TWELVE
Test Bed and Demonstration Planning

R. Lo Cigno[1], V. Ammirata[2], M. Brunato[1], D. Di Sorte[3], M. Femminella[3], R.G. Garroppo[4], D. Giustiniano[2], A. Ordine[2], G. Reali[3], S. Salsano[2], D. Severina[1], I. Tinnirello[5], L. Veltri[6]

[1]Dipartimento di Informatica e Telecomunicazioni – Università di Trento
[2]Dipartimento Di Ingegneria Elettronica – Università di Roma II "Tor Vergata"
[3]Dipartimento di Ingegneria Elettronica e dell'Informazione – Università di Perugia
[4]Dipartimento di Ingegneria dell'Informazione: Elettronica, Informatica e Telecomunicazioni – Università di Pisa
[5]Dipartimento di Ingegneria Elettrica – Università di Palermo
[6]Dipartimento di Ingegneria Informatica – Università di Parma

***Abstract*— We present the detailed planning of the test-bed, experimental, implementation and demonstrative activities that will be carried out within the TWELVE PRIN Project.**

During the entire second year of the project, in parallel with fundamental research, all the partners of the project will carry out an extensive campaign of experiments and demonstrations with the aim of disseminating the more mature results of the project that have major practical impacts. The overall organization of the experimental activities is a nation wide test-bed of HotSpots, all coordinated among them, that enables the presentation and demonstration of innovative services, algorithms, protocols and management techniques developed within the project.

This document describes and discussed (nearly) all the activities foreseen within the TWELVE experimental Work Package, outlining the expected results and the temporal development.

I. Introduction

Although the PRIN TWELVE Project is focused on fundamental research, from the very beginning we devoted an entire work package to experimental activities aimed at the demonstration of the achieved results that are more mature and close to possible practical and/or commercial implementation and exploitation. This contribution defines the road-map of the demonstrative activities devised within TWELVE.

Making justice of the large amount of research done in the project with demonstrations is a gruesome task. The easy approach would be the selection of some appealing and 'aesthetic' result, setting up a demo room in the coordinating University (University of Rome II – 'Tor Vergata,' very central and convenient), and invite colleagues to visit it.

We took the hard way. The 'demo' will indeed be a nation wide test-bed, involving all the research units that will demonstrate different activities and results, unified by a framework for users and service management that we call *UniWireless*.

UniWireless is a collection of coordinated HotSpots, all managed by the *Uni-Fy* gate provided by the University of Trento and described with some detail in Sect. II.

This work is supported by the Italian Ministry for University and Research (MIUR) under the PRIN project TWELVE (http://twelve.unitn.it/)

The Universities of Pisa and Palermo provides add-on features focused on QoS (802.11e) and advanced scheduling techniques, described in Sect. III.

The Universities of Perugia and Trento integrate in their HotSpots advanced services based on distribution, publish/subscribe architectures, and advanced localization, as described in Sect. IV.

Additional activities on advanced authentication and mesh networking are integrated in the nation-wide test-bed by the Universities of Roma II and Parma, as discussed in in Sect. III-B and Sect. IV-B

Sect. V ends this contribution giving the Gantt chart of the testbed implementation and some conclusive remarks on the contribution of the Demonstration to the state of the art. Fig. 1 gives a snapshot of the demonstration activities in the six universities realizing the *UniWireless* test-bed.

II. Demo Coordination and Common Activities

The burden of the demonstration coordination is on the University of Trento, which provides the main support and the initial software for the management and interconnection of the *UniWireless* HotSpots.

The idea of a set of HotSpots realized following the philosophy of Open Access Networks (see [1] for a detailed description of this approach), where the local University plays the role of access network provider, while all the remote Universities play the role of remote authenticators or service providers, is the leading theme under all the demonstration activities. All the other, sometimes more complex experiments and demonstrations are, as far as possible, integrated within the local HotSpot of the University that manages the specific demonstration. For instance, a visitor to the University of Perugia will have access to the local HotSpot, registering and authenticating using his account at his home University[1], and, in some specific are of the HotSpot will be able to access the

[1]The University of Trento will manage an additional, Radius-based authenticator devoted to visitors wishing to access the Demo, but not belonging to any of the accredited Universities.

Publish-Subscribe experimental services set up by the research group in Perugia.

The remaining part of this section is devoted to the description of the philosophy and overall architecture of the *UniWireless* HotSpot management.

A. The Uni-Fy gate User Management Framework

Uni-Fy gate is a Wireless LAN and HotSpot management system built starting form the source software named WilmaGate [2]. Starting from WilmaGate, the TWELVE team in Trento realized the *Uni-Fy* gate, adding functionalities and enhancing its flexibility, to support the *UniWireless* system. The *Uni-Fy* gate is implemented in C++ as a collection of user-space applications. Its modular design is intended for easy implementation of new features: the general framework is that of an open test-bed, where innovative algorithms and procedures can be implemented and tested.

The whole architecture pivots around an authentication system following the Open Access Networks philosophy [1]: all the users who want to access the public network must be authenticated by an ISP or agency which is not (normally) the same entity owning and controlling the access network. The goal of our system is the feasibility demonstration of flexible and secure management in WLAN and public HotSpots, including the integration with 3G networks.

The main features of the system are: i) support of multiple external authentication providers, ii) support for several authentication techniques, iii) firewalling, and iv) accounting.

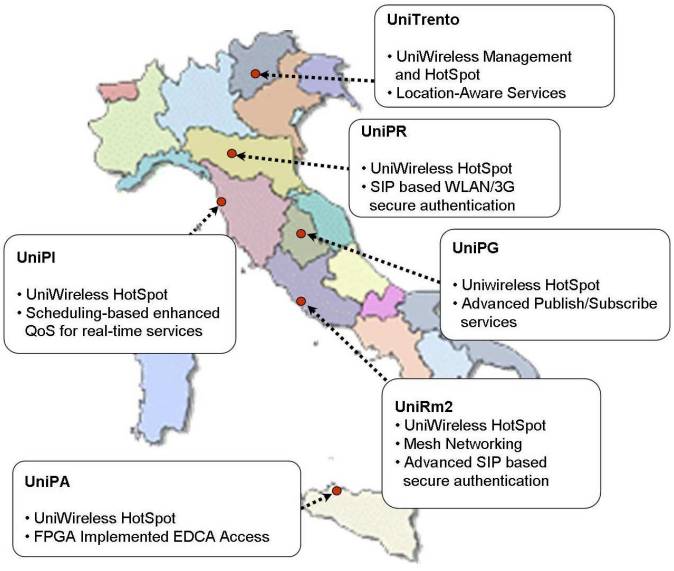

Fig. 1. Overview of *UniWireless* deployment within the TWELVE project.

Currently the implemented authentication procedure is based on the "captive portal" technique via web pages. A user opens his browser and requests a web page; the system intercepts the request and redirects the user's browser to an authentication page where the user selects the preferred authentication provider. Then the user inserts his data (i.e. user-id and password) in a form and sends them to the remote authenticator through a secure connection. The authentication procedure ends with two notification of "user identified" from the remote provider: one is sent to the user and one is sent to the management system. The authorization renew is managed automatically by a pop-up window, refreshed with specific periodicity, ensuring protection from spoofing.

As shown in Fig. 2, the system architecture is made up of two main components: Gateway and Gatekeeper.

The *Gateway* is a layer-3 switch that receives all the packets from the WLAN and puts them on the right interface. If the received packet belongs to an authenticated user, it is forwarded, or otherwise if it belongs to a non-authorized user, it is forwarded to the Gatekeeper.

The routing policies are taken according to an internal firewall table: each firewalling rule is made up of a pair of client identifiers (IP, MAC) and an action that can be one among: i) "allow packet forward to external LAN," ii) "send packet to Gatekeeper," and iii) "drop the packet". The firewall table is kept as simple as possible, to avoid computational bottlenecks.

The *Gatekeeper* performs all procedures that require more processing than packet inspections. This component manages packets received from the Gateway and these can be classified in three categories:

- DHCP requests;
- packets belonging to unauthorized users. The system manages the request with a web page redirect to an authentication web page;
- packets not belonging to any previous class: the module discards them.

The modularity of the system allows adding new plugin modules to extend the collection of supported protocols and supported authentication methods.

The Gatekeeper interacts also with any remote authentication provider to receive information about authorization of users. The remote provider must be a trusted entity with a previous agreement between the provider and the manager of the local *UniWireless* HotSpot. The Gatekeeper maintains a list that stores information about users: this is a superset of information of the Gateway's access list. When the Gatekeeper receives information from a remote authenticator, it updates its table and forces also an update in the Gateway's list.

Users' privacy and database-sharing are worth a few comments. The system does not manage user's private data (user-id and password), neither it can spoof them because personal data are encrypted on a secure connection (HTTPS). Furthermore the remote authentication provider does not allow the system to access its own database: the authentication procedure inquires an authentication server of a provider and only this server can interacts with the private database. For further and in depth information about the features of Gateway and Gatekeeper we refer the reader to [2] and [3].

The modularity of the authentication system allows different network configurations for the system: in this way each implementation of the system can be customized to meet each

HotSpot requirements. The Gateway and Gatekeeper are two different processes that interact one another, but they can be run on the same machine or in different machines. The possible configuration are distinguished by the number of hardware component involved in the system and the requirements about the number of network interface cards (NICs) and the availability of public IP addresses. For instance the configurations can be:

- one PC with two NICs and one public IP address;
- two PC (both with two NICs) and one public IP address;
- two PC (both with two NICs) and two public IP address.

Other configurations can be used and each of them can be programmed through a plain text file, describing the system architecture (association of NIC MAC and IP address and port). With "public IP address" it is meant that the provider deploying the *Uni-Fy* gate needs to ensure that the public interface of the Gatekeeper is reachable from the external network (specifically from remote authentication servers).

B. UniWireless: A Nation Wide Experiment

A generic example of deployment of the *Uni-Fy* gate is shown in Fig. 2. The entities that interact in this configuration can be classified in two categories:

- Connectivity Provider. An entity deploying the *Uni-Fy* gate to grant Internet access to authenticated users;
- Authentication Provider. An entity supporting a secure database of users entitled to access HotSpots managed with *Uni-Fy* gate.

The collection of all Connectivity and Authentication providers forms the *UniWireless* test-bed. Fig. 1 shows the Universities forming *UniWireless* and their specific additional role. Obviously a provider can be simultaneously a connectivity and an authentication provider.

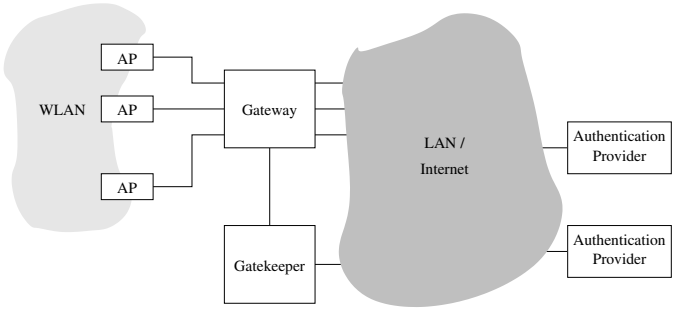

Fig. 2. Overview of the *Uni-Fy* gate system

Each University will be an authentication provider (it shares its user database) and a connectivity provider (it uses *Uni-Fy* gate system). In such scenario the mobility of researchers will not a problem: nomadic users belonging to any University involved in TWELVE project, can access from all the networks belong to the *UniWireless* test-bed without the need to request access credential to different network managers.

III. Enhanced Medium Management

The enhancement of QoS must start for the medium usage, hence in the TWELVE project a lot of activity is devoted to this subject and the demo emphasizes some of the most interesting activities undertaken.

The medium management has many different facets and implications: MAC scheduling, QoS and priority mapping between the network and the link layer, management of distributed resources.

In the TWELVE test-bed we concentrate on three different topics: i) scheduling between the IP and link layer; ii) service differentiation within the DCF (EDCA); and iii) distributed resource management via mesh networks.

A. Improved Scheduling and QoS Support

On the one hand, we will explore some differentiation techniques, which do not require any hardware upgrade of the existing Access Points (APs) and network cards, but are obtained by means of software upgrade only. On the other hand, we will explore the actual differentiation capabilities of the 802.11e MAC extensions, by means of a proprietary implementation of 802.11e cards. Commercial 802.11e cards are not yet available, since the Task Group 802.11e finished its work in July 2005 and the final draft is still under approval. Our implementation is based on version D8.0 [4] of the draft standard.

1) Enhanced Scheduling: The first demonstration will be carried out in Pisa. The solutions do not require any hardware upgrade of the AP and NICs. In particular, the demonstration is aimed at showing the performance improvements produced by the insertion of an appropriate scheduling algorithm able to take into account both the transport requirements of traffic flows and the quality of radio channel experimented by the receivers. The rationale in the set up of the test bed is to show the negative effects on the following two aspects of the 802.11 MAC on some important services such as video distribution,Voice over IP and Web browsing. The theoretical framework at the basis of this activities was published in [5], [6].

The first aspect is related to the characteristic of the Distributed Coordination Function (DCF) to provide a fair sharing of the medium in terms of channel access probability. In ideal conditions (i.e., when all frames are of equal length and transmitted at the same data rate, and none is lost due to collision or low radio channel quality), this characteristic implies that all the stations are provided with an equal share of medium occupancy time, which reflects in an equal share of the overall bandwidth. However, in real scenarios some frames may not be received correctly, since the radio channel quality may be poor and collisions may occur. Lost frames are retransmitted until the Maximum Retry Limit (MRL) is reached or the frame is correctly delivered. Also, the transmitter can adopt more robust, but less efficient modulation schemes.

The second aspect is that commercial APs use a single FIFO queue. All the buffered frames must wait for the currently served one to be dequeued; event that occurs after either the

successful frame transmission or the frame discard due the MRL. This phenomenon, known as "head-of-line" blocking, implies a frame transmission delay that depends not only on the radio channel quality experienced by the particular station to which the frame is addressed, but also on other factors such as the position of other stations and the number of frames enqueued in front of it.

The combination of these two aspects leads to have all the downlink flows affected by increased delay, possible packet loss due to buffer overflow and reduced throughput at application level. As a consequence, all users can experience bad service quality, even in the case a single station is subjected to bad radio conditions. In order to overcome this drawback, we have developed the Deficit Transmission Time (DTT) scheduling algorithm, and we have implemented it in the AP at device driver layer. The DDT scheduler is able to take into account the actual radio channel quality in terms of the amount of time needed to correctly transmit a single frame from the AP to a station. DTT permits to achieve a better utilization of the medium and to guarantee the isolation among flows addressed to different stations.

Demonstration Network Scenario

The test bed used in the demonstration is shown in Fig. 3 and is composed by the following key elements:

- the prototype AP including the DTT scheduler;
- a notebook to access FTP download services (Station1);
- a notebook utilized as client to video-streaming services (Station2);
- a SIP IP phone with 802.11 NIC (Station3);
- an FTP server (Host4);
- a video-streaming server (Host3);
- an IP phone (Host2);
- a Proxy and Registrar SIP Server (Host1).

The prototype AP is an Access Cube by 4G-system GmbH. The choice to use this device for the development of our prototype is connected with the flexibility of the considered AP, given by its particular Linux embedded system architecture. In particular, the main feature of this system is the availability of an open source driver that permits the insertion of new functionalities to the basic 802.11 functions. Furthermore, the possibility to insert new software modules is supported by the 32MB flash memory and a 64 MB RAM memory of the Access Cube.

The demonstration will show and evaluate the improvements introduced by the DTT scheduler on the users perceptive quality of service. In particular the service quality protection when users move around will be shown.

2) 802.11e-like HotSpot Building: The activities on EDCA effectiveness for service differentiation and QoS support build on the DCF developed in Palermo on Field Programmable Gate Array (FPGA) in a previous national project (the PRIMO FIRB project). By interfacing the programmable DCF to a MAC-less network card, we can easily manage the upgrades required for EDCA implementation. The ongoing development of the minimal EDCA functionalities required to run the experimental part is documented in [7] The resulting network

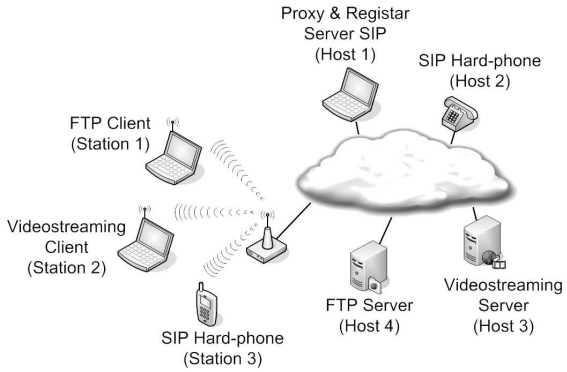

Fig. 3. DTT HotSpot Test-bed; the AP is enhanced with the DTT soft scheduler

card does not provide all the 802.11 functionalities (such as Association and Authentication). In order to avoid the implementation of the complete set of 802.11 functions, which do not have any role in EDCA testing, we are planning to build a special HotSpot with QoS capabilities, according to the architecture shown in Fig. 4. The QoS-AP is made up of a standard AP and a computer equipped with our EDCA card, working on the same channel. Legacy 802.11 cards can associate to the standard AP card, while our 802.11e clients are programmed to associate to the AP EDCA card. The LLC layer in the computer connected to the EDCA card is modified in order to classify different traffic flows in different access categories. The best effort traffic (the traffic for the legacy cards) is routed to the AP, while the other traffic is routed to the EDCA card. Note that the two network cards at the QoS-AP contend for the channel access as two separate stations. Thus, the virtual collision resolution is not implemented, since no coordination is provided among the two different cards and only the frame transmissions on the radio channel can reveal collisions among the best effort and QoS traffic. However, this architecture can be easily deployed with our technology and allows a first experimental evaluation of EDCA.

In the TWELVE test-bed, we are planning two main experimental activities:

- *Study of the prioritization provided by EDCA, with different MAC parameter settings, in case of contention among upstream traffic flows.* The demonstration regards both the measure of MAC-layer performance figures, such as bandwidth repartition and packet loss rate, and the evaluation of the QoS perceived at the application layer. For example, we compare the video-streaming quality of two different traffic flows from two 802.11e clients (exploiting different access parameters) to the QoS-APs, while varying the best effort load conditions.
- *Study of coexistence among upstream and downstream traffic flows.* The demonstration shows how the download/upload bandwidth splitting can be easily tuned through EDCA. Specifically, we evaluate the MAC-layer and transport layer performance figures, in a scenario in which the legacy stations generate upstream traffic, while

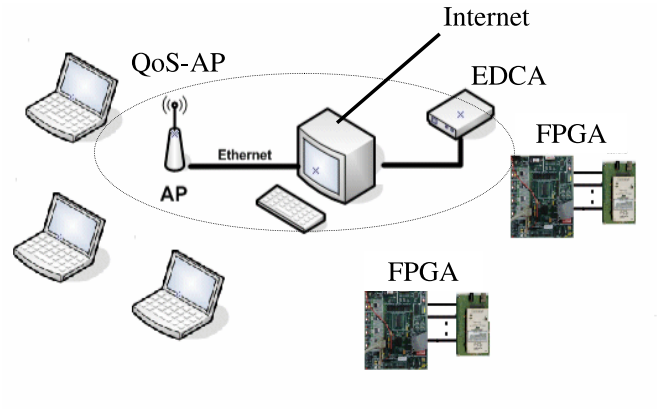

Fig. 4. 802.11e HotSpot Test-bed

the 802.11e clients are downloading data.

B. Mesh Networks

The following terms taken from [8] are used to describe IEEE 802.11 Mesh Network basic concepts .

1) WLAN Mesh - A WLAN Mesh (previously known as ESS Mesh) is an IEEE 802.11-based Wireless Distribution System (WDS) which is part of a Distribution System (DS), consisting of a set of two or more Mesh Points interconnected via IEEE 802.11 links.
2) Mesh Point - Any IEEE 802.11 entity that contains an IEEE 802.11-conformant Medium Access Control (MAC) and Physical Layer (PHY) interface to the Wireless Medium (WM), that is within a WLAN Mesh.
3) Mesh AP - Any Mesh Point that is also an Access Point.
4) Mesh Link - A bidirectional IEEE 802.11 link between two Mesh Points.
5) Mesh Path - A concatenated set of connected Mesh Links from a source Mesh Point to a destination Mesh Point.
6) Mesh Path Metric - Criterion used for Mesh Path Selection.

An 802.11 WLAN Mesh is a network where access points are interconnected through wireless links, typically based on 802.11 themselves [9]. Mesh networks have been recently developed by commercial vendors (e.g. Tropos Network, Firetide, MeshDelivery, etc.), community networks [10] and academy campuses (e.g. the MIT RoofNet [11]) By academic deployments are ongoing for research purposes. WLAN Meshs are under standardization within the activity of 802.11s Task Groups [8].

In the TWELVE project, our goal is to develop a Mesh Access Point characterized by the capability to support service differentiation. The idea is to provide differentiated routing paths, dedicated to different, dynamically instantiated classes of packets. This is accomplished by combining the two following key approaches.

First, different routes are quasi-statically deployed by means of multiple overlapping spanning trees. Our ultimate goal is to reuse, as much as possible, the features of the well known 802.1Q Multiple Spanning Tree Protocol (MSTP). Our current research activity is dedicated to assess the impact of different Mesh Path Metric functions which not only depend on the link rate, as in the traditional STP protocols, but also depend on low-level measurements which, gathered from the PHY and MAC Management Information Base (MIB) statistics, allow taking into account channel quality, interference, and transmission/collision effectiveness.

Second, the routing decision on available MSTP paths is very flexible and dynamic. Packets can be differentiated not only on the basis of static information (as traditionally done in 802.1Q Virtual LAN), such as IP-level tagging, MAC/IP source/destination addresses, protocol information, etc. Packet tagging may be dynamically performed taking into account feedbacks from the wireless network. For example, upon congestion, or for load balancing purposes, packets from the same station, or addressed to a same station may be diverted to an alternative path simply by tagging them with a different MSTP indicator.

The work described above has been just started, and its conducted within the development of a layer-2 Mesh demonstrator. Different instance of spanning tree will be managed in the prototype using an appropriated tool, called “distributors”. Our testbed implementation is currently based on 802.11 a/b/g wireless interfaces that use the Atheros 5213 chipset, which are driven by the open-source Multiband Atheros Driver for Wifi (MADWiFi) [12].

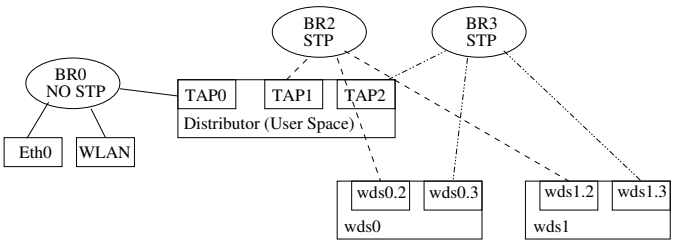

Fig. 5. Basic scheme of the distributor

1) Distributor tool: A preliminary implementation of a Linux-based Mesh AP is in progress. At the moment, we have developed a tool that can distribute the marked packets within a specific bridging tree and manage multiple independent instances of layer 2 spanning tree protocol. The resulting Mesh AP architecture is exemplified in Fig. 5. All packets coming from the air at the hardware interface (WLAN) are collected and handled by the BR0 bridge. The packets that are not for the ethernet interface are sent to TAP0 interface. Packets sent to this interface are handled by a user-space process, called “distributor”, which has the task to map packets onto different bridges (BR2 and BR3 in the Fig. 5). Such mapping is the core of the proposed operation, and can be based on either static or dynamic policies (static mappings based on information gathered from the MAC frame and the IP packet header have been tested; the extension to dynamic tagging is straightforward from an implementation point of

view). Packets written by the user-space process are injected back into the kernel networking subsystem and are forwarded using the appropriate bridge table: BR2 and Br3 are two concurrent bridge instances characterized by two independent forwarding tables. In the final version of the experiment we plan to maintain them dynamically via the MSTP protocol). Finally, frames exiting the bridge modules are re-tagged at the input of the wds cards and transmitted over the air. Frame tagging is based on the 802.1Q features already available in the Linux kernel. The advantage of the proposed implementation is fourfold:

- it uses kernel features already implemented;
- it takes advantage of TAP devices;
- it facilitates debugging and runtime diagnostics ;
- it is simple to upgrade other configurations.

2) Link level measurement campaign: Our research has the additional goal of defining Mesh Path Metric for the spanning tree bridge instances. To achieve this task, we deem crucial to lead experimental evaluations, because, to the best of our knowledge, no adequate model is present in the literature.

Furthermore some reported results are fuzzy and not convincing as the reported independence of packet loss with respect to the received signal strength [13]. A more elaborated and convincing explanation may turn into a simple characterization of Mesh Links, and thus of the Mesh Path Metric model.

The campaign uses laptops with a Linux O.S. version 2.6 equipped with a 802.11 a/b/g compliant card with a 5 dBi extern antenna driven by the AR5213 chipset from Atheros (via the MADWIFI driver). The available technology enables the comparison of 802.11b PHY implementation, based on Barker code at the transmitter and RAKE receiver, with 802.11a/g systems, where the OFDM modulator has the enormous advantage of robustness to multipath, which is commonly considered as the main reason of packet loss in outdoor channels.

The detailed results of the measurement campaign are presented in [14] and are used int the test-bed to tune the Mesh Path Metric.

IV. Services, Localization and Distribution

The last aspect of service differentiation we decided to include in the TWELVE demo is related to high-level, enhanced service provisioning. We explore both the possibility of differentiation the access based on service publishing and enhancing the usage experience via advanced, location based recommending systems.

A. Service publishing in 802.11 access networks

In advanced HotSpot scenarios the "service" may go well beyond simple connectivity. On the one hand operators may like to differentiate their offer in terms of security, Quality of Service (QoS), application services. On the other hand, users may want to select the Wireless ISP (WISP) and/or the AP to attach to according to a number of factors beyond the signal strength (e.g., security, enrollment, QoS, supported services, price). It is clear that first associating/authenticating, and only afterwards discovering these factors may be inefficient and bothering for users. An effective AP selection is challenging because the 802.11 standard does not provide an efficient support in this direction. This is the reason why there is a strong necessity to develop a mechanism that allows the Mobile Terminal (MT), and thus the user, to access a larger set of information before association/authentication. Clearly, this need becomes more and more urgent if there are a large number of APs/WISPs available to users in a given area, each of them with a different service offer.

1) An 802.11-tailored solution: Our goal is to provide users with a subset of service-related information via 802.11 prior to association/authentication to make the access selection work properly. In literature there are very few documents on this subject [15]. The IEEE 802.11 Task Group u (TGu, [16]) has just started the work to standardize a mechanism to retrieve from layer 2 information useful for the access selection.

Since the typical procedure of a TG to add new functions to the standard is to define new mandatory or optional IEs within management frames, we propose to set a new IE containing service-related information useful for network selection.

It is worth noting that the set of information broadcasted within beacons should be limited to avoid stuffing them and has to be helpful for a preliminary screening of the service peculiarities of the accesses. In this view, interesting inputs to the access selection could be:

- the price charged for the access;
- the class of users permitted to access;
- the type of enrollment, authentication, and ciphering;
- the application service offers;
- the IP address management;
- the type of support for service discovery/configuration.

Once the MT has connected to the target AP, in order to get more refined service attributes (e.g., configuration info) to enjoy a given service, the use of service discovery protocols (such as Service Location Protocol, SLP [17]) is needed.

Although the above mentioned solution is perfectly backward compatible with the standard, the implementation requires firmware updates not only to wireless cards, but also to APs. Thus, to have a working solution compliant with existing equipments, our choice for the demonstrator has been to encode this information within the SSID (i.e., the name of the access), which is a standard IE broadcasted into IEEE 802.11 beacon frames. We used the character '@' as separator between the network name and encoded service-related information. Thus, the format of the SSID is "networkname@<Code,Value><Code,Value>...", where the pair <Code,Value> identifies the service feature (Code), and the relevant configuration (Value). The network name can be the name of the operator providing the access (TWELVE, in our case). This solution allows emulating the presence of a new IE in beacons. In the following, the APs with this kind of SSID are called "TWELVE-compliant APs".

2) Current demo architecture: Our objective is to set up a network configuration able to publish and provide a multi

service 802.11 access environment. The components of the demo architecture are:

- in the network side: standard AP, Virtual Access Point (VAP), SLP Directory Agent, DHCP server, video server, Radius server, and TWELVE Data Sharing (TDS) server;
- in the terminal side: TWELVE Wireless Selector (TWS) tool, TDS client, Radius client, and SLP User Agent.

There are a number of 802.11 accesses (one provided by the standard AP and the others provided by the VAP). The service differentiation is provided on a per-access basis in terms of:

- network service offer (QoS and security);
- service publishing (beacon-based and SLP-based);
- application service offer;
- access permission.

A VAP is a logical entity that exists within a physical AP [18]. Each VAP appears to be an independent, physical AP and emulates the operation of a standard AP at the MAC layer (it represents an instantiation of a complete 802.11 MAC including BSSID, SSID, and capability set). One of the main advantages of this architecture is that a WISP can differentiate the offered services within the same physical AP. Furthermore, a number of WISPs can share the same physical device. Thus, a VAP device is quite suited for the deployment of a multi-service WLAN. In our lab we make use of a Colubris MSC3200 with VAP technology with up to 16 concurrent SSID/BSSIDs [19].

The main components of a SLP architecture are [20]: (i) User Agents (UAs), which discover services; (ii) Service Agents (SAs), which advertise the services they represent along with the relevant attributes; (iii) Directory Agents (DAs), which accumulate service information and respond to service requests from UAs. Clearly, service information may be also statically stored within DAs. Also, services may be grouped in a number of scopes according to specific policies. In this regard, we assume that services are grouped on the basis of the different wireless network accesses (i.e., SSIDs). This means that each service is associated with one or more SSIDs, and we implement the differentiation of the service publishing at the SLP level as well.

We offer a number of different application services:

- classic Internet access;
- video service;
- TDS service;
- printing service.

The TDS service is an advanced data sharing mechanism exploiting the broadcast, band-limited 802.11 channel, useful when there are non-confidential, hot contents. The service architecture consists of a TDS module on the server (master) and a TDS module on the client(s). The server TDS is a SQUID-based proxy [21] able to generate, upon HTTP data requests from a client, unicast UDP advertisements towards the AP supporting the service. This allows publishing the transmission and providing MTs with the information needed to enable the data acquisition process. This architecture allows all TDS clients under the same AP sharing the same contents. When a user decides to attach to the AP supporting the TDS service, then the TWS is in charge to automatically retrieve from the SLP DA all the information needed to configure the service.

3) The TWELVE Wireless Selector (TWS): The TWS is a Java-based graphic control tool running on the MT and it is able to (i) perform wireless network scanning, (ii) present to the user the list of surrounding APs (both legacy and TWELVE-compliant) (see Fig. 6) and the service peculiarities of the TWELVE compliant APs (see Fig. 7), (iii) perform user-driven handovers, (iv) configure network parameters, and (v) show the current network configuration (from MAC to DNS).

Fig. 6. TWELVE Wireless Selector GUI

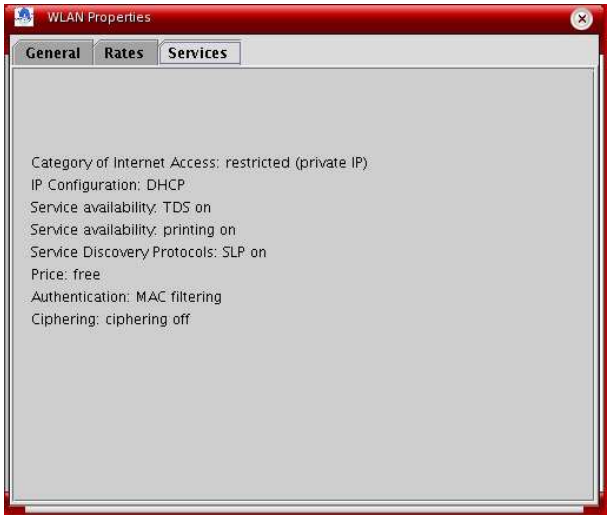

Fig. 7. Service-related information of a TWELVE-compliant AP

An additional important characteristic of the TWS is the capacity to perform wireless network scanning and to present to the user only the list of the points of access that match user preferences. This allows the user to further speed up the selection process. The preferences may be edited by the user in a window of the TWS.

The TWS control panel is also integrated with the SLP UA to get from a remote SLP DA more refined information

relevant to the services provided through the AP the MT is currently attached to. To this end, the TWS issues SLP queries with the value of the scope field set to the SSID of the current AP.

Thus, once the user has selected the AP to attach to from the rough service information obtained by beacons, then a given service may be properly configured through the TWS.

B. WLAN/3G secure authentication based on SIP

In this section we describe the portion of TWELVE test-bed that aims at demonstrating the feasibility of the integration within the *Uni-Fy* gate system of other authentication and securing schemes based on SIP. In particular, we will use the same scheme used in the 3GPP/IMS security framework [22].

Authentication is provided at service level (SIP) through standard SIP-Digest-AKA mechanisms, while confidentiality can be applied at IP layer through IPSec and packet filtering functions.

The basic idea is that the SIP based authentication mechanisms can interact with the *Uni-Fy* gate system improving the authentication capacities:the SIP based mechanism will coexist with the captive portal mechanism.

An authentication agent runs on the terminal and is capable to automatically perform the authentication procedure using SIP. The Authentication procedure can be based on stronger mechanism than user-name/password, for example we will emulate the same mechanism used in 3GPP network which is based on secret keys stored in the user USIM.

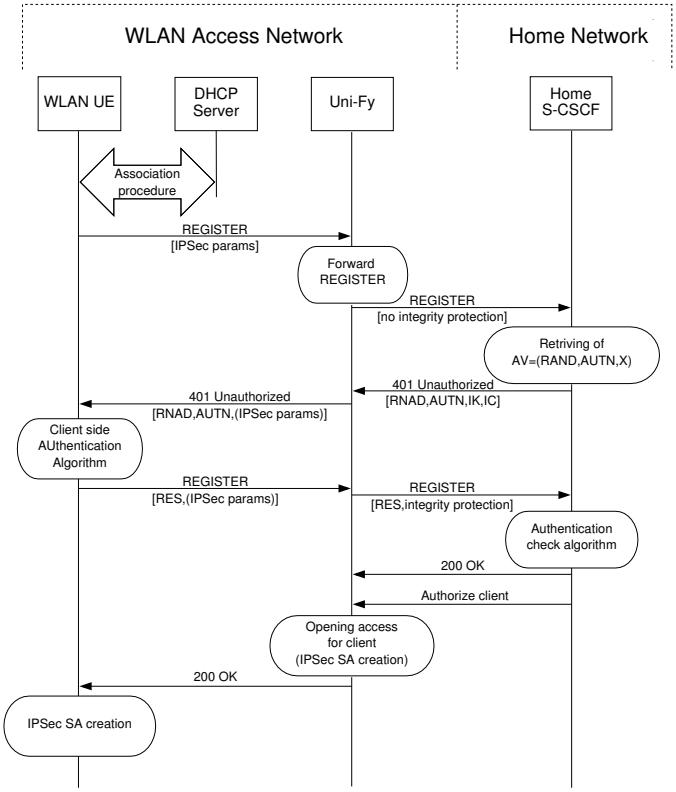

Fig. 8. Successful authentication procedure

The authentication procedure is depicted in Fig. 8. When the MT roams onto a new network, it tries to associate with a wireless AP. After the association the MT configures its network layer through local DHCP server. At this point the MT starts the SIP registration and SIP-Digest-AKA authentication procedure towards its home S-CSCF. The *Uni-Fy* gate system will be configured to forward the SIP registration messages towards the home SIP register server, which will perform the authentication procedure with the MT. If the SIP authentication procedure succeeds, the SIP server contacts the *Uni-Fy* gate system in the same way used by the ”traditional” web based authentication server. Therefore we assume that a trust relationship exists between the SIP server and the *Uni-Fy* gate. Optionally, a bidirectional IPSec Security Association(SA) between the MT and the *Uni-Fy* gate can be established.

In the demonstrator a MT (a laptop PC) accesses to a visited WLAN through the *Uni-Fy* gate system. The SIP server is running on a Linux box and can be located anywhere in the Internet. The SIP functions (both server and user authentication agent) have been developed in Java based on MjSip open source SIP stack [23]; they fully implement standard 3GPP/IMS SIP signalling. Fig. 9 shows the overall demonstrator layout.

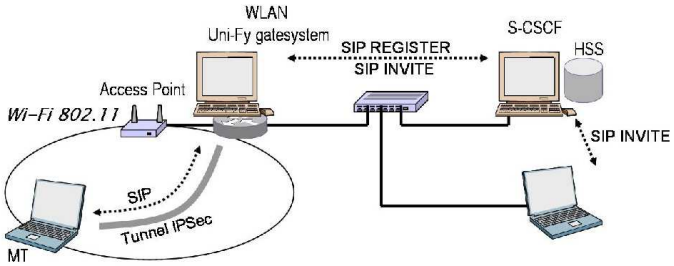

Fig. 9. SIP based WLAN/3G authentication demonstrator layout

C. Context-enhanced service provisioning

The common *Uni-Fy* gate AAA platform enables participants to fully exploit new services that take advantage of context information. Such services provide additional benefits to wireless users, promoting mobility and helping network operators and researchers to collect usage data.

A sample system, aimed at providing context data (in particular position) to user applications at the Faculty of Science of the University of Trento, is described in Fig. 10. The system is structured into three layers: *network*, *middleware* and *application*.

The *network*, at the bottom of the picture, is the collection of all networking equipment, in particular devices related to wireless access: APs, AAA gateways and mobile clients. We only consider components that are “context-aware”, in the sense that they are able to provide information that helps to build context representations. For instance, APs usually provide association information that can be used to estimate clients’ positions; clients, on the other hand, may optionally localize themselves by means of specialized hardware (GPS

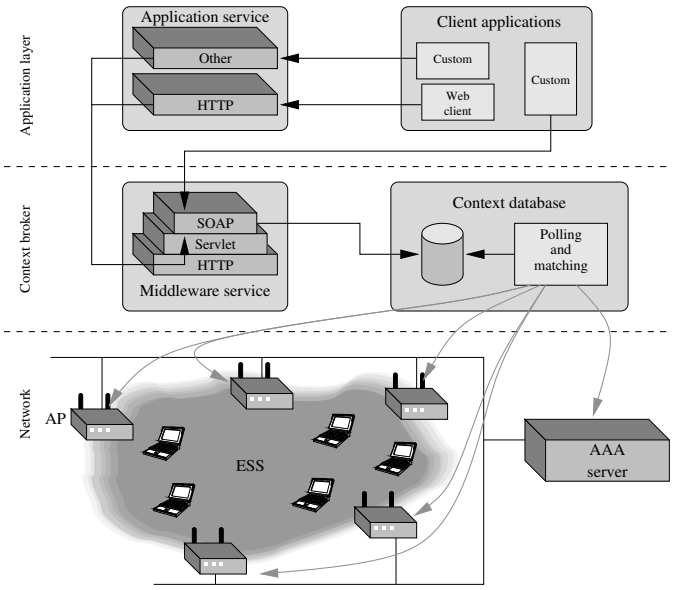

Fig. 10. Block diagram of the context-enhanced service framework

receivers, RFID tags) or by exploiting radio properties of the wireless medium [24]; the AAA gateway can complement location data with the identity of the users, their activity and time since login.

The *context broker* is the main component. Its purpose is to obtain context data by correlating information obtained by polling network devices and provide coherent context information to applications. It is composed of two logically separate components: the *context database* and the *middleware service*. The first basically contains a database of context information about users. Currently, time, identity and location data are considered. Such data are built by an application that receives information from network devices by using a mixture of methods, including SNMP polling and Syslog trapping; data are matched by common tags (for instance, APs and the AAA gateway both report IEEE802 MAC addresses, while client applications use IP addresses that are also known by the AAA gateway). Currently, user locations are inferred in an asynchronous manner. In fact, active logging functionalities of some AP models have been found to be unreliable, so users are not localized as soon as they connect, but shortly thereafter by a polling cycle. Information stored in the database is not directly accessed by applications. A *middleware service* is interposed in order to provide a standard protocol for querying, independence from the DB representation and additional functions such as location tracking, confidence estimates or search functions. The service is implemented as a SOAP Java servlet, thus appearing as a standard web service to upper layer applications.

Finally, in the *application* layer (upper part of Fig. 10) several types of user services can be found. The context middleware can be queried by a web server running a (PHP, ASP, JSP, CGI) script, by some other application service, or directly by context-aware custom applications.

An example of context-aware service currently available at the University of Trento is shown in Fig. 11. The captive-portal authentication mechanism implemented in the access network allows the system to offer the user a link to a local web page whose contents are tailored according to the user's identity, position and, possibly, preferences (either declared or inferred). The web page contains the current user's location, suggestions about the nearest available classroom, a poll system about university services, tools for the organization of student communities. Other envisioned services include the suggestion of less congested wireless locations, restaurant menu (for students nearby the cafeteria and only during lunch hours) and teachers' availability.

Another longer-term service will be a wireless video broadcasting system which shall be used to diffuse lectures and talks to students. The system shall take advantage of a mixture of coding techniques in order to ensure good video quality even in the case of high packet loss ratio, while offering optimum performance to properly positioned clients. In depth description can found in [25]

V. Implementation Timeline

Fig. 12 shows the availability of the different parts of the test-bed referred to the "project time." TWELVE started Dec. 1 2004, and will finish Nov. 30 2006. The first 12 months are compressed in the column "$<$ 13," since experimental activities were foreseen for the send year only; however, since many of them started well before, it seems correct to indicate the one that are already started.

Experiments and demonstrations are identified with a letter (**A, B, . . .**) whose meaning is shortly outlined in the following. All activities in the Gantt chart in Fig. 12 are split in two different parts (e.g. **A1** and **A2**), referring respectively to the deployment phase and the availability phase. During the deployment the experiment is available locally to the TWELVE partners and it is not integrated in the *UniWireless* framework. In the availability phase the experiment can be accessed by anyone visiting the *UniWireless* test-bed, albeit, clearly, only in the HotSpot supporting the specific activity.

- **A —** The first experimental activity has been the testing of the *Uni-Fy* gate in Trento, using it for the management of the WLAN in the Faculty of Science, comprising more than 15 APs and hundreds of daily users.
- **B —** The HotSpot in Trento becomes the first HotSpot of *UniWireless*, allowing remote authentication to users belonging to the other *UniWireless* partners. Proper authentication servers will be set up in the other partner Universities.
- **C —** The other HotSpots in Palermo, Pisa, Parma, Perugia and Roma Tor Vergata are set up and tested. Each one will comprise at least two AP managed by a local version of *Uni-Fy* gate.
- **D —** The team in Palermo integrates in the HotSpot a modified AP, with an FPGA based enhanced MAC that support EDCA features. It is still not clear if

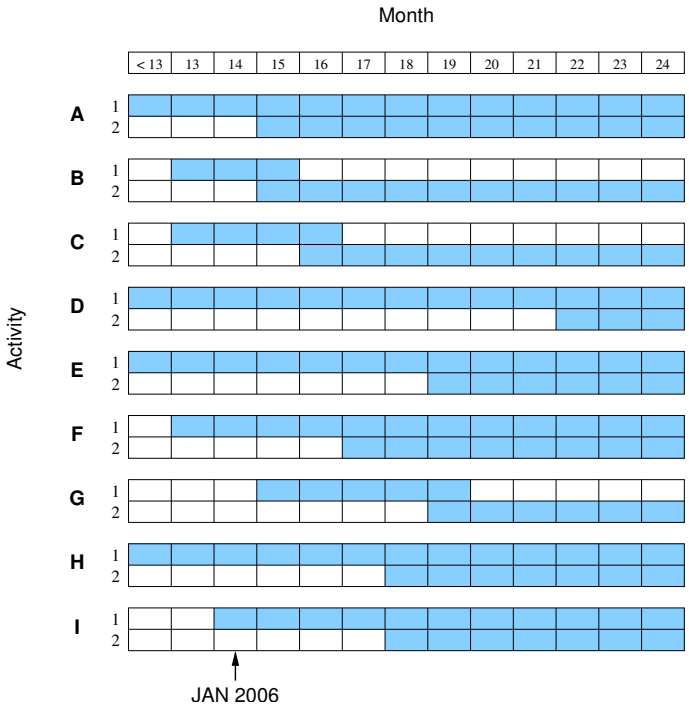

Fig. 12. Timeline of the test-bed and demonstration activities

Uni-Fy gate will be able to integrate and manage this modified AP.

E — Advanced scheduling that do not require the enhancement of hardware is deployed in Pisa for the support of streaming and real-time services.

F — APs enhanced for the management of mesh networks developed in Roma II are integrated within the *Uni-Wireless* test-bed.

G — A joint work of the teams in Roma II and Parma leads to a SIP-based signaling system for the secure cross-authentication of customers between WLAN and 3G systems.

H — Publish/Subscribe services are made available in Perugia.

I — Advanced Localization service support is integrated in the Trento HotSpot.

ACKNOWLEDGMENTS

As authors of this paper we wish to thank all the people that are working in the TWELVE project, making this test-bed a reality. Listing all names is not possible, but we are urged to mention Prof. Giuseppe Bianchi, coordinator of the TWELVE project and one of the principal promoter of this activity. He should be the first of the authors, but he pretended to be removed on the basis that he's "just controlling our work," and not implementing anything himself, as if planning and management are no activity at all — Thanks Giuseppe!!

REFERENCES

[1] R. Battiti, R. Lo Cigno, M. Sabel, F. Orava, B. Pehrson, "Wireless LANs: From WarChalking to Open Access Networks," *Mobile Networks and Applications*, Vol. 10, 2005, pp. 275–287, Springer Science

[2] R. Battiti, M. Brunato, R. Lo Cigno, D. Severina, A. Villani, "WilmaGate: an Overview", available at `http://netmob.unitn.it/files/WilmaGate_Overview.pdf`

[3] M. Brunato, D. Severina, "WilmaGate: a New Open access Gateway for Hotspot Management", *In Proc. WMASH2005*, Cologne, DE Sept. 2, 2005, pp 56-64.

[4] *Wireless MAC and PHY specifications: MAC Enhancements for Quality of Service (QoS)*, IEEE 802.11e/D8.0, Draft Supplement to Part 11, Feb. 2004.

[5] R.G. Garroppo, S. Giordano, S. Lucetti, L. Tavanti, "Providing Air-time Usage Fairness in IEEE 802.11 Networks with the Deficit Transmission Time (DTT) Scheduler," Accepted for publication on *Wireless Networks (WINET)*, Springer.

[6] R.G. Garroppo, S. Giordano, S. Lucetti, L. Tavanti, "Performance evaluation of the Deficit Transmission Time scheduler for Voice over WiFi," submitted for publication.

[7] L. Scalia, I. Tinnirello, "Architectural solutions for the implementation of an IEEE 802.11e EDCA MAC," TWELVE Internal Report

[8] Wi-Mesh Alliance, `http://www.wi-mesh.org`

[9] R. Bruno, M. Conti, E. Gregori, "Mesh networks: commodity multihop ad hoc networks", *Communications Magazine*, IEEE, March 2005.

[10] R. van Drunen, J. Koolhaas, H. Schuurmans, M. Vijn, "Building a wireless community network in the Netherlands," *In Proc. USENIX/Freenix Conference*, June 2003.

[11] J. Bicket, D. Aguayo, S. Biswas, R. Morris, "Architecture and evaluation of an unplanned 802.11b mesh network," *In Proc. ACM MobiCom 2005*, Cologne, Germany.

[12] Multiband Atheros Driver for Wifi (MADWiFi), `http://madwifi.org`

[13] D. Aguayo, J. Bicket, S. Biswas, G. Judd, R. Morris, "Link-level measurements form an 802.11b Mesh Network," *In Proc. ACM SIGCOMM 2004*, Portland, Oregon USA

[14] D. Giustiniano, "Link Level Measurements of a Wireless University Community Network," available as TWELVE TR_2005_RM2_R02, at `http://twelve.unitn.it/`.
Link Level Measurements of a Wireless University Comunity Network *Link to measurements campaign, Roma II.*

[15] Y.W. Lee, S.C. Miller, "Network selection and discovery of service information in public WLAN hotspots," *in Proc. of the 2nd ACM international workshop on Wireless mobile applications and services on WLAN hotspots*, Philadelphia, PA, USA, 2004.

[16] IEEE P802.11, TASK GROUP U, `http://grouper.ieee.org/groups/802/11/Reports/tgu_update.htm`.

[17] E. Guttman et al., "Service Location Protocol, version 2," *IETF RFC 2608*, 1999.

[18] Virtual AP Technology Multiplies WLAN Services, Whitepaper, Colubris Networks, March, 2004, available at `http://www.colubris.com/downloads/whitepapers/wp_vap.pdf`

[19] The Colubris Web Site, MultiService Controllers, `http://www.colubris.com/downloads/datasheets/DS_MSC_3000.pdf`

[20] The OpenSLP Project, `http://www.openslp.org`

[21] The Squid Web Proxy Cache, `http://www.squid-cache.org`

[22] S. Salsano, L. Veltri, "WLAN/3G secure authentication based on SIP", available as TWELVE TR T2.1_2005_PR_R02, at `http://twelve.unitn.it/`. Presented at Courmayeur meeting, January 11-13, 2006.

[23] MiSip, GPL Java implementation of SIP, `http://www.mjsip.org`

[24] J. Hightower, G. Borriello, "Location Systems for Ubiquitous Computing," *IEEE Computer*, Vol. 34, No. 8, August 2001.

[25] M. Brunato, S. Sgorlon, J. Fox, P. Toldo, "Context-enhanced service provisioning", TWELVE internal report.

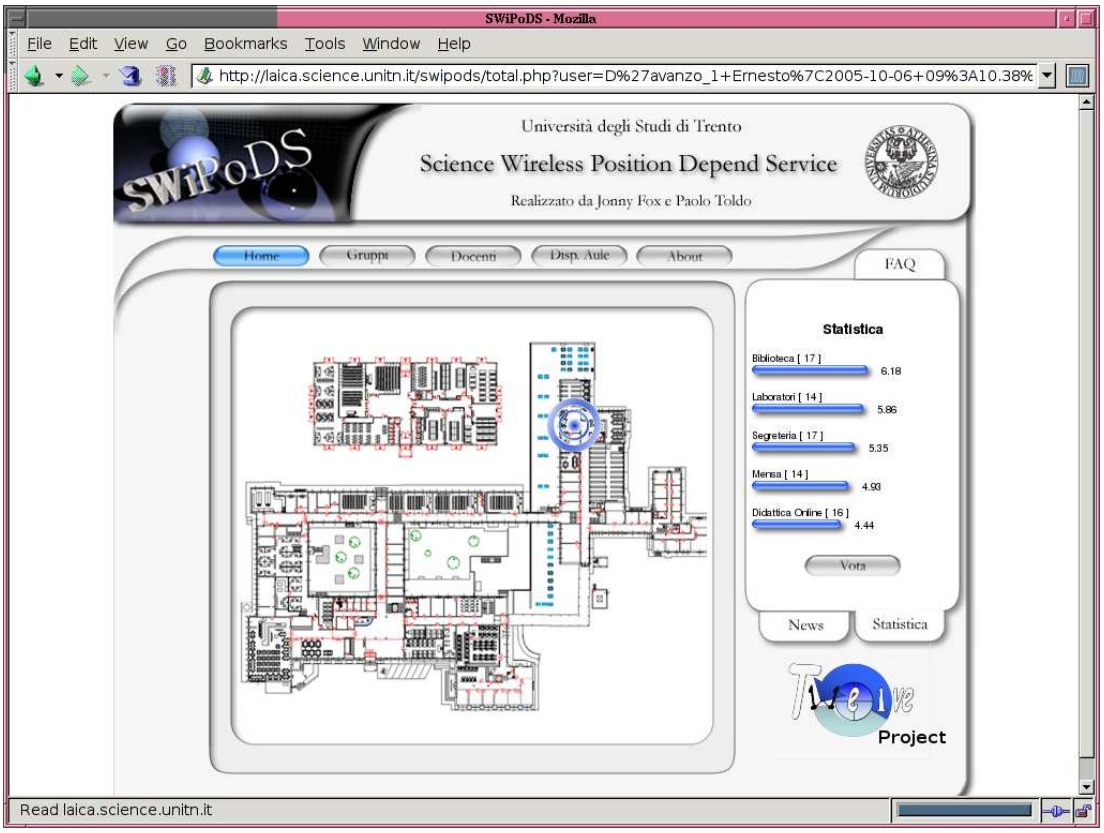

Fig. 11. Example of location-aware service developed at the University of Trento. A location map and a context-dependent poll are shown as sample services.