Simulation Framework for the Evaluation of Access Selection Algorithms over Heterogeneous Wireless Networks

Johnny Choque, Ramón Agüero, and Luis Muñoz

University of Cantabria, Santander, Spain
`{jchoque,ramon,luis}@tlmat.unican.es`

Abstract. This work presents the design of a flexible, scalable and easy-to-configure simulation platform, which is primarily conceived so as to evaluate access selection algorithms. As opposed to other similar tools, the simulator offers the possibility to deploy highly configurable scenarios, with various types of users, services, terminals and technologies. It also enables the analysis of large and complex scenarios (comprising many users and access elements), thanks to the abstraction techniques which have been considered during its design phase, without incurring in a high computational overhead. In addition, it can be used to evaluate algorithms using multi-operator strategies, thus leading to multi-access, multi-interface, multi-service and multi-operator scenarios.

Keywords: Access Selection, Heterogeneous, Simulation.

1 Introduction

The wide range of *Radio Access Technologies* (RAT), together with the increasing presence of multi-RAT terminals, lead to a large variety of scenarios in which the access selection procedures have become rather complex tasks. This aspect becomes more relevant considering the great number of parameters which both the operator and the end-user might consider when taking a decision about the most appropriate access.

In this sense, this work presents the design of a simulator tool which aims at covering all the requirements which might be asked by the network, the end-users, the particular services, etc. so as to create an environment to evaluate, on a flexible and scalable way, a great variety of access selection algorithms without imposing a large computational complexity/overhead. In order to show the goodness and potential of the simulator, we establish a highly heterogeneous network scenario, with a broad range of access technologies, user types, services and operators. By using a generic and open access selection algorithm, the obtained results clearly show the benefits brought about by the simulator, which can be used so as to extract interesting conclusions about the behavior of the analyzed network.

K. Pentikousis et al. (Eds.): MONAMI 2011, LNICST 97, pp. 46–60, 2012.

In order to cover the previously mentioned objective, the paper has been structured according to the following points: Section 2 offers a perspective of the related state of the art, establishing the main differences with this work. Section 3 introduces the design principles which have been considered during the simulator implementation to deal with the scalability requirement. Section 4 depicts the simulator software architecture, its internal operation, as well as the tools which have been added so as to guarantee the wanted degree of flexibility. Section 5 introduces a generic access selection algorithm, whose performance will be analyzed with the implemented tool (Section 6), reporting the main results in Section 7. Finally, Section 8 concludes the paper, advocating the lines of research which will be fostered based on the platform.

2 Related work

At the time of writing there are a great variety of simulation tools which might be grouped (on a high level) according to the specific OSI layers they deal with [5]. On the one hand, we can find those frameworks designed to reliably characterize the performance and behavior of the physical and link layers (as well as the interaction between them), like transmitting waves, radio-frequency, propagation, etc; and, besides, there are others which mostly deal with the rest of layers, mainly addressing the analysis of network protocol performances, and they are usually referred to as network simulation tools. This work focuses on this latter category, and we will restrict to such platforms from here onwards.

Another division which can be made deals with the licensing issue of the available tools: some of them are open source, while there are some commercial alternatives. Both of these groups have their own advantages and disadvantages, which the researcher should consider while selecting one alternative. Many tools are initially developed during a research project and, therefore, usually belong to the open source group, like GloMoSim [17] and OMNet++ [15], although in some cases they evolve to commercial versions, like QualNet [13] and OMNEST [14], respectively. Other relevant simulation platforms are *ns-2* and its evolution *ns-3* [3] (open source) and OPNET [10], as the most relevant representative of the commercial tools.

The ultimate objective of all these tools is to facilitate the analysis (by means of simulation models) which is being carried out, offering a set of integrated modules to ease a dynamic and quick interaction; however, in many cases, it is hard (or not possible) accessing the internals of the platform, and thus it can not be adapted to a particular scenario. In this point it becomes sensible asking whether a proprietary/tailored design might be more appropriate. The researcher would have the advantage of knowing the exact characteristics, capacities and limitations of the designed tool. He/she could make the design according to his/her specific goals, and refine the models and results with his/her needs.

On the other hand, it is also clear that the design and implementation of a simulation tool from scratch might require a great effort and temporal investment, so it becomes of great importance the abstractions which are adopted during its development, without the need of thoroughly modeling all the details. Even in the most reputable simulation platforms, some abstraction is done, mostly due to the intrinsic resource limitation (memory, processing time, etc) of the machines to execute the simulations. Although there are parallel processing techniques and distributed strategies [7] to perform large-scale simulations, the requirements (in terms of both hardware and software) usually make them an unsuitable alternative. Besides, adopting any abstraction has the disadvantage of limiting the reliability of the results. Therefore, it is of paramount relevance the tradeoff between the abstraction degree to carry out and the loss of precision which it brings about. Previous works have already analyzed this tradeoff, like [4], which studies the effect of the level of detail for the radio propagation models using various use cases. In such work, the authors state that a simpler model might be a more sensible choice, in those cases in which the main goal of the simulation does not heavily depend on the physical layer abstractions, while being an important part of it.

When deciding on a greater precision when modeling the system to analyze usually leads to a notable increase on the simulation time. This aspect is even more relevant when working with many terminals, base stations or with a wide range of restrictions to be applied by the access selection algorithms. For these reasons, there are a large number of works, see e.g. [16], which have been forced to reduce the number of elements to be considered for the simulations. This work describes the initial design phases, in which we justify the selected abstraction mechanisms, so as to overcome the aforementioned limitations and, therefore, to be able to work with a much greater number of terminals, base stations and constraints.

3 Design Principles

The tool which has been designed and implemented is named *multi-Constraint Access Selection in heterogeneous Environments (mCASE)*, and can be described as an event-based simulator, based on an object-oriented programming language (C++). It allows the creation of different network scenarios, based on the specification, both number and type, of the various elements which are involved in the simulation (access technologies, terminals, base stations, users, services, etc). It has the capacity of replicating the previously analyzed scenarios, since we store not only the main characteristics of the scenario and the number of the different elements, but also the events (with the corresponding information), keeping traces of the node mobility and service dynamics. This would allow to assess the impact of different access selection strategies under exactly the same circumstances.

In order to ensure the pursued flexibility and scalability, it becomes necessary (as was already discussed) taking a number of abstractions so as to allow *mCASE* handling (with sensible computational resources) the vast number of constraints, network elements and events which will be generated on a single simulation run.

3.1 Traffic Model Abstraction

Many works, like [6], have studied traffic modeling according to different degrees of detail, being able to distinguish three levels: session, connection and packet. Each of them can be characterized by a different statistical behavior and thus should be modeled accordingly. Session level is related to the behavior of the user while connecting and disconnecting to/from the system. This should be indirectly handled by *mCASE*, based on the mobility patterns of the end-users. Packet level characterizes the distribution of packets within a particular connection. We decide that this is not needed for *mCASE*, since its main goal does not consider traffic internals. Therefore, traffic will be modeled at a connection level, with different distributions for the calls each of the end-users can initiate (being independent the calls of different service types). Furthermore, in order to abstract the various load units which might be used by each of the involved technologies, we define a generic discrete capacity unit, the so-called *Traffic Unit* (TU) [11], used so as to characterize both the access element capacity and the requirements of the requested services.

3.2 Radio Propagation Abstraction

Taking into account that the main goal of *mCASE* is not to precisely study the propagation channel, but it focuses on the evaluation of access selection algorithms [8], we propose a high level abstraction of the propagation models to be used in the simulator. This implies that we will use non-complicated alternatives, although they might represent, as much as possible, the most relevant characteristics of more reliable models. This strategy has been used in other works, like in [9] which, due to the intrinsic complexity of a complete WiMAX system, proposes a relatively simple model, although mimicking the overall characteristics.

4 Simulator Architecture

mCASE comprises a number of various C++ classes, related between them and which take a specific role within the simulator. As can be seen on Figure 1, the *scenario* class is the one which compiles all the objects which are part of *mCASE*. In this sense, it stores all the information about the terminals and base stations which are created during any simulation run and also coordinates the interaction between the rest of elements. During the network deployment phase, all the objects which represent base stations (BS), terminals and users are instanced. Every user carries a single terminal, but the two objects maintain their particular properties. Besides, a BS has a single RAT, while a terminal can incorporate one or more RATs. Furthermore, during the development phase, all BSs are associated to the operator they belong to; each operator has a number of BSs (which might also differ in the technology they use). In order to be able to analyze situations in which a terminal needs to make a handover between BSs belonging to different operators and assuming that there might be cooperation

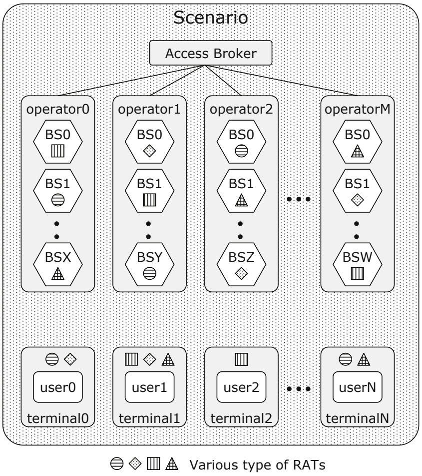


Fig. 1. mCASE high level architecture

agreements between them, we have incorporated in the architecture an *Access Broker* functionality, which will be able to manage the strategies between operators. In this sense, *mCASE* is able to deal with multi-RAT and *multi-operator* scenarios.

4.1 Simulator Configuration

mCASE is a flexible simulation platform, scalable and easy-to-configure. It allows to specify all simulation parameters by a general configuration file, `mCASE.cfg`. This file groups (in various sections) all the properties for each of the objects involved in the simulation; in addition, it also defines the other parameters which are needed for the simulator.

Each of the employed radio technology is modeled with a RAT object, characterized by its coverage area and the load it can support, in TUs. The possibility of being able to use different RAT types can be used to deploy urban scenarios (with a wide range of access alternatives) or rural environments (with few base stations and technologies).

The different terminal types are implemented with the *terminal* object. This object has a *probability* parameter, which indicates the probability for any user to carry such type of terminal. Each of them are characterized by the RATs it incorporates (list of *RATid*), so that a wide range of terminals can be easily added to the scenario (from advanced devices to more modest ones). Likewise, the *probability* parameter of the *user* object is used so as to specify the percentage of each user type in the simulation, which differ on the services they are able to support, so as to include various traffic requirements depending on the type of user. The link between a particular user and the terminal he/she uses is done through the *userTerminal* object, which gives a complete degree of freedom to combine different types of user and services. Regarding the services associated to a user type, each of them is represented with the *service* object, which includes the following properties: time between arrivals (t_{ia}), service time (t_s)[1], requested capacity (TU), as well as a number of additional features, like particular real time requirements, etc. In this sense, it becomes possible incorporating various types of services, like video, voice and data. Finally, *mCASE* offers the possibility that each user has different mobility patterns. This can be used to analyze high-speed users (within vehicles) or pedestrians. The *movement* object is used for that, and it incorporates different mobility models (e.g. *Random WayPoint* and their characteristics).

Regarding base stations, each of the them are represented by the *basestation* object. These have only one RAT and, by means of the *mindistance* parameter, we can carry out a more sensible deployment, by fixing a minimum distance between BSs of the same type and operator. Furthermore, each operator has its own base stations, bringing about the possibility to reflect a great variety of scenarios. Finally, the *access broker* object is added to manage the cooperation strategies between operators and therefore, it includes all the operators which are defined in the scenario.

4.2 Simulator Operation

The *mCASE* modular design allows adding or modifying any part of its structure, so as to add the simulator the possibility to incorporate new functionalities, if this is deemed necessary. It is mostly constituted by the phases which are briefly described below.

1. *Terminal deployment.* During this phase, all the *userTerminal* objects which will be included in the simulation are created. Each terminal is assigned a unique identifier, together with the type of terminal and user, the operator it is subscribed to and a movement pattern; these three parameters are randomly selected (based on the corresponding configuration); to establish the operator, we use the *MarketShare* section of the `mCASE.cfg` file, which establishes the market distribution between the involved operators. Finally, each of the users is randomly placed in the scenario, and the characteristics of the corresponding movement and service patterns are also fixed.
2. *Base station deployment.* During the deployment of the BS, the corresponding *basestation* objects are created, each of them identified by a unique ID, the type of BS [2] and the operator it belongs to. Each operator has a specific number of base stations, and thus the deployment basically assigns their position within the scenario, considering the minimum distance to be kept between BS of the same operator and technology.
3. *Movement and service patterns creation.* Before starting the simulation, we generate, for every user, all the events which represent the movements he/she will make during the simulation life-time. Each movement has a set of parameters to characterize it (identifier, starting and end positions, direction

---

[1] The service uses an *ON-OFF* model, where t_s corresponds to the average duration of the *ON* state and t_{ia} is the average time at the *OFF* state.

[2] Each BS type has a number of parameters: e.g. capacity and range (cell-site).

- angle, speed, etc), as well as the corresponding time event, which is stored in the single queue managed by *mCASE.* Similarly, all services are created for every user; each of them will have as many traces as service types he/she supports, characterized by a unique identifier and the current state (*on*, *off*), as well as by an event indicating the time when the state is changed.

4. *Simulation start.* The event manager stores all the events which have been generated during the previous phase. They are sorted according to the executing time. Then, at the beginning of the simulation, the first event is dispatched, calling the appropriate handler. Depending of the particular event type, there might be cases where other events are generated, being stored at the event queue. When all events are handled or when the finish time is reached, the process is stopped and all the required statistics are collected in output files (according to the configuration file).

It is important to highlight that the first three phases could also be done by means of external files (previously generated by *mCASE*) which would fully characterize a previous setup.

4.3 Access Selection Process

When a service enters its active (*on*) state, this implies that the terminal starts to generate traffic, according to the particular characteristics of such service. Therefore, it becomes necessary to ask the network for resources so as to satisfy such service, triggering an access selection procedure. It is worth mentioning that this process is also started whenever a terminal with an on-going service crosses the coverage boundaries of its current serving BS and also during the application lifetime, periodically, to check whether more appropriate alternatives have become available. The strategy which has been adopted to streamline this process were originally proposed within the *Ambient Networks* project [12], implying the steps which are described below.

1. *Access Detection.* According to the actual position of the terminal, it establishes the set of available base stations, without considering (at this stage), the operator they belong to, or whether they have enough resources to handle the request. The only aspect which is considered is thus the physical connectivity. This is the *Detected Set* (DS).
2. *Access Validation.* Taking the DS as an input, we apply the rules and strategies that the operators might have over the base stations. Based on the type of applied policy, the DS can be reduced, filtering those BSs which do not fulfill it, or it can also modify some of the BS parameters. For instance, it might happen that an operator applies a security constraint that the terminal can not cope with (and therefore it is discarded), or it can apply some rules to modulate the offered price, based on the current load situation. In this latter case, this type of policies involve the BSs of different operators, and the *Access Broker* entity could come into play. To sum up, this phase refines the DS and validates the various parameters of the selected BSs, building the *Validated Set* (VS).

3. *Candidate Accesses.* This latter phase is the one which has the intelligence of the access selection procedure. *mCASE* is flexible enough so as to incorporate different strategies or access selection algorithms, even the more elaborated ones, like those based on multiple attribute decision techniques (*Multi-Attribute Decision Making*, MADM) [16]. It includes a default algorithm based on a weighted sum of various constraints, which will be further depicted in the next section. The outcome is a set of sorted BSs (according to the aforementioned sum). Each of the base stations on this set is a candidate to handle the connection, and it is thus called *Candidate Set* (CS).

Finally, in order to establish the BS which will handle the request, each of the CS elements is asked (in order) about whether it has enough resources to handle the service. If such is the case, those are reserved and if not, the next BS is interrogated. If none of them can handle the service request, then the connection is rejected, assuming that the terminal does not have any available BS to satisfy the particular service demand.

5 Generic Access Selection Algorithm

The simulator includes an access selection algorithm which is based on a utility function $\varPhi_{ij}$, between user i and base station j, which is based on the weighted sum of the various constraints which either the end-user or the network might have. Each of the constraints can be modulated with a different weight, and the access alternative which maximizes the utility will be selected. The use of these weights is a way to provide a great degree of flexibility, since it can give more or less relevance to a particular constraint of the utility function (establishing different access selection strategies). The constraints represent particular aspects which are related to the preferences any end-user (or operator) might have while deciding between various access alternatives. In particular we have considered the constraints which are briefly introduced below.

- *Preferred operator.* This parameter reflects the willingness any user might have to connect, whenever this is possible, to his/her preferred operator (η_i), due to the existence of a contract, better fees, etc. This parameter depends on the particular operator which manages the BS (ζ_j). We will use B_{ij} in the corresponding utility function, defined as:

$$\mathrm{B}_{ij} = \begin{cases} 1 & \text{if } \eta_i = \zeta_j \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

- *Handovers.* Once an end-user is connected to a base station, he/she will prefer to keep it as much as possible, so as to avoid the degradation and overhead which might happen during a handover process. This way, knowing the BS to which the end-user was previously connected, we define $\varGamma_{ij}$ as:

$$\varGamma_{ij} = \begin{cases} 1 & \text{if user } i \text{ was connected to BS } j \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

- *Link quality.* While deciding between various access alternatives, one of the parameters which is traditionally used is the quality of the radio link. Obviously, this is an aspect which heavily depends on the radio technology and the propagation model. In general, we can model it with a decreasing function of the distance to the base station (d_{ij}), in this case, we will use a triangular function [11], which takes the maximum value (1) at the base station position and the minimum (0) at its coverage area edge (ω_j), so that we define the $\mathit{\Delta}_{ij}$ as follows:

$$\mathit{\Delta}_{ij} = \begin{cases} 1 - \dfrac{d_{ij}}{\omega_j} & \text{if } d_{ij} < \omega_j \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

- *Load.* This is possibly the aspect most favored by the network when establishing the CS; the goal here is to balance the load of the various base stations; the current relative load (θ_j) is used, so as when all their resources ($\theta_{\max}$) are available it gets the maximum value (1), taking the minimum one (0) when all the capacity is being used; we define the E_{ij} parameter as:

$$\mathrm{E}_{ij} = \begin{cases} 1 - \dfrac{\theta_j}{\theta_{\max}} & \text{if } \theta_j < \theta_{\max} \\ 0 & \text{otherwise} \end{cases} \qquad (\forall i) \tag{4}$$

From the previous parameters, we define the utility function ($\mathit{\Phi}_{ij}$), which combines them so as to allow a quick classification of the available base statations.

$$\mathit{\Phi}_{ij} = \beta \cdot \mathrm{B}_{ij} + \gamma \cdot \mathit{\Gamma}_{ij} + \delta \cdot \mathit{\Delta}_{ij} + \epsilon \cdot \mathrm{E}_{ij} \tag{5}$$

In order to make such function as flexible as possible, each of the aforementioned parameters is modulated by a different weight; in this sense, β favors the use of a base station belonging to the preferred operator; γ aims at minimizing the handover processes; δ strengthens the use of a base station which has a high link quality; finally, ϵ tries to balance the load of the base stations, by favoring those BS which have more available resources. All the previous definitions assume that the corresponding parameters are within the interval $[0, 1]$, so if we fix that the sum of the four weights equals 1.0, $\beta + \gamma + \delta + \epsilon = 1.0$, we can bound the value of the utility function within the same interval.

A similar algorithm was analyzed in [2], but there are three main differences: (1) service models are added, and users only try to establish a connection when

Table 1. Involved technologies

Operator	**ID**	**Coverage (m)**	**Capacity**	**# Elements**	**Technology**
B	ρ_0	80	5	20	WLAN-B
B	ρ_1	60	8	30	WLAN-A
A	ρ_2	600	20	2	GSM

required; (2) load balancing is added to the list of considered constraints; (3) the decisions are based *only* on the local information available to a particular user, as opposed to [2], in which the optimization problem assumed global information.

6 Using mCASE to Analyze Heterogeneous Access Networks

The wide range of parameters which can be configured within the simulator framework gives *mCASE* the capacity of accepting a great variety of network scenarios. As a starting point, we propose a heterogeneous network scenario, with various technologies and operators. In particular, we will use the three technologies which are depicted in Table 1. The last one (ρ_2) mimics a technology whose characteristics are similar to those of traditional cellular communications (GSM), since it has a notably wider coverage and, in addition, it offers a greater capacity. The two other (ρ_0 and ρ_1) technologies are closer to WLAN access points, with more limited coverage and capacity. The capacity is modeled with the abstraction presented before, based on discrete load units (TUs).

We also assume that there exist two operators. The first one (A) is the traditional one, which manages the base stations of cellular technology, while the second (B) would mimic a novel operator, offering a less-conventional access, by means of WLAN technologies.

We consider a square are of 1000 *m* side, in which the base stations are deployed without any particular previous planning (although limiting the minimum distance between them, when they belong to the same operator and are of the same type). Taking all of this into consideration, the network which will be analyzed is shown in Figure 2. As can be seen the two GSM BSs cover most of area, without a relevant overlap. The area covered by operator B is notably lower, but it provides access alternative within an area the traditional operator does not reach (left top corner).

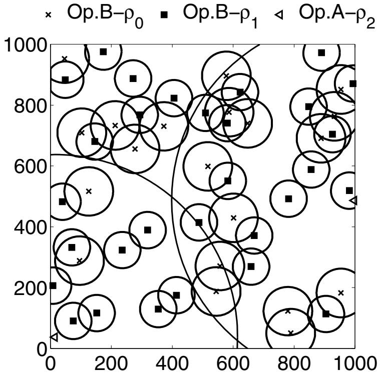


Fig. 2. Network deployment used during the analysis

Table 2. Involved service types

ID	T$_{ia}$ (*s*)	T$_s$ (*s*)	Capacity (TUs)	Service Type
0	120	60	1	Data
1	120	180	1	Voice
2	200	180	3	Video

Table 3. Access selection strategies

Parameter	A	B	C	D	E	F	G	H	I	J	K
β	0.25	1	0	0	0	0.5	0.5	0.5	0.0	0.0	0.0
γ	0.25	0	1	0	0	0.5	0.0	0.0	0.5	0.5	0.0
δ	0.25	0	0	1	0	0.0	0.5	0.0	0.5	0.0	0.5
ϵ	0.25	0	0	0	1	0.0	0.0	0.5	0.0	0.5	0.5

We deploy 200 users, assuming that 60% of them are clients of operator A, while the rest would rather connect to operator B. We also define three types of terminals: a basic one which only has a GSM interface; a medium one, which has two interfaces: GSM and WLAN-A; the third one would be the more advanced one, having the three RATs which are considered within the simulation[3]. The assignment of a terminal to every user is done based on certain probabilities, which were $0.3, 0.4$ and 0.3 for basic, medium and advanced terminals, respectively.

We also define two types of users: regular and business, depending on the services they would invoke. The traffic is modeled as *ON-OFF* processes, defining one or more services which the users might use simultaneously, according the particular configuration of the scenario. In this work, we have established three different services, whose characteristics are summarized in Table 2. Based on them, the regular user (70% of the overall) uses voice and data services, while the business-type also employ the video application. Users are randomly placed within the simulation area and afterwards they move according to the *Random Waypoint* model [1], with a speed selected within the interval $[1, 3]$ (m/s).

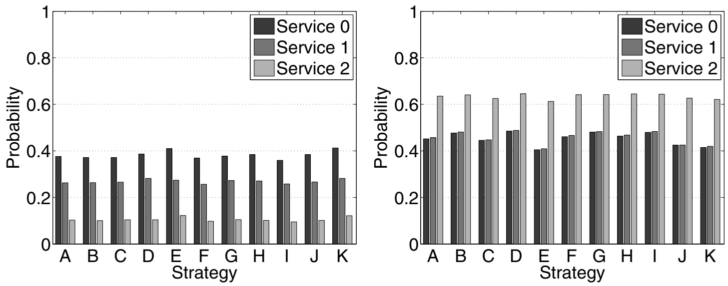
Once the scenario has been described, Table 3 present the access selection strategies which were analyzed. As can be seen, we modify the value which is given to each of the weights, so that every strategy would prioritize some of the aforementioned constraints. Strategy **A** provides the same weight to all the parameters, as a way to see the consequences of a fair weight distribution within the utility function. Strategies **B**, **C**, **D** and **E** focus (each of them) on a single parameter, so as to study their individual effect. Finally, strategies **F**, **G**, **H**, **I**, **J** and **K**, favor two of the used parameters to assess the effect of some of the combinations which can be formulated.

7 Discussion of Results

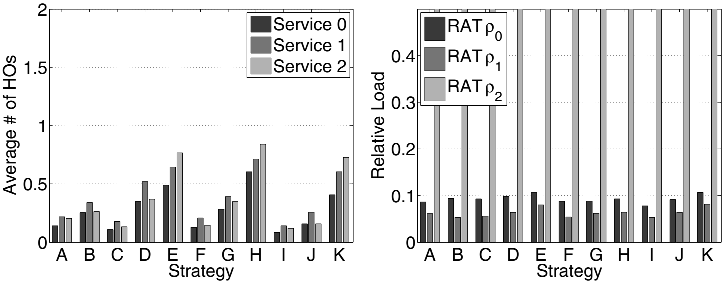
In this section we describe the results which were obtained when using the 11 access selection strategies which have been previously presented. The simulation lasts 2000 seconds, and 100 independent runs are executed, so as to ensure the statistical validity of the results. In addition, since one of the main goals of this paper was to assess the validity and flexibility of *mCASE*, we have made two complementary configurations of the same scenario. In the first one (Figure 3), we use the values provided in the previous section for the terminal distribution,

---

[3] Note that this is just an illustrative example and *mCASE* would allow any combination of the various RATs, according to the configuration depicted in `mCASE.cfg`



(a) Successful services (b) Rejected services



(c) Handovers (d) Load per RAT

Fig. 3. Access selection strategies performance (three terminal type configuration)

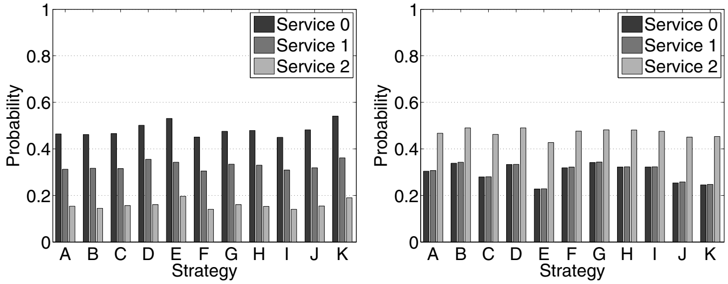
while for the second case (Figure 4) we assume that all the users are equipped with the advanced terminal.

Figure 3(a) shows the probability that a service successfully finishes, while Figure 3(b) represents the reject probability[4]. We can see that the different strategies do not have a great influence on the probability that a service appropriately finishes, but it is clear that services with fewer requirements (in terms of capacity) show a greater probability of being successful (service 0). On the other hand, from the results of Figure 3(b), it can be inferred that there is a certain influence of the strategies over the reject probability, which is lower for strategies **E**, **J** and **K**, which aim at balance the load between the various base stations. Following this way of thinking, we could have expected the same behavior from strategy **H**, but in this case, the influence of the preferred operator constraint leads to higher reject probability (since its base stations get easily saturated).

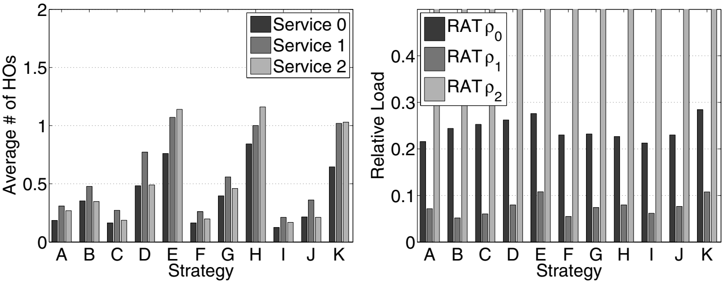
On the other hand, Figure 3(c) yields a great influence of the strategies over the number of handovers. In this case, **C**, **F** and **I** show a lower average number of handovers per service, since they prioritize their minimization in the corresponding utility function. On the contrary, for **J**, the impact of the load balancing weight causes a slight increase on the number of handovers.

---

[4] The sum of both probabilities does not equal 1.0, since there might be some calls which are initiated, but are not properly finished, since there were not resources after a handover; *mCASE* treats these as dropped services.



(a) Successful services (b) Rejected services



(c) Handovers (d) Load per RAT

Fig. 4. Access selection strategies performance (single terminal type configuration)

Finally, Figure 3(d) shows the relative load of all the BS of the same type. For all the strategies ρ_2 BS are almost saturated (load is above 90% for all cases), since they almost cover the complete scenario and, in addition, all users are able to connect to them (all terminals have a ρ_2 interface). On the other hand, the load of the BSs of the novel operator is rather low (for most of the cases it stays below 10%); this is due to the fact that these BSs do not cover a great part of the area under analysis, and (in addition) there are some users which are not able to use such technologies, since they might be carrying a basic terminal (which only has the GSM interface). In any case, it can be seen that for strategies **E** and **K**, the load is slightly higher, since in this case the load balancing parameter is prioritized in the corresponding utility function. Although this could have been expected for strategies **H** and **J**, but the influence of the preferred operator and handover constraints compensate this effect.

On the other hand, Figure 4 can be used so as to assess the influence of changing one aspect of the scenario configuration. The use of an advanced terminal by all the users increase the connectivity changes, and we can see how this is reflected in the corresponding results (by comparing to those obtained with the original configuration). The probability that a service successfully terminates is increased (approximately 5%) for all service types (Figure 4(a)). On a similar way, rejected services (Figure 4(b)) are sharply reduced, ≈ 20% for all strategies, due to the increase of connectivity possibilities. On the other hand, the number of handovers is notably higher for strategies **E**, **H** and **K**, since they prioritize load balancing, and therefore, end-users might be able to use alternative accesses (being equipped with a terminal having all the involved technologies). Finally, the effect of the advanced terminal penetration appears very clearly in the load results (Figure 4(d)), which shows a sharp increase on the load for RATs ρ_0 and ρ_1 (being slightly lower for the latter one, which has less overall coverage) for all cases. The results for ρ_2 are rather similar (above 90%) and the conclusions which were extracted before also applies here. In this case, it is interesting to compare strategies **E** and **K**; it could have been expected a better load balancing for the former one, since the corresponding utility function only prioritized such constraint, but we can see that favoring higher quality links (**K**) also favors a better load balancing.

8 Conclusions

This work has introduced *mCASE*, a proprietary simulation tool which has been designed in order to analyze algorithms in the field of access selection within heterogeneous network environments. We have identified the specific requirements which called for a proprietary tool, as opposed to other available alternatives.

In order to assess the validity and operation of *mCASE*, we have presented a first analysis about the performance of various access selection strategies, which give different priority to a number of parameters of merit. The obtained results not only validates the implementation, but they can be used to establish some tradeoffs between the various constraints which might be considered during the access selection procedures.

We will use the framework provided by *mCASE* so as to thoroughly analyze different strategies for resource management in heterogeneous wireless access environments. We will also study cooperation strategies between operators, price policies, etc. For the sake of completeness, these results will be corroborated and complemented with analytical studies, which will be based on various mathematical techniques, like linear programming [2] or game theory.

Acknowledgements. The authors would like to express their gratitude to the Spanish government for its funding in the following project: “Cognitive, Cooperative Communications and autonomous SErvice Management”, C3SEM (TEC2009-14598-C02-01). Ramón Agüero and Luis Muñoz would also like to thank the European Commission for its funding through the “Scalable and Adaptive Internet Solutions”, SAIL Project (FP7-ICT-2009-5-257448).

References

1. Camp, T., Boleng, J., Davies, V.: A survey of mobility models for ad hoc network research. Wireless Communications and Mobile Computing 2(5), 483–502 (2002)
2. Choque, J., Agüero, R., Hortigüela, E.-M., Muñoz, L.: Optimum Selection of Access Networks within Heterogeneous Wireless Environments Based on Linear Programming Techniques. In: Pentikousis, K., Agüero, R., García-Arranz, M., Papavassiliou, S. (eds.) MONAMI 2010. LNICST, vol. 68, pp. 135–149. Springer, Heidelberg (2011)

3. Free Software GNU GPLv2: The ns-3 network simulator (2010), `http://www.nsnam.org/`
4. Heidemann, J., Bulusu, N., Elson, J., Intanagonwiwat, C., Chan Lan, K., Xu, Y., Ye, W., Estrin, D., Govindan, R.: Effects of detail in wireless network simulation. In: SCS Multiconference on Distributed Simulation Society for Computer Simulation, Phoenix, Arizona, pp. 3–11 (January 2001)
5. Kasch, W., Ward, J., Andrusenko, J.: Wireless network modeling and simulation tools for designers and developers. IEEE Communications Magazine 47(3), 120–127 (2009)
6. Klemm, A., Lindemann, C., Lohmann, M.: Traffic modeling and characterization for umts networks. In: Global Telecommunications Conference, GLOBECOM 2001, vol. 3, pp. 1741–1746. IEEE (2001)
7. Lee, H., Manshadi, V., Cox, D.: High-fidelity and time-driven simulation of large wireless networks with parallel processing. IEEE Communications Magazine 47(3), 158–165 (2009)
8. Lucas-Estan, M., Gozalvez, J., Sanchez-Soriano, J.: Common radio resource management policy for multimedia traffic in beyond 3G heterogeneous wireless systems. In: IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC 2008, pp. 1–5 (September 2008)
9. Miozzo, M., Bader, F.: Accurate Modelling of OFDMA Transmission Technique Using IEEE 802.16m Recommendations for WiMAX Network Simulator Design. In: Pentikousis, K., Agüero, R., García-Arranz, M., Papavassiliou, S. (eds.) MONAMI 2010. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 68, pp. 258–269. Springer, Heidelberg (2011)
10. OPNET Technologies, Inc.: Opnet network r&d simulator (2010), `http://www.opnet.com/solutions/network_rd/`
11. Poyhonen, P., Tuononen, J., Tang, H., Strandberg, O.: Study of handover strategies for multi-service and multi-operator ambient networks. In: Second International Conference on Communications and Networking in China, CHINACOM 2007, pp. 755–762 (August 2007)
12. Sachs, J., Prytz, M., Gebert, J.: Multi-access management in heterogeneous networks. Wirel. Pers. Commun. 48, 7–32 (2009)
13. Scalable Network Technologies, Inc.: QualNet simulation software (2010), `http://www.scalable-networks.com/products/qualnet/`
14. Simulcraft Inc.: OMNEST simulation software (2010), `http://www.omnest.com/`
15. Varga, A.: The OMNeT++ discrete event simulation system. In: Proceedings of the European Simulation Multiconference, ESM 2001 (June 2001)
16. Xing, B., Venkatasubramanian, N.: Multi-constraint dynamic access selection in always best connected networks. In: The Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, MobiQuitous 2005, pp. 56–64 (July 2005)
17. Zeng, X., Bagrodia, R., Gerla, M.: GloMoSim: a library for parallel simulation of large-scale wireless networks. In: Proceedings of Twelfth Workshop on Parallel and Distributed Simulation, PADS 1998, pp. 154–161 (May 1998)