. Another important conclusion extracted from the above is that the performance of the load-balancing mechanism is not affected by the number of the reconfigurable and the autonomous devices that each network node handles, with an exception for the edges of the population mix. The first two simulations proved that for each case the percentage of the dropped devices is almost the same for each class of mobile device. At this point, we should mention that the involvement of "dummy" devices, namely legacy devices with no intelligence or self-$x$ mechanisms, does not affect the load-balancing mechanism for the two classes of "intelligent" devices. Although they consume system resources, such "dummy" devices do not produce decision-making requests and cannot trigger the reallocation

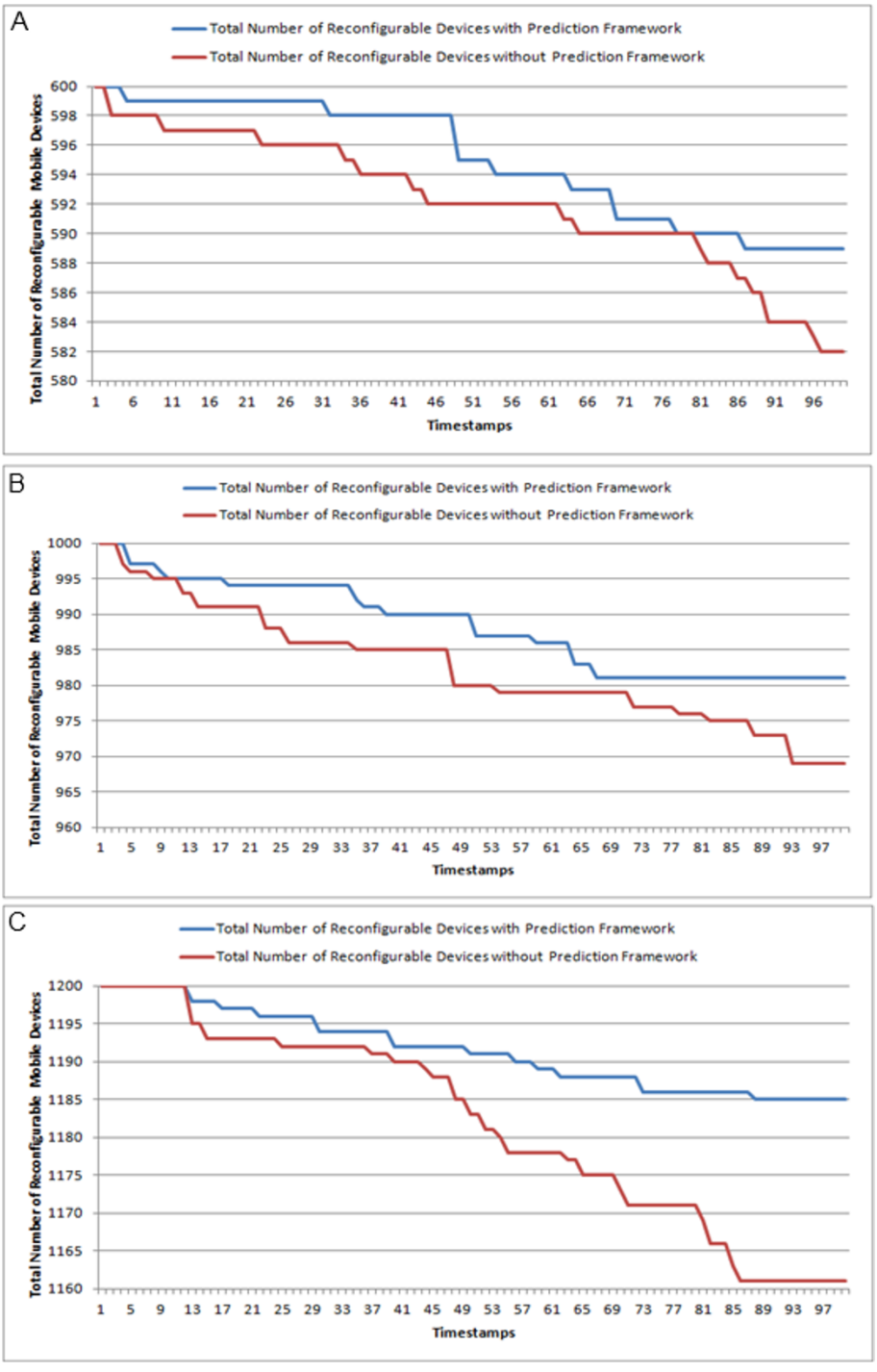


**Figure 3 Total number of reconfigurable mobile devices with and without the application of the prediction framework in the load balancing system for three different simulation scenarios**. **(a)** Population mix of 600 reconfigurable and 600 autonomous mobile devices, **(b)** population mix of 1,000 reconfigurable and 200 autonomous mobile devices, and **(c)** population mix of 1,200 reconfigurable and 0 autonomous mobile devices.

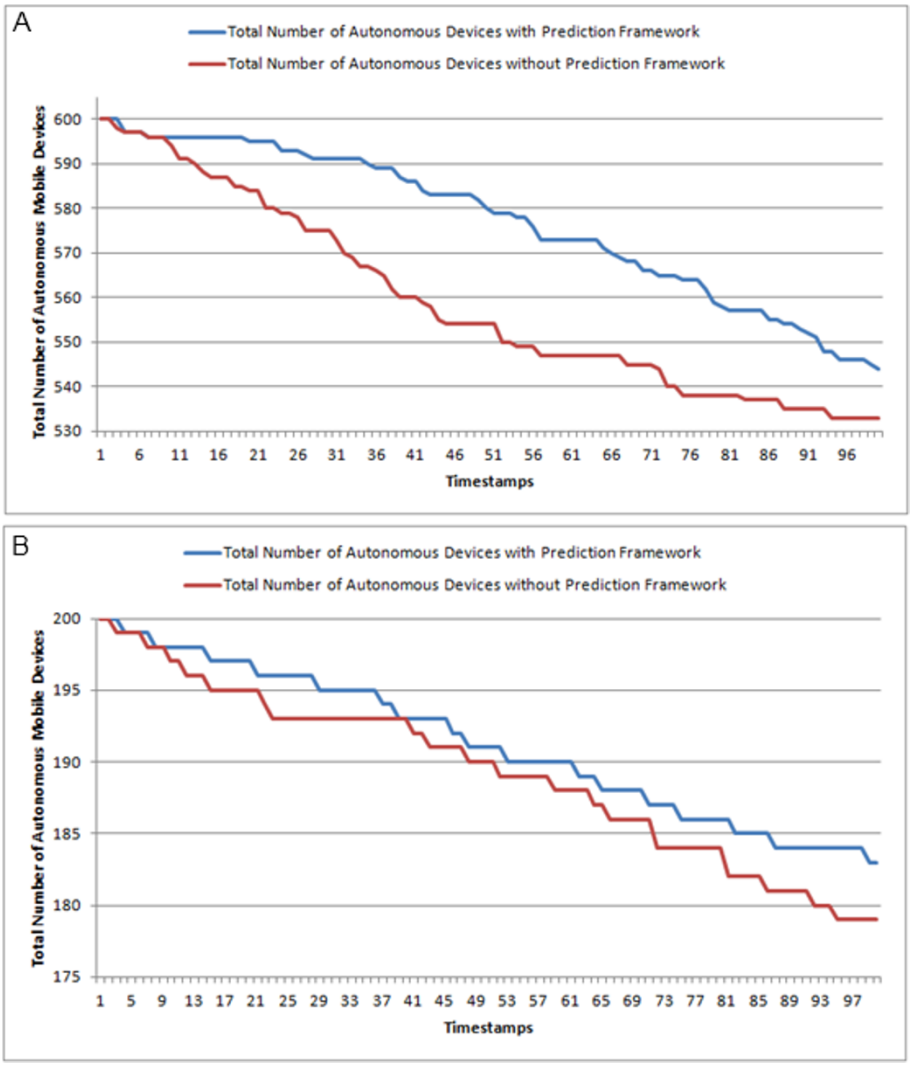


**Figure 4 Total number of autonomous mobile devices with and without the application of the prediction framework in the load balancing system**. **(a)** Population mix of 600 reconfigurable and 600 autonomous mobile devices, **(b)** Population mix of 1,000 reconfigurable and 200 autonomous mobile devices.

procedure; therefore, they are not considered as sources of control-plane signalling. In conclusion, the load-balancing procedure for the autonomous and reconfigurable devices is not affected at all from the existence of legacy devices.

## 6. Conclusions

In this article, we have discussed an advanced, proactive resource management framework to effectively handle the burden of the network resource management. We have analysed the administration of the users' decision-making requests according to their originating class, considering requests coming from reconfigurable and autonomous mobile devices. The main contribution of this article is the introduction of a novel algorithmic framework for load balancing of user decision-making requests, integrating prediction schemes. This framework is based on the actual and predicted values of the user satisfaction: a metric of the satisfaction of the user based on the network response time for serving its requests. In addition, in order to forecast the future values of the user satisfaction, we employed and fine-tuned the prediction models proposed by Andreolini. The latter are not based directly on the resource measures but on the representation of the load behaviour of system resources (load trackers). Such models ensure that not only a limited view of the resources is provided, but also a view of the behavioural trend. In our analysis, we evaluated the need to employ load trackers to smoothen the behaviour of the load trend of the network response time. The $EMA_{30}$ load tracker was selected and appropriately parameterised. Next, the future values of the

network response time have been forecasted based on a set of load tracker values for the network response time. The key part of this work is that the predicted values of the user satisfaction are used to proactively trigger the load balancing of the decision-making requests, in order to avoid the saturation of the computational resources. As regards the requests reallocation, the neighbouring nodes are considered, taking into account the utilisation status of their own resources. Results demonstrate the gains of the introduced system in terms of efficient resource management. In detail, it is proved that the prediction-based proactive reallocation of system requests minimises the requests failure rates. This outcome is confirmed for both reconfigurable and autonomous mobile devices. Better optimisation is achieved for autonomous mobile devices. This is a very important outcome that proves in practice the theoretical assumption that autonomic systems pose less administration burden in the network. Next steps of this work include the introduction of traffic prioritisation patterns [20], using policy rules.

## Endnote

[a]In order to be in accordance with a real system as regards the dynamic alteration of the network response time, we consider that it follows a gamma distribution.

**Competing interests**
The authors declare that they have no competing interests.

Received: 15 September 2011 Accepted: 17 April 2012
Published: 17 April 2012

**References**

1. E Patouni, N Alonistioti, L Merakos, Cognitive decision making for reconfiguration in heterogeneous radio network environments. IEEE Trans Veh Technol. **59**(4), 1887–1990 (2010)
2. P Chatzimisios, Ch Verikoukis, I Santamaria, M Laddomada, O Hoffmann, *Mobile Lightweight Wireless Systems* (2010). ISBN: 978-3-642-16643-3
3. E Patouni, N Alonistioti, Lightweight mechanisms for self-configuring protocols, in *the Proceedings of the 2nd International Conference on Mobile Lightweight Wireless Systems (Mobilight 2010)*, (Barcelona, Spain, 10–12 May 2010), vol. 45 (Springer 2010) pp. 112–123
4. M Andreolini, S Casolari, M Colajanni, Models and framework for supporting run-time decisions in web-based systems. ACM Trans Web. **2**(3), 1–43 (2008)
5. R Lancellotti, M Andreolini, C Canali, M Colajanni, Dynamic request management algorithms for web-based services in cloud computing, in *Proc of the IEEE Computer Software and Application Conference (COMPSAC 11)*, Munich, Germany, 401–406 (July 2011)
6. M Andreolini, S Casolari, M Colajanni, Load prediction models in web-based systems, in *VALUETOOLS*, Pisa (2006)
7. W Song, W Zhuang, Multi-service load sharing for resource management in the cellular/WLAN integrated network. IEEE Trans Wirel Commun. **8**(2), 725–735 (2009)
8. W Song, W Zhuang, Y Cheng, Load balancing for cellular/WLAN integrated network. IEEE Netw. **21**(1), 27–33 (2007)
9. SKD Kandula, S Sinha, A Berger, Flare: responsive load balancing without packet reordering. ACM Comput Commun Rev. **37**(2), 51–62 (2007). doi:10.1145/1232919.1232925
10. T Wauters, J Coppens, B Dhoedt, P Demeester, Load balancing through efficient distributed content placement, in *EuroNGI Conference on Next Generation Internet Networks - Traffic Engineering*, Rome, Italy, 99–105 (2005)
11. R Bolla, R Bruschi, A Cianfrani, M Listanti, Enabling backbone networks to sleep. IEEE Netw Mag. **25**(2), 26–31 (2011)
12. F Yu, V Krishnamurthy, Efficient radio resource management in integrated WLAN/CDMA mobile networks. Springer Telecommun Sys. **30**(1-3), 177–192 (2005)
13. CC Holt, Forecasting seasonal and trends by exponentially weighted moving averages. Elsevier Int J Forecast. **20**, 5–10 (2004)
14. E Patouni, S Gault, M Muck, N Alonistioti, K Kominaki, Advanced reconfiguration framework based on game theoretical techniques in autonomic communication systems. Annal Telecommun J. **62**(9-10), 1099–1120 (2007)
15. M Litoiu, J Rolia, Object allocation for distributed applications with complex workloads, in *11th Int Conf Comput Perform Eval.:Modelling Tech Tools, London,1786*, 25–39 (2000)
16. G Balbo, G Serazzi, Asymptotic analysis of multiclass closed queueing networks: multiple bottlenecks. Perform Eval J. **30**(3), 115–152 (1997). doi:10.1016/S0166-5316(97)00005-9
17. M Litoiu, A performance analysis method for autonomic computing systems. ACM Trans Auton Adapt Syst. **2**(1) (2007)
18. P Brockwell, RA Davis, *Introduction to Time Series and Forecasting*, (Springer, New York, 2001)
19. A Antonopoulos, C Verikoukis, Traffic-aware connection admission control scheme for broadband mobile systems. IEEE Commun Lett. **14**(8), 719–721 (2010). ISSN 1089-7798

doi:10.1186/1687-1499-2012-144
**Cite this article as:** Patouni *et al.*: A lightweight framework for prediction-based resource management in future wireless networks. *EURASIP Journal on Wireless Communications and Networking* 2012 **2012**:144.