Telecommun Syst
DOI 10.1007/s11235-011-9473-9

Improving quality of experience in wireless VoIP through novel call scheduling

Lambros Lambrinos · Constantinos Djouvas

Abstract In the last few years we have experienced a dramatic increase in the use of IP networks for voice applications (VoIP) over wireless networks due to increased bandwidth availability and enhanced device capabilities. Since demand often exceeds available capacity, Call Admission Control mechanisms are in place to prevent the uncontrolled usage of bandwidth. Through the use of an intermediary gateway, VoIP calls are in many cases terminated to a normal landline or cellphone; the capacity of such a gateway is also a finite resource since the number of users can vary significantly as many are mobile. In this article we propose an enhanced scheme that aims to manage access to the lines available so that they are used in a fair manner and utilized to the highest degree possible. This management is facilitated by enhancing a proxy implementation with a number of call scheduling policies. The ability to satisfy pending call requests as soon as lines become available, results in increased user service satisfaction. Moreover, it increases line utilization which is crucial from an economic viewpoint. The ultimate goal is to improve Quality of Experience which is deemed as highly important especially considering that wireless network users experience opportunistic and intermittent connectivity.

Keywords Call admission control · Voice over IP · Call scheduling · Quality of service · Quality of experience · Line utilisation

L. Lambrinos (✉) · C. Djouvas
Department of Communication and Internet Studies, Cyprus University of Technology, 31 Arch. Makarios Avenue, Limassol, Cyprus
e-mail: lambros.lambrinos@cut.ac.cy

C. Djouvas
e-mail: costas.tziouvas@cut.ac.cy

1 Introduction

Computer networks were originally designed to facilitate the sharing of data and computer resources [29] and were suitable for a limited number of applications with specific characteristics. Their unsurpassed success is somewhat due to their use by a multitude of applications with varying characteristics and requirements and in areas that researchers were not able to predict in advance.

One application that has drawn extensive research and commercial interest in recent years involves the exchange of multimedia data in real time. The timely delivery requirements of real-time traffic impose strict bandwidth and delay requirements to the underlying network. In order to meet these requirements different approaches have been proposed, aiming to manage finite network resources, with network bandwidth being perhaps the most crucial. The management of finite resources is of high importance considering that the performance of real-time applications can not be predicted especially in cases where the sharing of vital resources among multiple applications occurs in an uncontrolled manner.

In this article we focus particularly on the transport of voice data over IP networks (VoIP). An approach that has long been advocated for managing network bandwidth by controlling the number of simultaneous calls is Call Admission Control (CAC), and within this scope different schemes have been proposed [7, 9, 21]. Despite the different strategies used, the underlying principle is the same: the traffic of a new call is admitted only if enough bandwidth is available, so that the new admission will not affect the quality of ongoing calls.

In addition, the high demand for ubiquitous access to network services has caused a rapid expansion of wireless networks to complement existing wired networks. Roaming

Published online: 29 June 2011

users utilize these new advances in wireless networks for different purposes, one of which is making VoIP calls to the Public Switched Telephone Network (PSTN). Such a service is offered thought the use of a VoIP-to-PSTN gateway; in this context, such a gateway may provide PSTN connectivity to a large number of users who may be independently accessing the service through various network paths.

The unique characteristics of wireless networks and their highly unpredictable nature mandate the development of more effective and efficient CAC techniques. The need for novel CAC techniques is even more apparent in wireless environments; this is especially the case in some types of wireless networks (e.g., mesh networks), in which packets have to travel through many hops in order to reach an access point. Moreover, another resource which is also deemed as finite is the number of PSTN termination lines available via the gateway which as expected has a given capacity.

In this work, we propose a Call Admission Control mechanism that uses the number of available lines at the VoIP to PSTN gateway as a criterion taken into account for the admission of a new call. We concentrate on the scenario where "network-level" CAC schemes "guarantee" that bandwidth is available at the IP network (and hence allow the call) but all lines that interconnect to the PSTN are busy. We pay particular attention to periods where the gateway lines are saturated and introduce the application of different novel policies suitable for scheduling call requests. Through the use of such policies we expect to enhance Quality of Experience (QoE); instead of the standard call blocking method (i.e. request rejection) scheduling mechanisms will enable call requests to be accepted with a slight delay. It is important to note that our results show that applying our policies increases overall line utilization.

The rest of the article is organized as follows: Sect. 2 provides an overview of existing CAC mechanisms and in Sect. 3 the rationale behind the design of our call scheduling scheme is described; Sect. 4 presents the overall architecture of the system whereas Sect. 5 describes the simulations carried out and presents the evaluation of our findings. Finally, in Sect. 6 we present our conclusions and future work considerations.

2 Background

Call Admission Control is the dominant strategy currently used to manage bandwidth resources by controlling the number of calls allowed over an IP network. Throughout the years, different CAC techniques have been proposed which can be grouped into two main categories: statistical-based and reservation-based.

Statistical-based approaches [10, 14, 20, 21] try to estimate the current status of the underlying network and use this estimation to decide upon the acceptance or rejection of a new call. The generic method involves probing the network with a set of messages and then calculating the packet loss ratio, the delay, and the delay jitter. Based on these statistical measurements, the congestion level in the network is estimated; if it lies below a given threshold the call is accepted otherwise the call is rejected. Generally, statistical-based approaches are simple to implement, as they only require minimal or no changes to the current infrastructure. However, they do not provide absolute guarantees since they (a) rely on heuristics for estimating the bandwidth availability which are not always accurate and (b) do not perform any kind of bandwidth reservation.

The main alternative category of approaches for providing QoS in VoIP applications is based on resource reservation. Proposed solutions in this category [13, 16, 28] follow the same principle as traditional telephony: each call has its own dedicated end-to-end reserved bandwidth along a specific path. The simplest approaches for resource reservation utilize RSVP [6] to reserve an end-to-end path before accepting a new call. Clearly, this approach offers absolute guaranties since a designated path is reserved for each call; however, such guaranties are offered at the expense of network utilization. Moreover, to facilitate resource reservation it is necessary to have additional equipment in place that is responsible for setting up and maintaining reserved paths is required.

The generic CAC approaches just described, are complemented by CAC techniques that were designed for specific environments. One such environment that exhibits some unique characteristics is wireless networking, wherein different approaches have been proposed. The majority of the approaches proposed for wireless networks rely on the fact that traffic in a wireless environment is handled by a single entity, the Access Point (AP); they aim to improve QoS by enhancing voice quality inside the wireless LAN. One such approach is described in [21] and is based on statistical CAC. The mechanism proposed divides the probing phase into two distinct stages: (a) probing inside the wireless LAN (from the wireless device to the AP) and (b) probing the rest of the network (from the AP to the final destination). Thus, distinct strategies can be used in each phase, optimizing the overall performance of the network.

In addition to CAC techniques that were derived from approaches proposed for wired networks, there are techniques specifically designed for wireless networks. Researchers in [31] suggest the integration of SIP and 802.11e to conduct resource reservation as a means of CAC and thus support QoS for VoIP in IEEE 802.11e WLANs. A different approach is proposed in [33], where the responsibility for CAC was shifted to the wireless clients; by observing the network, clients estimate the load on the AP and then perform CAC accordingly.

Other approaches provide CAC through the utilization of a model that allows them to estimate the impact of a new call to the wireless network. Such an example is a model that estimates the voice capacity in an IEEE 802.11b WLAN [7, 29]. Utilizing this model, CAC can maintain voice quality by managing network capacity.

CAC techniques applicable to specific categories of wireless networks also exist in the literature. A type of wireless network that exhibits some unique characteristics that make it more challenging is the increasingly popular wireless mesh network. An effect that has an adverse result on network capacity in wireless mesh networks is interference; different CAC solutions have been particularly devised to help alleviate this effect. Utilizing graph theory, the CAC scheme proposed in [8] admits calls on the basis of reducing the number of edges (wireless nodes) that interfere with each other. A similar approach is proposed in [17], where researchers developed a model that approximates the mutual interference between wireless nodes. The CAC scheme proposed in [30] is based on a model that can be used for estimating the number of calls a wireless mesh network can tolerate given the number of nodes transmitting and the number of networks that a message has to traverse.

The fact that users of wireless networks are inherently mobile, imposes an additional challenge on CAC techniques specifically developed for wireless networks: handovers between access points. The general principle followed in addressing this issue is to favor handoff calls against other calls. A very simple approach is described in [26], where researchers suggest the assignment of higher priority to handoff calls. A different and more sophisticated approach is proposed in [23]; a designated mobile agent migrates to the new AP before the actual client enters its coverage area and reserves bandwidth on behalf of the client based on its needs. A number of approaches (known as "thinning schemes") aim to reduce the probability of handoff failures. One such approach is described in [27] where the model proposed selects the admission probability to provide fairness among new calls and handoff calls.

Finally, a number of approaches propose joint scheduling and call admission control policies. In [12] and [19] a data scheduler is used to control the load of delay-tolerant traffic in favour of real-time traffic and thus maintain the total load the network can tolerate. An adaptive scheduling scheme that allocates an optimum rate to each traffic queue in order to minimize scheduling delay is proposed in [22]. Finally in [32], CAC is used for improving the network's throughput, while the scheduler minimizes the difference of individual delay performance among admitted flows.

Based on the above discussion, it is clear that most applications of CAC use bandwidth availability as a means of controlling the admission of new calls. To the best of our knowledge, there is no CAC scheme that utilizes scheduling for optimizing the process of selecting the most appropriate call to be accepted in order to improve overall user satisfaction. Hence, we focus our work in this area and concentrate on a resource which is also sometimes deemed as finite; the number of lines available when a gateway is used to terminate calls to the PSTN. A single gateway may provide PSTN connectivity to a large number of users who may be independently accessing the service through various network paths. Such users are likely to be moving so demand is unpredictable. Thus, efficient line management can play a vital role in satisfying such varying demand, increasing service satisfaction as well as optimizing line utilisation.

3 Scheduling-based call admission control

The different CAC approaches described above, along with numerous others that have been proposed over the years, aim to improve QoS for real time traffic by either predicting the state of the underlying network or by reserving bandwidth. In this work, we consider a different resource which is sometimes deemed as finite and previous CAC schemes do not consider: the number of lines available when a gateway is used to terminate calls to the PSTN. To enhance existing solutions, we use the number of available lines at a VoIP to PSTN gateway as an additional criterion taken into account in the call admission decision process. Moreover, during line saturation periods we propose the use of a scheduler to optimize the process of selecting the "most appropriate" call request to be admitted from a list of pending requests.

Different scheduling policies can be defined and used, facilitating adaptive schemes. We aim to improve user Quality of Experience (QoE) which depends to a large extent to the provision of QoS but also imposes its own requirements on a system. Assuming that QoS is more or less guaranteed once a call is allowed to go through (using network-level CAC mechanisms) we want to concentrate on devising techniques that contribute towards enhancing QoE; such techniques can define extendable and adjustable strategies for handling new call requests with respect to the number of outgoing lines and the state of users. Adaptive schemes can assist in providing more efficient services for end users, thereby improving their QoE.

One potential strategy can be implemented in the form of distinct policies, responsible for scheduling call requests in effective and efficient ways. These scheduling policies will be applied to call requests that are currently in the "pending calls" queue and will be responsible for assigning different priorities to each call; the goal is to optimize the utilization of termination lines in a fair way with QoE in mind.

Our work proposes a combined Call Admission and Scheduling scheme that can be applied to wireless environments. Its aim is the optimization of different aspects of the

CAC process in order to improve end user satisfaction. We assume that a single VoIP-to-PSTN gateway is used to terminate VoIP-based calls initiated by clients that reside inside a wireless LAN to a PSTN telephone. The wireless LAN consists of a collection of access points (APs), forming a single wireless network. Some of the users within the range of the wireless network attempt to make a VoIP-to-PSTN call. Under normal circumstances, if PSTN connectivity is saturated (i.e., outgoing lines are busy), new call requests will be rejected. To avoid this, we suggest that unsatisfied call requests are placed in a queue and get processed when lines become available. Moreover, the definition and utilization of different policies for selecting calls from the queue (e.g. prioritisation) will result in more effective processing of the pending calls, something that is expected to improve user satisfaction.

In earlier work [18], we defined and implemented two distinct scheduling policies. The first one (defined as "Order Received") gives higher priority to call requests waiting longer in the queue, i.e., the request received first will be processed first. This scheduling policy is simple to implement and apply since it does not require additional information about the requesting user's location or call history. The second policy we defined (the "First Exit" policy), assumes that the scheduler can accurately predict the time that a user will exit the coverage area of the network (and hence lose service). Utilizing this information, priorities are assigned based on the estimated user's exit time; a call request from a user that is expected to exit first has the highest priority.

However, after further analysis of the "First Exit" policy and simulating more complicated scenarios, we realized that it is extremely difficult, if not unrealistic, to assume that a client's exit time can be accurately predicted by the scheduler. For this particular reason, in this work, we defined a new scheduling policy (named as "Further Away") that replaces the "First Exit" scheduling policy. The new policy gives higher priority to users that are further away from the center of the network, i.e., close to its boundaries. Thus, users that have just entered or are about to exit the coverage area of the network have the highest priority for establishing a call; users are more likely to place a call once they gain access to a calling service or are about to lose such access.

To implement the new policy, we assume that a client can accurately estimate its current position (e.g. utilizing GPS, signal strength or a combination) and that the client includes this information (i.e., its current location) in an INVITE message. Assuming that the scheduler knows both the area that the wireless network covers and the current position of a client, it can prioritise calls based on the client's proximity to the network's border. Clearly, the assumptions of the new scheduling policy can be easily satisfied and implemented. Moreover, the scheduler is simplified since the volume and complexity of the data it must maintain and process in order to calculate the user's position is low.

In addition to the aforementioned policies, in this paper we introduce another simple and effective policy that contributes towards a fairer usage of resources. The newly defined policy aims to simplify the scheduling process, reduce the information required by the scheduler, remain completely transparent to the end user, and maintain the high satisfaction rate offered by other scheduling policies. This policy only requires information on the number of calls a user has established. Upon the registration of a new user, the scheduler creates a designated counter for maintaining this information; this counter remains active until the exit of the user from the coverage area of the network. We assume that our users are "on-the-move" and will not stay in the coverage area for a large period of time. Potentially, a sliding window mechanism could be applied to remove a user's "old" calls.

Utilizing historical information on outgoing calls, the scheduler gives higher priority to call requests from users that have made the fewest calls. In contrast to the previous scheduling policy where users indirectly participate in the queuing process by providing their current position, this policy is completely transparent to the end user, requiring no changes to the SIP User Agent. Depicting its rule of operation, this policy is named as: "Fewer Calls".

All the policies defined are further explained in the following section where we describe the enhancements made to a SIP proxy in order to facilitate the provisioning of CAC and call request scheduling services.

4 System architecture and major entities

The system architecture (Fig. 1) comprises of three major entities: the user agents (calling clients), a SIP proxy and a PSTN gateway. The role of each is explained in the sections below along with a detailed explanation of the system's operation with respect to the call establishment procedure.

4.1 User agent

The user agent is a normal SIP client (e.g. a softphone running on a mobile device) that includes functionality for establishing, accepting and terminating calls. To fully take advantage of the potential functionality offered by our system, the SIP client has to be slightly modified. This is because the scheduler when using the "Further Away" policy requires the current location of the user in order to assign priorities to call requests. Thus, in cases were this policy is used, the client should include its current position in the INVITE message. The method used by the client for acquiring this information, is beyond the scope of this work.

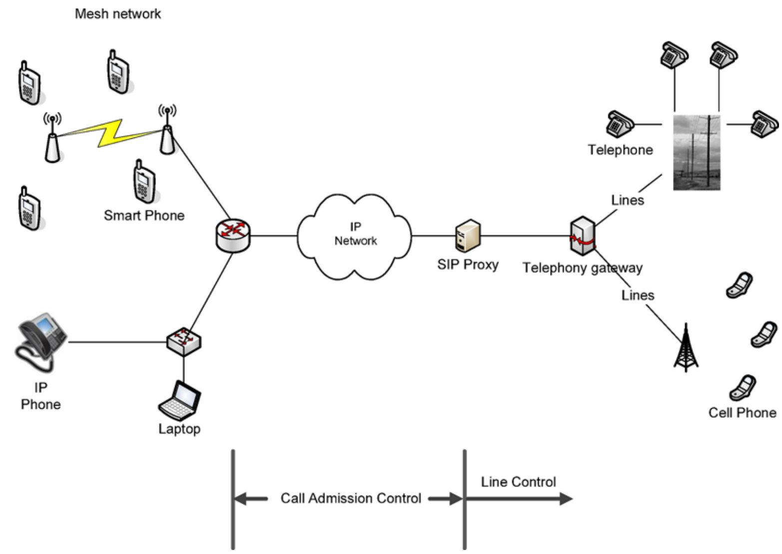

Fig. 1 Architecture for PSTN access for VoIP users

4.2 Call admission control SIP proxy

The core of our system is the Call Admission Control SIP Proxy (CACSP). The CACSP is an enhanced SIP proxy that incorporates CAC and scheduling functionality. The extended functionality enables the proxy to act in an intelligent manner and perform call request scheduling when no lines are available. Currently, four distinct policies are implemented:

1. *Order Received*: Prioritises call requests according to the order they have been received.
2. *Fewer Calls*: Orders call requests based on the number of calls the user has already made; requests from users with the fewest calls have higher priority.
3. *Further Away*: Utilises the current position of the client and gives higher priority to call requests from users who are furthest away from the center of the network (i.e., closer to its borderline and leaving or have just entered the network).
4. *No Scheduling*: This is the standard scheme that uses no scheduling; when no line is available the call is dropped.

For implementing the above policies we extended the functionality and the data structures of a SIP proxy; the amended proxy stores information relating to its users and their calls in two distinct lists. The information in each list, consists of data that is necessary to uniquely identify each client and its calls, but also includes some statistical information regarding user's attempted and completed calls. More specifically, the lists maintained by the SIP proxy are the following:

1. *In-Range Users List*: Under normal circumstances, users are included in this list once they are within range of an access point. These users can potentially initiate a call by sending an INVITE message; however, they must first register to the SIP proxy by sending a SIP REGISTER message.
2. *Pending Calls List*: Contains all call requests (i.e. INVITE messages) that couldn't be satisfied in the first place and are currently pending. The ordering of this list is controlled by the scheduling policy that is currently in use. As expected, when the "No Scheduling" policy is used this list remains empty.

In addition to the above data structures, we extended the functionality of the SIP proxy so that it can handle received SIP messages in a special manner; the SIP INVITE, BYE, and CANCEL messages affect the content of the queues and are hence highly important for the correct operation of our mechanisms. All other messages are irrelevant to the extended proxy and are forwarded intact without any processing. Below, we describe the actions the proxy takes when each one of the special messages is received and Fig. 2 depicts this functionality graphically.

INVITE: When receiving an INVITE message the proxy first checks if the caller is in the "In-Range Users List". If not, the message is dropped. Otherwise, it increases the user's attempted calls by one and checks if a line is available. If so, the INVITE message is forwarded immediately and the user's successful calls total is increased by one. If no lines are available, the call is added to the "Pending Calls List" to a position determined by the scheduling policy. In order to prevent premature request timeouts from the client, the proxy replies with a TRYING message regardless of how the INVITE message is handled; this postpones the timeout and subsequent retransmission of the INVITE message by the client.

BYE: After receiving a BYE message, the proxy "releases" the occupied line and assigns it to the pending call with the highest priority. In this way a previously received call request that has been placed in the "Pending Calls

List" is satisfied and the user's successful calls counter is increased by one.

CANCEL: When a CANCEL message is received, the proxy checks if a corresponding INVITE message exists in the "Pending Calls List", i.e., the user tries to cancel a pending call request. If so, the request is removed from the "Pending Calls List", and on OK message is sent back to the user without forwarding the CANCEL message. However, if there is no corresponding pending call, the CANCEL message is forwarded to the SIP proxy for further processing (i.e., cancel the call request which is currently under processing).

4.3 PSTN gateway

The PSTN gateway provides the means of terminating calls originating in a VoIP network (in our case from users within a wireless network) to the PSTN. The gateway will work in conjunction with a SIP proxy that is responsible for passing calls for termination to the gateway. The PSTN gateway has a fixed capacity; as expected, the higher the capacity the higher the cost for its acquisition and operation.

4.4 System operation

The system is expected to operate as normal with the policies being applied to enhance QoE. Users will follow the normal procedure for establishing calls and as mentioned above, the SIP proxy maintains some information regarding the state of call requests; this information is necessary in order to apply scheduling. All INVITE messages that can not be immediately satisfied (due to non-availability of lines) are placed in the pending calls queue and are ordered according to the prioritization policy currently in place. When a line becomes available the INVITE message with the highest priority is immediately processed and its call request is forwarded to the PSTN gateway.

It is important to note that in the case where an INVITE message is not forwarded (due to a shortage of lines), the SIP Proxy sends back to the User Agent a TRYING message. This message is sent in order to prevent premature retransmission timeouts on behalf of the User Agents. This postpones the retransmission of the INVITE message from the User Agent, and leaves the client in a waiting state. This suggests that a possible extension of the proposed scheme might include modifications to the SIP protocol and the User Agent; such modifications will enable the User Agent to distinguish between the regular TRYING message and a message indicating that the call is held in a queue and will be established as soon as possible giving the user the option to cancel the call.

5 Simulations and results

To evaluate the performance of the proposed CAC scheme and the effectiveness of the different scheduling policies, we carried out extensive simulations. In this section, an extensive description of the Simulation Environment, the Simulation Entities, the Simulation Execution, and the Simulation Results are presented.

5.1 Simulation environment

In the early stages of our work [18], we used the NS-2 [2] simulator. Results acquired from those simulations, allowed us to prove the effectiveness of the proposed approach. However, we later switched to the network simulator developed at the National Chiao Tung University (NCTUns) [1]. This is due to some advantages that in our opinion NCTUns has over NS-2 when simulating Application Layer implementations. Those advantages include:

1. Simple integration of application layer protocol implementations.
2. Utilization of existing, and possibly commercial implementations; this reduces the possibility of simulating implementations that contain bugs.
3. Easier transition from the simulation phase to a testbed deployment; code used in simulations can be used in a real environment without any modification.
4. Easier extensions and modifications of the application; no modifications are required on the simulator.

The aforementioned advantages derive from the fact that NCTUns utilizes the actual protocol stack of a Linux machine during simulations; this feature allows its use in conjunction with any application that uses IP-based network communication. In our case, we considered this as a distinct advantage since the use of a complete implementation of the Session Initiation Protocol (SIP) [24] allowed us to acquire more realistic results.

Our experiments are based on PJSIP [3], an implementation of SIP written in C++. PJSIP comes with an implementation of a User Agent that is capable of making and receiving calls and also an implementation of a SIP proxy. In our case we extended the existing implementations by adding our own functionality. Below, we present all the different entities we implemented and used during the simulations. Note, that new code was developed that can readily run in both the simulated environment as well as a real setting.

5.2 Simulation entities

To run the simulations we deployed three distinct entities: the user agent, the enhanced SIP proxy and the callee. It is important to note that we had to make some compromises in

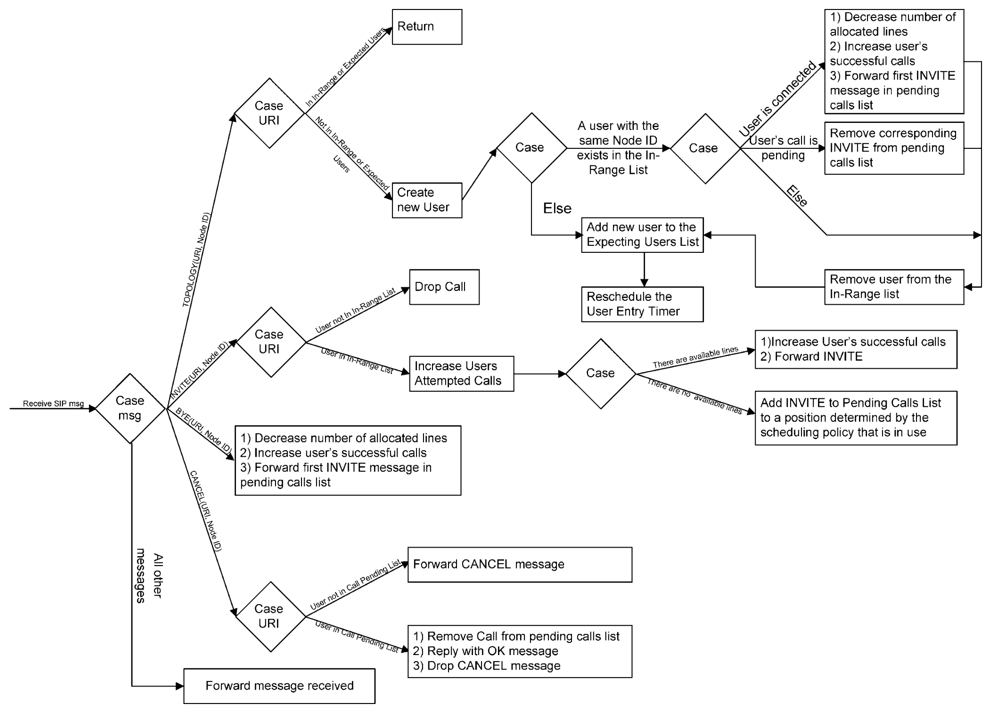

Fig. 2 The functionality of the extended SIP proxy

order to emulate access to the PSTN; more specifically, the functionality of the PSTN gateway is integrated within the SIP proxy and a PSTN user is replaced by the callee which is just a "dummy" SIP client that simply accepts incoming calls. The caller and the SIP proxy are described in detail below.

5.2.1 Caller

The caller is a SIP user agent; its primary responsibility is to mimic real user behavior by randomly establishing and terminating calls. Like real users, each caller acts autonomously and changes its behavior regularly; for this, we use a model that reflects the behavior of a real user. To create the caller behavioral model, we utilized statistical information acquired from different studies [11, 15, 25]. These studies try to analyze the behavior of users when using their mobile devices. Based on the results of these studies and considering that we want to concentrate on moving users, we created two distinct user categories: commuters and pedestrians. Note, that the percentage of people using their mobile phones while commuting is bigger that when people are involved in other activities. Furthermore, while commuting, people tend to make longer calls [4, 5].

5.2.2 Enhanced SIP proxy

The SIP proxy operates as described in Sect. 4.2 offering call scheduling capabilities based on call request prioritization policies. The SIP proxy also acts as the PSTN gateway by relaying call requests to the callees. To emulate PSTN connectivity capacity, the SIP proxy keeps track of ongoing calls and imposes a limit on the number of simultaneous calls that can occur at any time. At times of saturation, the proxy activates its call request scheduling mechanisms which operate according to the policies described earlier.

5.3 Simulation setup

To model the above mentioned behavior we defined the following parameters:

1. *The percentage of commuting users:* The percentage of users that are commuting. For example, setting this parameter to 25% implies that 25% of all users are commuting and the remaining (i.e., 75%) are pedestrians.
2. *The percentage of commuting users that make a call:* The average percentage of commuting users that are using their phone at any given time.
3. *The percentage of pedestrian users that make a call:* The average percentage of pedestrians that are using their phone at any given time.
4. *The average call duration for commuting users:* The average call duration (in seconds) for commuting users.
5. *The average call duration for pedestrian users:* The average call duration (in seconds) for pedestrian users.

As callers are moving, in addition to the above parameters, one must provide the minimum and maximum speed of each user category (pedestrians and commuters).

Utilizing the above information, each user agent adopts some random behavior attempting to resemble the behavior of a real user. The algorithm used for creating this behavior is described below; steps 1–2 are used for creating a random user with a random motion behavior; in step 3 the decision whether to place a new call is made; steps 4–6 are used for creating random calling behavior for each user based on the category it has been assigned to and finally, step 7 restarts the calling process. More specifically, with regards to the user agent, the system operation is as follows:

1. Use the "Percentage of commuting users" parameter to randomly assign a user to a category (i.e., pedestrian or commuter).
2. Based on the category that the user has been assigned to, assign a random speed to this user in the range [MinSpeed, MaxSpeed] of the particular category.
3. Randomly determine whether a user will place a call based on the user's particular category.
4. Calculate the potential call duration. This duration is derived using a normal distribution with a mean value equal to the average call duration of the user category.
5. Based on the decision of step 3 (place or not place a call) choose the next step of the algorithm. If the decision of step 3 was to make a call go to step 6. If not, wait for a time period equal to the call duration calculated at step 4 before going back to step 3.
6. Send an INVITE message and wait to receive both a TRYING and an OK message. Then start a timer that will expire after the call duration time. When the timer expires go to step 7.
7. Wait for some random time before going back to step 3.

As mentioned earlier, the above program mimics the behavior of a real user. Due to the unpredictable nature of real users, our results may only be accurate for a subset of users but this is expected to a large extent.

5.4 Simulation execution

To evaluate the proposed CAC scheme, we carried out simulations using different scenarios that mimic the (randomly characterized) behavior of humans in real life environments. The general network configuration used is depicted in Fig. 3.

The simulation scenarios used for acquiring the results presented include an average of 35 user agents with a 3:1 pedestrian:commuter ratio. In order to facilitate a varying number of users in the network, clients enter and exit the network coverage area at random intervals and exhibit random calling behaviour with an average call duration of 2.5 minutes for pedestrians and 4 minutes for commuters.

Concluding this section, we have to note that various line—number of users combinations were simulated. However, the combination of 8 lines (i.e., the maximum number of calls at any given time could not exceed 8) with 35 users appears to produce the most interesting results since there was a high but manageable demand for calls, which activated the application of our scheduling policies. Moreover, the same ratio of lines and users was used in two different network settings; the first one had one base station and the second had three base stations (Fig. 3 depicts the latter case). In this way, we were able to evaluate the effect of handoff to the proposed scheme. Finally, each simulation scenario was allowed to run for 50000 seconds. The acquired simulation results along with their evaluation are presented in the following section.

5.5 Simulation results and evaluation

For evaluating the performance of the proposed methods, each simulation produces a detailed log file containing information about the activity of each client. This log file includes all the actions a client performed (i.e. registered, made a call, hang-up a call, exited the network, etc.), along with some statistical information like the duration of each call, the total number completed and rejected calls and the overall satisfaction of each user. To quantify user satisfaction we used the User Satisfaction Rate metric, which is defined as the percentage of a user's successful calls over the total number of attempted calls. For example, a satisfaction rate of 50% occurs when a user had two call requests and only one of them was successful. We decided to use the User Satisfaction Rate metric because it creates data that can be utilised in acquiring additional metrics and statistics necessary for the evaluation of the proposed approach.

One other metric that can be derived from the User Satisfaction Rate is the Weighted User Satisfaction Rate. This metric adjusts the User Satisfaction Rate according to the time a user had to wait for a call request to be processed. More precisely, we decided to reduce the user's satisfaction by half every 32 seconds that a call remains in the pending calls queue; this time period, which we define as *invite_timeout* is the suggested INVITE retransmission timeout in the SIP protocol definition. The exponential reduction of the user's satisfaction was decided on purpose; using an aggressive approach will allow us to identify policies that can cause excessive call waiting times something which clearly is not desirable.

Formally, the Weighted User Satisfaction (per call) and the Weighted User Satisfaction Rate (per call) are defined as follows:

$$\begin{aligned} &\textit{Weighted Users Call Satisfaction} \\ &\quad = 100/\textit{total\_call\_wait\_time}/\textit{invite\_timeout} \end{aligned} \tag{1}$$

$$\begin{aligned} &\textit{Weighted User Satisfaction Rate} \\ &\quad = \frac{\sum_{Calls}(\textit{Weighted User's Call Satisfaction})}{\textit{Calls}} \end{aligned} \tag{2}$$

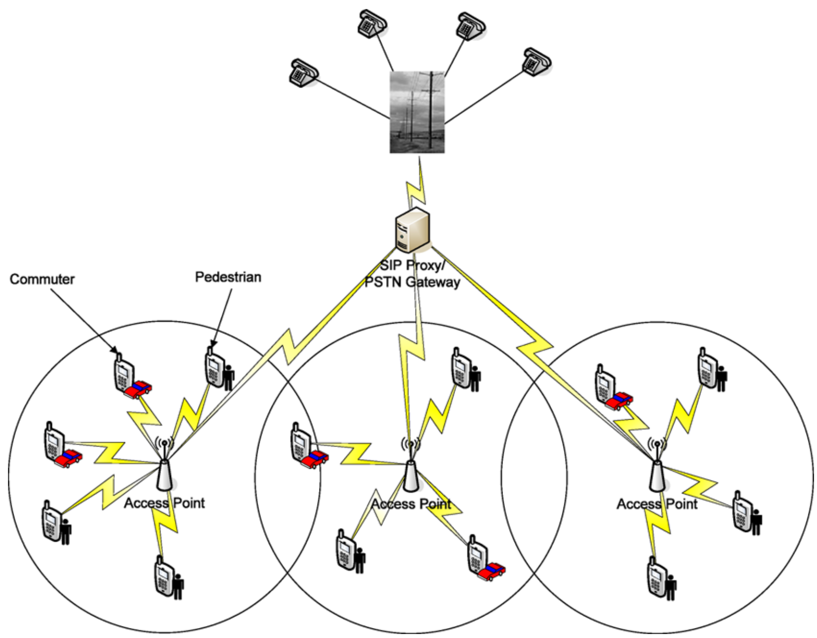

Fig. 3 Simulation topology

The results presented in Table 1 are calculated based on the above mentioned metrics. It is important to note that these results are derived from simulations which include multiple base stations. Results acquired using a single base station where very similar, and thus they didn't provide any additional information regarding the performance of the proposed scheme. This behaviour was expected since the bottleneck in our experiments is PSTN gateway line availability.

Using the table above, one can analyze the performance of the proposed scheme and its associated policies. However, before proceeding to the analysis of the performance of the new scheme we must explain the big difference on the number of rejected calls between the scheduled based approaches (Order Received, Further Away, Fewer Calls strategies) and the no scheduling approach. To explain this difference one must recall the definitions of the user agent and the extended SIP proxy. As mentioned earlier in this paper, when scheduling is used and no lines are available, the proxy places the INVITE message received in a queue and replies with a TRYING message. Upon receiving the TRYING message, the UA goes into a "waiting" state, pausing its operation until the establishment of the call. This behavior blocks the UA and thus postpones future call attempts. On the contrary, when no scheduling is used and no lines are available, the proxy immediately sends a BUSY message to the client. Thus, clients do not go into any blocking state; rather, they are allowed to start the process of making a new call immediately. This difference on the behavior of the Simulation Agent results in a bigger number of call attempts when no scheduling is used.

Next, we present the performance analysis of the proposed scheme. We will start this analysis with some simple observations extracted from Table 1. The first observation is that scheduling based CAC techniques have a better call acceptance rate (Percentage Accepted/Total Calls column). This was an expected result since scheduling schemes were specifically designed to increase this percentage. Influenced by the high acceptance rate, users are also more satisfied (Average User Satisfaction column) from the performance of the network when the scheduling base approaches were utilized. The small differences on the total talk time one can observe in the three scheduled based approaches (Order Received, Further Away, and Fewer Calls) are due to the random selection of the talk time by the UA and the different number of calls completed.

Furthermore, by examining Table 1 one can estimate the utilization of the lines when different CAC approaches are used. This information is derived from the Total Talk Time values. Considering that the simulation time was approximately 14 hours, dividing the total talk time of each method by 50000 gives the average line utilization. Approaches that perform request scheduling use an average of 7.2 lines out of the 8 available (90%), whereas the average line usage when no scheduling is applied is only 5.8 out of the 8 lines available (72.5%). Thus an improvement on line utilization of about 25% has been achieved something which is highly significant!

The last and more interesting observation, which also relates to the performance of the different policies, is an outcome of the analysis of the weighted user satisfaction (last column). As mentioned above, the weighted user satisfaction was introduced to correlate user satisfaction with the waiting time for establishing a call. Note here that the User Satisfaction Rate and the Weighted User Satisfaction Rate are equal when no scheduling approach is used since calls

Table 1 Simulation results for multiple BSs and 35 users

Policy	Calls accepted	Calls rejected	Total talk time (sec)	Percentage accepted/ total calls	Average user satisfaction	Average weighted user satisfaction
Order received	901	141	363301	86.46%	80.04%	13.09%
Further away	990	161	357784	86.01%	71.16%	37.89%
Fewer calls	1141	151	357146	88.31%	88.87%	54.45%
No scheduling	1059	1906	291743	35.71%	38.16%	38.16%

are either accepted or rejected immediately without any delay. Results suggest that there are cases where relatively long call establishment times can reduce user weighted satisfaction, making the "no scheduling" approach the most suitable choice. This phenomenon is more apparent when the Order Received policy is used as every new call request is placed at the end of the queue and is forced to wait in the queue for long time. Even though they were eventually successful, the weighted satisfaction is reduced for the majority of the calls, making the average weighted user satisfaction of this approach low. In order to clarify how an approach that never drops calls (i.e., the user satisfaction for a call is always above 0), can result in a lower user satisfaction from an approach that drops calls (i.e., user's satisfaction for a call can be 0) we present the following example that illustrates this phenomenon:

In a hypothetical scenario, at time t_0 all lines are occupied and a new INVITE message is received. With scheduling in place, this call will be placed in the pending calls queue, whereas without scheduling the call request will be denied completely. At this point, the user satisfaction value for both approaches will be 0. Then assume that at time t_1 a user hangs up and one line becomes available. This event will trigger the processing of a pending call when scheduling approaches are used. Processing a pending call will result in weighted user satisfaction for the scheduling approaches equal to $(100/(t_1 - t_0/\mathit{invite\_timeout}))$. Then assume that at time t_2 and before any line becomes available, a new INVITE message arrives at the proxy. The scheduling approaches will place the call in the pending calls list. However, the no scheduling approach will be able to immediately process the call using the available line previously released. Thus, the average user satisfaction for the no scheduling approach will be 50% (0% for the first call request and 100% for the second and successful call request). In the long run, this scenario usually results in a higher average weighted satisfaction when no call scheduling is applied.

To complement the results presented in Table 1, in the following two figures (Figs. 4 and 5) we present the Average and Average Weighted Satisfaction as they change over time. When starting a new simulation there are some unique conditions that are unlikely to occur in any later stage; all lines are available since no calls have been established yet, all queues are empty, and there is a huge demand for calls since a bunch of clients are generated simultaneously. These unique conditions are reflected at the leftmost part of each graph. We can clearly observe that at the beginning of a simulation both the Average and Average Weighted Satisfaction are very high; this is because all lines are available and the queues are empty. However, after some time into the simulation, we experience a significant drop in user satisfaction; this is due to the high demand which resulted in line saturation and longer queue delays. Moreover, we can observe that after the simulation comes into a "stable" condition, all graphs except the Average Weighted Satisfaction for the Order Received policy converge to a particular value. This clearly shows that the processing time of a call when the Order Received policy is used increases over time. This is because the size of the pending calls list increases, causing long processing delays which decreases User Satisfaction.

The above result suggests the extension of the scheduling-based CAC schemes in order to drop call requests when those cannot be processed within a reasonable amount of time. Possible solutions to this problem are suggested in the following section.

6 Conclusion and future work

In this work, we presented a new and novel mechanism that utilizes different call scheduling policies to improve user Quality of Experience when gateways with limited number of lines are used for terminating VoIP calls to PSTN phones. The implementation of such scheduling policies was made possible through the extension of a SIP-based proxy with call scheduling functionality. Proxies with such functionality are able during periods of non-availability of PSTN interconnecting lines, rather than denying outgoing call requests, to postpone the actual call establishment for some later time.

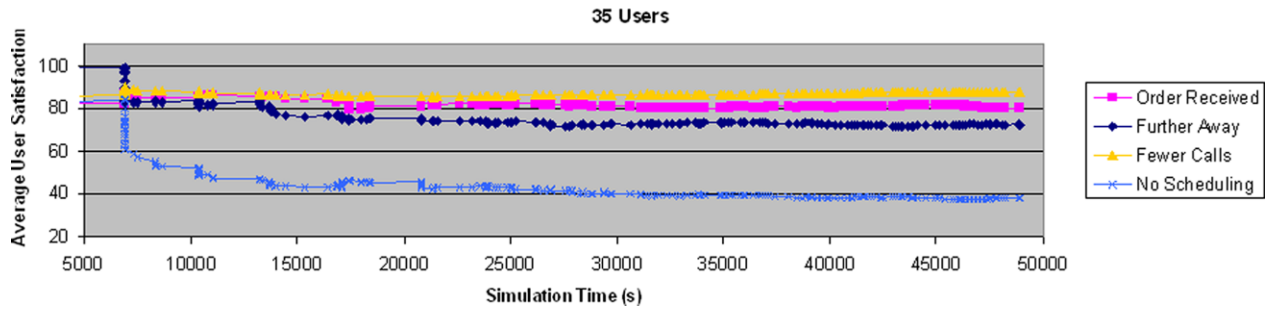

Fig. 4 Average user satisfaction

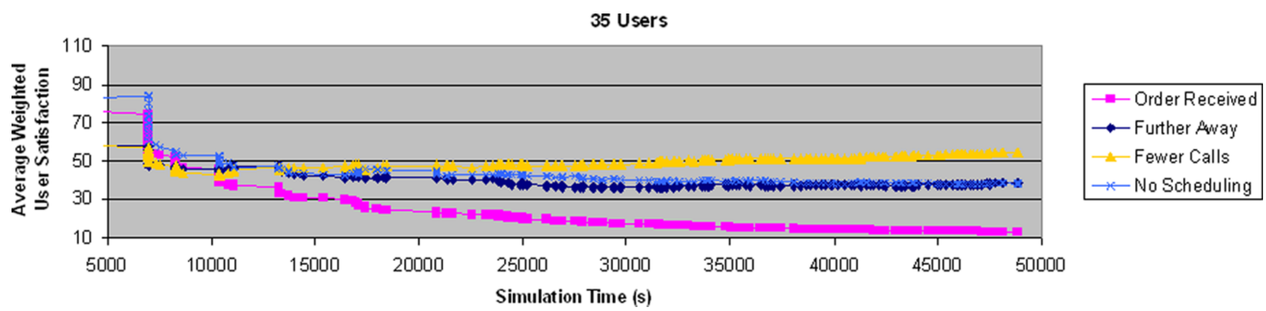

Fig. 5 Average weighted user satisfaction

To achieve this, the proxy places outgoing call requests in a queue; calls are established as soon as lines become available. The fact that call requests can be assigned different priorities, allows for the application of flexible call scheduling policies which can also be applied simultaneously.

For extending our work even further we plan to include more parameters in the call processing mechanism. These parameters can include the detailed categorization of sources and destinations (e.g. prioritizing calls that generate higher revenue or calls to emergency services) or even the assignment of different classes of service to different users who are perhaps willing to subscribe to them for a fee. Furthermore, we will consider proposing extensions to existing protocols in order to facilitate further improvements of our schemes and policies.

In addition, we will investigate possible extensions and improvements of SIP User Agents. The results of this work were acquired using SIP User Agents unaware of the scheduling capabilities of the proxy. Future work will evaluate the proposed schemes with clients aware for these new capabilities of the SIP proxy. This will allow us to ascertain whether any changes to the clients are required in order to improve the performance of the proposed scheme even further.

Finally, solutions to overcome the problem of low user satisfaction when calls remain pending for long time must be proposed. A possible solution to this problem can be derived by limiting the size of the pending calls queue. However, in order to improve the effectiveness of this approach, solutions that support a dynamically adjustable pending calls queue size (e.g. using a window-based method) may be deployed. This implies the existence of an algorithm capable of learning the calling habits of users that utilizes queue theory to accurately predict the waiting time of a call request that is about to be placed in a queue.

References

1. NCTUns 6.0 network simulator and emulator (Mar 13th 2010). http://nsl.csie.nctu.edu.tw/nctuns.html.
2. The network simulator—NS-2 (Mar 13th 2010). http://www.isi.edu/nsnam/ns/.
3. PJSIP—open source sip stack and media stack for presence, IM/instant messaging, and multimedia communication (Mar 13th 2010). http://www.pjsip.org/.
4. Statistics and cell phones (Mar 13th 2010). http://www.articlesbase.com/computers-articles/statistics-and-cell-phones-538109.html.
5. Berger, J. (Mar 13th 2010). Analysis and modeling of structures in telephony conversations. http://portal.etsi.org/stq/Workshop2005/presentations2005/P13_SpeechQuality_CellularNetworks.pdf.
6. Braden, R., Zhang, L., Berson, S., Herzog, S., & Jamin, S. (1997). Resource reservation protocol (RSVP)—version 1 functional specification (RFC 2205).
7. Carvalho, G. H., Francês, C. R. L., Costa, J. C. W. A., Rodrigues, R. C. M., & Carvalho, G. H. (2005). Optimal call admission control and bandwidth adaptation in multimedia cellular mobile networks. In *International conference on analytical and stochastic modelling techniques and applications (ASMTA)*, Riga, Latvia.
8. Chan, A., & Liew, S. C. (2007). VoIP capacity over multiple IEEE 802.11 WLANs. In *IEEE international conference on communications*, Glasgow, Scotland (pp. 3251–3258).

9. Chatzimisios, P., Raptis, P., Vitsas, V., Amanatiadis, D., & Paparrizos, K. (2008). Admission control for VoIP and data traffic in IEEE 802.11 WLANs. In *3rd international conference on performance evaluation methodologies and tools*, Athens, Greece. p. 1537056 1-7, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).
10. David, H., & Gopal, M. (2002). Call admission control and load balancing for voice over IP. *Performance Evaluation*, *47*(4), 243–253 586427.
11. Davis, A. (Mar 13th 2010). Letstalk annual cell phone etiquette survey. http://www.letstalk.com/company/release_031406.htm.
12. Dini, P., & Minni, R. (2004). A joint scheduling/admission control policy for multimedia services in utra FDD. In *European wireless (EW)*, Barcelona, Spain.
13. Doshi, B. T., Eggenschwiler, D., Rao, A., Samadi, B., Wang, Y.-T., & Wolfson, J. (2003). VoIP network architectures and QoS strategy. *Bell Labs Technical Journal*, *7*(4), 41–59.
14. Ezaki, T., Kawakami, H., & Asatani, K. (2004). A new VoIP call admission control scheme with use alternate routing for low call loss probability. In *IEEE international conference on communications* (Vol. 4, pp. 2209–2213).
15. Glassbrenner, D. (Mar 13th 2010). Driver cell phone use in 2005. http://www-nrd.nhtsa.dot.gov/Pubs/809967.PDF.
16. Houck, D. J., Kim, E., Uzunalioglu, H., & Wehr, L. A. (2003). A measurement-based admission control algorithm for VoIP. *Bell Labs Technical Journal*, *8*(2), 97–110.
17. Kashyap, A., Ganguly, S., Das, S., & Banerjee, S. (2007). VoIP on wireless meshes: Models, algorithms and evaluation. In *26th IEEE international conference on computer communications (INFOCOM)*, Anchorage, AK (pp. 2036–2044).
18. Lambrinos, L., & Djouvas, C. (2009). Applying scheduling policies to improve quality of experience in voice over IP. In *IEEE international performance computing and communications conference (IPCCC)*, Phoenix, AZ.
19. Lee, H.-W., & Chong, S. (2007). Combined packet scheduling and call admission control with minimum throughput guarantee in wireless networks. In *IEEE transactions on wireless communications* (Vol. 6, pp. 3080–3089).
20. Mase, K., Toyama, Y., Bilhaj, A. A., & Suda, Y. (2001). Management for VoIP networks with edge-to-edge admission control. In *Global telecommunications conference (GLOBECOM)*, New York, USA (Vol. 4, pp. 2556–2560).
21. McGovern, P., Chung, S., Murphy, S., & Murphy, L. (2006). Endpoint admission control for VoIPoWLAN. In *ICT 2006*, Funchal, Madiera Island.
22. Mohan, N., & Ravichandran, T. (2009). An efficient multiclass call admission control and adaptive scheduling for WCDMA wireless network. *European Journal of Scientific Research*, *33*(4), 718–727.
23. Rong, B., Qian, Y., Lu, K., & Kadoch, M. (2008). Call admission control for mobile agent based handoff in wireless mesh networks. In *IEEE international conference on communications (ICC)*, Beijing (pp. 3274–3278).
24. Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., & Schooler, E. (2002). SIP: Session initiation protocol (RFC 3261).
25. Samkange-Zeeb, F., Berg, G., & Blettner, M. (2004). Validation of self-reported cellular phone use. *Journal of Exposure Analysis and Environmental Epidemiology*, *14*, 245–248.
26. Tewari, M., & Jamadagni, H. (2003). A new call admission control scheme for real-time traffic in wireless networks. In *IEEE TENCON* (Vol. 4, pp. 1585–1589).
27. Tsiropoulos, G., Stratogiannis, D., Kanellopoulos, J. D., & Cottis, P. (2009). Fairness optimization of thinning call admission control in wireless networks. In *WMCNT 2009* (pp. 1–6).
28. Uzunalioglu, H., Houck, D. J., & Wang, Y. T. (2006). Call admission control for voice over IP: Research articles. *International Journal of Communication Systems*, *19*(4), 363–380.
29. Wang, S., Mai, Z., Xuan, D., & Zhao, W. (2006). Design and implementation of QoS-provisioning system for voice over IP. *IEEE Transactions on Parallel and Distributed Systems*, *17*(3), 276–288.
30. Wei, H.-y., Kim, K., Kashyap, A., & Ganguly, S. (2006). On admission of VoIP calls over wireless mesh network. In *IEEE international conference on communications (ICC)*, Istanbul, Turkey.
31. Wu, P.-Y., Chen, J.-J., Tseng, Y.-C., & Lee, H.-W. (2005). Design of QoS and admission control for VoIP services over IEEE 802.11e WLANs. In *National computer symposium*, Taiwan.
32. Yang, O., & Lu, J. (2006). Call admission control and scheduling schemes with QoS support for real-time video applications in IEEE 802.16 networks. *Journal of Multimedia*, *1*(2), 21–29.
33. Yasukawa, K., Forte, A. G., & Schulzrinne, H. (2008). Distributed delay estimation and call admission control in IEEE 802.11 WLANs. In *Mobile systems, applications, and services (MobiSys)*, Colorado.

Lambros Lambrinos is a lecturer at the Department of Communication and Internet studies at the Cyprus University of Technology. He holds a Computer Engineering degree from the University of Manchester and conducted his PhD at University College London in the area of Real-time Multimedia Communications. Dr Lambrinos has worked on several large projects funded by the European Commission. He has published various papers related to his research work and peer-reviewed papers for various conferences and journals. Currently, his research interests include applications of large-scale multimedia streaming architectures, voice-over-IP over wireless networks as well as applications of delay-tolerant networks.

Constantinos Djouvas is a Postdoctoral Fellow in the Department of Communication and Internet Studies at the Cyprus University of Technology. He holds a PhD degree from the City University of New York, a MSc degree from the University of Cyprus, and a BSc from the University of Crete, Greece. His doctoral research concentrated on Network Testing and was funded by the National Science Foundation (NSF). During his PhD studies, he collaborated with leading institutions in the USA including the Naval Research Lab (NRL) and the Massachusetts Institute of Technology (MIT). His current research focuses on VoIP and involves the investigation and proposal of efficient and effective techniques that extend current Call Admission Control (CAC) approaches with provisioning and scheduling capabilities.