

Is It Worth Involving Several Cooperating Agents for Multimedia User's Admission in Cellular Networks?

Youssef Iraqi and Raouf Boutaba

University of Waterloo, DECE, Waterloo, Ont. N2L 3G1, Canada,
{iraqi, rboutaba}@bbcr.uwaterloo.ca

Abstract. In this paper, we enhance our multi-agent system for resource management developed for cellular mobile networks. The new scheme allow agents to dynamically adapt to changes in the network load in order to maintain a target call dropping probability. We investigate the impact of the number of neighboring agents involved in a call admission decision in addition to the agent receiving the call request. The neighboring agents provide significant information about their ability to support the new mobile user in the future. This distributed process allows the original agent to make a more clear-sighted admission decision for the new user. Simulations are presented with a detailed analysis of a comparison between two schemes involving different numbers of agents.

1 Introduction

Cellular mobile networks have to continue supporting their mobile users after they leave their original cells. This rises a new challenge to resource management algorithms. For instance a call admission (CA) process should not only take into consideration the available resources in the original cell but also in neighboring cells as well.

Mobile users are in a growing demand for multimedia applications, and the next generation wireless networks are designed to support such bandwidth greedy applications. The (wireless) bandwidth allocated to a user will not be fixed for the lifetime of the connection as in traditional cellular networks, rather the base station will allocate bandwidth dynamically to users. Many evolving standards for Wireless Broadband Systems, UMTS and IMT2000 have proposed solutions to support such capability.

In [4] we have proposed a Multi-Agent system for call admission management designed for wireless mobile multimedia networks with dynamic bandwidth allocation. The call admission process involves the cell that receives the call admission request and a cluster of neighboring cells. The agents share important resource information so the new admitted user will not be dropped due to hand-offs. Consequently, the network will provide a low call dropping probability while maintaining a high resource utilization.

In this paper, we propose an enhancement of the Multi-Agent system and

propose a mechanism for dynamic adaptation to obtain a target call dropping probability (CDP). We investigate the impact of the number of involved agents in the CA process on the achieved performance, in terms of average bandwidth utilization and call dropping probability. Other parameters such as call blocking probability are also investigated.

The paper is organized as follows. In section 2, we describe the Multi-Agent architecture considered in this paper. Section 3 defines the dynamic mobile probabilities used by our Multi-Agent system and presents the call admission process performed locally by agents in our system. It also introduces the overall admission process and agent's cooperation. Section 4 describes the algorithm used to dynamically achieve a target call dropping probability. Section 5 discusses the conducted simulation parameters and presents a detailed analysis of the obtained results. Finally, section 6 concludes the paper.

2 The Multi-agent Architecture

We consider a wireless/mobile network with a cellular infrastructure that can support mobile terminals running applications which demand a wide range of resources. Users can freely roam the network and experience a large number of handoffs during a typical connection. We assume that users have a dynamic bandwidth requirement. The wireless network must provide the requested level of service even if the user moves to an adjacent cell. A handoff could fail due to insufficient bandwidth in the new cell, and in such case, the connection is dropped.

To reduce the call dropping probability, we have proposed in [4] a multi-agent system that allows neighboring cells to participate in the decision of a new user admission. Each cell or base station has an agent running on it. The agent keeps track of the cell's resources and shares information with neighboring agents to better support mobile users. Each involved agent in an admission request will give its local decision according to its available resources and information from other agents and finally the agent at the cell where the request was issued will decide if the new request is accepted or not. By doing so, the new admitted connection will have more chances to survive after experiencing handoffs.

We use the notion of a cluster similar to the shadow cluster concept [5]. The idea is that every connection exerts an influence upon neighboring base stations. As the mobile terminal travels to other cells, the region of influence also moves. The cells influenced by a connection are said to constitute a cluster (see figure 2). Each user¹ in the network, with an active connection has a cluster associated to it. The agents in the cluster are chosen by the agent at the cell where the user resides. The number of agents of a user's cluster depend on factors such as user's current call holding time, user's QoS requirements, terminal trajectory and velocity.

¹ in the rest of the paper the term "user" and "connection" are used interchangeably

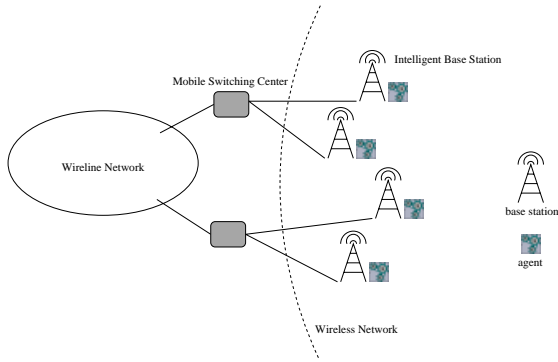


Fig. 1. A wireless Network and the Multi-Agent system

3 The Call Admission Process

3.1 Dynamic Mobile Probabilities

We consider a wireless network where the time is divided in equal intervals at $t = t_0, t_1, \dots, t_m$. Let j denote a base station (and the corresponding agent) in the network, and x a mobile terminal with an active wireless connection. Let $K(x)$ denote the set of agents that form the cluster for the active mobile terminal x . We denote $P_{x,j,k}(t) = [P_{x,j,k}(t_0), P_{x,j,k}(t_1), \dots, P_{x,j,k}(t_{m_x})]$ the probability that mobile terminal x , currently in cell j , to be active in cell k , and therefore under the control of agent k , at times $t_0, t_1, t_2, \dots, t_{m_x}$. $P_{x,j,k}(t)$ represents the projected probabilities that mobile terminal x will remain active in the future and at a particular location. It is referred to as the Dynamic Mobile Probability (DMP) in the following. The parameter m_x represents how far in the future the predicted probabilities are computed.

Those probabilities may be function of several parameters such as: handoff probability, the distribution of call length for a mobile terminal x , cell size, user mobility profile, etc.

For each user x in the network, the agent that is responsible for, decides the size of the cluster $K(x)$, which is the set of agents involved in the admission process, and sends the DMPs to all members in $K(x)$. The agent must specify whether the user is a new one (in which case the agent is waiting for responses from the members of $K(x)$) or not. As the user roams the network, the agents belonging to his cluster change. Only those agents that belong to a user's cluster will receive the DMPs and hence will have to deal with bandwidth reservation for the user. Agents that are no more part of the user's cluster will release any reserved bandwidth made for that user.

A method for computing dynamic mobile probabilities taking into consideration mobile terminal direction, velocity and statistical mobility data, is presented in [1]. Other schemes to compute these probabilities are presented in [2] [3]. To

compute these probabilities, one can also use mobiles' path/direction information readily available from certain applications, such as the route guidance system of the Intelligent Transportation Systems with the Global Positioning System (GPS).

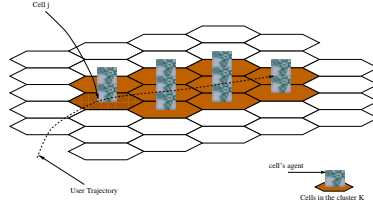


Fig. 2. Example of a user's cluster

3.2 Local Call Admission Decision

At each time t_0 each agent, in a cluster $K(x)$ involved in our CA process for user x , makes a local CA decision for different times in the future (t_0, t_1, \dots, t_{m_x}). Based on these CA decisions, we call Elementary Responses, the agent makes a final decision which represents its local response to the admission of user x in the network. Elementary responses are time dependent. The computation of these responses is different depending on the user location and type. The user can be either a local new user or a new user that has a non null probability to be in this cell in the near future.

The network tries first to continue supporting old users and uses the DMPs to check if a cell can accommodate a new user who will possibly come to the cell in the future. The cell's agent can apply any local call admission algorithm to compute the elementary responses. We write $r_k(x, t)$ the elementary response of agent k for user x for time t . We assume that $r_k(x, t)$ can take one of two values: -1 meaning that agent k can not accommodate user x at time t ; and $+1$ otherwise. A detail description of how to compute the elementary responses is presented in [4].

Since the elementary responses for future foreign users are computed according to local information about the future, they should not be assigned the same confidence degree. Indeed, responses corresponding to the near future are more likely to be more accurate than those of the far future.

We write $C_k(x, t)$ the confidence of agent k in its elementary response $r_k(x, t)$. Agent k has to compute (or simply choose) the confidence degree $C_k(x, t)$, typically between 0% and 100%.

The confidence degrees depend of many parameters. It is clear that the time in the future for which the response is computed has great impact on the confidence of that response. The available bandwidth when computing the elementary response also affects the confidence.

To compute the confidence degrees we propose a formula that uses the percentage of available bandwidth when computing the elementary response as an indication of the confidence the agent may have in this elementary response. The confidence degrees are computed using:

$$Confidence = e^{(1-p)} * p^n \quad (1)$$

where p is a real number between 0 and 1 representing the percentage of available bandwidth at the time of computing the elementary response. And $n \geq 1$ is a parameter that is chosen experimentally to obtain the best efficiency of the call admission routine.

If for user x , agent k has an elementary response $r_k(x, t)$ for each t from t_0 to t_{m_x} , those elementary responses are weighted with the corresponding DMPs $P_{x,j,k}(t_0)$ to $P_{x,j,k}(t_{m_x})$, to compute the final response. The final response from agent k to agent j concerning user x is then :

$$R_k(x) = \frac{\sum_{t=t_0}^{t=t_{m_x}} r_k(x, t) \times P_{x,j,k}(t) \times C_k(x, t)}{\sum_{t=t_0}^{t=t_{m_x}} P_{x,j,k}(t)} \quad (2)$$

where $C_k(x, t)$ is the confidence of agent k in the elementary response $r_k(x, t)$. To normalize the final response, each elementary response is also divided by the sum over time t of the DMPs in cell k . Finally, agent k sends the response $R_k(x)$ to the corresponding agent j . Note that $R_k(x)$ is a real number between -1 and 1 .

3.3 Distributed Call Admission Process

Here the decision takes into consideration the responses from all the agents in the user's cluster. The admission process concerns only new users seeking admission to the network, not already accepted ones. We assume that agent j has already decided the cluster $K(x)$ and that agent j has already assigned to each agent k in the cluster $K(x)$ a weight $W_k(x)$. Each weight represents the importance of the contribution of the associated agent to the global decision process. Usually, the more an agent is involved in supporting the user, the higher is its weight value. Weights $W_k(x)$ depend on the DMPs. We propose the following formula to compute the weights $W_k(x)$:

$$W_k(x) = \frac{\sum_{t=t_0}^{t=t_{m_x}} P_{x,j,k}(t)}{\sum_{k' \in K(x)} \sum_{t=t_0}^{t=t_{m_x}} P_{x,j,k'}(t)} \quad (3)$$

Relevance. In this paper, we introduce a new parameter that we call spatial relevance or simply relevance of an agent. To explain the idea of relevance, let's take the following example: consider a linear highway covered by 10 square cells as in figure 3. Assume that a new user, following the trajectory shown in figure 3, is requesting admission in cell number 0 and that the CAC process involves

5 cells. Responses from agents number 1, 2, 3 and 4 are relevant only if agent number 0 can accommodate the user. Similarly, responses from agents 2, 3 and 4 are relevant only if agent 1 can accommodate the new user when it hands off from cell 0. And the same principle applies to the other agents. This is because a response from an agent is irrelevant if the user can not be supported until that agent. We write $\Phi_k(x)$ the relevance of agent k for user x .

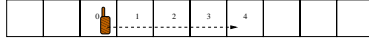


Fig. 3. An example of a highway covered by 10 cells

The relevance depends only on the topology of the considered cellular network. For the linear highway example of figure 3, we propose the following relevance formula:

$$\Phi_0(x) = 1 \text{ and } \Phi_k(x) = \prod_{l=1}^k \frac{(1 + R_{l-1}(x))}{2} \quad (4)$$

Note that for each $k \in K(x)$ we have $0 \leq \Phi_k(x) \leq 1$. Note also that in eq. 4, agent j (the agent receiving the admission request) has the index 0 and that the other agents are indexed in an increasing order according to the user direction as in figure 3.

The agent computes the sum of $R_k(x) \times W_k(x) \times \Phi_k(x)$ over k . The final decision of the call admission process for user x is based on:

$$D(x) = \frac{\sum_{k \in K(x)} R_k(x) \times W_k(x) \times \Phi_k(x)}{\sum_{k' \in K(x)} W_{k'}(x) \times \Phi_{k'}(x)} \quad (5)$$

Note that $-1 \leq D(x) \leq 1$. If $D(x)$ is higher than a certain threshold, we call acceptance threshold, the user x is accepted; the user is rejected otherwise. The more higher is $D(x)$ the more likely the user connection will survive in the event of a handoff.

Agent's Cooperation. At each time t , an agent j decides if it can support new users. It decides locally if it can support old users as they have higher priority than new users. This is because, from a user point of view, receiving a busy signal is more bearable than having a forced termination. Agent j also sends the DMPs to other agents and informs them about its new users requesting admission to the network (step 2 in figures 4, 5). Only those new users who can be supported locally are included. New users that can not be accommodated locally are immediately rejected.

At the same time, agent j receives DMPs from other agents and is informed about their new users requests. Using equation 2, agent j decides if it can support

theirs new users in the future and sends the responses to those agents (step 3 in figures 4, 5). When agent j receives responses from the other agents concerning its own new users, it performs the following for each of these users (step 4 in figures 4, 5): If it can not accommodate the call, the call is rejected. If the agent can accommodate the call, then the CA decision depends on the value of $D(x)$ as computed in eq. 5. A detailed description of the algorithm is presented in [4].

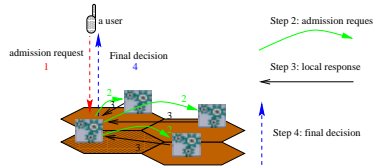


Fig. 4. Agent’s cooperation for the admission of a user

Figure 4 shows the different steps of agent’s cooperation when processing an admission request. Figure 5 depicts the admission process diagram at the agent receiving the admission request and at an agent belonging to the cluster. Because the admission request is time sensitive the agent waiting for responses from the agents in the cluster will wait until a predefined timer has expire then he will assume a negative response from all agents that could not respond in time.

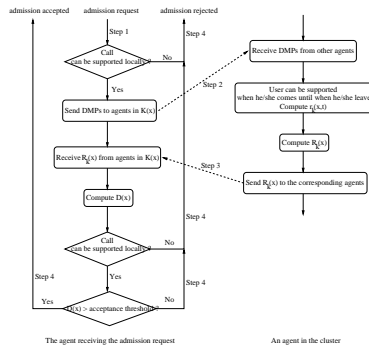


Fig. 5. Admission process diagram

It is worth noting that as the user roams, the corresponding dynamic mobile probabilities in certain cells will decrease; and as a consequence the bandwidth allocation made in these cells will also be decreased and eventually released. This is because a cell makes new reservations for each time slot. If the cell receives

new DMPs for a user, the reservations are updated with the new requirements. And if the cell does not receive the DMPs for a user that had some bandwidth reservation in the past time slot, the cell assumes that it does not make part of the user's cluster any more and hence releases the reserved bandwidth.

4 Maintaining a Target Call Dropping Probability

In this section we explain how our algorithm varies the value of the acceptance threshold to maintain a target CDP value. We assume that each Mobile Switch Center (MSC), controlling a set of agents in the network, modifies the acceptance threshold of the agents it controls in order to maintain a target CDP. The following is the pseudo-code of the algorithm for adjusting the acceptance threshold, we will refer to as algorithm 1 in the remaining of the paper.

Algorithm 1: Pseudo-code for adjusting the acceptance threshold T_{acc}

```

 $w = \lceil \frac{1}{target\ CDP} \rceil$ ;  $wobs = w$ ;  $nA = 0$ ;  $nD = 0$ 
if a user is accepted
  {  $nA++$ ;
  if ( $nA \geq wobs$ )
    {
    if ( $nD == wobs/w$ ) { $wobs = w$ ;  $nA = 0$ ;  $nD = 0$ ;}
    else { $wobs++ = w$ ; if ( $T_{acc} > -1.0$ )  $T_{acc}- = 0.01$ ;}
    }
  }
if a user is dropped
   $nD++$ ;
  if ( $nD > wobs/w$ ) { $wobs++ = w$ ; if ( $T_{acc} < 0.95$ )  $T_{acc}+ = 0.01$ ;}

```

The MSC begins by selecting a reference observation window w according to the target CDP as follows: $w = \lceil \frac{1}{target\ CDP} \rceil$. Note that we do not include the case where the target CDP is equal to zero, since this one is almost impossible to achieve and not practical from the provider point of view.

The variable representing the observation window $wobs$ is set to w , and the number of accepted users nA as well as the number of dropped users nD are set to zero.

If a new user is accepted in the system then nA is incremented by one. If we have observed at least $wobs$ accepted users ($nA \geq wobs$) then, if the number of users dropped is equal to the maximum allowed dropping value, we set $wobs$ to w and set nA and nD to zero and restart from the beginning. If the number of users dropped is less than the maximum allowed, then we increase $wobs$ and decrease the acceptance threshold. This means that we will allow more users to be admitted in the system.

In case a user is dropped then nD is incremented by one. If the number of dropped users exceeds the maximum allowed value, then we increase $wobs$ and increase the acceptance threshold. This means that we increase our observation

window and will allow less users to be admitted in the system.

Note that the proposed algorithm aims to achieve exactly the target CDP. This can easily be modified to let the actual CDP lay between a maximum and a minimum allowed values. Note also that the maximum allowed acceptance threshold is set to 0.95 in algorithm 1 in order for the network to accept a minimum number of users even if a congestion occurs.

5 Performance Evaluation

We evaluate here our Multi-Agent system with different numbers of agents involved in the CA process. We compare the performance of the scheme in the two following scenarios:

1. two agents are involved in the CAC process. This scheme will be referred to as **SC1**.
2. five agents are involved in the CAC process. This scheme will be referred to as **SC2**.

5.1 Simulation Parameters

For simplicity, we evaluate the performance of our Multi-Agent system for mobile terminals which are traveling along a highway as in figure 3. This is a simplest environment representing a one-dimensional cellular system. In our simulation study we have the following simulation parameters and assumptions:

1. The time is quantized in intervals $\tau = 10s$
2. The whole cellular system is composed of 10 linearly-arranged cells, laid at 1-km intervals. Cells are numbered from 1 to 10.
3. Cell 1 and 10 are connected so that the whole cellular system forms a ring architecture. This allows to avoid the uneven traffic load that will be experienced by cell 1 and 10 otherwise.
4. Each cell has a fixed capacity of 100 bandwidth units except cells 3, 4 and 5 which have 50, 30 and 50 bandwidth units respectively. This is to create a local congestion that will remain for a long period. An example of such case is a temporary increase in the interference level which prevents the cells from using all their capacity.
5. Connection requests are generated in each cell according to Poisson process. A newly generated mobile terminal can appear anywhere in the cell with equal probability.
6. Mobile terminals can have speeds of: 70, 90, or 105 km/h. The probability of each speed is 1/3, and mobile terminals can travel in either of two directions with equal probability.
7. We consider three possible types of traffic: voice, data, and video. The probabilities associated with these types are 0.3, 0.4 and 0.3 respectively. The number of bandwidth units (BUs) required by each connection type is: voice = 1, data = 5, video = 10. Note that fixed bandwidth amounts are allocated to users for the sake of simplicity.

8. Connection lifetimes are exponentially-distributed with a mean value equal to 180 seconds.
9. We simulate a total of 10 hours of real-time highway traffic, with a constant cell load equal to 720 new calls/h/cell.
10. The DMPs are computed as in [1].
11. The confidence degree is computed using eq. 1 with $n = 3$.
12. The relevance is computed using eq. 4.
13. All users with a specific type of service have the same acceptance threshold. Algorithm 1 is used to adjust the acceptance threshold T_{acc} of all 10 agents and the target CDP is 10%. We assume that all 10 agents are under the control of one Mobile Switching Center. The accepted thresholds for voice, data and video users are set to $1.7 * T_{acc}$, $1.2 * T_{acc}$ and T_{acc} respectively. This is to achieve fairness between voice, data and video users. Indeed, if we use the same acceptance threshold for all users irrespective to their class of service, very few video users will be admitted to the network. This is because video users require more capacity than the other users, and hence it is more difficult to obtain high responses ($D(x)$).

The following additional simulation parameters are used for the **SC1** scheme:

- m_x is fixed for all users and is equal to 18. This means that the DMPs are computed for 18 steps in the future.
- The size of the cluster $K(x)$ is fixed for all users and is equal to 2. This means that one agent in the direction of the user and the agent of the cell where the user resides form the cluster.

For **SC2** scheme, the following additional simulation parameters are assumed:

- m_x is fixed for all users and is equal to 25. This means that the DMPs are computed for 25 steps in the future.
- The size of the cluster $K(x)$ is fixed for all users and is equal to 5. This means that four agents in the direction of the user and the agent of the cell where the user resides form the cluster.

5.2 Simulation Results

In our simulations, a user x requesting a new connection is accepted into a cell only if the final decision $D(x)$ is above the acceptance threshold corresponding to the user class of service (voice, data or video). Figure 6 depicts the call dropping percentage achieved when using scheme **SC2**. The call dropping percentage represents the ratio of dropped users to the number of admitted users in the system. This is the aggregate call dropping percentage including all types of service. We can notice that algorithm 1 allows the actual CDP to approach the target CDP by varying the value of the acceptance threshold T_{acc} .

In figure 7, we compare the percentage of refused calls, given the offered load, when using scheme **SC1** and **SC2**. We can notice that **SC2** refuses less users than **SC1**. Indeed, **SC2** accepts about 8% more users than **SC1**. At a first sight,

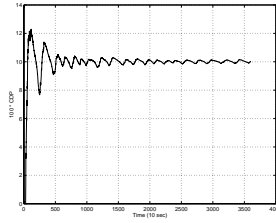


Fig. 6. Call dropping percentage

this result may seem abnormal. Indeed, scheme **SC2** involves five agents in the CAC decision process (3 more agents than **SC1**), and thus it is more difficult for a new user to be admitted by **SC2** than **SC1**. However, as we will see later in this section, **SC2** has the ability to avoid admitting those users who are most likely to be dropped and can use the saved bandwidth to accept more users who can most likely be supported.

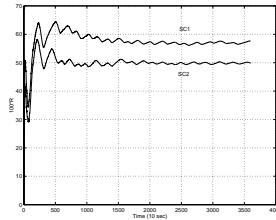


Fig. 7. Percentage of refused calls

Figure 8 shows that **SC2** not only accepts more users than **SC1** but also allows for a better resource utilization. In fact, **SC2** uses almost 10 bandwidth units more than **SC1**. It is worth noting that the low resource utilization experienced by the two schemes is due to the number of video users in the system and to the assumption that the whole system is controlled by one MSC. The latter assumption means that when a part of the network experience a congestion, the whole network is affected by refusing more users (since the MSC increases the acceptance threshold for all the agents in the network). Although the simulated one MSC configuration is not likely to happen in the real-world, simulation results show the potential benefit of using scheme **SC2** compared to scheme **SC1**.

To further compare the two considered schemes, we compute the individual dropping percentage among the three considered classes of service, namely voice, data and video. The simulation results are shown in figure 9.

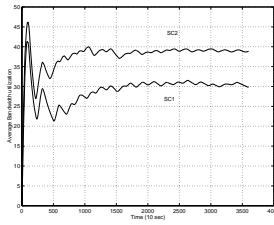


Fig. 8. Average bandwidth utilization

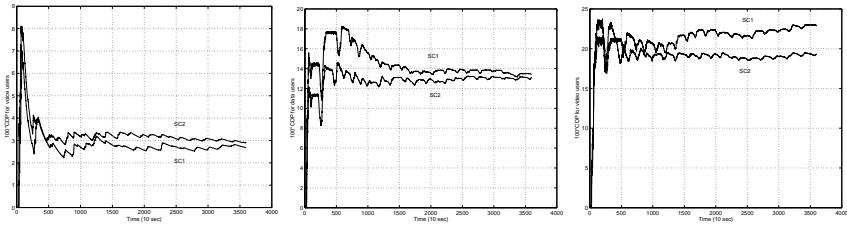


Fig. 9. Percentage of dropped voice, data and video users

In this figure, we can observe that the two schemes, **SC1** and **SC2**, achieve almost the same dropping percentage for voice and data users respectively, with a slightly better performance of **SC2** in case of data users. However, **SC2** drops almost 4% video users less than **SC1**.

As the percentage of dropped users depicted in figure 9 is computed according to the number of accepted users in each class of service, the comparison will not be fair if we do not observe the number of admitted users within each class of service for the two schemes. Figure 10 shows the percentage of refused calls within each class of service, and figure 11 plotted the number of accepted users within each class of service when using the two schemes.

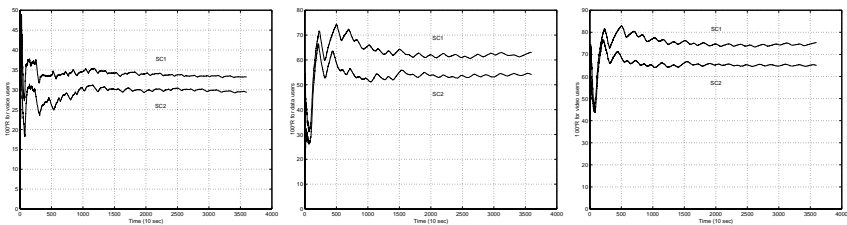


Fig. 10. Percentage of refused voice, data and video calls

According to figure 10, **SC2** refuses less users than **SC1** irrespective of users classes of service. This means that **SC2** accepts more users while achieving the same CDP in case of voice and data users, and that it allows more video users to be admitted to the network while achieving a lower CDP compared to **SC1**. According to figure 11, **SC2** accepts about 1500 video users more than **SC1** for the 10 real-time hours considered.

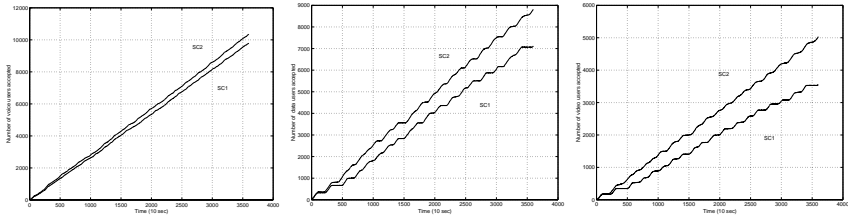


Fig. 11. Number of accepted voice, data and video calls

The bad performance achieved by **SC1** is explained by the fact that this scheme can not differentiate between those users who can be supported and those who can not. Its short sight prevents the scheme from being informed about a far congestion. Thus, the only way for **SC1** to reduce the CDP to the target value is to accept less users in the network, which results in a poor resource utilization.

On the other hand, since **SC2** involves more agents in the CA process than **SC1**, the scheme is able to distinguish between those users who can be supported and those who are most likely to be dropped due to some congestion. This has the two following benefits: (1) the scheme can accept more users without sacrificing the CDP; (2) the bandwidth saved from not allowing some “bad” users to be admitted in the network, can be used to admit more “good” users.

We have conducted several other simulations with different offered loads and different simulation parameters (such as different mean holding time). The main observation worth highlighting here is that the two schemes **SC1** and **SC2** achieve almost the same performance in case of no congestion or in case of a uniformly distributed congestion. The latter case is less important since it can be solved off-line by increasing the network capacity. We have observed in the simulations presented in this paper, **SC2** achieves a better performance in case of a local congestion. The fact that the two schemes achieve the same results in case of a non congested network or in case of a uniformly distributed congestion is foreseeable. This is mainly because the responses from the three additional agents in **SC2** (agents 2, 3 and 4 in figure 3) only confirm what the two involved agents in **SC1** (agents 0, 1 in figure 3) have decided.

Of course, **SC2** does not have only advantages. As **SC2** involves more agents in the CAC decision process, it induces more communications between base stations and also requires more processing power than **SC1**. These resources are

less critical compared to the wireless network bandwidth. A good compromise is to use **SC1** when the network is not congested and use **SC2** when a congestion is detected. The process of selecting the good scheme is out of the scope of this paper and is subject to future work.

6 Conclusion

We have described a Multi-Agent system for resource management suitable for wireless multimedia networks. The proposed system operates in a distributed fashion by involving, in a call admission decision, not only the agent receiving the admission request, but also a determined number of neighboring agents. We also proposed an algorithm that dynamically adjusts users acceptance threshold to achieve a target call dropping probability. We also presented an analysis of the comparison between two call admission schemes involving a different number of agents in the decision process. We have observed that it is worth involving more agents in the CA decision in case of local congestion. This allows the scheme to take a more clear-sighted admission decision for new users, hence, achieving better resource management and quality of service.

In this paper, we have compared the performance of our Multi-Agent system for resource management when involving two and five agents in the admission decision respectively. However, our system can involve any number of agents. We have demonstrated that in some cases it is worth involving several agents in the admission process. The choice of the number of agents to involve and when this should happen is an important issue that will be addressed in futur work.

References

1. D. A. Levine, I. F. Akyildz and M. Naghshineh, 'A Resource Estimation and Call Admission Algorithm for Wireless Multimedia Networks Using the Shadow Cluster Concept,' *IEEE/ACM Transactions on Networking*, vol. 5, no. 1, Feb. 1997.
2. Sunghyun Choi and Kang G. Shin, 'Predictive and Adaptive Bandwidth Reservation for Hand-Offs in QoS-Sensitive Cellular Networks,' in *Proc. ACM SIGCOMM'98*, pp. 155-166, Vancouver, British Columbia, Sep. 2-4, 1998.
3. Songwu Lu and Vaduvur Bharghavan, 'Adaptive Resource Management Algorithms for Indoor Mobile Computing Environments,' in *Proc. ACM SIGCOMM'96*, pp. 231-242, Aug. 1996.
4. Y. Iraqi and R. Boutaba, 'A Multi-Agent System for Resource Management in Wireless Mobile Multimedia Networks,' in *Proc. IFIP/IEEE DSOM 2000*, Dec. 2000.
5. D. A. Levine, I. F. Akyildz and M. Naghshineh, 'The shadow cluster concept for resource allocation and call admission in ATM-based wireless networks,' in *Proc. ACM MOBICOM'95*, pp. 142-150, Nov. 1995.