Article

Autonomic Control and Personalization of a Wireless Access Network[1]

N. Blefari-Melazzi[+], D. Di Sorte[*], M. Femminella[*], G. Reali[*]

[+]D.I.E. - University of Rome "Tor Vergata", blefari@uniroma2.it

[*]D.I.E.I. - University of Perugia, {disorte, femminella, reali}@diei.unipg.it

ABSTRACT

As ICT services are becoming more ubiquitous and mobile and access technologies grow to be more heterogeneous and complex, we are witnessing the increasing importance of two related needs: i) users need to be able to configure and personalize their services with minimal effort; ii) operators desire to engineer and manage their networks easily and efficiently, limiting human agency as far as possible. We propose a possible solution to reach these goals. Our vision, developed in the so-called Simplicity project, is based on a personalization device, which, together with a brokerage framework, offers transparent service configuration and runtime adaptation, according to user preferences and computing/networking context conditions. The capabilities of this framework can be exploited: i) on the user side, to personalize services, to improve the portability of services over heterogeneous terminals and devices, to adapt services to available networking and terminal technologies; ii) on the network side, to give operators more powerful tools to define new solutions for distributed, technology-independent, self-organizing, autonomic networking systems. Such systems could be designed so as to be able to react autonomously to changing contexts and environments.

In this paper, we first describe the main aspects of the Simplicity solution. We then want to show that our approach is indeed viable. To prove this point, we present an application which exploits the capabilities of the Simplicity system: a mechanism to drive mobile users towards the most appropriate point of access to the network, taking into account both user preferences and network context. We use simulation to evaluate the performance of this procedure in a specific case study, where the aim is to balance the load in an 802.11b access network scenario. The numerical results show the effectiveness of the proposed procedure when compared to a legacy scenario and to another solution from literature.

To give ample proof of the feasibility of our solution, we also designed and implemented a real prototype. The prototype enables not only the load to be balanced among different 802.11 access points, but also network and application services to be differentiated as a function of user profiles and network load. The main aspects of this prototype are presented in this paper.

Keywords: service personalization, service portability, service adaptability, user profile definition and handling, user mobility, auto-configuration of terminals, middleware, access network context, load balancing, IEEE 802.11b, numerical analysis, prototype development

Corresponding Author:

Nicola Blefari-Melazzi, Ph.D.

Dipartimento di Ingegneria Elettronica, University of Rome “Tor Vergata”

Via del Politecnico, 1 – 00133 Rome – Italy

Tel: +39 06 7259 7501, Fax: +39 06 7259 7435, e-mail: blefari@uniroma2.it

---

[1] A preliminary, shorter, version of this paper was published in the Proceedings of International Teletraffic Congress (ITC19), Beijing, China, August 2005: N. Blefari-Melazzi, D. Di Sorte, M. Femminella, L. Piacentini, G. Reali, “A distributed brokerage framework to balance the load in wireless networks”.

I. INTRODUCTION

As technology develops, people are using an increasingly broader and heterogeneous range of ICT (Information and Communication Technology) devices and network-based services. The result is an enormous burden of complexity on the shoulders of users, service providers and network operators. Excessive complexity, in turn, creates obstacles to effective exploitation and acceptance of beyond 3G systems and paradigms such as ambient intelligence, context-aware services, pervasive computing and novel access technologies. Heterogeneous services, terminals and networks create a complexity barrier not only to end-users, but also to operators, who have to devise and deploy tools and procedures to engineer and manage their networks efficiently.

The strategic goal of the European Union co-funded Simplicity project[2] [1][2] is to simplify the process of using and managing current and future services, by designing and deploying a brokerage level allowing i) an easy personalization of services to match user preferences and needs, ii) a seamless portability of services, applications and sessions across heterogeneous terminals and devices, iii) a smooth adaptation of services to available networking and support technologies and capabilities.

A key attribute of Simplicity is re-configurability at various levels. Re-configurability is typically understood as operating at lower layers (e.g., software defined radio, which indeed may be an "add on" to our solution). However, to integrate different paradigms from the user point of view, it is necessary, wherever possible, to break the logical wires which still tie mobile users to networks and services at upper layers as well. Thus, heterogeneous and mobile access networks can be fully integrated, as IP has glued heterogeneous networks. In order to move towards total re-configurability, the Simplicity project proposes a personalization approach, based on a user profile.

In our view, each user will be characterized by a personalized profile providing access to different services and networks, and using different classes of terminals. Users will enjoy the automatic selection of services appropriate to specific locations, the automatic adaptation of information to specific terminal devices and user preferences, and the easy exploitation of different telecommunications paradigms and services. The user profile will be stored in a so-called Simplicity Device (SD). Alternatively, storage in the SD might be limited to a "pointer", making it possible to download the whole profile from the network. Though it seems natural (from our own

[2] The Simplicity project lasted 26 months (January 2004 - February 2006), and included 11 major European industrial organizations, network operators, small and medium enterprises, research labs and universities [1]. The project ended on the 10th of February with a final review that classified it as "highly successful". The project web site contains papers, public deliverables, movies, presentations and other material that describes our work in the last two years.

everyday experience of 2G systems) to think of the SD as a physical device (e.g., an enhanced SIM card, a Java card, a Java ring, a USB stick, a sensor, etc.) the SD could also be implemented as a network location or a software agent. If the SD is a physical device, users could personalize terminals and services by simply plugging the SD into the chosen terminal. One of the novelties of the SD is that it is not tied to a single networking environment, or to a single class of user terminals (such as the GSM SIM, which, incidentally, is rarely used to routinely transport preferences from one terminal to another, except when a new phone is purchased).

The architecture of the Simplicity system encompasses three main components: the Simplicity Device, the Terminal Broker and the Network Broker.

The role of the SD, as discussed above, is to store a user's profiles, preferences and policies. It also stores and enables the enforcement of user-personalized mechanisms to exploit service fruition, to drive automatic adaptation to terminal capabilities, and to facilitate service adaptation to various network technologies and related capabilities.

The Terminal Broker (TB) manages the interaction between the information stored in the SD and the terminal, into which the SD is plugged. The SD enables the TB to perform actions such as the adaptation to networking capabilities and to the environment, service discovery and usage, and the adaptation of services to terminal features and capabilities. The TB also caters for user interaction with the overall Simplicity system (including network technologies and capabilities).

The goal of the Network Broker (NB) is to provide support for service advertisement, discovery and adaptation. Moreover, it orchestrates service operation among distributed networked objects, taking into account issues related to the simultaneous access by several users to the same resources, services and locations. Other functions of NB include sharing/allocating available resources, and managing value-added, networking functionality, such as service level differentiation and quality of service, location-context awareness and mobility support.

The enhanced capabilities of this distributed brokerage framework can be used not only to simplify the ICT users' life, but also to give operators more powerful tools to define new solutions for distributed, technology-independent, self-organizing, and autonomic networking systems. Such systems could be designed so as to be able to react autonomously to changing contexts and environments in a heterogeneous framework.

To sum up, the operators' possible interest in a Simplicity-like solution could be twofold: simplify service fruition on the user side, and service management on the network side.

With Simplicity, users will be able to personalize and customize their services with minimal effort, and operators will be able to engineer and manage their networks easily and efficiently.

In this paper, to show the potential of our solution (and to also show that it works), we apply it to a concrete problem as perceived by an operator: network selection. In other words, we select a specific case study, the problem of the network selection, and show how this problem can be solved with our approach.

Network selection may be defined as the process of selecting the most appropriate network point of access. If the process is user-driven (e.g., as in Wi-Fi hot-spots), the decision is taken by the final user and the metrics and criteria to select the access are typically user-oriented (e.g. price minimization, data rates maximization). If the process is operator-driven (e.g., as in cellular GSM/GPRS/UMTS systems, once the user has selected an operator), the decision is taken by the system (i.e., terminal+network) and not directly by the user. In this case, metrics and criteria to select the access are typically operator-oriented (e.g., maximization of the revenue, load balancing target, and more in general, service and network management objectives).

In our case, we assume that the system drives Mobile Nodes (MNs) to the most suitable point of access (Access Point, AP) according to operator’s policies, which, in principle, may consider a number of inputs to the selection process from both the user (e.g., user profile and preferences) and the network side (e.g., network and application service status). Obviously, MN-AP association control can be used to achieve a number of different goals. In this paper, in order to show the capabilities of the Simplicity system in a specific case study, we assume that the target of the network selection process is to improve load balancing performance. The scenario we consider for both simulative and experimental analysis is that of a single administrative domain offering a 802.11 wireless access. In principle, the mechanism can be extended to a heterogeneous scenario in which mobile terminals may have more than one wireless interface (multi-mode terminals). We introduce a monitoring procedure capable of tracking the access network context, including the state of resources, the amount of service demand and the available access points. As regards the latter, we assume that the TB is able to perform frequency scanning and to listen to Layer 2 (L2) beacons periodically transmitted by surrounding wireless access points, and to learn their identities (L2 IDs) in accordance with [5]. These access points are possible candidates to attach to and are communicated to the NB. Then, an appropriate selection procedure running in the NB uses this set of information to drive MNs towards the most appropriate point of access among those available in the surrounding area. Moreover, user information contained in the SDs can be used to predict users' behavior and to differentiate network services according, for instance, to the users' roles and tariff profiles. The whole procedure is run by the system in a way that is completely transparent to the users, who could be unfamiliar with networking aspects. Of course, in principle, users could also be involved in the decision process, depending on local policies and administrative rules/constraints.

To sum up, the contribution of this paper is twofold.

First, we present the Simplicity approach and its system architecture. Thanks to this solution and to its personalization/customization/configuration features, it is possible to satisfactorily and efficiently solve many problems and deploy several “applications”.

The second contribution of this paper is the description, implementation, and analysis of an example of a Simplicity “application”: the system-driven access network selection. Our simulative and experimental studies have shown that exploiting the Simplicity framework can indeed improve performance. In more detail, as regards the network selection procedure, we show:

1. its functional and architectural characteristics;
2. its performance evaluation, in terms of load balancing, in an 802.11b scenario. The numerical results, obtained by means of an NS-2 based simulator [11], show the effectiveness of our architectural approach, when compared to other solutions under different network conditions and mobility models. We remark that our main goal is to compare different *architectural* solutions without any specific focus on load balancing metrics, which have been widely discussed in many previous publications and standardization groups. In more detail, three different approaches were compared: (i) the legacy mechanism, where the point of access is selected by the terminal according to the signal power level; (ii) the Simplicity solution, according to which the point of access is selected by the network exploiting the brokerage framework distributed among the network and the terminal; (iii) the approach described in [4], based on a completely distributed architecture based on agents running at each access point without any support from the terminal;
3. its feasibility “proof”, obtained by designing and implementing a true prototype of the Simplicity system, which can manage the access network by also taking into account user context (read from the SD). This prototype enables not only the load to be balanced among different network accesses, but also network and application services to be differentiated as a function of user profiles and network load. We also report some measurement results concerning signaling load, handover time and packet losses.

As a final comment, we point out that our network selection procedure is backward compatible with current systems, in the sense that legacy devices can continue to work without problems, even though they will not benefit from the “new” features.

The paper is organized as follows. In the next section, we give an overview of the Simplicity system. In section III, we frame our activity as regards network selection within the current state of the art. In section IV, we introduce the Simplicity access network control procedure. Section V presents and discusses both a quantitative (from simulation analysis) and a qualitative comparison between the Simplicity approach and other architectural solutions. In section VI, we describe the prototype developed in our laboratory and present some results from lab experiments. Finally, in section VII, we give some final remarks.

II. THE SIMPLICITY SYSTEM

In this section, we describe the core Simplicity system architecture.

A number of applications may be deployed on top of the Simplicity system. There are four application categories: (i) applications explicitly developed for the Simplicity system; (ii) web-based applications, showing how Simplicity can complement existing web applications and enhance user experience; (iii) external applications, showcasing how existing, standalone applications can be integrated with the system; (iv) operator-centric services. To demonstrate a representative subset of the capabilities of the system, a number of scenarios were considered within the framework of the Simplicity project. In this paper, we will present the design and performance evaluation of an operator-centric access network control application in a so-called Campus Network scenario. The interested reader may find the detailed description of other key Simplicity applications and relevant scenarios in [28][33]. Such applications include: Adaptive Multimedia Messaging (an enhanced chat application that allows users to exchange pictures and video), MyPC (it customizes the software environment according to the user's own settings), Home Entertainment Service (it complements normal web applications and enhance the user's experience with personalization features), Auto-form filling (specifically designed for mobile phones), Tour Guide (an example of how an existing external application can be adapted to exploit the Simplicity vision).

The following sub-sections briefly describe the Simplicity layered architecture and architectural and functional entities. More details on the Simplicity architecture can be found in [32]. The project also produced a thorough analysis of the state of the art of related technologies, standards and works, which is available in [43].

A. High level architecture

This section describes a logical, layered, middleware architecture for user-centric service provisioning, supporting simplified use and personalization of heterogeneous services, networks, and devices. The key enablers for this vision are:

- personalization and adaptation functions allowing the personalization of services to match user profiles (including user-defined preferences), the current user context (the environment in which the user is located, the status of the user), and the resources available on user devices and in the user environment;
- policy management allowing users, service providers and network providers to manage policies for various aspects of service provisioning, (e.g., personalization, adaptation, authentication/authorization etc.).

Most traditional middleware architectures are based on a layered approach. To facilitate the re-use of functions already present in traditional middleware, we also follow a layered approach, adding a specific sub-layer to providing users with 'simplicity'.

The overall functionality of Simplicity is divided into Functional Entities (FEs) that bear no relationship to its physical components. FEs are implemented by a set of interacting software and hardware components, distributed across the Simplicity system. They are grouped in three layers (see Fig. 1): Network Layer, Service Support Layer and User Layer. The layering concept we use is looser than the layering in the classical OSI modeling. Each layer consists of a set of FEs. FEs in one layer will collaborate with entities in other layers to provide the required functionality. The main goal of the layering is to group functions related respectively to the network side, to services and to users. This approach follows the same guidelines proposed by the Wireless World Research Forum [46] in WG2.

The network support layer provides functions for network communications control in heterogeneous networks. The service support layer contains most of the functions of traditional middleware, such as AAA (authentication, authorization and accounting) and profile management. The user support layer supports autonomous, proactive agent behavior that is not present in traditional service middleware. This layer is designed to simplify user interactions with the system, exploiting user context, preference-based personalization and autonomous coordination to provide user-centric services.

B. Architectural entities

B.1. Interfaces

The interfaces between the system entities must, of course, be clearly defined. In particular, three fundamental interfaces were specified (refer also to Fig. 2): 1) the interface among brokers, handled by means of a Simplicity Asynchronous Event Protocol (SAEP); 2) the interface between brokers and the external applications (3rd party applications, 3pA) willing to exploit the system, called Simplicity Applications Interface (SAI); 3) the interfaces between the TB and the SD, called SD Access Interface (SDAI). Some details about these interfaces are given below, while describing system entities.

B.2. The Simplicity Brokers

Simplicity brokers are software systems instantiated in user terminals (Terminal Brokers, TBs) and in network servers (Network Brokers, NBs). Each broker consists of a central entity called Mediator and a number of loosely coupled subsystems attached to it, each one offering different functionality (see Fig. 2). Broker subsystems are stand-alone components, which provide functionality to other subsystems and specialize in a particular task. This design approach has the advantage that it enables a flexible encapsulation of a new functionality within a broker, without restricting pre-existing functionality. Communication between the subsystems is asynchronous and event-based. It is carried out by means of the Mediator, which has the ability to invoke the appropriate event distribution policies according to the "event context" (message type and sender) in order to filter, adapt and relay events among subsystems. A Mediator, according to the enforced policy, may reject an event or forward it to the intended receiver (with or without modification). It can also send it to a remote broker in a way which is transparent to the subsystem. This is done by passing the event to a specific subsystem, the Simplicity Broker Communication (SBC). In this case, the SBC has to decide whether the event needs to be targeted to a specific broker or broadcast to a group of brokers.

The implementation of a distributed architecture requires close inter-working of software running on different machines. Fig. 3 depicts the protocol stack for the interaction within the Simplicity framework.

Simplicity specifies the inter-SBC interface as an asynchronous XML based protocol, called SAEP. SAEP describes the structure of the messages that SBCs exchange, along with the necessary exchange patterns and bindings with underlying protocols, which in our case is SOAP (Simple Object Access Protocol, [27]); SOAP messages are transported by using the HTTP protocol. SOAP was selected because it is an XML based protocol, and this favors interoperability between different implementation platforms.

In principle, SAEP protocol messages can be exchanged between brokers using transport facilities provided by protocols such as ASAP [47], or alternative publish/subscribe communication mechanisms such as “tuple spaces” (e.g., [48]). This means that in principle different mappings can be defined.

The current architecture and implementation ensures that communication may take place only among brokers that have already registered over an IMS (IP Multimedia Subsystem) infrastructure (e.g., see [49]), which can be co-located with the NB. The IMS architecture was originally defined and standardized by 3GPP and can be regarded as a collection of FEs and interfaces used by a network service provider to offer SIP (Session Initiation Protocol)-based services to IMS subscribers.

The authentication mechanisms, therefore, ensure that a broker will only contact, on its own initiative, brokers that are authenticated, and, therefore, legitimate. Thus, the Simplicity services and Simplicity enabled 3rd party services are provided after successful access and registration of the terminal with the network.

B.3. The Simplicity User Profile and the Simplicity Device

The Simplicity User Profile (SUP) was designed taking into account two complementary views regarding structure and content. The first one is represented by the 3GPP Generic User Profile, in particular the Data Description Model (DDM) [34], which describes a syntax to express a “generic user profile” (XML-based). The second view was inspired by the so-called Simplicity Information Model, which defines the different types of user/services/capabilities information that a profile for 3G & beyond systems must contain. It is expressed in UML diagrams and is independent of any specific application, protocol, platform, data storage and access technology and can be extended to include additional information. The SUP data are physically distributed for efficiency, but semantically represent a whole, characterizing each user. The SUP is stored in the SD in a safe, secure way. The ideal SD would have an unbounded, secure and reliable memory space for storage, powerful processing capabilities to manage data and a minimal physical size. The project developed four prototypes using devices very familiar to end-users: i) a Bluetooth phone SD (BTSD), which exploits the memory, connectivity and processing capabilities of J2ME and Bluetooth-enabled phones; ii) a Java Card SD (JCSD), which implements the SD as an applet deployed on a JavaCard; iii) a Flash Memory SD (FSD) which uses memory cards or memory sticks and iv) a Virtual SD (VSD), consisting of a pure software implementation (see Fig. 2).

The heterogeneity of the different SD implementations was addressed by a special subsystem residing in any TB, the Simplicity Device Access Manager (SDAM). The SDAM uses controllers that provide a unique interface for developers, regardless of the particular SD implementation. A single TB may host more than one controller, enabling the use of different kinds of SD with the same TB. Depending on the nature of the SD, communication with the SD itself may be based on asynchronous messages over Bluetooth connections, exchanges of Java Card APDUs, or other mechanisms. The SDAM offers to other Simplicity subsystems an interface based on the XQuery/XPath language specifications. The SDAM also enables the SD to exploit functions offered by the NB and TB. For example, when the SD has limited or no memory capacity, data are stored in a network repository called Simplicity Data Storage, SDS, and a pointer is used to link the SD to the location where data are stored.

It is worth noting that, beyond the terminal registration to the network described above, the Simplicity system also foresees the authentication of the user towards the SD, for instance when the latter is inserted in the terminal. User authentication is the most common kind of authentication and it is typically based on a secret login-password or a Personal Identifier Number (PIN).

B.4. The Simplicity Personal Assistant

The Simplicity Personal Assistant (SPA) supported by a number of “core” Simplicity subsystems, offers mechanisms to proactively assist users in their interaction with the system. It supports personalization and adaptation for services and user interfaces, as well as automatic service subscription/invocation, including configuration of services for used devices and session transfer across devices.

The SPA interacts with users via a User Interface and it is attached to the broker system through the Simplicity Application Interface of the relevant manager (SAIM). The SPA User Interface combines several functions, e.g., a desktop manager, a task manager and a system tray, to name only a few. A reduced version of the SPA, named miniSPA, was also developed for Bluetooth phones embedding an SD (BTSD).

Note that, in order to integrate third party applications, the TB may also offer the Simplicity Applications Interface to third party applications. The SAI API allows applications to benefit from Simplicity features.

B.5. Other core subsystems in the Simplicity framework

Profile Management Subsystems: Upon a subsystem request to retrieve/modify the SUP, the Profile Management Subsystem interprets the request, checks the requester credentials, checks the access rights associated to the requester contained in the SUP, and consequently allows or denies access to profile data. The profile data is then presented in a suitable format which may be, depending upon the request, plain text, xml, or even custom structured data. In order to do this, it exploits the aforementioned interfaces exposed by the SDAM. Security and data presentation are in this way ensured.

Policy Subsystems: One important goal of Simplicity is to build a flexible adaptive system. This can be achieved by "Subsystem Policies", which define choices in the behavior of a system, using context information. Simplicity provides two core subsystems dealing with policies. The policy subsystem, representing the Policy Decision Point (PDP), makes decisions based on a specific subset of context information and policies, which are provided by other Simplicity subsystems. The policy management subsystem is in charge of administrating all policies that are used inside the whole Simplicity environment.

Service Manager Subsystems: The functionality of the Service Manager on the TB is to request services from the Service Manager on the NB, based on the user's profile and on current needs. The Service Manager searches for the services from service registries and provides the Service Subsystem with the necessary information from the services which have been found. Finally, if the user chooses to subscribe to a service, the SUP is suitably updated.

User contracts and pricing manager: This subsystem is responsible for providing information as regards the costs of services and network accesses.

Capability Manager: The Capability Manager stores and provides the hardware and software configuration of the user terminal system. The complete terminal configuration is stored in an xml file.

Access Network Subsystems: The purpose of the Access Network Subsystem on the TB (ANS-TB) is to auto-detect network parameters (IP address, subnet mask, gateway IP, etc.) of every terminal network interface. The Access Network Subsystem on the Network Broker (ANS-NB) is, on the other hand, responsible for providing information about a given user's current connection with the system.

Location Manager: The Location Manager subsystem uses e.g. RFID technologies to provide information as regards a user's position.

III. ACCESS NETWORK SELECTION: RELATED WORK

Network selection may be performed by the human user or by a software tool running in the terminal on his/her behalf and according to his/her policies (user-driven selection). Alternatively, the selection process can be executed by the system (in the terminal and/or in the network), according to operator policies (operator-driven selection). In the latter case, the network selection can be triggered either by the terminal or the network [44].

The choice of the "best" network Access Point (referred to below as AP) has different implications, depending on whether the reference environment is made of homogeneous or heterogeneous access networks, but is of interest in both cases, and is of growing importance as the offer in terms of number of wireless accesses and services increases. In fact, several standards and technological solutions take this issue into account.

The IEEE 802.21 [16] working group aims to specify media-access independent mechanisms which optimize handovers between heterogeneous 802 systems and between 802 systems and cellular networks. The 802.21 draft standard specifies a set of handover-enabling functions within the mobility-management protocol stacks of the mobile nodes and of the network elements that provide mobility support. These functions are performed by the so-called Media Independent Handover Function (MIHF), which is logically defined as a shim layer between L2 and L3. Its goal is to help the higher layer mobility management protocols to have a global view of heterogeneous networks to be found in a given area and to perform effective network selection for both horizontal and vertical handovers. In particular, the MIHF has to collect information (denoted as 802.21 Information Service Elements, ISEs) relevant to the heterogeneous network accesses existing within a geographical area. The 802.21 standard classifies the ISEs into three categories: (i) General Network Information (e.g., network identity, location, network operator); (ii) Link Layer Information (e.g., channel, frequency, physical types, data rates, security, quality of service); (iii) Higher Layer Information (e.g., IP configuration, Virtual Private Network, types of applications, pricing, roaming partners). This set of information may be retrieved by the MN (and thus from the user) from an information server in the network by means of MIH message exchanges, and be used as input to the AP selection process. Note that some information can also be made available directly by the L2 (e.g., by means of beacons broadcasted by 802.11 APs). In the latter case, the MIHF can get this information via a properly defined, local interface between MIHF and L2.

As an example of information provided with the aim of facilitating network selection, we quote the decision of 802.11 TGu to address the distribution of the following information [19][20]: authentication and enrolment methods, roaming agreements, and application service offer. Price-related information is set as an optional requirement.

It is also worth mentioning the work carried out by the IETF Seamoby Working Group, which proposed a Candidate Access Router Discovery (CARD) protocol [5]. This is a high-level protocol, which enables a network-assisted mechanism for the quick discovery of the surrounding wireless environment, and in particular of IP addresses and service capabilities of candidate access routers to hand over to. The procedure is distributed and involves all access routers (ARs). The rapid acquisition of IP addresses enables MNs to speed up the (horizontal or vertical) handover process and thus to perform a seamless handover [21][22]. On the other hand, information about service capabilities (e.g., in terms of security and quality of service) is important for the selection of the most appropriate wireless access (target access router). The main aim of this solution is to improve the efficiency of attachment procedures in the case of a change in location, and it does not provide any support when an MN is not already attached to a network and wants to find a suitable access. In fact, since link-layer decisions depend on information retrieved from a high-level protocol, the MN needs prior association and authentication with a network point of access, in order to enable the mechanism. A performance analysis of different CARD solutions in terms of discovery time and signaling burden may be found in [23], whereas an analysis of the capabilities needed to drive L2 and L3 handovers is shown in [24].

As for the selection of the best point of access, it is worth noting that a very critical point is the definition of metrics able to quantitatively compare heterogeneous points of access, to drive optimal vertical handovers. In fact, metrics which account for physical layer parameters differ greatly in different wireless networks. Such a topic has been widely studied over the last decade; the interested reader should refer to [35][36][37][38].

Other works on this topic focus on architectures and protocols to manage layer 2 handovers in a homogeneous 802.11 access section, which is the reference access network that we consider in both simulation (section V) and experimental (section VI) analyses.

In the typical implementation of real 802.11b systems, a terminal selects the AP exclusively on its received signal strength, without taking into account the current traffic load.

In [4], the Authors propose a distributed architecture based on agents running on 802.11 APs, called Load Balancing Agents (LBAs). In the following, we will refer to this approach as LBA-based solution. The APs exchange traffic load information to cooperatively balance the traffic among them, by forcing the handover of a subset of MNs associated with an overloaded AP. The drawback of this approach is that the procedure is unaware of which APs are available to an MN, and this implies that an MN which is forced to leave the current AP could be denied a subsequent association and could remain connectionless. The advantage of this approach is that it works with standard wireless LAN stations. However, in our opinion, if load balancing decisions are taken on the network side, information about the access network context obtained from the terminal side is necessary to make the approach effective. The performance of the LBA-based solution will be analyzed in section V by means of simulations.

Another alternative, proposed in [6], is that users could be explicitly requested to cooperate actively, by physically moving towards specific locations within the network for load balancing purposes. In [8][9] Authors assume that selection decisions are taken by MNs and propose to embed load balancing information in 802.11 beacons, which are periodically transmitted by APs and can be received by MNs. This approach would require some modifications to the structure of the IEEE 802.11 beacon. As a result of this approach, some vendors have introduced proprietary solutions for load balancing purposes in 802.11 wireless networks (e.g., see [10]). However, this solution leads to a lack of interoperability between different vendors. In addition, moving all context information, together with the decision process, towards terminals has two main drawbacks. The first one is that, if an operator does not want to divulge network-related information for security and/or business reasons, it is not possible to embed such information into beacons, which are accessible by everyone. The second drawback is that, if the decision is taken by the terminal, it is not possible to perform advanced management operations to control the wireless access network.

Another approach is proposed in [39], where the Authors assume that each mobile device is equipped with client software to monitor the wireless channel quality that the user is experiencing from each of its nearby APs. The client gives this information to a network control centre (NOC) which determines the users' associations and updates the clients about its decisions. Accordingly, the users switch their associations. The MN-AP association control is an NP-hard problem and the Authors present several approximation algorithms under different, firm assumptions (greedy users and advanced scheduling algorithms in the APs). In principle, the algorithm is run offline and synchronously for all MNs and is to be repeated each time a user arrives or departs. The drawbacks are the strong assumptions on which the association control algorithm is based and the fact that the approach does not prove to match a dynamic service pattern.

IV. ACCESS NETWORK CONTROL: THE SIMPLICITY APPROACH

A typical goal of a network manager is to optimize network operation in terms of quality of service (users' side), throughput and load balancing (operator's side), to minimize failure probability by deploying well-designed redundancy and pre-empting critical situations, and minimizing the load on human operators. In general, network control actions depend on user-related information, on the spatial distribution of users over the coverage area, and on the characteristics of the network, such as network topology, network resources, and available tuning capabilities. If this data is largely unknown, network management has to be essentially reactive. Our goal is to collect the largest possible amount of information and to use it as input for a decision engine. In this regard, we argue that the goals mentioned above could be achieved more easily by exploiting certain specific features of the brokerage framework defined within the Simplicity architecture.

The reference environment is an IP network managed by a single operator (e.g., a Campus Network) with a number of access routers (ARs) which control a set of heterogeneous APs. Handovers can be both inter and intra-technology, intra and inter AR, at both layer 2 and layer 3. As regards mobility among ARs, this can be managed with the Mobile IP protocol or by means of higher layer solutions (e.g., SIP) [15]; here, without loss of generality, we assume Mobile IP is used in order to maintain session continuity. We assume that mobile terminals may be provided with a number of wireless interfaces (multi-mode terminals).

In the following, we first give details about the network control procedure and frame such an approach within the Simplicity system architecture. We then present some considerations about the NB deployment.

A. The access network control process within the Simplicity framework

For access network control purposes, the most important characteristics of the Simplicity brokering framework are:

- the capabilities of the SD-enabled terminals, which can be exploited to assist the network in monitoring availability and performance of wireless coverage;
- the user information retrievable from SDs (profiles and preferences), which can be useful to predict users' behavior and to differentiate network services according, for instance, to users' roles and tariff profiles.

We assume that the network selection process is operator-driven. Users may only provide the operator’s management actions with implicit inputs by means of the information stored within the SDs. The NB is the entity acting as decision maker, i.e., it is the functional entity in charge of taking decisions on the operator side. The TB is in charge of assisting the NB, by providing inputs regarding the access network context (i.e., the radio access technologies currently perceived by the terminal through a frequency scanning), and by acting as Policy Enforcement Point (PEP), e.g., to force handovers. MNs are able to perform frequency scanning and listen to L2 beacons periodically transmitted by surrounding APs, and learn the L2 IDs of the APs (i.e., the unique identifier of an AP [5]); these APs are the candidates to hand over to. Note that the ability to perform frequency scanning is a minimum requirement for all wireless technologies. In addition, we assume that a monitoring process provides the NB with information concerning the current status of availability of wireless resources at the APs.

The NB output consists of network control decisions. By applying properly designed algorithms, the NB can not only dynamically optimize the distribution of mobile users within the wireless section, but also activate/deactivate APs, control AP transmission power, limit the fruition of application services and so on.

In this paper, we test the ability of the Simplicity system to balance the load by driving MNs towards the most appropriate AP. We assume that the wireless coverage is such that several wireless accesses are available (dense wireless coverage), so that a load balancing mechanism makes sense. Balancing the traffic load is obviously convenient not only for management purposes, but also for the users, who can thus experience an improved level of service. The selection process may be invoked: (i) when the terminal is switched on; (ii) periodically, according to operator's policies; (iii) when a handover is needed. The overall process is sketched in Fig. 4. When a terminal is switched on, the TB exchanges information with the NB through a default network connection and the NB drives the terminal towards the most appropriate wireless access.

The distributed nature of this procedure raises a number of security and privacy considerations, both for users and for service providers. These aspects are beyond the scope of this paper; however, the Simplicity project did, in fact, analyze them (see [32]).

We point out that our proposed approach can be used in a heterogeneous wireless access network environment, including several different access technologies.

With reference to the detailed Simplicity system architecture described in section II, the access network control process is accomplished by new application-specific entities: the Campus Network Access Control (CNAC) subsystems, one in the TB and one in the NB (see Fig. 5).

The CNAC subsystem in the TB is in charge of

- communicating with the CNAC in the NB via the mediator subsystems;
- retrieving user and terminal side information and sending them to the CNAC in the NB;
- interfacing with the heterogeneous Network Interface Cards (NICs) of the terminal with the aim of (i) driving frequency scanning operations and retrieving the access network context from the terminal side; (ii) monitoring the current wireless connection; (iii)

forcing the wireless connection of the terminal towards a given point of access according to the commands from the NB;

- triggering a new selection process (e.g., when the quality of the current access is definitely decreasing).

As regards the CNAC in the NB,

- it communicates with the CNAC in the TB via the mediator subsystems;
- it is configured with specific management policies, defined by the network operator with the aim of improving network performance while maintaining user satisfaction;
- it executes a network monitoring function to periodically collect information concerning the status of the network points of access, for instance from bandwidth monitors installed within them;
- it periodically triggers the network selection process.

Thus, all the user and network context is stored by the CNAC in the NB. As regards the act of taking decisions, in principle such a task may be performed by either the CNAC or the PDP subsystem.

B. The NB architecture

In principle, the NB may be implemented either as a centralized entity or distributed over several network entities. In the latter case, we assume that the NB instances are co-located with ARs. This means that each AR is in charge of directly managing only the APs under its control.

As regards the issues specifically related to the case of 802.11 access networks which we analyze in the remainder of the work, a large number of vendors currently envision the presence of a centralized controller entity addressing management issues. The interested reader should refer to the work of the IETF CAPWAP Working Group [17][18], the main goal of which is to define a standardized, interoperable interface between the APs and the controller without involving user terminals. An important feature of the Simplicity architecture, and a difference with respect to the CAPWAP centralized architecture, is that the NB may also communicate with Simplicity-enabled terminals.

The centralized implementation of the NB is a reasonable choice for small networks. On the other hand, if the network is large, the distributed solution could, for reasons of scalability, be mandatory. In this case, each AR has to be able to retrieve the network status of its APs and of APs belonging to neighboring ARs, the coverage area of which overlaps with its own. In this way, MNs can be driven by the current AR to perform handovers at both layers 2 and 3. The discovery of the surrounding wireless coverage map at each AR (discovery phase) may be either manual or automatic. In the latter case, the network has to be able to self-discover its wireless coverage. In operation (steady phase), neighboring ARs have to exchange the service capabilities of their APs. The analysis of the self-learning discovery phase and of the steady phase, along with a quantitative evaluation of discovery time and signaling burden, can be found in [23]. In this regard, we remark that the signaling burden associated with both the discovery and the steady phase in the whole network is definitely low (few Kbps). Thus, the benefits which can be obtained by implementing the Simplicity load balancing mechanism may easily justify the corresponding cost in terms of network resource consumption and complexity of implementation.

V. MODEL DEFINITION AND PERFORMANCE ANALYSIS

In this section, we define the parameters of the model and show some numerical results obtained with a simulation campaign. Our aim is to highlight the effectiveness of the proposed load balancing mechanism. We carried out many experiments by using a NS-2 [11] simulator, in a homogeneous 802.11b access network.

We compare three approaches:

- the legacy approach, where the point of access is selected by the terminal according to the signal strength (Received Signal Strength Indicator, RSSI);
- the Load Balancing Agent (LBA)-based approach described in [4], consisting of a completely distributed architecture based on agents running at each AP without any support from the terminal. In more detail, each LBA periodically broadcasts the load level of its AP to other APs. Using the reports from other LBAs, each LBA assesses whether the load is balanced among neighboring APs. If it is not balanced, it determines its AP state with respect to load sharing. Under-loaded APs are willing to accept new MNs. Overloaded APs do not accept additional MNs and force the disconnection (sending a de-association notification) of some MNs to reduce their load level;
- the Simplicity solution, according to which the point of access is selected by the network exploiting the brokerage framework distributed among the network and the terminal, as described in section IV.

These approaches do not require any modification to the IEEE 802.11 standard, but only application-level software within either APs and/or MNs. A qualitative comparison among the three approaches is reported in Table 1.

It is to be noted that the Simplicity approach requires the installation of new software in both user and network elements. One might argue that this is a heavy requirement. However, installation of this additional software in our business model for the Simplicity approach is not merely for the sake of load balancing. It is put there for a variety of reasons, primarily to personalize and customize services. Then, it may *also* be used to balance the load.

Table 1: Architectural/functional comparison among legacy, Simplicity and LBA approaches.

Feature	LBA-based approach	Simplicity approach	Legacy approach
Compatibility with IEEE 802.11 standard	YES	YES	YES
Additional software in APs	YES (LBA)	YES (bandwidth monitor)	NO
Additional software in MNs	NO	YES (Terminal Broker)	NO
Network-MNs cooperation	NO	YES	NO
Communication protocol	Needed (among LBAs)	Needed (among brokers: NB-NB and TB-NB)	Not needed
Centralized controller	NO	YES (a Network Broker for each IP subnet)	NO
Decision maker	LBA	Network Broker	Mobile Node
Decision output	De-association of MNs	Target AP	Target AP
Inputs to the selection metric	Traffic load	Traffic load, RSSI, etc	RSSI
Network service differentiation on the basis of user profile	Not possible	YES	Not possible
Autonomic mobility management	NO	YES (system driven)	NO
Operator-oriented	YES	YES	NO

A. Simulation scenario

We simulated a network scenario with 25 APs (802.11b). Each AP is controlled by an AR. This is due to an intrinsic limitation of NS-2, which allows connecting only one AP to each AR. Thus, in the following, we will refer to AR and AP indifferently, since they are co-located. Nevertheless, this limitation does not affect the generality of our results on the load balancing performance.

We distributed the ARs following a regular, hexagonal, cellular pattern over a rectangular area of 100×130 m^2, representing a portion of a campus area. This implies that the ARs located at the centre of the simulated area always have six neighboring ARs. The distance between any two neighboring ARs is set at 28.8 m. The coverage radius is equal to 22.4 m (corresponding to a coverage area equal to 1576 m^2), and the overlapping area between two adjacent APs is equal to 381 m^2. We classify the coverage area of an AP into two zones: the *optimal* zone and the *border* zone. The former is characterized by a received power level in the range [PW_{opt}, P_T], where

P_T=34 mW is the transmission power, and PW_{opt}=9.889 nW is the minimum received power for an optimal reception (corresponding to a radius equal to about 19.4 m). The border zone is characterized by a received power level in the range [PW_{min}, PW_{opt}], where PW_{min}=6.677nW is the receiver sensitivity. The overlapping area between the optimal zones of two neighboring APs is equal to 125 m^2.

In order to simulate a real setting, we adopted a frequency reuse strategy based on a triangular structure, and selected channel numbers 1, 6, and 11 of the IEEE 802.11b standard [12]. This implies that at any point in the simulation area, there is only one AP active for a given frequency channel. The Mobile IP version used to manage inter-AR mobility is MIPv6.

MIP advertisements are sent each second, whereas the L2 beacons are sent each 100 ms. Since three 802.11b channels are used, the duration of the beacon listening phase is bounded by 200ms (i.e., the time needed to scan the two channels different from the current one).

To analyze the load balancing capability of the Simplicity-enabled system, we loaded the network with constant bit rate traffic (modeling VoIP calls, [26]) with an overall bit rate equal to B=64 Kbit/s. The call duration is exponentially distributed with an average duration of 5 minutes. Call arrivals are modeled as a Poisson point process. The value of the average arrival frequency is set so as to have a traffic load within the network equal to ($0.6 \cdot N_{MN}$) Erlang, where N_{MN} is the number of MNs in the area. Such a number is variable in the different simulation settings and ranges from 50 to 200. We do not implement any call admission control scheme to limit the number of VoIP calls in the network.

As regards the mobility model, as stated in [40], "in the absence of established properties of real mobility patterns, it is not yet clear today what the requirements on a mobility model should be". However, according to the review presented in [3], we selected the Gauss-Markov mobility model, because "the Gauss-Markov Mobility Model also provides movement patterns that one might expect in the real-world". In fact, the Gauss-Markov model avoids sharp changes of direction, thus allowing previous speed and direction to influence future mobility; in addition, it forces MNs away from the edges of the simulation area, thus avoiding undesired edge effects [3].

The equations controlling the motion of a MN mobility are:

$$\begin{cases} s_n = \alpha \cdot s_{n-1} + (1-\alpha) \cdot s_{ave} + \sqrt{(1-\alpha^2)} s_{x_{n-1}} \\ d_n = \alpha \cdot d_{n-1} + (1-\alpha) \cdot d_{ave} + \sqrt{(1-\alpha^2)} d_{x_{n-1}} \end{cases}, \quad (1)$$

$$\begin{cases} x_n = x_{n-1} + s_{n-1} \cos(d_{n-1}) \\ y_n = y_{n-1} + s_{n-1} \sin(d_{n-1}) \end{cases}, \quad (2)$$

where s_n and d_n are the speed and the direction of the MN at time n, $0 \le \alpha \le 1$ is the tuning parameter used to modify randomness, s_{ave} and d_{ave} are values representing the average of s_n and d_n, $s_{x_{n-1}}$ and $d_{x_{n-1}}$ are Gaussian random variables with zero mean and variance 1 and σ, respectively. Clearly, x_n and y_n represent the surface coordinates of the MN position in the area at time n.

We have used this model with directional parameter $\alpha = 0.5$, variance of the direction $\sigma = \pi/2$, initial average direction $d_0 = 0$ and average speed $s_{ave} = 1$ m/s, with the step fixed at 1 s (i.e., the position of MNs is updated every meter, on average). When an MN is near the edge of the simulated area, its average direction is inverted.

In addition, we also considered a modified version of this mobility model. This modification aims to model situations in which a number of MNs move towards the same set of points (which we define *attractors*). In particular, if (x_a, y_a) are the coordinates of an attractor, the average direction (d_{ave}) of an attracted MN at step n is modified as follows:

$$d^{a}_{ave,n} = \tan^{-1}\left(\frac{y_a - y_{n-1}}{x_a - x_{n-1}}\right). \tag{3}$$

In our simulations, we have included five attractors, the coordinates of which are (65,85), (57.5,85), (65,77.5), (72.5,85), and (65,92.5). When the attractors are switched on, a percentage X_{ACC} of the total number of MNs moves towards them with a number of attracted MNs equal to $(X_{ACC} \cdot N_{MN})/5$ for each attractor. In addition, we also simulated a Stop&Go behavior to model the case when MNs are both in movement and motionless. The rationale of these choices may be found in the mobility features which may be typical of a campus network [41]. The movement towards attractors may be representative of a number of users (e.g., students and professors) gathering in laboratories or classrooms for lessons. On arrival, it is clearly necessary to emulate the length of stay in such a place.

B. Selection metric

As regards the process of selecting an AP, we implemented three models in the simulator: the legacy 802.11b system, the LBA-based mechanism, and the Simplicity-enabled 802.11b system.

As regards the formal definition of the load of an AP, there is no common consensus in literature (e.g., see [4][7][36][25][37][39][42]). Possible definitions are the following: (i) the number of MNs associated with the AP; (ii) the throughput of the AP; (iii) the percentage of time during which the medium is busy; (iv) the sum of the inverse of the effective bit rates of MNs associated with the AP; (v) the number of competing stations. In this work we assume that the load is the throughput of the AP, including both uplink and downlink traffic, since such a value can be easily computed by the AP without any assistance from the MNs.

B.1. The legacy approach

The legacy mechanism was designed according to the typical implementation of real 802.11b systems, in which a terminal remains attached to the current AP until the received power level goes below PW_{min}. When the terminal detects it is going to be disconnected, it performs a beacon scanning and then selects and attaches itself to the AP with the highest signal strength among the set of discovered APs.

B.2. The Simplicity approach

In the Simplicity mechanism, when the power level of the current AP goes below PW_{opt}, the TB performs a L2 beacon scanning, communicates the relevant outcome to the NB, and requests a driven handover. We also assume that, when the terminal turns on, it initially selects the AP with the highest signal strength to enable the message exchange between TB and NB. In the simulator, we implemented the distributed version of the NB. Thus, each AR acts as an NB and is in charge of controlling the MNs under its coverage. Each NB identifies the "best" AP among the set of candidates by using both the context information retrieved from the TB (available APs and relevant power level) and the measurements collected each T_{BW}=6 s at the neighboring APs (i.e., the amount of available bandwidth, which is measured on a time window equal to 6 s). An AP is considered a candidate if the power level of its L2 IDs is above PW_{min}.

As mentioned above, the selection process may also be triggered periodically in a way which is asynchronous for each MN. The period, T_{SEL}, is set to T_{SEL}=60 s. The choice of the values of T_{BW} and T_{SEL} was made according to the results of the analysis carried out in [25], where we presented results showing the ability of the system to distribute the MNs among points of access, without explicitly considering network traffic.

Finally, the NB notifies the target AP to the TB, and the TB associates with such an AP.

The Simplicity enhanced system selects the best AP by using a cost function, M_{AP}, which depends on the current load (L) of the AP and on the power level (PW) perceived by the MNs during beacon listening.

The choice of the cost function was motivated by the following considerations (see Fig. 6). Whenever possible, the MN has to be driven towards the APs with a power level of over PW_{opt}, since an MN in the border zone of an AP is in a precarious situation (even a small movement can lead to a loss of the network connection). We define this set of APs as *top level candidates*. Among them, the selection is made by taking into account the load balancing criterion. Thus, load balancing influences the AP selection if the MN is in either the optimal zone of more than one candidate APs or the border zone of all candidate APs; we define the APs whose signal is received by an MN with a power level in the range from PW_{opt} to PW_{min} as *low level candidates*.

Thus, the AP selection is a two-stage decision. In the first stage, APs are classified on a power-basis. In the second stage, the least loaded AP is chosen from among those selected in the first stage. In other words, the selected AP is the one with the lowest cost, defined as

$$M_{AP}(i) = f_1(L_i) \cdot f_2(PW_i). \tag{4}$$

The function $f_1(L_i)$ represents the cost related to the amount of used bandwidth on AP_i, and is given by

$$f_1(L_i) = \max\left(1/a, \frac{L_i + H \cdot B \cdot j}{C}\right), \tag{5}$$

where L_i is the current average load of AP_i, whereas j is a parameter set equal to 0 for the AP to which the MN is currently attached and equal to 1 for the other candidate APs.

Parameter C is a normalization value and is set equal to 11Mbit/s (i.e., the nominal bandwidth of an 802.11b AP); B=64 Kbit/s is the amount of bandwidth associated with a call. Parameter H (hysteresis) is introduced to avoid annoying ping-pong effects (i.e., continuous switches among overlapping APs). Of course, the higher the value of H, the lower the load balancing effect of the procedure, and the higher the stability of the process. Thus, the choice of the value of H has to be made considering a trade-off between performance and stability. The value a>1 is a design parameter introduced to make the cost function $f_1(L) \neq 0$.

The function $f_2(PW_i)$ is the factor related to the power level of AP_i, and it is equal to

$$f_2(PW_i) = \begin{cases} 1 & \text{if } PW_i \geq PW_{opt} \\ a & \text{if } PW_{\min} \leq PW_i \leq PW_{opt} \end{cases}. \tag{6}$$

Thus, we separate the cost associated with top level candidate APs and the cost associated with low level candidate APs, for any value of L and for each choice of a. We choose a=1000 so as to make the bandwidth cost of the extreme case (the one corresponding to no traffic load) lower than the bandwidth cost relevant to a single call (i.e., to have $f_1(0)=1/a<B/C$). In this way, $f_1(L)$ ranges from 0.001 (all the bandwidth is free) to a value lower than 1, since the maximum net bandwidth of an 802.11b AP is far lower than 11 Mbps and is around 5 Mbps. The cost function is depicted in Fig. 6. We are aware that the proposed approach is a heuristic one, using a fairly simple metric,

and we stress that the goal of this work is to show the effectiveness of the architectural solution even with a non-optimized, selection algorithm, as illustrated in the following subsection. Clearly, the Simplicity architecture can easily support the use of a more complex and performing metric in both homogeneous and heterogeneous wireless environments.

B.3. The LBA-based approach

We assume that each AP compares the value of its load (L) with the one (L_{ave}) obtained by averaging the loads of its neighboring APs. Each LBA has to be configured manually with the list of neighboring APs. Note that, since the MN cannot communicate the set of APs it listens to, L_{ave} has to be evaluated on all the APs in the list. In principle, the MN may not be under the coverage of them all.

If $L < L_{ave}+B$, then the MN is allowed to remain connected with the current AP, otherwise it is forced to de-associate. This operation is performed periodically in a way which is asynchronous for each MN; the period is set equal to 60 seconds. Neighboring APs periodically exchange information about their current load with a period equal to 6 seconds. When the MN realizes to be disconnected, it performs a beacon scanning and then selects and attaches itself randomly to one of the discovered APs.

C. Numerical Results

The total simulated time is 3000 s. It consists of a transient period (lasting for the first 1000 s, during which we do not collect statistics) and two observation phases. In phase 1, lasting from the time instant 1000 s till the time instant 2000s, MNs move according to the Gauss-Markov mobility model; subsequently they stop until the end of the simulation (phase 2, from time instant 2000 s until the time instant 3000s). This enables the load balancing mechanism to be test both when MNs move and when they are motionless.

Please note that all the curves, if not differently specified, are obtained by averaging 20 simulation runs. Confidence intervals, which are definitely low, are not shown to improve the neatness of the figures. In the case of the Simplicity approach, we ran simulations for values of the hysteresis H ranging from 0 to 4; in the figures, we only report curves relevant to the value of H=2, which yields the best performance and provides the system with sufficient stability.

Our goal is twofold. Firstly, we aim to minimize the traffic load of the most loaded AP in the network area. In other words, the load balancing procedure has to be able to minimize the maximum utilization coefficient among the APs. A preliminary observation is that, due to the mobility model adopted, MNs tend to concentrate in the centre of the simulated area. Consequently, the APs in the central region of the network are the most loaded ones. Secondly, we aim to maximize the throughput of the whole network (i.e., to minimize packet losses on the wireless section due to congestion).

We define the average throughput of the most loaded AP during a simulation phase (either 1 or 2) as

$$max_{AP-load} = \max_{i=1,..,N}\{\overline{L_i}\}, \tag{7}$$

where $\overline{L_i}$ is the average throughput of the ith AP for a given simulation run and N is the number of APs. In addition, we define the average throughput of the overall system as

$$L_{TOT} = \sum_{i=1}^{N}\overline{L_i} \tag{8}$$

In the first set of simulations, attractors remain inactive throughout the entire simulation.

Fig. 7a shows the mean value of $max_{AP-load}$ during phase 1 (i.e., when MNs move) for the legacy, the LBA and the Simplicity approach as a function of the number of MNs, N_{MN}. Fig. 7b plots the same quantities with reference to phase 2 (i.e., when MNs move) Clearly, the throughput increases with $N_{MN,}$ since the traffic load also increases with N_{MN}, as mentioned above. When the MNs move, the performance of the three mechanisms are quite close to each other, whereas the improvement of the Simplicity load balancing mechanism is evident (around 25% for N_{MN}=150 with respect to both legacy and LBA solutions) when MNs stop.

Fig. 8 shows the average throughput of all the APs in the network area during phase 2 for the three approaches, for a single run, when N_{MN}=150. Histograms show that the traffic load is definitely well distributed among the APs when Simplicity load balancing is used. The APs in the central area of the network are definitely the most loaded ones, whereas the APs located on the border of the network area deliver a lower amount of traffic and are almost useless for load balancing purposes.

Fig. 9 illustrates the average packet loss experienced in the network as a function of the number of MNs during phase 1 and phase 2. As expected, the Simplicity mechanism presents the best performance. This is especially due to the fact that the Simplicity system is able to drive the MNs connection by taking into account information coming from the terminals (i.e., the set of surrounding APs) and it is also able to drive handovers before MNs are disconnected from their previous link. This allows limiting the time during which MNs are disconnected, thus reducing packet losses. This clearly means that the Simplicity system guarantees a higher value of the overall throughput L_{TOT}.

The improvement of the Simplicity approach with respect to the legacy system is quite noticeable and is more evident when MNs stop and the Simplicity system is able to reach a steady state; when N_{MN}=150, the improvement in terms of losses is around 60% in phase 1 and 80% in phase 2.

As for the LBA approach, we can see that it performs even worse than the legacy system when MNs stop. This is due to the fact that the LBA mechanism forces the de-association of an MN without any knowledge about the coverage status surrounding the MN itself. Consequently, the number of "blind" handovers of LBA is definitely higher than that of the legacy system (in principle equal to zero), thus implying a higher disconnection time and more packet losses. This negative result would only be avoided if the coverage area of neighboring APs were completely overlapped. On the other hand, when MNs are in movement, the LBA load balancing action has a positive effect on the overall throughput and the performance is better than the legacy approach, although definitely worse than the Simplicity approach.

Beyond the advantages in terms of throughput and packet losses, we remark that the Simplicity approach improves the load balancing performance when MNs are motionless (as illustrated in the figures above). In addition, it is also worth noting that the current AR knows the surrounding wireless coverage and is able to communicate to the MN the IP address of the AR to hand over to. This enables the overall handover process to be speeded up and fast handover procedures to be executed ([13][14]).

Now, let us analyze the case in which attractors are switched on during phase 1 (i.e., from time instant 1000 s until time instant 2000 s), when MNs move according to the modified Gauss-Markov mobility model. During phase 2 (i.e., from time instant 2000 s till time instant 3000s), MNs are motionless. The situation is quite different from the one in the previous simulation setting. In fact, a number of MNs move towards a small portion of the simulation area, and thus a subset of network resources are more stressed during phase 1 and especially during phase 2.

Fig. 10 shows the average value of $max_{AP\text{-}load}$ for all solutions in both phase 1 (Fig. 10a) and phase 2 (Fig. 10b), as a function of the percentage of attracted MNs, X_{ACC}. X_{ACC} ranges from 20% to 50% and the number of MNs is set equal to N_{MN}=150. Since the concentration of MNs in a small area represents a critical condition for the APs covering such an area, in this case the advantage of the Simplicity solution with respect to both the legacy and LBA solution is always noticeable, even in phase 1, with a gain of up to 6÷7%; in phase 2 the gain reaches values of up to 30%.

Looking at Fig. 10a (relevant to phase 1, [1000 2000] s), for X_{ACC} equal to 20% and 30%, the higher the value of X_{ACC}, the higher the throughput of the most loaded AP. This is an expected result, since the higher the number of attracted MNs towards a small area, the higher the load of the most stressed AP, which is the one in the centre of that area. However, for X_{ACC} equal to 40% and 50%, this trend is inverted.

This behavior can be explained by also taking into account the results shown in Fig. 11a, which reports the average packet losses experienced in the network as a function of X_{ACC} in phase 1. As expected, packet loss rapidly increases with X_{ACC}, since, when a number of MNs are concentrated in a small area, they offer a large amount of traffic to a small number of APs. This implies that these APs become highly congested and start losing packets due to collisions on the shared wireless medium[3].

Thus, if we return to Fig. 10a, when the percentage of MNs concentrated in a small area is not high, only a few packets are lost, and the throughput of the most loaded APs (i.e., the ones in the crowded area) increases with X_{ACC}. For higher values of X_{ACC}, this phenomenon increases, and the packet loss due to collisions becomes dominant and $max_{AP\text{-}load}$ begins to decrease.

As regards phase 2 ([2000, 3000] s), if we look at Fig. 10b, the throughput of the legacy system for low values of X_{ACC} strongly increases with X_{ACC}, as in phase 1. When the number of MNs concentrated in a small area increases up to 75, due to the same phenomena mentioned above, the value of $max_{AP\text{-}load}$ for both the legacy and LBA systems rapidly decreases. The behavior of the Simplicity system when compared with that of phase 1 is slightly different. The value of $max_{AP\text{-}load}$ still increases for X_{ACC} equal to 40%, whereas its value definitely decreases less than in Fig. 10a for higher values of X_{ACC}. This is due to the load balancing function performed by the Simplicity system, which also enables approximately 6% of the overall system throughput to be saved, when compared to the legacy and LBA systems, for a value of X_{ACC} equal to 40%. Fig. 11b shows that the gain in terms of packet losses is approximately 30% for X_{ACC}=40%. When attractors are switched on and MNs are motionless, the improvement in terms of aggregated throughput is due to the fact that network resources are stressed, and a better load distribution implies a lower amount of packet losses. Another comment is that Fig. 10b shows that, as expected, when the number of attracted MNs increases, the action of the load balancing function is less efficient, even though the performance of the Simplicity approach in terms of packet losses and load distribution among APs is always better than those of the other solutions. In this regard, Fig. 12 shows results obtained with a single run, for N_{MN} equal to 150 and X_{ACC} equal to 30%. It illustrates the average load in phase 2 for the legacy, LBA and the Simplicity system. With respect to Fig. 8 (attractors inactive during phase 1), the effects of the load concentration in a very small number of APs and

[3] This also means that the overall throughput, L_{TOT}, delivered by the wireless network decreases with X_{ACC}.

of the load balancing function in the Simplicity system are definitely more evident.

As for the LBA solution, we remark that, contrary to the case with attractors switched off, in this case in phase 2 it experiences performance in terms of packet losses similar to the legacy system, whereas its performance in terms of load balancing capabilities are slightly better. This behavior is due to the fact that, under these new conditions, wireless resources are definitely more stressed and, contrary to the case when attractors are switched off, the positive effect of the load balancing is able to compensate the negative effect caused by network disconnections.

To sum up, as already mentioned for the case with attractors switched off, the Simplicity load balancing mechanism is definitely more effective when MNs do not move, and the system dynamics is due to call arrival/departure processes only.

VI. Prototype Design and Development

The prototype described in this section is part of the test-bed developed within the framework of the Simplicity project (see e.g., [28][33]), according to the architecture guidelines described in section II. The goal of the scenario we focus on in this paper (the so-called Campus Network scenario) is to show the feasibility of a mechanism which enables a network operator in the Campus to control the wireless access network, by personalizing both network and application services. For this purpose, and contrary to the simulations discussed in section V in which users are considered equal to each other, the system also takes into account the user profile information retrieved from the SD, which is a basic input of the personalization process of the service.

In principle, service personalization is a general concept, which can be extended to any access network and application services. Here, we limit the scope of the demonstration to an 802.11b access network, two classes of users (professors and guests/students) and to a web browsing service. We consider two classes of APs. APs of class A are open to everyone, irrespective of their load status and, if the load is above a given threshold, guests/students are allowed to browse in limited mode only (i.e., web pages are downloaded without images and flash videos[4]; an example is reported in Fig. 13 and Fig. 14). APs of class B are open to everyone only if their load status is below a pre-defined limit. Beyond this value, guests/students are not allowed to access all class B APs.

[4] The rationale of the choice of forbidding low priority users from downloading web images and videos under critical network conditions is twofold. The first is that this policy enables wireless bandwidth to be saved and a basic service to be provided at the same time. The second is that, even though it is clearly possible to define more effective policies to personalize application services, this policy is impressive and easily visualized.

The objective of the access network control process is twofold:

- the procedure enables the operator to perform network access differentiation and load balancing, based on the user role, which is a specific field of the user profile. High priority users (professors) perceive a better service in the case of wireless network congestion. In more detail, they have full access to all APs, whereas low priority users (guests/students) have access to a set of "restricted" APs (class B), only if the congestion level of these APs is low. In addition, high priority users can always browse the web in normal mode, whereas students can be forced to browse in the limited mode in the case of a high traffic load.
- the access network control process provides users with an automatic mechanism able to manage network connection without requiring any additional effort on the part of the user.

In the following sections, we first describe the demo architecture, giving some details concerning hardware and software choices, and then we report the results obtained from a measurement campaign performed to evaluate the overall procedure in terms of signaling burden, handover time and packet losses.

A. Test-bed description

The physical demo architecture is illustrated in Fig. 15. We assume that the users are located in the same IP network, under the coverage of two APs connected to the Internet through a gateway. The coverage areas of the two APs are partially overlapping. The NB is implemented in a Linux server located in the same IP network.

On the terminal side, the Campus Network scenario comprises a Linux-based Simplicity laptop hosting the TB. We remark that the TB is in charge of (i) driving the mobile terminal towards the most appropriate AP, and (ii) setting the web-surfing mode (limited or normal). Such network control decisions are taken by the NB on the basis of the wireless traffic load, retrieved from the APs (implemented by means of Linux-based laptops), and of the user role, retrieved from the SD through the TB.

In this scenario, the SD is implemented by means of an USB stick. The exchange of messages is illustrated in Fig. 4.

As depicted in Fig. 15, we use two laptops to generate the traffic that loads the two APs. For this purpose, we used the D-ITG software [31].

As mentioned in section IV, the specific access network control functions are executed by the CNAC subsystems. In more detail, the CNAC Subsystem in the TB communicates both network and user context to the CNAC in the NB in order to allow it to select not only the best AP among those discovered by the MN, but also the proper web surfing mode. All Simplicity subsystems were developed in Java. In addition to the Java-based broker software, we developed a number of specific software modules to implement the load balancing and the web surfing adaptation functions. These additional application-specific modules are located both in the MNs and in the APs (see Fig. 16). As regards the MN, these modules are:

- Wi-Fi Manager: this is a C module developed to interact with the 802.11b wireless card (WLAN 802.11b Digicom Palladio Wave with chipset Prism2). It is driven by the CNAC in the TB, which orders to perform frequency scanning and to attach to the target AP. Interaction with the wireless card is handled by means of the Linux Wireless Extensions, available in the kernel of the Mandrake 10.0 distribution, able to fully drive the Prism2-based card with the Host AP driver [29] in managed mode. The interface with the CNAC is implemented with a TCP local socket.
- Web Surfing Manager (in the TB): this is a modified version of Muffin [30], which is a Java-based proxy driven by the CNAC in the TB with the aim of setting the selected web surfing mode. The interface with the CNAC is implemented with a TCP local socket.
- A graphic user interface (the SPA, see Fig. 17), in the user terminal reports information relevant to the current connection, and in particular the name of the current AP, the current web surfing mode, and the current received power level.

The APs were implemented by means of the Host AP daemon [29] in master mode, using WLAN 802.11b Digicom Palladio Wave with chipset Prism2. In addition, we developed a module named Bandwidth (BW) Monitor. It is a C module which measures the used bandwidth on the wireless interface of each AP within a sliding window, and communicates this information periodically to the NB via a UDP network socket.

As regards the NB, we developed a graphic control panel enabling the network operator to set the system configuration parameters. In Fig. 18 we show a screenshot of this control panel, which includes the following fields:

- The list of managed APs: it is possible to add, remove or edit any entry in the list of APs, specifying the L2 ID, the Class (A: everyone or B: restricted), and the IP address of the AP. The amount of bandwidth currently used in each AP (in Kbit/s) is periodically updated.
- IP address and port number of the NB: this specifies the socket on which the CNAC in the NB receives the updates from each AP in the list.

- BW Measurement Window: the length of the measurement window used by the BW Monitor module located in the APs to compute the average load.
- BW Update Period: APs send updated information to the CNAC in the NB with this period. This parameter corresponds to T_{BW}, already defined in V.B.
- Selection Period: the time period used by the CNAC in the NB to check the status of the controlled terminals and, if needed, to change their settings. This parameter corresponds to T_{SEL}, already defined in V.B.
- Bandwidth Thresholds (Kbit/s): the meaning of this parameter depends on the class of the AP (class A or class B, as defined at the beginning of this section). If the traffic load of an AP of class A is higher than the Everyone threshold (L_E), guests/students connected to that AP are allowed to browse in a limited mode only. If the traffic load of an AP of class B is higher than the Restricted threshold (L_R), then guests/students are not allowed to access that AP.
- Power Thresholds (these parameters have been already defined in section V.A):
  - $PW_{\min}$ (dBm): this is the power value under which an AP is not considered a candidate to be selected as a target access to the network;
  - PW_{opt} (dBm): this is the power value beyond which an AP is considered an optimal candidate to be selected as a target access to the network.
- Hysteresis levels:
  - BW (Kbit/s): used by the CNAC in the NB in the bandwidth metric (refer to (5)) to avoid ping-pong effects. This has the same meaning of the product $H{\cdot}B$, used in section V.B;
  - PW (dBm): used by the CNAC in the NB in the power metric (see (10) below) to avoid ping-pong effects, when a terminal is near the border zone (i.e., it receives a power level around PW_{opt}).

The metric used to select the target AP in the demonstrator is more complex than that used in the simulations, since it also takes into account inputs from the user profile. It is an enhanced version of the one proposed in section V.B and is defined as follows. If the user is low priority, the cost of the *i*-th AP discovered by the MN is equal to

$$M_{AP,LP}(i) = f_{1,LP}(L_i)\cdot f_{2,new}(PW_i), \tag{9}$$

where $f_{2,new}(PW_i)$ is a slightly modified version of the $f_2(PW_i)$ presented in section V.B, whereas $f_{1,LP}(L_i)$ is a cost function specific for low priority users that takes into account the AP load L_i. In more detail, the new power-based cost function is

$$f_2(PW_i)=\begin{cases}1 & \text{if } PW_i \geq PW_{opt}-j\cdot PW\\ a & \text{if } PW_{\min} \leq PW_i \leq PW_{opt}-j\cdot PW\end{cases}, \tag{10}$$

where the parameter PW is the hysteresis value described above, a is equal to 1000 (as in the simulation trials),and j is equal to 0 for the current AP, and equal to 1 for all other candidates. This modification of the power-based cost function presented in (6) is due to the characteristics of the real wireless channel, which differs considerably from the one we used in NS-2. To be more specific, the signal strength of the current AP is monitored reliably, since disconnection from the current channel is not required to execute this measure, whereas the power level of possible candidates is gathered within a single frequency scan, and this measure could be very unreliable due to some transient, environmental factors. Thus, to limit the ping-pong effect due to unfavorable scan results, we added a hysteresis on the power level measured for candidates other than the current AP.

The new load-based cost function for low priority users is a bit more complex than that presented in (5), and is equal to

$$f_{1,LP}(L_i)=\begin{cases} if\ AP_i \in ClassA: \ \max\left(\dfrac{1}{a},\dfrac{L_i+j\cdot BW}{C}\right)\\ if\ AP_i \in ClassB: \ \begin{cases}\max\left(\dfrac{1}{a},\dfrac{L_i+j\cdot BW}{C}\right) & if\ L_i<L_R\\ \infty & otherwise\end{cases}\end{cases}, \tag{11}$$

where the parameter BW and L_R have been defined above.

If the user is a high priority one, the cost function of the i-th AP discovered by the MN is simply equal to

$$M_{AP,HP}(i)=f_1(L_i)\cdot f_{2,new}(PW_i), \tag{12}$$

where $f_1(L_i)$ is the function defined in (5).

In addition, the demonstrator implements another metric to decide whether the browsing sessions of low priority users connected to a class A AP, which is too heavily loaded, need to be set to the limited mode. To this end, we define for $AP_i \in ClassA$ the function $g(L_i)$, which establishes the web browsing mode of low priority users served by the AP according to the following law:

$$g(L_i)=\begin{cases}\text{normal} & if\ L_i<L_E\\ \text{limited} & otherwise\end{cases}, \tag{13}$$

We recall that the web browsing mode of high priority users is always set to "normal".

The functions of the demonstrator are described by means of a story-line, in Fig. 19.

B. Measurement results

We carried out a number of measurement trials to assess the performance of the Simplicity access network control procedure. In more detail, we evaluated the cost in terms of signaling rate, and the benefit in terms of reduced packet losses. In addition, we also measured the handover time.

As regards the traffic overhead, Fig. 20 reports the signaling rate per-user as a function of the selection period T_{SEL}. As expected, the signaling rate decreases with the selection period and its value is below 3 Kbps when the selection period ranges from 1 to 4 minutes.

As regards the benefit of the mechanism in terms of reduced packet loss, we considered three different kinds of users and performed a measurement trial for each of them: a non-Simplicity user, a Simplicity student, and a Simplicity professor. Each user generates a UDP traffic with a constant bit rate equal to 3 Mbps. We also generated a background traffic to load the network. The traffic pattern used to load the two APs of the test-bed during the experiment (lasting 210 seconds) is shown in Fig. 21a. Both APs are initially unloaded; at time t_1=30 seconds the user begins the UDP session; at time t_2=60 seconds the AP of class A is loaded with another UDP traffic flow with a rate equal to 4 Mbps; at time t_3=150 seconds another UDP flow charges the AP of class B with a load equal to 1.5 Mbps.

At the beginning of the experiment, all users are connected to the class A AP and the selection process occurs at times 60, 120, 180 seconds.

The Simplicity professor is always able to reach the less loaded AP, whereas the Simplicity student is forced to connect to the AP of class A from the time t_4=180 seconds (i.e., the first selection decision after time t_3) until the end of the experiment. Moreover, the non-Simplicity user is not driven by the system and can be connected to the AP of class A only.

Fig. 21b shows the percentage of packets which are lost during the experiment for the three class of users. As expected, the Simplicity professor perceives the best service, whereas the non-Simplicity user experiences the highest value of packet losses, even if he does not make any handover.

Finally, we investigated the total time needed to perform a handover triggered by the NB. The total handover time is defined as the time needed to execute the following steps (see Fig. 22):

1. the NB sends the TB the request to find the access network context;
2. the TB executes the frequency scanning, which lasts Δt_1 (see Fig. 22);
3. the TB delivers the results to the NB;
4. the NB selects the target AP;

5. the NB communicates to the TB the identity of the target AP;
6. the terminal executes the layer 2 (802.11) handover, which lasts Δt_2 (see Fig. 22).

Fig. 23 reports the total handover time as a function of the network load of the AP to which the terminal is associated, before the execution of the handover. As expected, the total handover time increases with the traffic load. The total handover time seems to be fairly high, however it is worth noting that the time during which the MN is actually disconnected (called handover interruption time) is equal "only" to $\Delta t_1+\Delta t_2$. In fact, the total handover time includes also message exchange among brokers to select the new AP. The handover interruption time, perceived by the user, is much less than the total handover time: our measurements indicate values around 130 ms and 2 ms for Δt_1 and Δt_2, respectively. These values are quite dependent on the network card and our results are in line with the analysis presented in [45].

Another important comment is that the handover interruption time with the Simplicity approach is definitely lower than the one perceived with the legacy mechanism, which can reach values of up to 2 seconds [45]. This improvement is due to the fact that our mechanism manages proactively the handover and avoids a connection interruption during detection time, which is the time taken by the network card to realize the disconnection and the need for handover [45].

As a general comment, the performance of the access network control process can be considered as satisfying.

VII. CONCLUSION

In this paper, we have presented our Simplicity vision for existing and emerging ICT services. The Simplicity project addressed a crucial issue for systems beyond 3G and proposed a solution to handle the increasing complexity of systems, services and technologies. We have proved this concept by designing our proposed architecture and implementing its main aspects, thereby showing its feasibility. The Simplicity system offers applications features of adaptation and personalization, based on an extensible, open and standardized Simplicity User Profile, conveniently expressed as an XML construct. It becomes the main source of personalization information, thus easing the spread of personalized services. In addition, the demonstrator showed that it is possible to support a wide range of terminals by developing different types of hardware devices. Last but not least, the project performed tests and measurements on the demonstrator to assess: i) the feasibility of our approach; ii) the usability, ergonomics, and human (and social) impact of the services provided by Simplicity; iii) some basic performance of the Simplicity system. Regarding applications, the Simplicity demonstrator proved that the Simplicity architecture is versatile enough to support both Simplicity-specific and external applications.

The application presented in this paper, the network selection procedure, is one of the “living proofs” of the validity of our approach. The proposed procedure exploits the enhanced capabilities of SD-enabled mobile terminals, which allows them to co-operate with a broker entity in the network. This co-operation in turn enables the knowledge of network context and of user profile/preferences to be exploited in order to: i) simplify the fruition of services; ii) define new, automatic, management tools for network control and self-configuration in a heterogeneous framework. Thanks to the distributed brokering system, all management operations are transparent from the user point of view.

As regards network management, our case study in this paper focused on load balancing. The simulation analysis showed that our proposed architecture improves performance in terms of both user-perceived levels of service (packet losses) and load balancing. In addition, our proposal should be judged not only in terms of the particular load balancing application described extensively in this paper, but also as a powerful tool to be used for a broader set of management operations and service customization/personalization.

Finally, we presented a prototype of our procedure, developed to show its feasibility and to gain more insight into its working operation. The demonstrator can differentiate network services according to user role, in addition to executing load balancing functions.

ACKNOWLEDGMENT

The authors sincerely thank the anonymous referees for their constructive suggestions and comments that have largely contributed to the improvement of the quality of this paper.

This work was co-funded by the European Union in the framework of the IST project SIMPLICITY and by the Italian Ministry of Education, University, and Research (MIUR) within the PRIN project TWELVE. In more detail, the definition of the architecture and the demo implementation were developed within the project SIMPLICITY, whereas the simulation study was carried out within the project TWELVE. The authors would like to acknowledge the contributions of their colleagues from the Simplicity and TWELVE projects.

REFERENCES

[1] Project IST SIMPLICITY, <http://www.ist-simplicity.org>.

[2] N. Blefari-Melazzi et al., "The Simplicity Project: Managing Complexity in a Diverse ICT World", in Ambient Intelligence, IOS Press, 2004.

[3] T. Camp, J. Boleng, V. Davies, “A survey of mobility models for ad hoc network research”, Wireless Communication & Mobile Computing, 2(5), 2002.

[4] H. Velayos V. Aleo, G. Karlsson, "Load Balancing in Overlapping Wireless LAN Cells," IEEE ICC 2004, Paris, France, June 2004.
[5] M. Liebsch et alii, “Candidate access router discovery”, IETF RFC 4066, July 2005.
[6] A. Balachandran, P. Bahl, G. M. Voelker, “Hot-Spot Congestion Relief in Public-area Wireless Networks,” IEEE WMCSA 2002, Callicoon, USA, June 2002.
[7] G. Bianchi, I. Tinnirello, “Improving Load Balancing mechanisms in Wireless Packet Networks,” IEEE ICC 2002, New York, USA, 2002.
[8] W. Zhang, J. Jähnert, K. Dolzer, "Design and evaluation of a hand-over decision strategy for 4th generation mobile networks," IEEE VTC 2003 Spring, Jeju, Korea, 2003.
[9] Yui-Wah Lee and Scott C: Miller, "Network selection and discovery of service information in public WLAN hotspots," ACM WMASH, Philadelphia, USA, 2004.
[10] Proxim Wireless Networks, Orinoco Access Point Family, <http://www.proxim.com/learn/library/datasheets/AP-2000_A4.pdf>.
[11] The Network Simulator Web Page, <http://www.isi.edu/nsnam/ns/>.
[12] IEEE 802.11b Standard, "IEEE 802.11, Part 11: Wireless LAN Medium Access Control (MAC) And Physical Layer (PHY) Specifications: Higher-speed Physical Layer Extension In The 2.4 GHz Band", 2000.
[13] R. Koodli, “Fast handovers for mobile IPv6”, IETF RFC 4068, July 2005.
[14] K. El Malki et alii, “Low latency handoffs in mobile IPv4”, Internet draft, June 2004.
[15] T. Taekyoung Kwon, M. Gerla, Sa. Das, Su. Das, "Mobility Management for VoIP Service: Mobile IP vs. SIP", IEEE Wireless Communications, October 2002.
[16] Draft IEEE Standard for Local and Metropolitan Area Networks: Media Independent Handover Services, July 2005.
[17] B. O'Hara, P. Calhoun, J. Kempf, “Configuration and Provisioning for Wireless Access Points (CAPWAP) Problem Statement”, IETF RFC 3990, February 2005.
[18] L. Yang, P. Zerfos, E. Sadot, “Architecture Taxonomy for Control and Provisioning of Wireless Access Points (CAPWAP)”, IETF RFC 4118, June 2005.
[19] M. Moreton, “Suggested TGu Functional Requirements”, IEEE Contribution, format , June 2005.
[20] M. Moreton, “Requirement Motions, IEEE Contribution”, doc.: IEEE 802.11-05/0643r2, July 2005.
[21] R. Koodli, “Fast handovers for mobile IPv6”, IETF RFC 4068, July 2005.
[22] K. El Malki, “Low latency handoffs in mobile IPv4”, Internet draft, October 2005.
[23] D. Di Sorte, M. Femminella, L. Piacentini, G. Reali, "Performance Evaluation of the Push-Mode-Multicast based Candidate Access Router Discovery (PMM CARD)," Computer Networks, Elsevier, 50(3), February 2006. A shortened version of this paper has appeared in Proceedings of IEEE ICC2005, Seoul, Korea, 2005.
[24] D. Di Sorte, M. Femminella, L. Piacentini, G. Reali, “Target access router selection in advanced mobility scenarios,” Computer Communications, Elsevier, 29(3), pp. 337-357, February 2006. A shortened version of this paper has appeared in Proceedings of IEEE ICC2004, Paris, France, 2004.
[25] N. Blefari-Melazzi, D. Di Sorte, M. Femminella, G. Reali, “Toward an autonomic control of wireless access networks,” IEEE GLOBECOM 2005, St. Louis, USA, November 2005.
[26] B. Goode, “Voice Over Internet Protocol (VoIP)”, Proceedings of the IEEE, 90(9), September 2002.
[27] Simple Object Access Protocol (SOAP), <http://www.w3.org/TR/soap/>.
[28] N. Blefari-Melazzi et alii, “The Simplicity System and its Demonstrator”, Wireless World Research Forum Meeting 15, 8-9 December, 2005, Paris, France, http://www.ist-simplicity.org/publicat.htm.
[29] Host AP Web Page, <http://hostap.epitest.fi/>.
[30] Muffin Web Page, <http://muffin.doit.org>.
[31] Distributed Internet Traffic Generator (D-ITG) Web Page, <http://www.grid.unina.it/software/ITG/index.php>.
[32] Project Simplicity, Deliverable D2202, “Final system architecture specification”, <http://server.ist-simplicity.org/deliverables.php>.
[33] N. Blefari-Melazzi et alii, “The Simplicity project and its demonstrator: improving ease of use and personalization of ICT services,” Proceedings of IEEE GLOBECOM 2006, San Francisco, USA, November 2006.

[34] 3rd Generation Partnership Project. Data Description Method (DDM) - 3GPP Generic User Profile (GUP). Technical specification of Technical Specification Group Terminals, Version 6.1.0. 2004.Reference example.
[35] H.J. Wang, R.H. Katz, J. Giese, "Policy-Enabled Handoffs Across Heterogeneous Wireless Networks," Second IEEE Workshop on Mobile Computer Systems and Applications, New Orleans, USA, February 1999.
[36] C. Guo, Z. Guo, Q. Zhang, W. Zhu, "A Seamless and Proactive End-to-End Mobility Solution for Roaming Across Heterogeneous Wireless Networks," IEEE Journal on Selected Areas in Communications, 22(5), June 2004.
[37] C. W. Lee, L. M. Chen, M. C. Chen, and Y. S. Sun, "A Framework of Handoffs in Wireless Overlay Networks Based on Mobile IPv6," IEEE Journal on Selected Areas in Communications, 23(11), November 2005.
[38] P. Vidales, J. Baliosian, J. Serrat, G. Mapp, F. Stajano, A. Hopper, "Autonomic Systems for Mobility Support in 4G Networks," IEEE Journal on Selected Areas in Communications, 23(12), December 2005.
[39] Y. Bejerano, S.-J. Han, L.E. Li, "Fairness and Load Balancing in Wireless LANs Using Association Control," MobiCom 2004, Philadelphia, USA, September-October 2004.
[40] J. Le Boudec and M. Vojnovic, "Perfect simulation and stationarity of a class of mobility models," IEEE INFOCOM 2005, Miami, USA, March 2005.
[41] D. Tang, M. Baker, "Analysis of a local-area wireless network," MOBICOM 2000, Boston, USA, August 2000.
[42] G. Bianchi, I. Tinnirello, "Kalman filter estimation of the number of competing terminals in an IEEE 802.11 network," IEEE INFOCOM 2003, San Francisco, USA, April 2003.
[43] Simplicity Deliverable D2103 "Final Use Cases and Business Models", <http://server.ist-simplicity.org/deliverables.php>.
[44] T. Melia, J. Korhonen, R. Aguiar, S. Sreemanthula, V. Gupta, "Network initiated handovers problem statement," Internet Draft, work in progress, June 2006.
[45] H. Velayos, G. Karlsson, "Techniques to reduce the IEEE 802.11b handoff time," IEEE ICC 2004, Paris, France, June 2004.
[46] The Wireless World Research Forum Web Page, <http://www.wireless-world-research.org/>.
[47] OASIS Asynchronous Service Access Protocol (ASAP) TC, <http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=asap>.
[48] B. Johanson, A. Fox and T. Winograd, "The Interactive Workspaces Project: Experiences with Ubiquitous Computing Rooms", IEEE Pervasive Computing, 1(2), 2002.
[49] A. Cuevas, J.I. Moreno, P. Vidales, H. Einsiedler, "The IMS service platform: a solution for next-generation network operators to be more than bit pipes," IEEE Communications Magazine, 44(8), August 2006.

List of Figure Captions

Fig. 1 - Functional Entities of High-level architecture.

Fig. 2 – The Simplicity system.

Fig. 3 – Protocol stack in the Simplicity framework.

Fig. 4 – Network control process exploiting a distributed brokering framework: an overall picture.

Fig. 5 – The CNAC subsystem within the Simplicity architecture.

Fig. 6 – Graphical representation of the cost function.

Fig. 7 – Average throughput of the most loaded AP versus N_{MN} for (a) moving and (b) motionless MNs.

Fig. 8 – Average throughput of APs when MNs are motionless (N_{MN}=150) in a single run.

Fig. 9 – Average packet loss versus N_{MN} for (a) moving and (b) motionless MNs.

Fig. 10 – Average throughput of the most loaded AP versus X_{ACC} for (a) moving and (b) motionless MNs (N_{MN}=150).

Fig. 11 – Average packet loss versus X_{ACC} for (a) moving and (b) motionless MNs (N_{MN}=150).

Fig. 12 – Average throughput of the APs for motionless MNs (X_{ACC}=0.3, N_{MN}=150) in a single run.

Fig. 13 – Aspect of a web page when the web surfing mode is set to normal.

Fig. 14 – Aspect of the same web page when the web surfing mode is set to limited.

Fig. 15 – Physical demo architecture.

Fig. 16 – Software demo architecture.

Fig. 17 – Screenshot of the GUI in the user terminal.

Fig. 18 – Screenshot of the control panel in the NB.

Fig. 19 – Demo story-line.

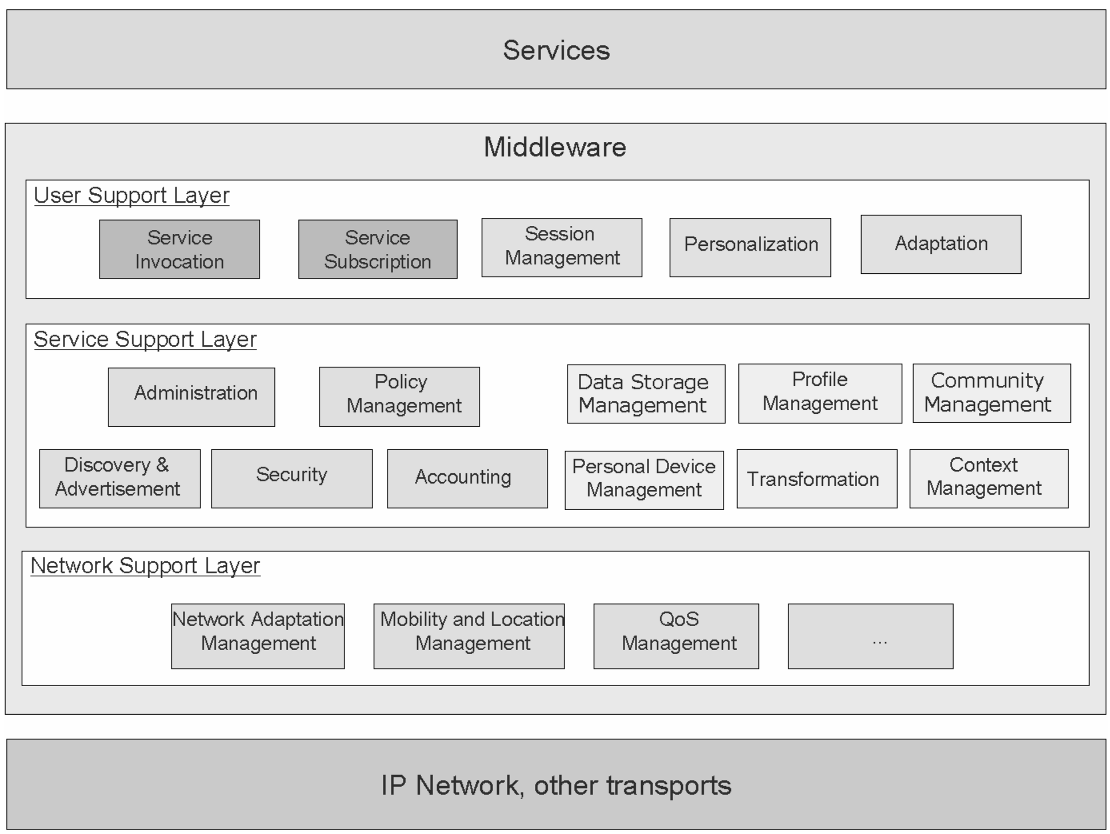
Fig. 20 – Signaling burden versus T_{SEL} in the Simplicity test-bed.

Fig. 21 – Packet loss as a function of the user profile in the Simplicity test-bed.

Fig. 22 – Handover steps.

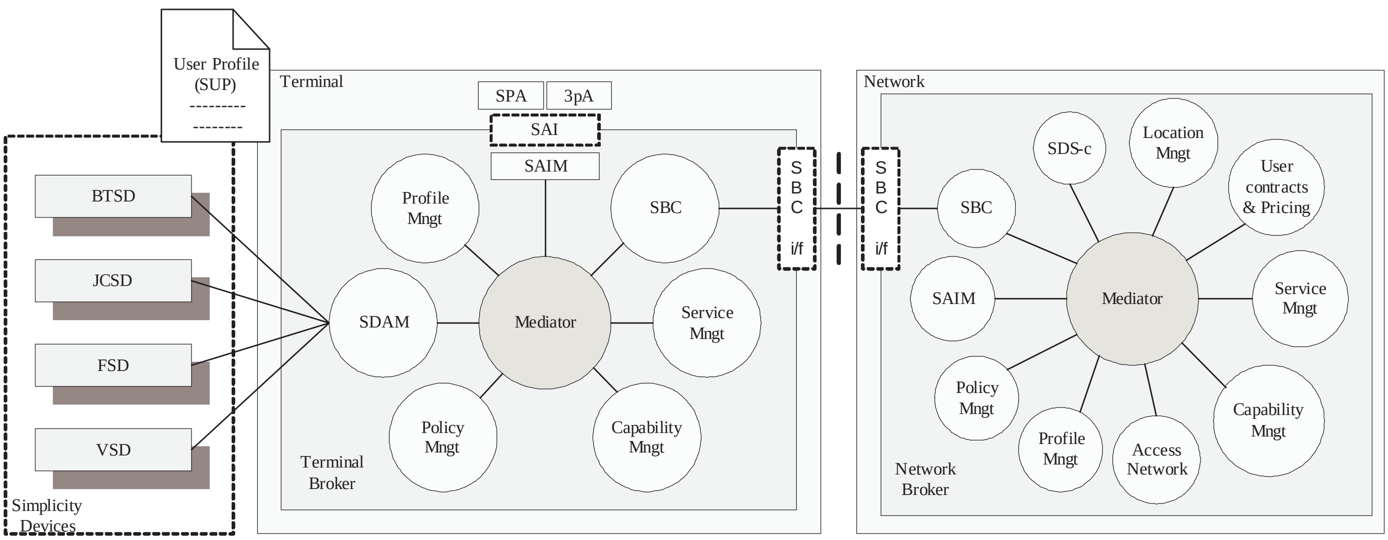
Fig. 23 – Handover time versus traffic load in the Simplicity test-bed.

Fig. 1



Services
Middleware
User Support Layer
Service Invocation
Service Subscription
Session Management
Personalization
Adaptation
Service Support Layer
Administration
Policy Management
Data Storage Management
Profile Management
Community Management
Discovery & Advertisement
Security
Accounting
Personal Device Management
Transformation
Context Management
Network Support Layer
Network Adaptation Management
Mobility and Location Management
QoS Management
...
IP Network, other transports

Fig. 2



User Profile
(SUP)
Terminal
SPA
3pA
SAI
SAIM
Profile Mngt
SBC
BTSD
JCSD
FSD
VSD
SDAM
Mediator
Service Mngt
Policy Mngt
Capability Mngt
Terminal Broker
Simplicity Devices
S B C i/f
S B C i/f
Network
SDS-c
Location Mngt
User contracts & Pricing
SBC
SAIM
Mediator
Service Mngt
Policy Mngt
Profile Mngt
Access Network
Capability Mngt
Network Broker

Fig. 3

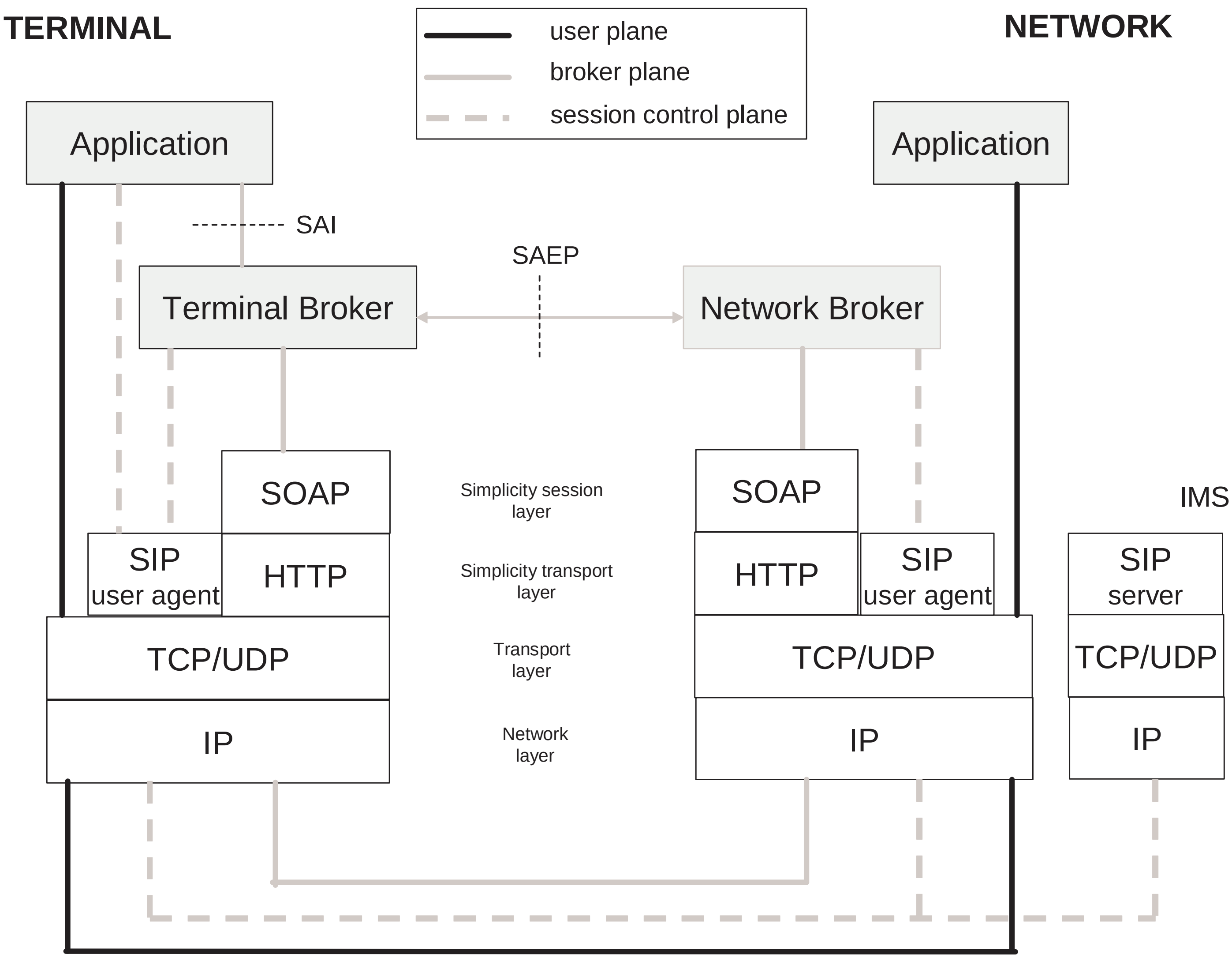
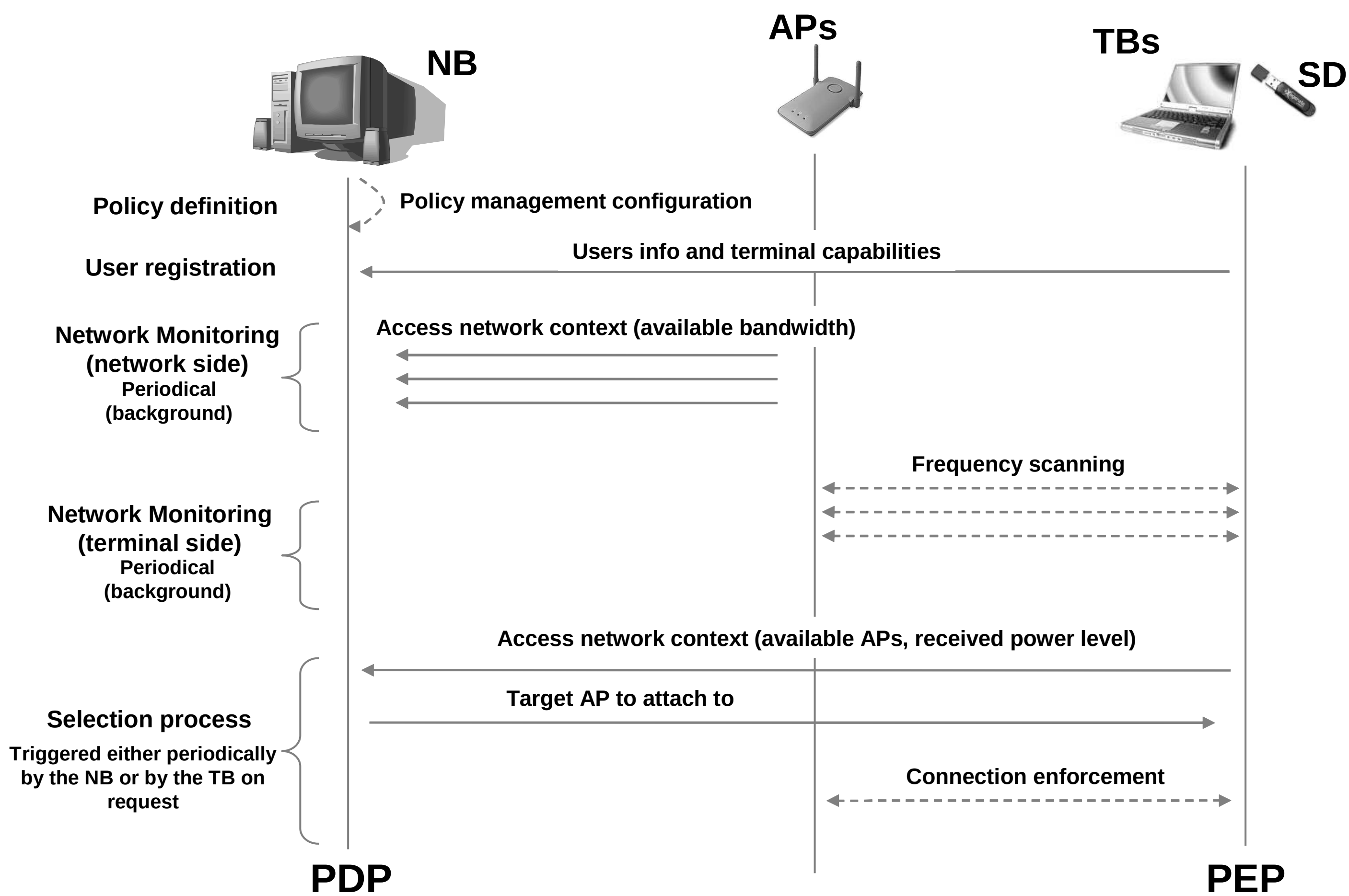

TERMINAL
user plane
broker plane
session control plane
NETWORK
Application
SAI
SAEP
Terminal Broker
Network Broker
Application
SOAP
Simplicity session layer
SOAP
IMS
SIP user agent
HTTP
Simplicity transport layer
HTTP
SIP user agent
SIP server
TCP/UDP
Transport layer
TCP/UDP
TCP/UDP
IP
Network layer
IP
IP

Fig. 4



NB
APs
TBs
SD
Policy definition
Policy management configuration
User registration
Users info and terminal capabilities
Network Monitoring
(network side)
Periodical
(background)
Access network context (available bandwidth)
Frequency scanning
Network Monitoring
(terminal side)
Periodical
(background)
Access network context (available APs, received power level)
Selection process
Triggered either periodically
by the NB or by the TB on
request
Target AP to attach to
Connection enforcement
PDP
PEP

Fig. 5

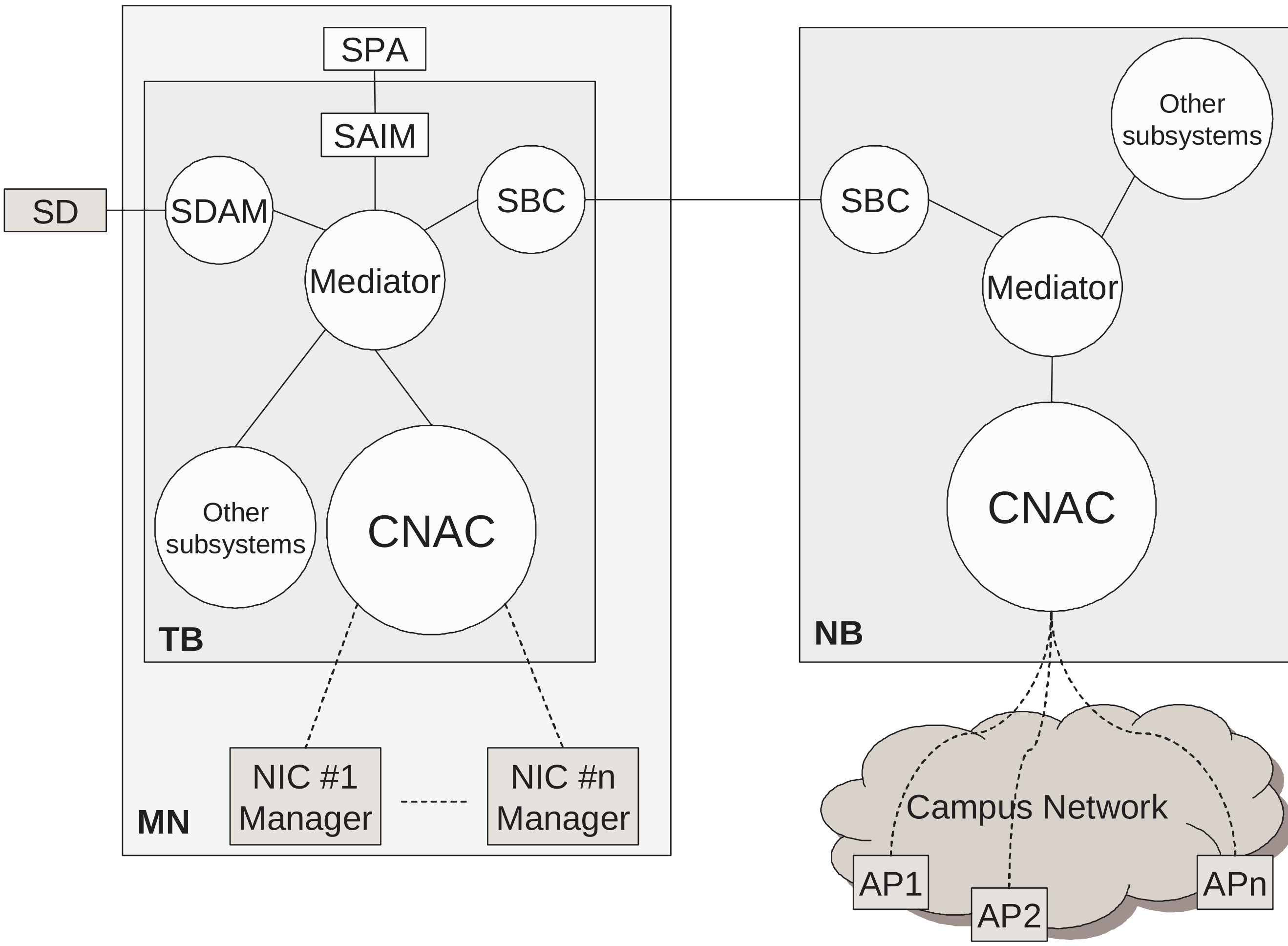

SPA
SAIM
SD
SDAM
SBC
Mediator
Other subsystems
CNAC
TB
NIC #1 Manager
NIC #n Manager
MN
SBC
Other subsystems
Mediator
CNAC
NB
Campus Network
AP1
AP2
APn

Fig. 6

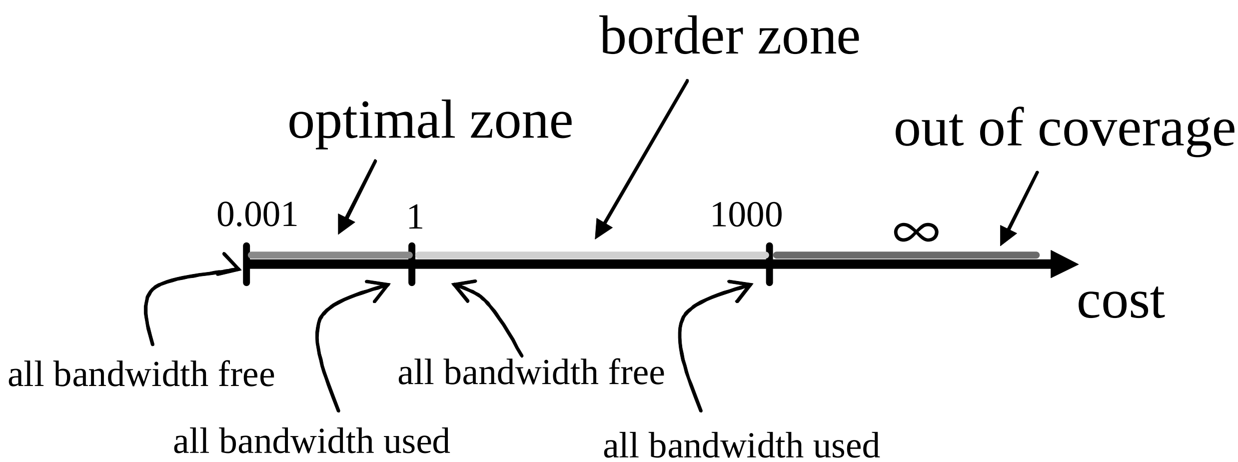

border zone
optimal zone
out of coverage
0.001
1
1000
∞
cost
all bandwidth free
all bandwidth used
all bandwidth free
all bandwidth used

Fig. 7

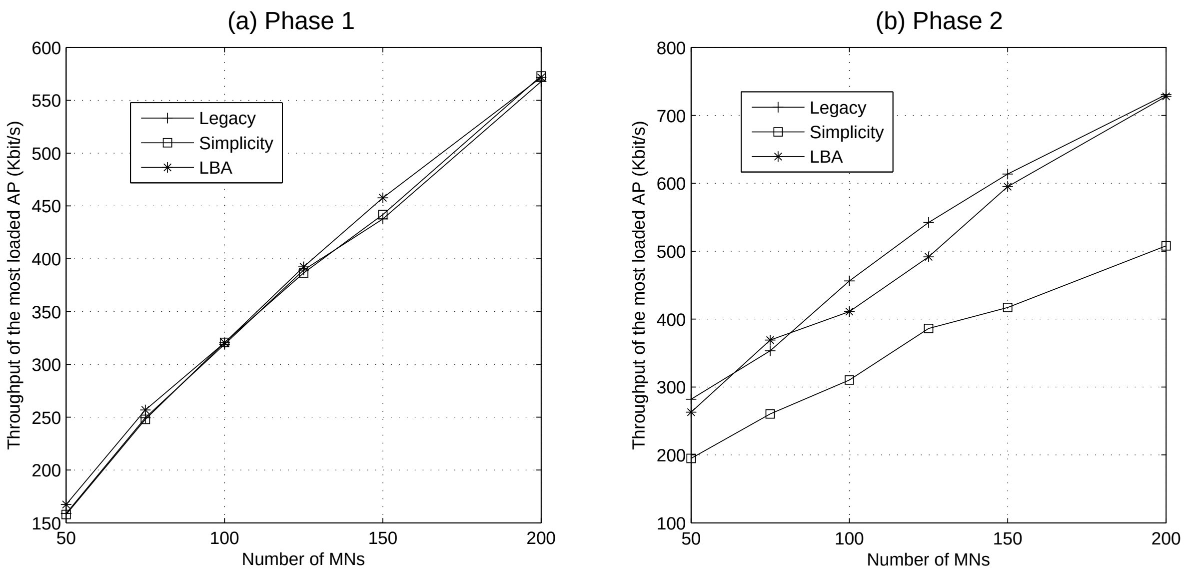
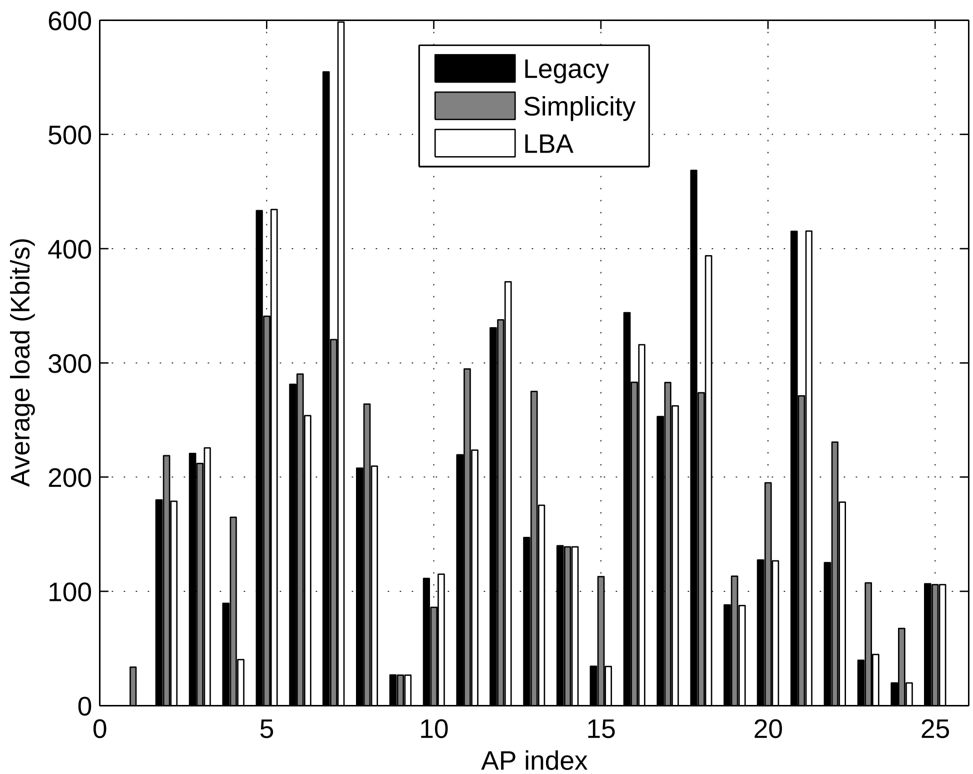
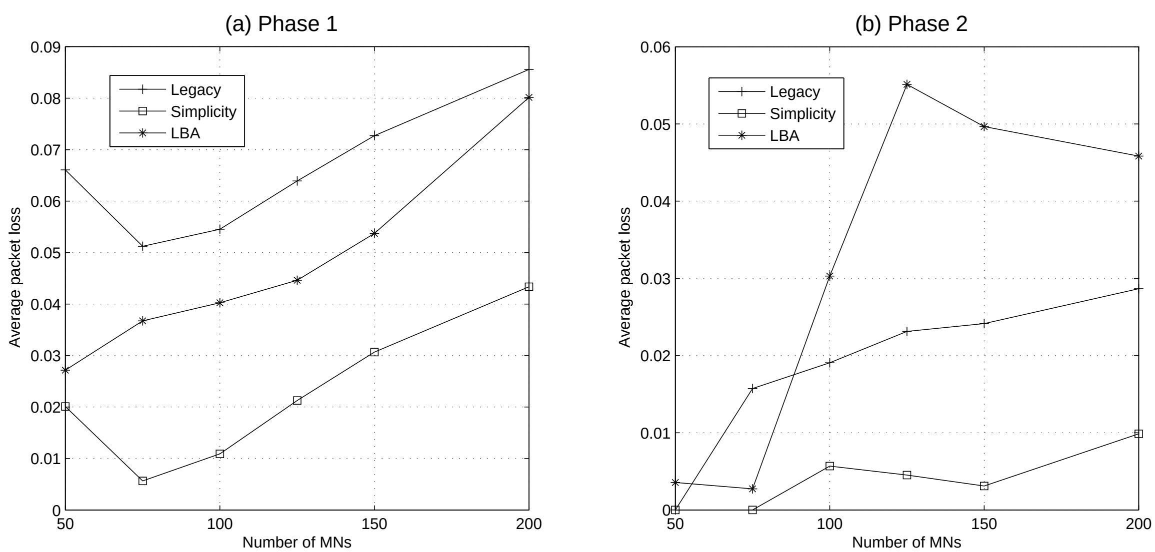

(a) Phase 1
Throughput of the most loaded AP (Kbit/s)
600
550
500
450
400
350
300
250
200
150
50
100
150
200
Number of MNs
Legacy
Simplicity
LBA
(b) Phase 2
Throughput of the most loaded AP (Kbit/s)
800
700
600
500
400
300
200
100
50
100
150
200
Number of MNs
Legacy
Simplicity
LBA

Fig. 8



Legacy
Simplicity
LBA
Average load (Kbit/s)
AP index
0
100
200
300
400
500
600
0
5
10
15
20
25

Fig. 9



(a) Phase 1
Average packet loss
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
50
100
150
200
Number of MNs
Legacy
Simplicity
LBA
(b) Phase 2
Average packet loss
0.06
0.05
0.04
0.03
0.02
0.01
0
50
100
150
200
Number of MNs
Legacy
Simplicity
LBA

Fig. 10

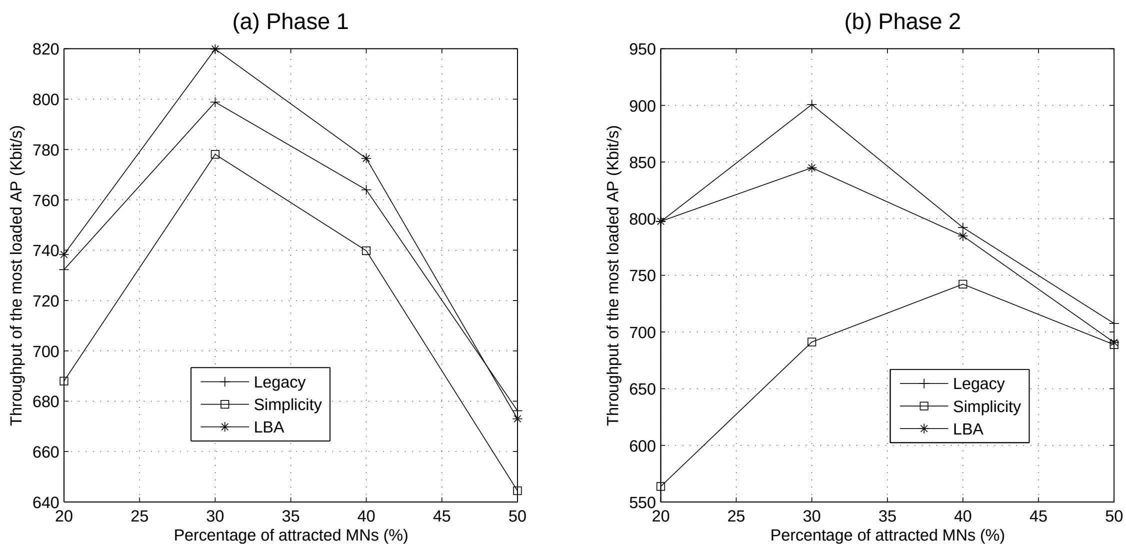

(a) Phase 1
Throughput of the most loaded AP (Kbit/s)
820
800
780
760
740
720
700
680
660
640
20
25
30
35
40
45
50
Percentage of attracted MNs (%)
Legacy
Simplicity
LBA
(b) Phase 2
Throughput of the most loaded AP (Kbit/s)
950
900
850
800
750
700
650
600
550
20
25
30
35
40
45
50
Percentage of attracted MNs (%)
Legacy
Simplicity
LBA

Fig. 11

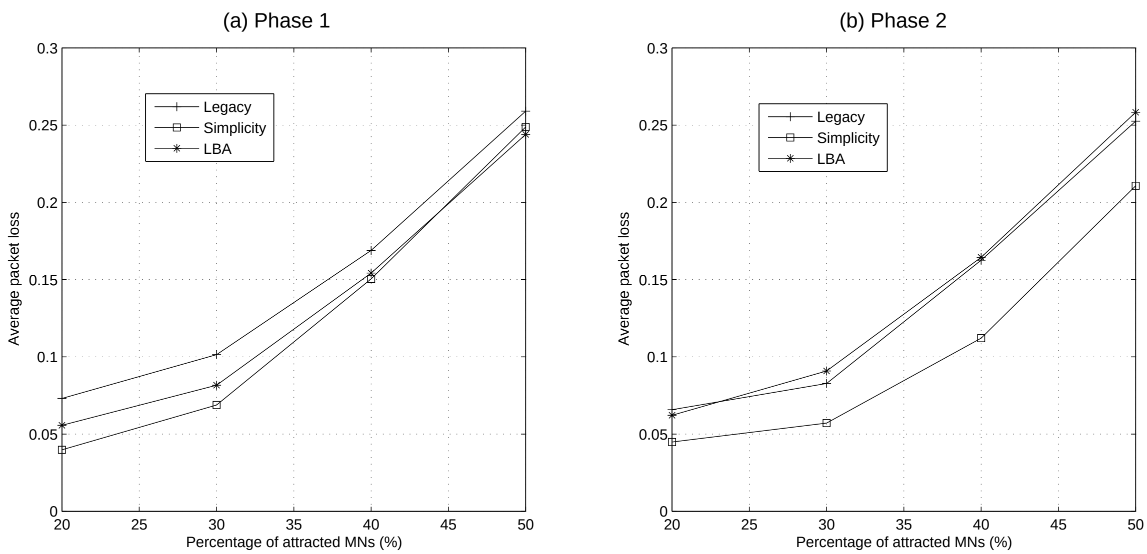
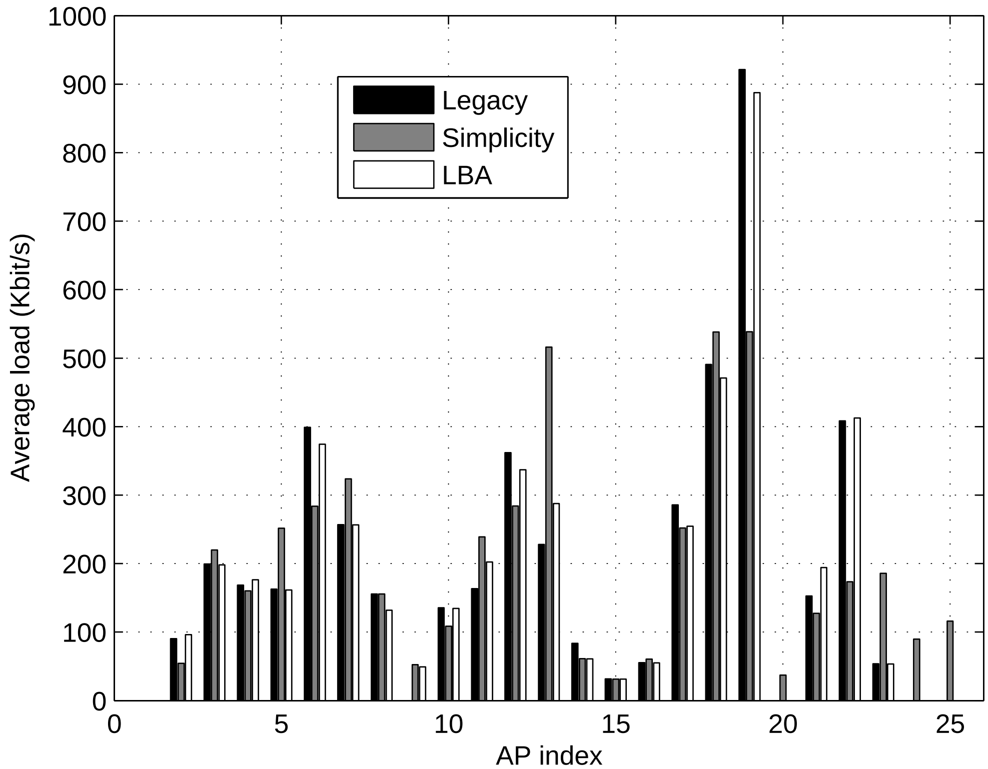

(a) Phase 1
(b) Phase 2
Legacy
Simplicity
LBA
Average packet loss
Percentage of attracted MNs (%)
0
0.05
0.1
0.15
0.2
0.25
0.3
20
25
30
35
40
45
50

Fig. 12

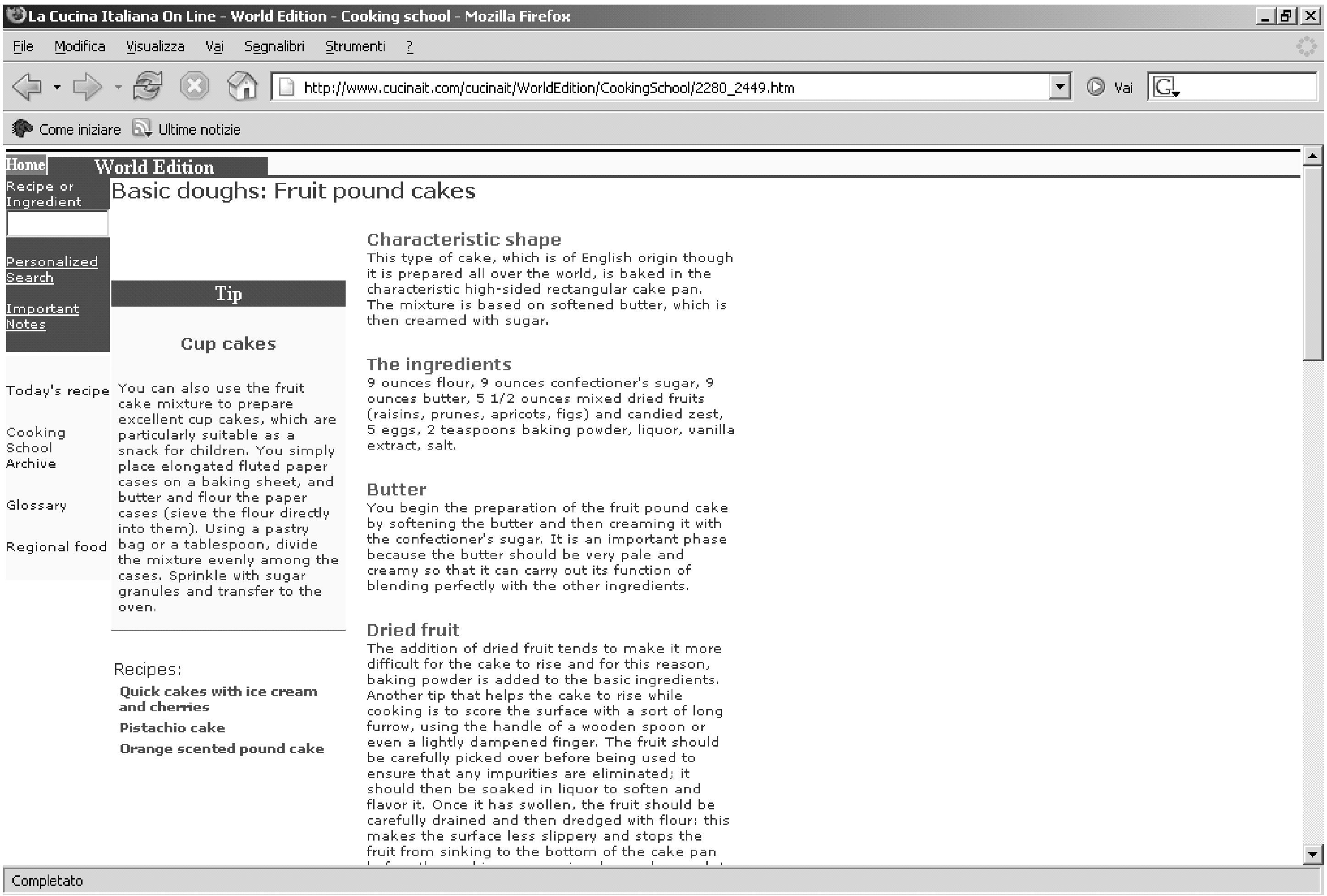


Legacy
Simplicity
LBA
Average load (Kbit/s)
AP index
0
100
200
300
400
500
600
700
800
900
1000
0
5
10
15
20
25

Fig. 13

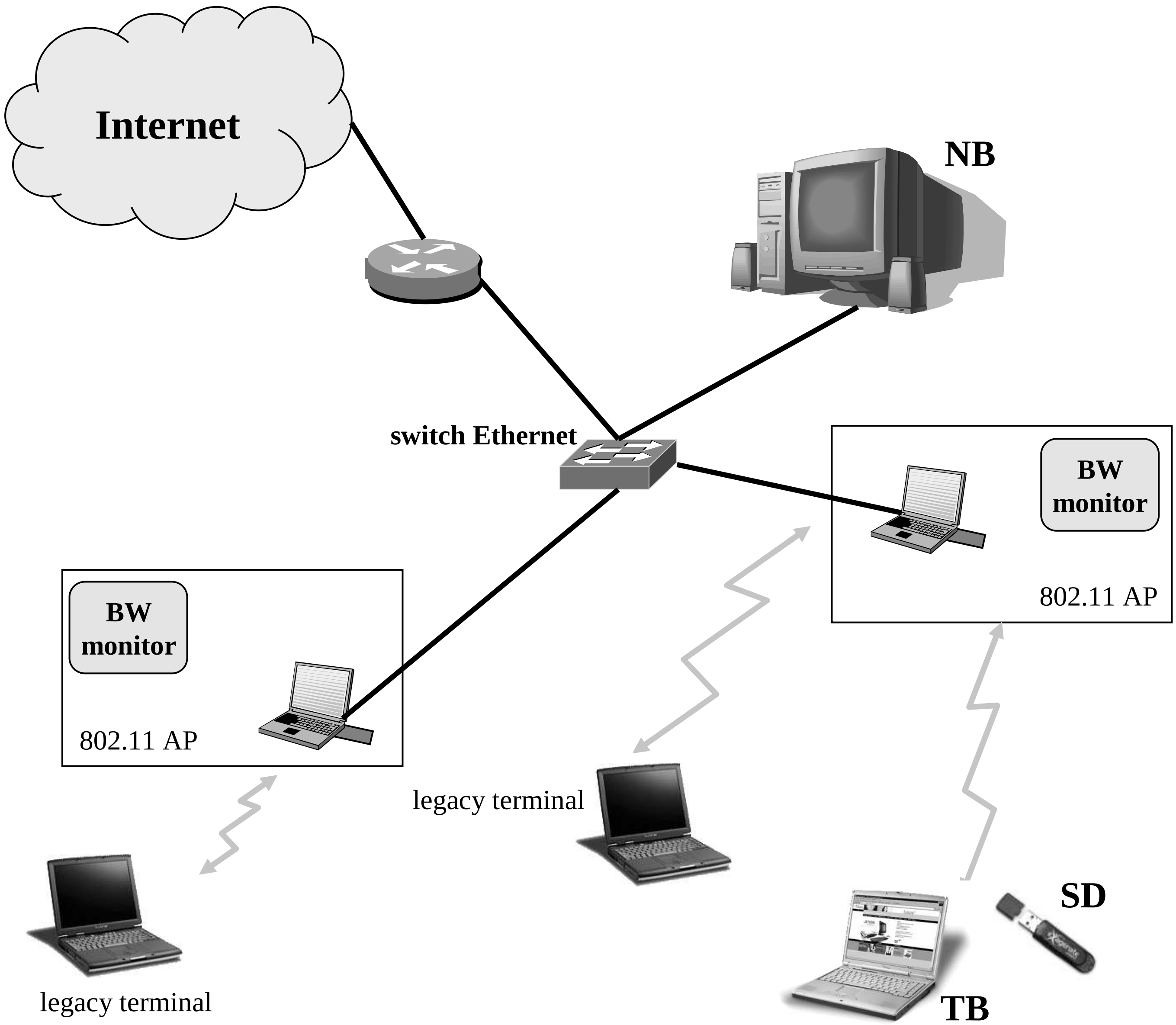

La Cucina Italiana On Line - World Edition - Cooking school - Mozilla Firefox
File Modifica Visualizza Vai Segnalibri Strumenti ?
http://www.cucinait.com/cucinait/WorldEdition/CookingSchool/2280_2449.htm
Vai
Come iniziare
Ultime notizie
con gusto.
In edicola con ...
con gusto.
LA CUCINA ITALIANA On Line
Home
World Edition
Recipe or Ingredient
search
Personalized Search
Important Notes
Today's recipe
Cooking School
Archive
Glossary
Regional food
Basic doughs: Fruit pound cakes
Characteristic shape
This type of cake, which is of English origin though it is prepared all over the world, is baked in the characteristic high-sided rectangular cake pan. The mixture is based on softened butter, which is then creamed with sugar.
The ingredients
9 ounces flour, 9 ounces confectioner's sugar, 9 ounces butter, 5 1/2 ounces mixed dried fruits (raisins, prunes, apricots, figs) and candied zest, 5 eggs, 2 teaspoons baking powder, liquor, vanilla extract, salt.
Butter
You begin the preparation of the fruit pound cake by softening the butter and then creaming it with the confectioner's sugar. It is an important phase because the butter should be very pale and creamy so that it can carry out its function of blending perfectly with the other ingredients.
Dried fruit
The addition of dried fruit tends to make it more difficult for the cake to rise and for this reason,
Tip
Cup cakes
You can also use the fruit
Completato

Fig. 14



La Cucina Italiana On Line - World Edition - Cooking school - Mozilla Firefox
File Modifica Visualizza Vai Segnalibri Strumenti ?
http://www.cucinait.com/cucinait/WorldEdition/CookingSchool/2280_2449.htm
Vai
Come iniziare
Ultime notizie
Home
World Edition
Recipe or Ingredient
Personalized Search
Important Notes
Today's recipe
Cooking School Archive
Glossary
Regional food
Basic doughs: Fruit pound cakes
Tip
Cup cakes
You can also use the fruit cake mixture to prepare excellent cup cakes, which are particularly suitable as a snack for children. You simply place elongated fluted paper cases on a baking sheet, and butter and flour the paper cases (sieve the flour directly into them). Using a pastry bag or a tablespoon, divide the mixture evenly among the cases. Sprinkle with sugar granules and transfer to the oven.
Recipes:
Quick cakes with ice cream and cherries
Pistachio cake
Orange scented pound cake
Characteristic shape
This type of cake, which is of English origin though it is prepared all over the world, is baked in the characteristic high-sided rectangular cake pan. The mixture is based on softened butter, which is then creamed with sugar.
The ingredients
9 ounces flour, 9 ounces confectioner's sugar, 9 ounces butter, 5 1/2 ounces mixed dried fruits (raisins, prunes, apricots, figs) and candied zest, 5 eggs, 2 teaspoons baking powder, liquor, vanilla extract, salt.
Butter
You begin the preparation of the fruit pound cake by softening the butter and then creaming it with the confectioner's sugar. It is an important phase because the butter should be very pale and creamy so that it can carry out its function of blending perfectly with the other ingredients.
Dried fruit
The addition of dried fruit tends to make it more difficult for the cake to rise and for this reason, baking powder is added to the basic ingredients. Another tip that helps the cake to rise while cooking is to score the surface with a sort of long furrow, using the handle of a wooden spoon or even a lightly dampened finger. The fruit should be carefully picked over before being used to ensure that any impurities are eliminated; it should then be soaked in liquor to soften and flavor it. Once it has swollen, the fruit should be carefully drained and then dredged with flour: this makes the surface less slippery and stops the fruit from sinking to the bottom of the cake pan
Completato

Fig. 15



Internet
NB
switch Ethernet
BW
monitor
802.11 AP
BW
monitor
802.11 AP
legacy terminal
legacy terminal
SD
TB

Fig. 16

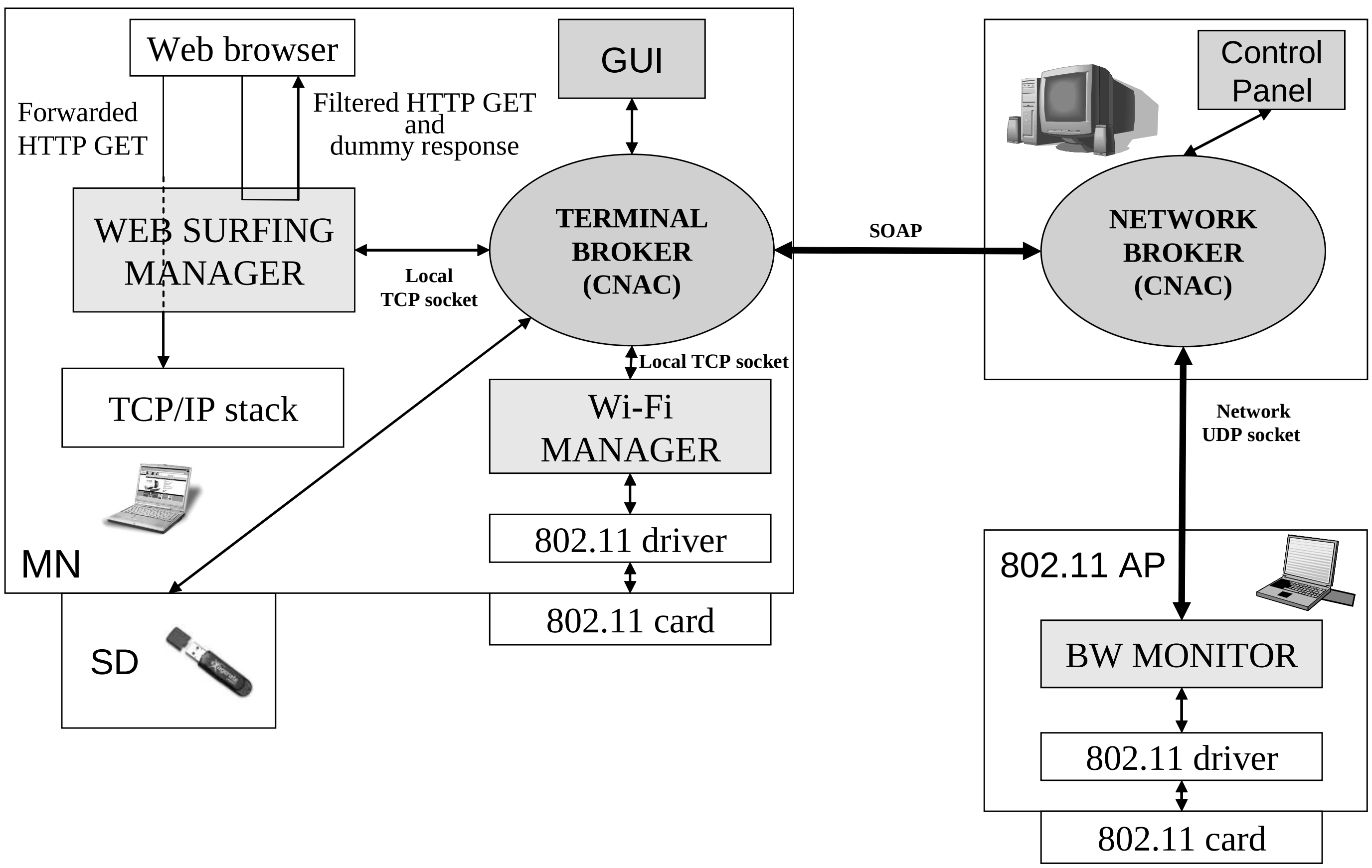

Web browser
GUI
Forwarded HTTP GET
Filtered HTTP GET and dummy response
WEB SURFING MANAGER
Local TCP socket
TERMINAL BROKER (CNAC)
Local TCP socket
TCP/IP stack
Wi-Fi MANAGER
802.11 driver
MN
802.11 card
SD
SOAP
Control Panel
NETWORK BROKER (CNAC)
Network UDP socket
802.11 AP
BW MONITOR
802.11 driver
802.11 card

Fig. 17

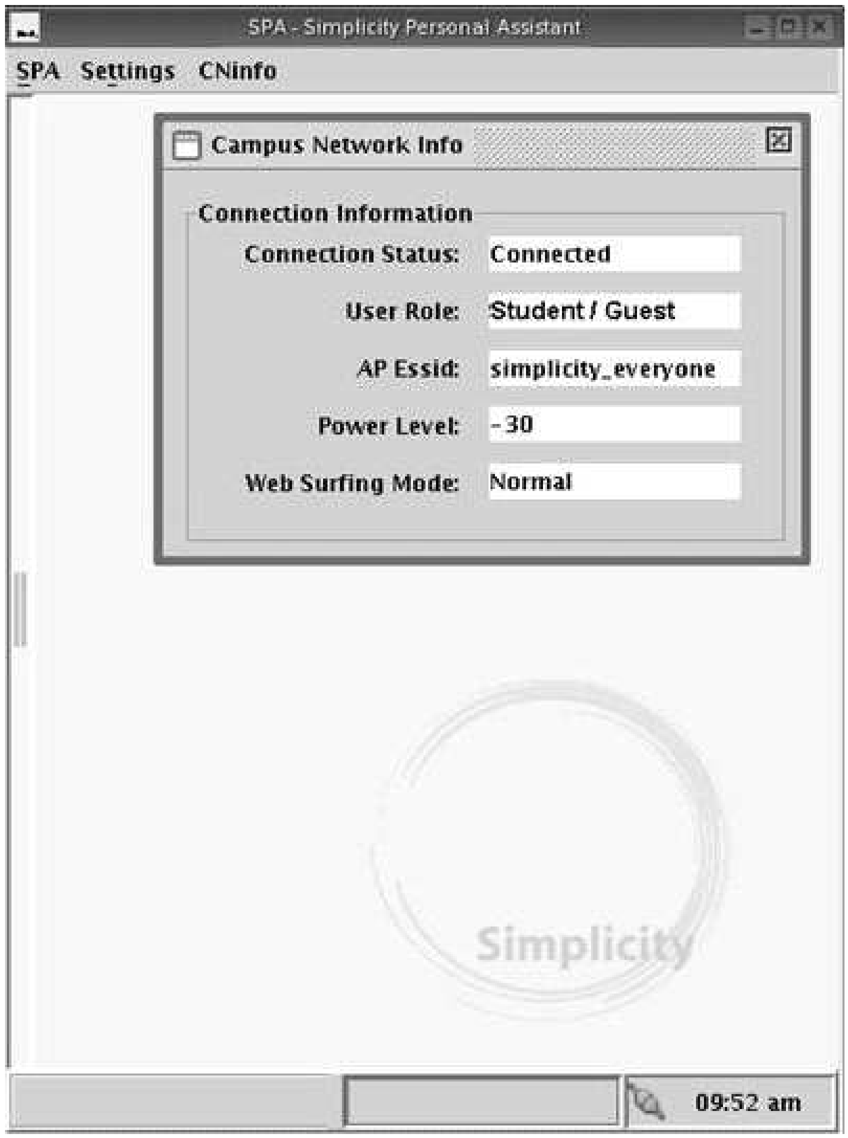
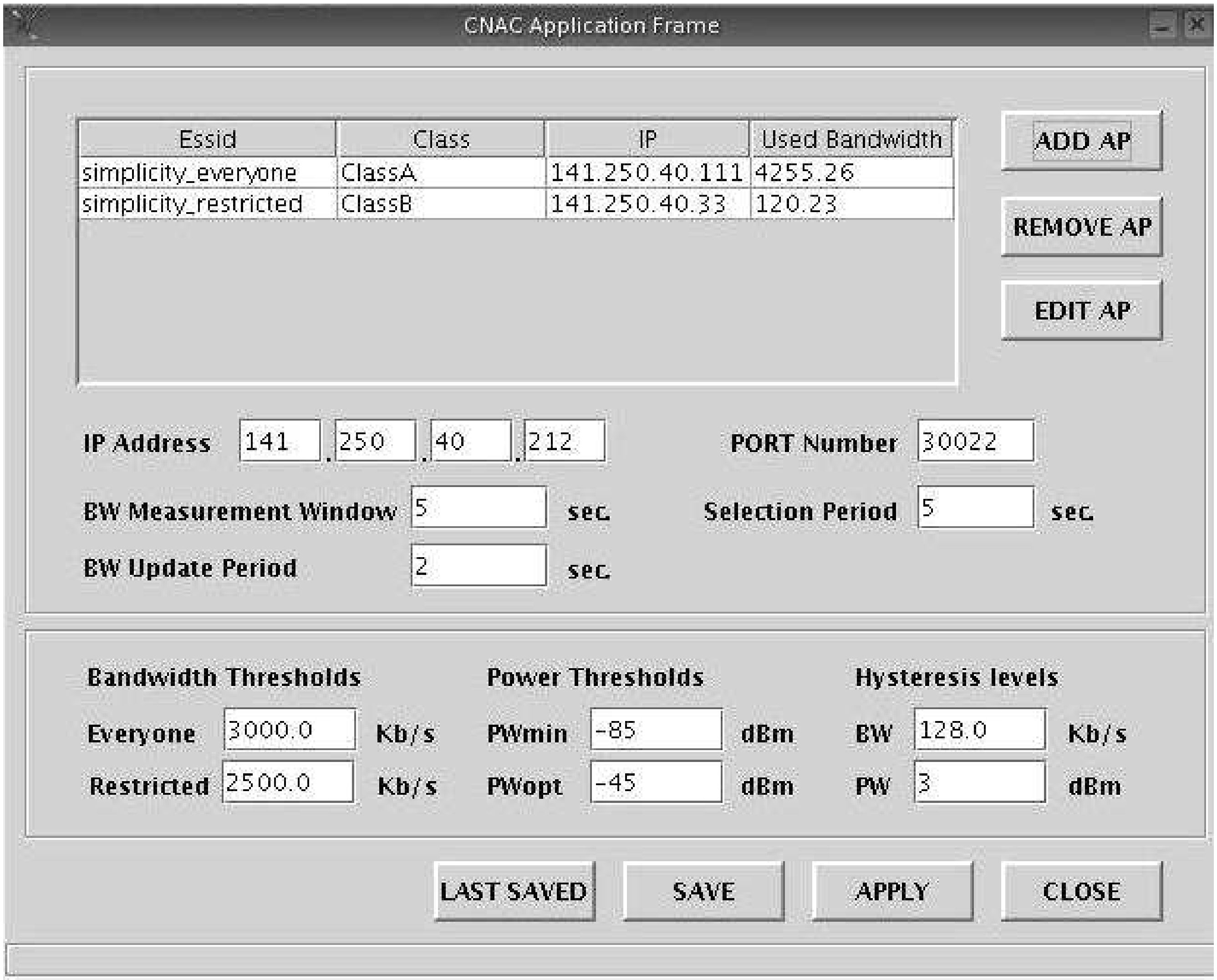

SPA - Simplicity Personal Assistant
SPA
Settings
CNinfo
Campus Network Info
Connection Information
Connection Status:
Connected
User Role:
Student / Guest
AP Essid:
simplicity_everyone
Power Level:
- 30
Web Surfing Mode:
Normal
Simplicity
09:52 am

Fig. 18



CNAC Application Frame
Essid
Class
IP
Used Bandwidth
simplicity_everyone
ClassA
141.250.40.111
4255.26
simplicity_restricted
ClassB
141.250.40.33
120.23
ADD AP
REMOVE AP
EDIT AP
IP Address
141
250
40
212
PORT Number
30022
BW Measurement Window
5
sec.
Selection Period
5
sec.
BW Update Period
2
sec.
Bandwidth Thresholds
Everyone
3000.0
Kb/s
Restricted
2500.0
Kb/s
Power Thresholds
PWmin
-85
dBm
PWopt
-45
dBm
Hysteresis levels
BW
128.0
Kb/s
PW
3
dBm
LAST SAVED
SAVE
APPLY
CLOSE

Fig. 19

- Jan goes to the Campus to meet his friend Nick, who is a professor of networking. Since Nick is temporarily busy, he offers to Jan his Simplicity-enabled laptop, provided with a wireless card, to browse the web. Jan plugs his SD into the laptop.
- The Simplicity system recognizes Jan as a guest.
- The Simplicity system drives the terminal towards the AP belonging to Class A (AP_A), which is currently unused, without requiring any effort/action from the user.
- Jan starts browsing the web in the normal mode.

Function: load-based access network selection

- Another user starts using AP_A, so that its load goes above L_E. In order to maintain high the quality of the browsing session, the mobile terminal used by Jan is automatically driven towards the AP belonging to Class B (AP_B), which is currently unused. No effort/action is required from Jan.
- Jan continues browsing the web in the normal mode.

Function: network/application service personalization

- A professor starts using AP_B, so that its load goes above L_R. Due to the limited privileges of Jan (he is a guest) and the current network status, the terminal used by Jan is driven back towards AP_A and Jan is constrained to browse in the limited mode.

Function: load-based access network selection

- At this stage, Nick kindly asks Jan for his laptop. Jan removes the SD from the laptop and Nick plugs his SD into it.
- The Simplicity system recognizes Nick as a professor.
- The terminal is now driven towards the best access, which currently is AP_B. No effort/action is required from Nick.
- Nick starts browsing the web in normal mode. Since Nick is a professor, he enjoys a service better than students/guests.

Function: power-based access network selection

- Nick moves from the current location and the terminal is automatically driven towards AP_A to maintain the network connection. No effort/action is required from Nick.
- Nick continues browsing the web in the normal mode.

Fig. 20

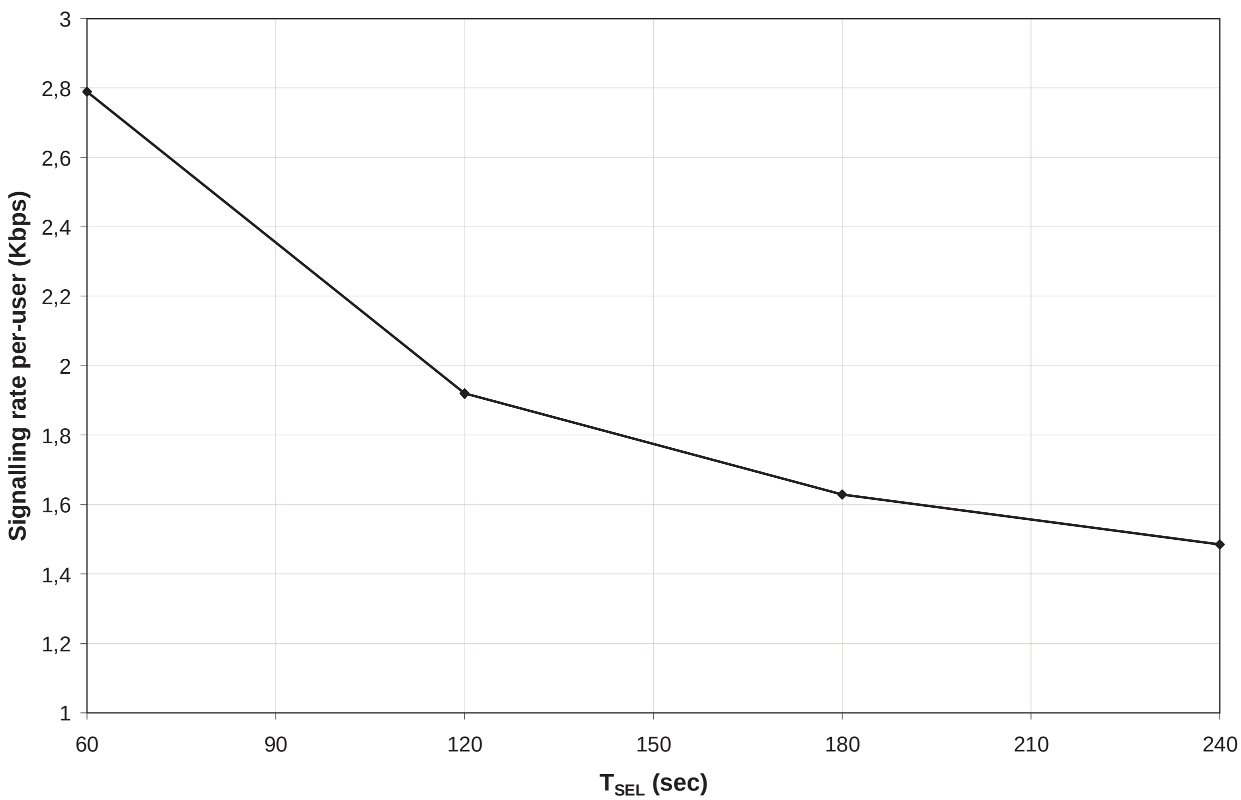

3
2,8
2,6
2,4
2,2
2
1,8
1,6
1,4
1,2
1
60
90
120
150
180
210
240
Signalling rate per-user (Kbps)
T SEL (sec)

Fig. 21

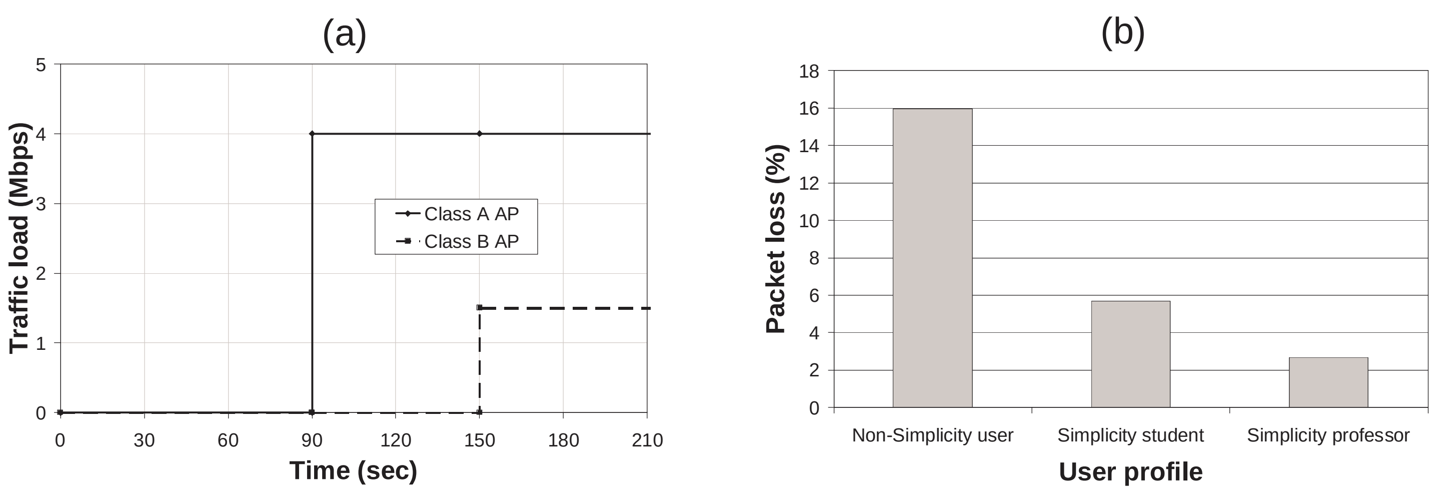
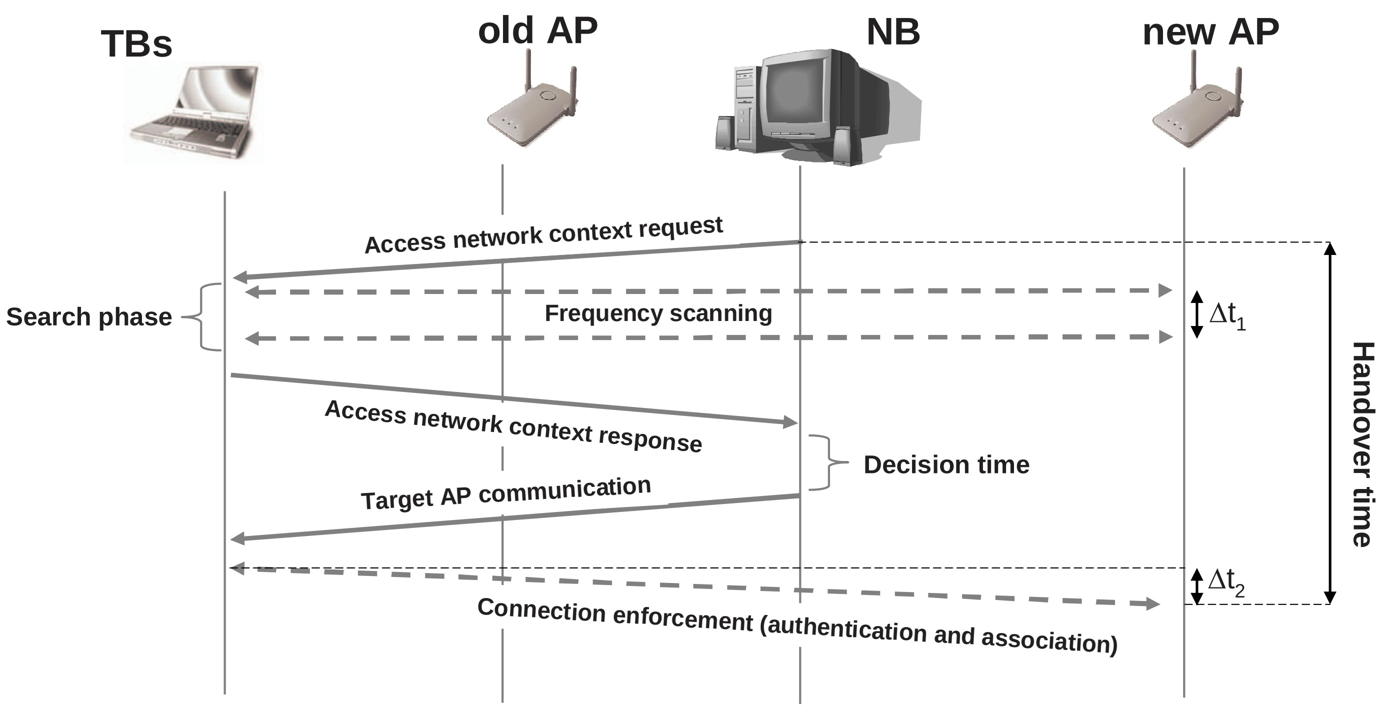
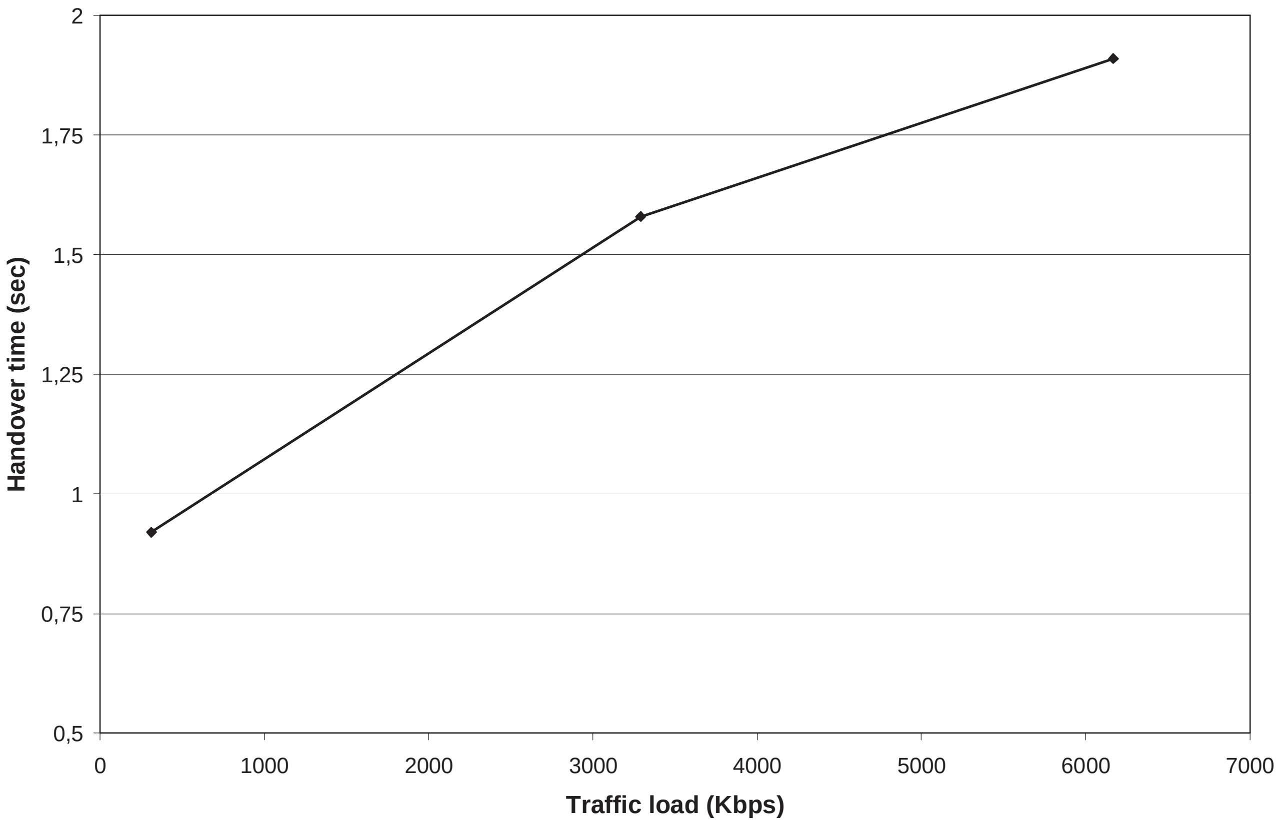

(a)
Traffic load (Mbps)
5
4
3
2
1
0
0
30
60
90
120
150
180
210
Time (sec)
Class A AP
Class B AP
(b)
Packet loss (%)
18
16
14
12
10
8
6
4
2
0
Non-Simplicity user
Simplicity student
Simplicity professor
User profile

Fig. 22



TBs
old AP
NB
new AP
Access network context request
Search phase
Frequency scanning
Δt1
Handover time
Access network context response
Decision time
Target AP communication
Δt2
Connection enforcement (authentication and association)

Fig. 23



2
1,75
1,5
1,25
1
0,75
0,5
Handover time (sec)
0
1000
2000
3000
4000
5000
6000
7000
Traffic load (Kbps)