Service publishing to support an efficient network selection in an 802.11 multi-access environment

D. Di Sorte, M. Femminella, G. Reali
D.I.E.I. - University of Perugia, Italy
{dario.disorte,mauro.femminella,gianluca.reali}@diei.unipg.it

Abstract— In a multi-access and multi-service 802.11 environment, the problem of providing users with useful service-related information to support a correct, rapid network selection is expected to become a very important issue. In this paper, we propose a feasible, short-term 802.11-tailored solution for service publishing. To achieve a working solution compliant with existing equipment, our suggestion is to publish service information encoded within the SSID Information Element within beacon frames. This makes it possible for an operator to implement service publishing in 802.11 networks while waiting for a standardized mechanism.

We present a quantitative comparison of service discovery times between the legacy scenario, where the user is forced to associate and authenticate with a network point of access to check its service offer, and the enhanced scenario where the set of service-related information is broadcasted within beacons. These discovery times are obtained by processing the results of a measurement campaign performed in a multi-access/service 802.11 environment. This analysis confirms the effectiveness of the beacon-based approach.

Keywords— 802.11, service publishing, network selection, performance measurement

I. INTRODUCTION

IEEE 802.11 [1] is emerging as a promising platform to provide users with networked services in public spaces. With the continuous increase of the offer in terms of number of 802.11 wireless accesses and services, the problem of network selection is expected to become a very important issue.

Network selection is defined as the continuous process of selecting the most appropriate network for any user operation at any given time [8]. This makes sense in both homogeneous and heterogeneous access networks. In this paper, we limit our analysis to homogeneous 802.11 access networks.

In a multi-access/service wireless environment, users may want to select the Wireless ISP (WISP) and/or the Access Point (AP) to attach to according to a number of factors beyond the signal strength (e.g., roaming agreements, security, QoS, supported services, price). Associating/authenticating and then discovering these factors (the "try-and-see" approach) may be inefficient (i.e., time consuming) and bothering for users. Thus, a quick, effective AP selection is a very challenging issue, because the 802.11 standard currently does not provide an efficient support in this direction. This is the reason why there is a strong need to develop a mechanism which allows the Mobile Terminal (MT), and therefore 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. On the other hand, operators may like to advertise their own services not only to attract customers in a multi-operator scenario, but also to push users and influence their behaviour in a single-operator scenario.

The multi-access/service scenario can be expected in the near future, and this is the reason why some work in this field has begun recently [2][8][9]. It is worth noting that the initial standardization process in this direction in the framework of the IEEE 802.11 within the TGu [9][10][11][12] is far from a conclusion.

As is well known, APs periodically broadcast (every 100 ms) management frames (beacons), which include a number of pieces of information within fixed-length mandatory frame body components (Fixed Fields, FFs), together with variable-length mandatory and optional frame body components (Information Elements, IEs). This mechanism provides a means for MTs to discover not only APs but also certain service capabilities. An IE consists of three fields: the ID of the IE, the length of the body, the body. The SSID (Service Set IDentity) is an IE (with ID=0) containing the name of the network, the maximum length of which is 32 octets (i.e., 32 ASCII characters). To acquire IEs which are not carried in beacons, a Probe Request management frame can be used. This carries the Request IE (with ID=10), in which the IDs of the requested IEs are listed. The AP supplies the requested IEs within the Probe Response frame.

To achieve a working solution compliant with existing equipment, our suggestion is to publish service information encoded within the SSID, where the network name is correctly separated from the other set of information. The choice, which regards not only the set of information, but also the type of coding, is up to the operator.

The reason for this proposal is to present a simple, feasible short-term solution compatible backwards with the current standard and not in contrast with the work of 802.11 TGu. We believe that the path towards standardization in this field will be very long and difficult, since the set of information of interest for network selection is large and may well increase in

the future. In principle, each operator would also like to advertise the characteristics of its service offer according to proprietary, flexible, and dynamic criteria.

In order to quantify the benefits that users would enjoy from the immediate (before the association/authentication) knowledge of these additional information, we set up a demonstrator in our laboratory. This test bed is composed of a number of 802.11 accesses providing differentiated services in terms of authentication, quality of service, applications services, ciphering and access permissions. Service-related information is published within the SSID. With this first-hand solution, we analyze the benefits perceived by users in terms of service discovery time (i.e., the time needed for a user to find a desired service), when the service information is broadcasted within beacons. In addition, we compare this solution with the legacy scenario, where the user is forced to a "try-and-see" approach.

To enable a user-friendly network discovery and selection, we developed a Java-based graphic control tool (Twelve Wireless Selector, TWS) running on the MT, the main tasks of which are (i) to perform wireless network scanning and to present the user with the list of surrounding APs together with their service peculiarities, (ii) to drive handovers. Users also have the possibility to set their preferences, so that the TWS directly present the APs that match these preferences. The logs of this tool have enabled us to collect measurement data. To the best of our knowledge, this is the first paper which presents a quantitative analysis of advanced mechanisms for service discovery in 802.11 access networks.

The paper is structured as follows. The following Section summarizes the state of the art for service publishing and network selection topics. In Section III, we illustrate our proposal with a number of implementation considerations. Section IV describes the configuration of our demo and presents the TWS tool in detail. Section V presents a quantitative analysis of service discovery times. Finally, we report some concluding remarks and insights for future work.

II. Related Works

The topic is to provide users with service-related information prior to association/authentication to enable an efficient wireless access. In principle, the selection can be based on various criteria and a large amount of information. Clearly, this need becomes increasingly urgent if there is a large number of network accesses available to users in a given area, each of them with a different service offer. This scenario is to be expected in the near future.

From this point of view, interesting inputs to the access selection could be: (i) the price charged for access, (ii) roaming agreements (to discover whether a set of credentials allows access to the network), (iii) the type of enrolment (e.g., credit card), authentication (e.g., 802.1X or UAM, typical of an 802.11 environment), and ciphering (e.g., WEP or 802.11i, typical of an 802.11 environment), (iv) the application service offer, (v) IP address management (i.e., DHCP, NAT, MIP) and so on.

IEEE 802.21 [8] attempts to specify media-access independent mechanisms which optimize handovers between heterogeneous 802 systems and between 802 systems and cellular networks. The 802.21 standard aims to specify a set of handover-enabling functions within the mobility-management protocol stacks of network elements. These functions are performed by a new entity, the Media Independent Handover Function (MIHF), between L2 and L3. Its goal is to help the higher layer mobility protocol to acquire a global view of the heterogeneous networks and to perform effective network selection for both horizontal and vertical handovers. In particular, MIHF entity has to be able to collect information (referred to, in the following, as 802.21 Information Service Elements, 802.21 ISEs) relevant to the heterogeneous accesses existing within a geographical area.

802.21 aims to standardize the format of ISEs and classifies them into three categories:

1. General Network Information (GNI): it gives a general overview of the network (e.g., network ID, location, network operator).
2. Link Layer Information (LLI): it includes the information related to link layer (e.g., channel, frequency, PHY types, data rates, security, QoS).
3. Higher Layer Information (HLI): it provides information relevant to higher layer services/applications that are supported by the access (e.g., IP configuration, Virtual Private Network, types of applications), pricing, service discovery protocols), roaming partners and so forth.

This set of information may be retrieved by the MT by means of MIH message exchange from an information server in the network. Note that some information is made available directly by L2. In the latter case, MIHF can obtain them from a properly defined local interface between MIHF and L2.

So far, 802.11 TGu has decided to address the distribution of the following information [10][11]: authentication and enrolment methods, roaming agreements, and application service offer. Price-related information is set as an optional requirement.

As regards roaming agreements, in [2] the Authors discuss in detail a number of approaches and propose to set a new IE, which includes a Roaming Information Code (RIC) with roaming information; in principle the RIC may be also included in the SSID IE. In [12], a solution based on the Network Access Identifier (NAI) [13] and probe request/response mechanism is also presented. With respect to the RIC-based solution, the main advantage is that any number of Subscription Service Provider Networks (SSPN) may be supported by a single AP.

In [2], the Authors also propose a new IE to carry price-related information. In [14], an IE is broadcasted within beacons with the main service characteristics of access (class

of Internet access, two bits, availability of automatic enrolment, one bit, and free/fee-based access, one bit). The MT can then discover more about the credentials needed to access the network and the cost of access by means of a probe request query.

In addition, the Authors also address issues specifically related to 802.21 in [12]. They present a probe request/response based mechanism to provide the MT with 802.21 ISEs. In more detail, a new IE (IS Request IE), to be carried within the Probe Request management frame, together with another new IE (IS Response IE) to be carried within Probe Response management frame, are proposed. The former includes all the IDs of the 802.21 ISEs in which the MT is interested, whereas the latter includes the set of requested 802.21 ISEs. The set of information is retrieved by the AP from an 802.21 information server.

As regards issues specifically related to 802.11 network management, the interested reader should refer to the work of the IETF CAPWAP WG [16][17]. Its main goal is to define a standardized, interoperable interface between APs and a centralized controller addressing additional WLAN services (centralized architecture). In other words, the final objective of the IETF CAPWAP WG is to develop a CAPWAP protocol, an open standard that all vendors can implement and show interoperability. This could also be needed to dynamically configure the set of service-related information to be published by the APs from a centralized controller. In a centralized architecture, procedures to control/configure APs from a remote entity clearly require confidentiality, integrity, and authenticity to be addressed.

Finally, it is also worth mentioning the work carried out by the IETF Seamoby WG, which proposes a Candidate Access Router Discovery (CARD) protocol [15]. This is a high-level protocol, which enables a network-assisted mechanism for the quick discovery of the surrounding wireless coverage, especially the discovery of IP addresses and service capabilities of candidate access routers to hand over to. The rapid knowledge of IP addresses enables MTs to speed up the (horizontal or vertical) handover process and thus perform a seamless handover, whereas information about service capabilities is important for the selection of the most appropriate wireless access (target access router). In principle, decisions can be either MT-driven or network-driven. This kind of solution is mainly focused on making the attachment procedures more efficient when moving from one location to another, and does not provide any support when an MT is not currently attached to the network and wants to select access correctly. In fact, since link-layer decisions depend on information retrieved from a high-level protocol, the MT needs to be associated and authenticated with a network point of access in order to enable the mechanism.

III. BEACON-BASED SERVICE PUBLISHING

Since the typical procedure of a TG to add new functions to the standard is to define new IEs, we propose to set a new IE containing useful service-related information for network selection. In more detail, we aim to standardize the IE (and thus reserve an IE ID), but not its content. This is similar to what happens with the SSID IE, which is filled with the network name by the network administrator according to its policies. In other words, we are saying that such a service-related IE, named Service IE, may be filled by the network administrator with the information relevant to the service offer of the operator that is considered important to be advertised.

The reason for this proposal, which is backward compatible with the current standard and does not contrast with the work of 802.11 TGu is that, in our opinion, the standardization activity in this field will be very long and difficult, since the set of information of interest for network selection is large and may well increase in the future. In principle, each operator would also like to advertise characteristics of its service offer according to proprietary, flexible, and dynamic criteria.

It is worth noting that the set of information broadcasted within beacons should be limited so as not to cram them and consume too much wireless bandwidth. This would help towards a preliminary screening of the service peculiarities of accesses.

Consequently, once the MT has connected to the target AP, service discovery protocols (see [19][23] for an overview), such as, for instance, the Service Location Protocol (SLP) [3] may have to be used in order to acquire more refined service attributes (e.g., configuration information for applications) to enjoy a given service.

Although the above mentioned solution based on the Service IE is perfectly backward compatible with the standard, the relevant implementation would require firmware/driver updates not only to wireless cards, which have to be able to acquire information from the Service IE, but also to APs, which have to be able to transmit such a new IE.

Thus, in order to have a working solution compliant with existing equipment, our suggestion is to publish service information encoded within the SSID, where the network name (necessarily a short one) can be separated from other information by the symbol @, using character stuffing for data transparency. Note that any other symbol which is not normally used can work for our goal.

The choice not only of the set of information, but also of the type of coding is left to the operator.

Clearly we do not claim that this is the best solution, but we simply present it as a possible way for an operator to implement service publishing in 802.11 networks while waiting for a standardized mechanism.

The main advantages of this solution are:

- it is simple and feasible in the short term;
- it is backward compatible with the standard;
- it does not require additional mechanisms and IEs;
- it is compliant with existing APs and network cards;

– it does not contrast with other solutions;
– it is flexible on the operator's side, since the network administrator is not constrained to publish standardized information, but can decide which services have to be advertised;
– if service-related information coding is efficient and the network name is not long, the bandwidth consumption is kept low.

On the other side, the drawbacks of this implementation are:

– the use of a structured SSID to carry other information beyond the network name is not provided by the standard;
– the constraint on the length of the network name is lowered from the original 32 ASCII characters (typically the length of network name is substantially lower than 32 ASCII characters). Clearly, in principle, the bigger the set of information to be published, the shorter the network name. Again, this choice is left to the operator's discretion;
– the dictionary to decode service-related information is operator-oriented. In other words, the specific network operator must provide subscribers with a software tool able to decode service-related information from the SSID field[1]. In principle, only the dictionary could be proprietary, whereas the software tool may be either a standard one or a proprietary one. In subsection IV.A, we will describe TWS tool, which is able to get service-related information, among other things, from the SSID field.

Finally, we report some considerations on security aspects. Service-related information which characterizes a given access is advertised before association/authentication procedures by using management frames, and therefore there is no level of security. A network administrator may decide which set of information has to be published, according to proprietary policies. If certain information is considered sensitive by the administrator, then he/she will not publish it. Another problem is that illegal APs can masquerade as real APs and present themselves as a network access with a set of peculiarities. This is a general problem of 802.11, which could only be solved by protecting the beacon, and is definitely beyond the scope of this paper.

IV. Demo Scenario

In this Section, we first describe the multi-access 802.11 network configured in our TLC laboratory, we then describe the TWS tool which enables network selection to be carried out according to the service characteristics of the different APs. The quantitative analysis as regards service discovery times presented in the following Section V, will be based on measurements performed in the depicted network scenario and on the capabilities of the TWS.

A. Demo architecture

Our objective is to set up a network configuration able to publish and provide a multi-service 802.11b access environment. Please consider that our final goal is to evaluate by measurements the performance of different service publishing solutions, and we have not focused on the optimization of the network configuration.

The components specific to the demo architecture (see Fig. 1) on the network side are:

- standard AP;
- Virtual Access Point (VAP);
- SLP Directory Agent;
- DHCP server;
- video server;
- Radius server;
- Twelve Data Sharing (TDS) server.

The components in the terminal side are:

- TWS tool;
- TDS client;
- Radius client;
- SLP User Agent.

A VAP is a physical AP in which a number of logical entities, named VAP profiles, exist [4]. Each VAP profile appears to be an independent, physical AP and emulates the operation of a physical 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. In principle, a number of WISPs can also share the same physical device. Thus, a VAP device is quite suitable for the deployment of a multi-service WLAN. In addition, VAP technology enables VLANs [20] to be extended to the wireless segment of the network. In our lab we make use of a Colubris MSC3200 with VAP technology which supports up to 16 concurrent SSID/BSSIDs [5].

In addition, we deploy an SLP architecture for service discovery and configuration, once the MT has attached to a network access. The main components of an SLP architecture are [3]:

- User Agents (UAs) which discover services;
- Service Agents (SAs) which advertise the services they represent together with their relevant attributes;
- Directory Agents (DAs) accumulate service information and respond to service requests from UAs. Clearly, service information may also be statically stored within DAs. Services may also be grouped in a number of scopes according to specific policies.

We offer a number of different services: Internet access; video service; TDS service; printing service.

[1] In the market, this solution is applied by the WISP Boingo, which provides customers with a software tool to be installed in their laptops; this tool is in charge of recognizing from the SSID of an AP whether the administrator of that AP is a Boingo partner.

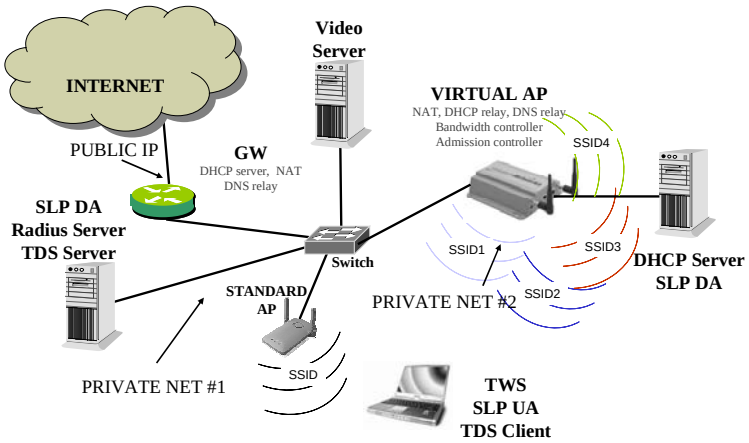

Fig. 1 – Demo architecture.

The TDS (Twelve Data Sharing) service is an advanced data sharing mechanism. We started from the consideration that, whenever confidentiality of personal data is not an issue, a given set of contents is particularly "hot", there is a broadcast and bandwidth-limited channel, then publishing the transmission of data requested by a user to enable content acquisition by other users enables: lower time to access contents, capability to consult hot contents off-line, improved channel efficiency, lower server load, emulation of reliable multicast procedures.

The service architecture deploying the data-sharing mechanism 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 [7] able to generate unicast UDP advertisements towards the AP supporting the service upon http data requests from a client. This enables the transmission to be published and MTs can be provided with the information needed to enable the data acquisition process. This architecture allows all TDS clients under the same wireless network access to share the same contents. Possible application scenarios of the TDS service in an 802.11 access network are: the sharing of files exchanged in a peer to peer mode, the sharing of contents served by a local/remote server (e.g., in library and lecture hall environments), whiteboard applications, group communications. The parameters to configure the TDS service are acquired by the MT by means of an SLP query.

As regards the video service, a dedicated PC is able to provide video-clips with two different levels of quality in terms of resolution and frame rate. The video-clips are made available via the web interface and the URLs are published by the SLP architecture.

In addition, we used two black/white laser printers (one in our lab and the other in the DSP lab of our department) and one inkjet colour printer (in our lab), all of them connected to a public IP address in the department network.

In the demo, there are five 802.11b TWELVE-compliant accesses (named TLC_i, i=0,…,4), one provided by the standard AP (TLC_0) and the others provided by the VAP. The characteristics of the different accesses are summarized in the following table.

Service differentiation is provided on a per-access basis in terms of:

- Network service offer, in terms of
  - QoS. In more detail, we exploit the capabilities of the CN3200 which enables four levels of priority to be defined for bandwidth management and admission control to be performed. By combining these two features, we are able to guarantee a minimum amount of bandwidth to deliver high quality videos.
  - Security. In more detail, we make use of the capabilities of the CN3200, which can work as an 802.1X authenticator [22] and can also locally control user accesses according to either the UAM (Universal Access Method, [21]) or the MAC based authentication. The device also supports ciphering.
- Service publishing:
  - beacon-based. Service-related information is included within the SSID IE, as explained at the end of this subsection.
  - SLP-based. We group services by means of the SLP scope mechanism 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 also have the differentiation of the service publishing at the SLP level. In other words, the query to the SLP server from an MT indicates the SSID of the AP the MT is currently attached to. Thus, the SLP server replies with the information of the services belonging to the scope with which the SSID is associated.
- Application service offer. Note that two VAP profiles (those providing the video service) are not allowed to access the public network, and therefore they do not offer Internet access and printing services. In addition, the video server is only accessible from the two mentioned VAP profiles. To this end, we deploy VLAN traffic isolation policies.

Table 1: APs service offer

Service characteristics	Standard AP	VAP Profile 1	VAP Profile 2	VAP Profile 3	VAP Profile 4
Network name	TLC_0	TLC_1	TLC_2	TLC_3	TLC_4
Authentication	MAC	Radius	Radius	Radius	UAM
User class	any	premium	premium	any	any
Ciphering	none	WEP	WEP	WEP	none
BW control	none	yes	yes	yes	yes
Admission control	no	no	yes	no	no
SLP support	yes	yes	yes	yes	no
SSID-based publishing	yes	yes	yes	yes	yes
IP configuration	DHCP	DHCP	DHCP	DHCP	DHCP
Price	free	free	flat	free	free
Internet access	restricted (private IP)	restricted (private IP)	no access	no access	restricted (private IP)
Video streaming	no	no	yes (high resolution)	yes (low resolution)	no
Printing	yes	yes	no	no	no
TDS	yes	no	no	no	no

As regards the coding of service-related information within the SSID field, as mentioned above, we have used the character @ as the separator between the network name and the encoded service-related information. The format of the SSID is *TLC_i@<Code,Value><Code,Value>...*, where the pair <Code,Value> identifies the service feature (Code), and the relevant configuration (Value) (see Table 2).

Table 2: service-related information within the SSID IE.

Category	Code	Value	
Class of users	A	0: basic users	1: premium users
Category of Internet access	B	0: unrestricted (public IP) 2: web	1: restricted (private IP); 3: no access
IP configuration	C	0: static 2: MIP	1: DHCP
Service availability	D	0: TDS off	1: TDS on
	E	0: printing off	1: printing on
	F	0: streaming off	1: streaming on
Service discovery protocols	G	0: none 2: Jini on	1: SLP on 3: UPnP on
Price	H	0: free 2: time-based	1: flat 3: volume-based
Enrolment	I	0: no credentials 2: credit card	1: username/password 3: certificates
QoS	L	0: QoS enabled	1: QoS disabled
Authentication	M	0: open system 2: 802.1X 4: UAM	1: shared key 3: MAC filtering
Ciphering	N	0: ciphering off 2: WEP (13 bytes)	1: WEP (5 bytes) 3: 802.11i

We assume that the specific structure of the SSID IE gives the user information about the operator managing the wireless access. In other words, TWELVE-compliant APs are considered to belong to the same administrator and are accessible with specific credentials. We are also conscious that the proposed information coding is not efficient, however, at this stage our objectives are to show the feasibility of the SSID-based solution and to quantitatively evaluate the relevant benefits in terms of service discovery times (see Section V).

In the following, the APs with this kind of structured SSID are referred to as Twelve-compliant APs.

B. The Twelve Wireless Selector

The TWS is a Java-based graphic control tool running on the MT. The hardware/software requirements of the TWS tool are: (i) wlan-ng driver for prism2-based 802.11 card[2]; (ii) openSLP v1.2.1 [6]; (iii) Linux Mandrake 10.0 OS; (iv) Java Virtual Machine v4.2.8.

Such a tool is in charge of

- performing wireless network scanning;
- presenting to the user the list of surrounding APs, both legacy and Twelve-compliant ones (see Fig. 2);
- showing the service peculiarities of the Twelve compliant APs (see Fig. 3) retrieved from the SSID IE;
- showing the more refined service information acquired via SLP (see Fig. 4);
- performing user-driven handovers;
- configuring network parameters;
- obtaining network configuration parameters from the DHCP protocol;
- showing the current network configuration (from MAC to DNS).

An additional important characteristic of the TWS is the capability to perform wireless network scanning and to present the user with only the list of accesses which 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 (see Fig. 5).

The TWS control panel is integrated with the SLP UA to acquire more refined application service information relevant to the current AP from a remote SLP DA. In more detail, the TWS issues an SLP query to acquire more refined information

[2] The TWS is also supported by the hostap driver, although this driver prevents users from accessing the TDS service.

relevant to services with the scope value set to the SSID of the AP the MT is currently attached to. Thus, once the user has selected the AP on the basis of the rough service information obtained by beacons, he/she has to attach to it. Then, a given service may be accurately configured by means of the TWS. For instance, in our demo, the TDS service is automatically configured with the parameters (IP address and proxy port) of the TDS server obtained via SLP. In addition, the TDS is also configured with the IP address of the MT dynamically acquired via DHCP. In other words, when a user decides to attach to the AP supporting the TDS service, then the TWS automatically retrieves the entire set of information needed to configure the service, and performs service configuration.

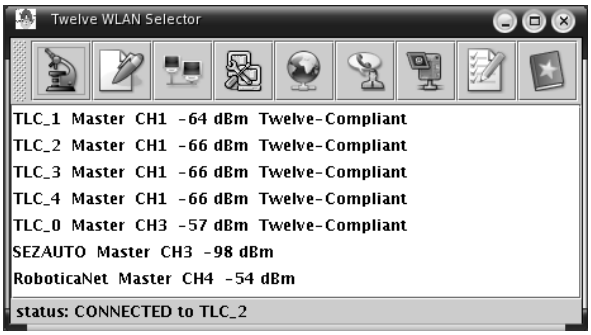

Fig. 2 – TWS: list of surrounding APs.

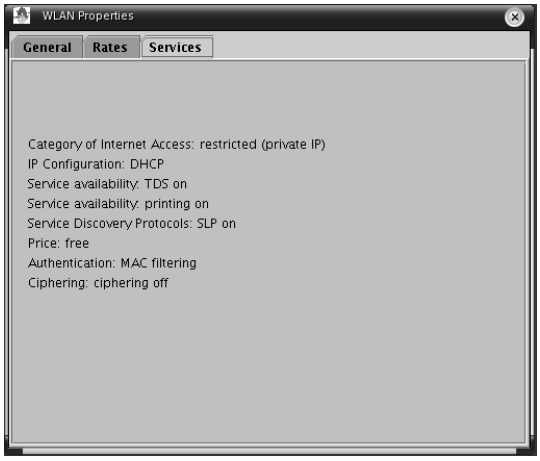

Fig. 3 – TWS: service characteristics of the AP.

V. Performance Analysis

In this Section, our goal is to present a quantitative comparison in terms of service discovery times provided by a number of solutions, and to show the effectiveness of beacon-based service publishing. We first summarize the solutions exploiting the features of a beacon-based mechanism, an SLP-based mechanism and a DHCP-based mechanism. We then propose an approach to quantitatively evaluate service discovery times. Finally, we present numerical results exploiting measurements performed in the network configuration described in the previous Section.

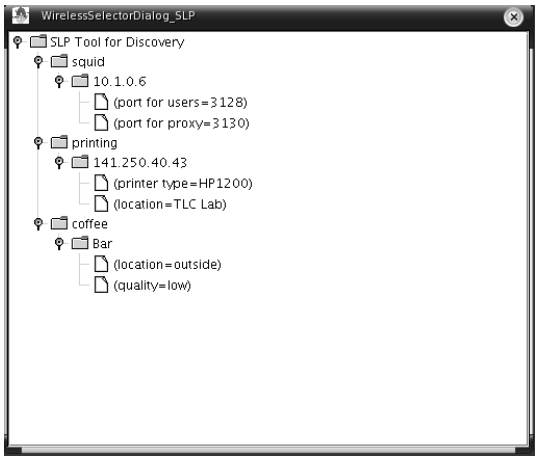

Fig. 4 – TWS: panel to set user preferences.

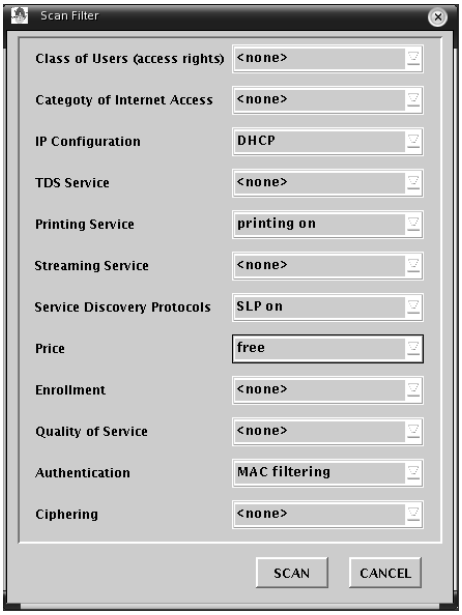

Fig. 5 – TWS: service information from SLP.

A. Service discovery solutions

Let us assume that a user is seeking an access providing a given feature, for instance an application service. If multiple accesses are available in the legacy scenario, there is no means of comparing them without manually visiting (i.e., associating and authenticating with) each one.

We assume that the user has some credentials (e.g., username and password) to access a subset of the surrounding APs. In more detail, we consider a premium user, able to enter all TWELVE accesses. We also clearly bear in mind that the user does not know the network a priori and that the MT is not configured with a static IP.

The process evolves as illustrated in Fig. 6. The user begins to attach to an AP. If the process is successful (i.e., the user is authenticated and receives the IP address, if needed), the user can immediately verify whether the service (e.g., Internet

access) is available. If more refined service information is needed to enjoy the service (e.g., the TDS service), the user may issue a query using a service discovery protocol, for instance SLP. Finally, if the user is unable to receive the service, he disconnects from the current access and attempts to attach to another access. The process ends when either the user finds the desired service or the accesses to monitor are exhausted and the service is unavailable.

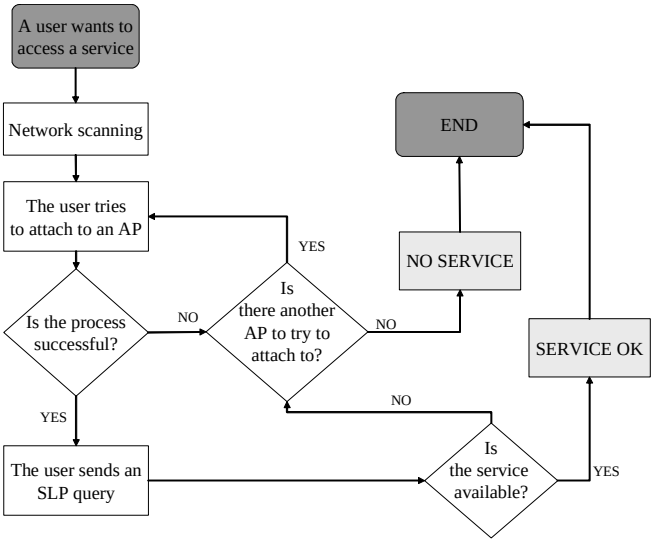

Fig. 6 – Service discovery: the legacy scenario.

Exploitation of the DHCP options [18], which enable service-related information to be included within DHCP messages as well, could speed up the entire process. In this case, the user is again forced to associate/authenticate with an AP, but the queries to acquire service attributes are not needed.

It is worth noting that SLP and DHCP-based mechanisms are not useful in the selection of the correct AP to receive the desired service. One means of identifying the APs providing the service is beacon-based service publishing. In this case, the user can enjoy the service immediately after completing the process of connection to the network for certain services (e.g., Internet access). If some configuration information is needed for other types of services, this set of information can be acquired either via a DHCP mechanism (needed anyway to obtain network configuration parameters if the MT is not already configured) or via an SLP query. We remark that, if the information relevant to a service (e.g., printing service) broadcasted within beacons is rough (the printing service is or is not available), then the user may need to acquire additional information to understand whether the desired service is provided by the access. For instance, if the user is looking for a colour printing service, he/she may access an AP providing a b/w printing service. The same occurs if the user is looking for a printer located in a given area. This means that, for some kind of services characterized by specific peculiarities, the beacon-based solution enables the identification of a subset of APs, which can provide the service with different, specific attributes.

To sum up, it is possible to identify four service discovery solutions:

1. the try-and-see approach with SLP support;
2. the try-and-see approach with DHCP support;
3. the beacon-based approach with SLP support;
4. the beacon-based approach with DHCP support.

B. Evaluation of service discovery times

In this subsection, we will evaluate the average discovery time to find a service with specific characteristics.

To this end, let us define the following parameters relevant to the underlying network environment:

- N: the number of surrounding network accesses discovered by means of a standard beacon scanning procedure;
- M: the number of network accesses for which the considered user has access privileges ($M \leq N$);
- X: the number of network accesses, from among the M accesses, which use the SSID mechanism to publish generic information concerning the service the user is searching for ($X \leq M$);
- Y: the number of network accesses by means of which the user can enjoy the service with the desired attributes ($Y \leq X$).

Now let us define the following time parameters needed to compute the average service discovery time for the different mechanisms:

- T_{SCAN}: the time needed to perform a standard beacon scanning procedure;
- T_{SCAN_FILTER}: the time needed to perform a beacon scanning procedure using the filtering mechanism described above (see Fig. 4);
- T_{CONN}: the time needed to associate (at layer 2) with an AP;
- T_{AUTH}: the time needed to perform user authentication;
- T_{DHCP}: the time needed to acquire an IP address via the DHCP mechanism (and eventually all the service-related information the operator wishes to provide via DHCP);
- T_{DHCP_FAIL}: the timeout of the DHCP client on the MT. This event occurs when the MT associates with an AP, but the MT does not succeed in obtaining a valid IP address, since the user is not allowed to access that AP. Note that this timeout also expires when a non-authorized MT tries to access an AP with MAC-based authentication, according to which only the MTs with authorized MAC are allowed to access the AP;
- T_{UAM_FAIL}: the time needed to acknowledge a failed web-authentication attempt. In the UAM-based authentication (also known as captive portal), the MT may access the AP, and its first HTTP request is forwarded towards an authenticator which is in charge of performing a web-based authentication procedure. The user is required

to provide his/her credentials via an SSL web-based login page. If this procedure succeeds, then packets from that MT are delivered towards the Internet, whereas if the procedure fails, an error message appears on the web browser;

- $T_{802.1X\_FAIL}$: the time needed to acknowledge a failed 802.1X authentication attempt. The user is requested to provide his/her credentials via a pop-up which appears in the laptop once the MT and the AP have negotiated the authentication mechanism. If the procedure fails, an error message appears on the user laptop;
- T_{SLP_SERV}: the time needed to perform an SLP query to discover supported services (with reference to Fig. 5, this step produces the results: "squid", "printing", and "Bar");
- T_{SLP_ATTR}: the time needed to perform two SLP queries to acquire the address of the server and the relevant configuration attributes (with reference to Fig. 5 and to a search for printing service, these steps produce the results: "141.250.40.43" as the printer IP address, and "printer type" and "location" as the server attributes).

Let us begin by focusing on the beacon-based procedure supported by the SLP protocol to retrieve additional service-related details, if needed. The time needed to discover an AP providing the desired service is equal to

$$T_{ave,FILTER} = T_{SCAN\_FILTER} + \sum_{i=1}^{X-Y+1} P(i) \cdot T(i) , \quad (1)$$

where $P(i)$ is the probability of finding the service at the i-th attempt and $T(i)$ is the time needed to acquire all service information at the i-th attempt. In more detail, it results that:

$$\begin{cases} P(1) = \dfrac{Y}{X} \\ P(i) = \left[\prod_{j=1}^{i-1} \dfrac{X-(j-1)-Y}{X-(j-1)} \right] \dfrac{Y}{X-(i-1)}, \quad i = 2,..,X-Y+1 \end{cases} , \quad (2)$$

$$T(i) = iT_{FULL} , \quad (3)$$

where T_{FULL} is the time needed to acquire all the service attributes, i.e.,

$$T_{FULL} = T_{CONN} + T_{DHCP} + T_{AUTH} + T_{SLP\_SERV} + T_{SLP\_ATTR} . \quad (4)$$

In fact, following a network scanning using the filter shown in Fig. 5, the TWS will provide a list of X network accesses, i.e., only those which publish the code relevant to the desired feature within the SSID. As an example, if the user is searching for the printing service, he will only obtain the list of networks carrying the code "E1" after the @ in the SSID (see Table 2).
The user now has to associate/authenticate and obtain an IP address. If the service desired can be used immediately (e.g., Internet access), no further steps are needed (i.e., T_{SLP_SERV} and T_{SLP_ATTR} are equal to zero). However, if additional information is needed to enjoy the service, he/she issues SLP queries to retrieve all the service attributes. Only at this stage is the user able to decide whether the service he/she has found fulfils its need (e.g., the user wants a colour printer, whereas he only found a black/white one). If he is unsatisfied with the specific service implementation, he has to try another network from among those supporting the desired service (thus, the maximum number of attempts is equal to X-Y+1).

A variant of this approach is the beacon-based procedure supported by the DHCP protocol to retrieve additional service-related details, if needed. In this case, the only difference with respect to the previous case is that the user does not have to issue SLP queries to get service information. For the computation of the average discovery time, we can use the above equations with T_{SLP_SERV} and T_{SLP_ATTR} set equal to zero for every kind of service.

The evaluation of service discovery times for the other two solutions (Try&See + SLP and Try&See + DHCP) is a little more complicated, since the user can stop the search process when attached to a given access at various stages. We also remark that in these cases the user is *unable* to use a smart network scanning which can (i) recognize the APs of its SSPN, (ii) take into account user preferences, (ii) restrict the set of surrounding APs to monitor. Therefore, each AP discovered through a standard beacon scanning procedure is a possible candidate.

Let us start with the Try&See + SLP solution. If the user tries to attach to one of the N–M networks to which he/she has no access privilege, the time associated with this attempt is equal to

$$T_{DENIED} = \begin{cases} T_{CONN} + T_{DHCP\_FAIL} & \text{for MAC authentication} \\ T_{CONN} + T_{DHCP} + T_{UAM\_FAIL} & \text{for UAM authentication} \\ T_{CONN} + T_{802.1X\_FAIL} & \text{for 802.1X authentication} \end{cases} . \quad (5)$$

We name this kind of failure event as a α-event. For instance, if at the first attempt the user selects a forbidden network, then the probability of this event is equal to $(N–M)/N$.

On the other hand, if the user attaches to one of the M–X networks which do not offer the service he/she is looking for, the time needed to acknowledge the failure is equal to

$$T_{NO\_SERV} = T_{CONN} + T_{DHCP} + T_{AUTH} + T_{SLP\_SERV} . \quad (6)$$

In fact, the user generally has to wait for the list of supported services provided by the SLP protocol to know that the desired one is not supported. Clearly, if we consider services such as Internet access, T_{SLP_SERV}=0, but we have to account for an additional time to acknowledge the failure (i.e., the time necessary to visualize an error message from the web browser, due to missed DNS response). We name this kind of failure event as a β-event. For instance, if at the first attempt the user selects a network not providing the service, then the probability of this event is equal to $(M–X)/N$.

Finally, if the user enters a network supporting the generic service he/she is searching for, but not the one with specific attributes, the time needed to acknowledge the failure is equal to T_{FULL}, the expression of which is given by (4). We name this kind of failure event as a γ-event. For instance, if at the first

attempt the user selects a network which does not provide the service with the specific attributes, then the probability of this event is equal to $(X–Y)/N$.

Summing up, the average discovery time to find a service is equal to:

$$T_{ave} = T_{SCAN} + \sum_{i=1}^{N-Y+1} T_w(i) \quad (7)$$

where $T_w(i)$ is the time needed to select the correct network access at the i-th attempt, weighted by the probability that this event occurs. It is easy to see that the maximum number of attempts to find an access satisfying the user's requirements is equal to $N–Y+1$. The values of $T_w(i)$ depend on T_{DENIED}, T_{NO_SERV}, T_{FULL}, and on the number of events α, β, and γ. More details can be found in the Appendix.

Finally, the average discovery time for the Try&See + DHCP procedure can still be computed starting from (7), setting T_{SLP_SERV} and T_{SLP_ATTR} equal to zero and considering that γ-events do not occur.

C. Numerical results

In the following table, we report the time values obtained from the measurement campaign performed in our lab by averaging five time samples. In more detail, Table 3 reports the times when the wireless accesses are unloaded and when they are charged with 4Mbps UDP traffic. We have measured the time for both UAM and 802.1X authentication (the time to perform a MAC-based authentication is assumed equal to zero). Please note that we do not consider the time needed by the user to write his/her username and password in the UAM and 802.1X authentication. This time is in the order of a few seconds.

Table 3: measurement results.

Time parameters	No traffic	4Mbps traffic
T_{SCAN}	803 ms	803 ms
T_{SCAN_FILTER}	865 ms	865 ms
T_{CONN}	1357 ms	1531 ms
T_{DHCP}	4785 ms	7340 ms
T_{DHCP_FAIL}	63465 ms	64310 ms
T_{SLP_SERV}	3435 ms	4365 ms
T_{SLP_ATTR}	5044 ms	7943 ms
T_{UAM}	6927 ms	10260 ms
$T_{802.1X}$	776 ms	1931 ms
T_{UAM_FAIL}	2390 ms	4159 ms
$T_{802.1X\_FAIL}$	4731 ms	8263 ms

Thus, starting from the theoretical analysis performed in the previous subsection and considering the time values measured, we are able to find the average service discovery times of the different solutions (the try-and-see approach with SLP support and with DHCP support, the beacon-based approach with SLP support and with DHCP support) for (i) the different type of services offered in the demo scenario, (ii) different network load conditions, (iii) different authentication mechanisms[3].

Each one of the following four figures refers to a specific application service provided in our TLC lab.

Fig. 7 refers to the TDS service, which requires configuration information from either a DHCP or an SLP protocol. In the demo scenario we consider, note that N=7, M=5, X=1, Y=1; only one access (TLC_0) publishes and provides the service. As expected, the beacon-based solution supported by DHCP is the best, whereas the solution with SLP support only is the worst. The advantage in terms of the discovery times of beacon-based with respect to try-and-see solutions is in the order of tens of seconds. An important consideration is that beacon-based solutions are more robust to traffic variation, since there are fewer message exchanges on average with respect to the try-and-see approaches. Note also that the highest discovery times evaluated for MAC authentication are those for the try-and-see approaches. This is due to the fact that a failed authentication attempt is acknowledged only after expiry of the DHCP client timeout, which is usually approximately one minute (in a Linux Mandrake 10.0 terminal). In any case, remember that our model does not take into account the times associated with all the interactions between the user and the MT, in particular the time needed to write the username and password.

Fig. 8 shows the discovery times of the colour printing service. In this case, rough information from beacons is not sufficient to understand whether an access is able to provide the service. Either SLP or DHCP support is needed. In the demo scenario we consider, note that N=7, M=5, X=2, Y=1; two accesses (TLC_0 and TLC_1) publish the printing service, but only one (TLC_1) of them provides the colour printing service. As expected, the discovery times increase compared with the TDS service, due to a possible attachment to the wrong access. All the other considerations made for the TDS service search also holds in this case. We can say that they are fairly generalised.

From the analysis of Fig. 7 (X=Y=1) and Fig. 8 (X=2 and Y=1), we can observe that the discovery times increase with (X-Y), i.e., the number of network accesses which publish the generic service and which do not offer the service with specific attributes. Note that the discovery times of the try-and-see + DHCP solution are also unaffected by the variation of (X-Y), since, once attached, the MT immediately obtains all the service information.

Fig. 9 refers to the discovery of a b/w printing service. In this case, X=2, and Y=2, i.e., both the accesses which publish the printing service via the SSID are also able to deliver the b/w printing. As expected, the service discovery times are lower compared with those of the colour printing service.

Note that, as regards the high/low quality video service (X=2, and Y=1), the discovery times are the same as those of the colour printing service.

[3] All accesses are assumed to perform the same authentication method.

Finally, Fig. 10 reports the discovery times relevant to the web browsing service (more generally, the Internet access). In the demo scenario we consider, note that N=7, M=5, X=3, Y=3. Note that, as also outlined in previous Sections, the fruition of this service is not subject to the acquisition of configuration information. For this reason the discovery times of solutions with DHCP support and SLP support (which is not needed) are the same. In addition, discovery times are lower compared to those of b/w printing, as expected, since there are three and not two accesses which publish and provide the service. We also remark that, if the MT has network connection properties (i.e., the IP address, gateway, DNS) already set, then DHCP support would be unnecessary and discovery time would be even lower.

Looking at Fig. 7 (X=Y=1), Fig. 9 (X=Y=2), and Fig. 10 (X=Y=3), it is clear that the advantage of beacon-based over try-and-see based solutions increases with the value of (N-X). This is due to the fact that the service-related information retrieved before association/authentication prevents users from attaching to network accesses which do not provide the desired service.

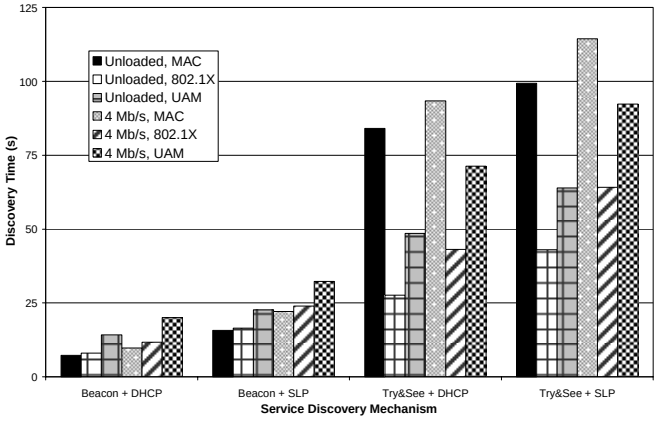

Fig. 7 – TDS: service discovery times.

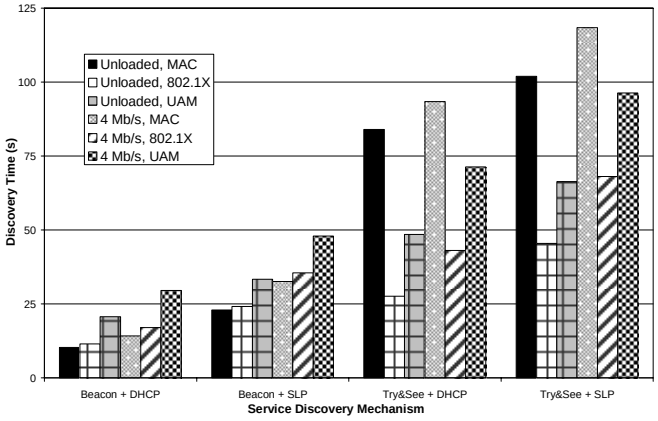

Fig. 8 – Colour printing: service discovery times.

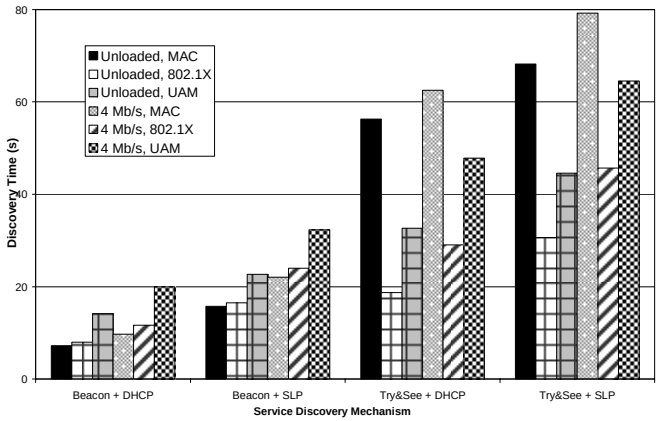

Fig. 9 – B/W printing: service discovery times.

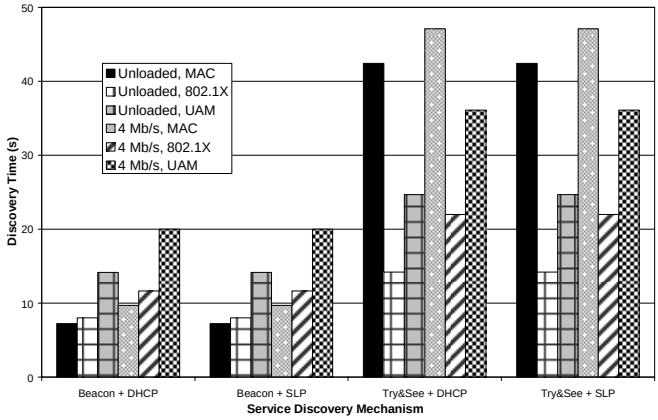

Fig. 10 – Web browsing: service discovery times.

VI. CONCLUSION

Providing nomadic users with some kind of service-related information is quite urgent in an expected future scenario where a large number of 802.11 APs/WISPs are available in a given area, each with a different service offer. This would help users carry out a rapid, correct network selection and would also allow operators to advertise their own services to attract new customers.

We have proposed and implemented an SSID-based solution, the main advantages of which are that it provides the network operator with a ready-to-use operational solution, backward compatible with the standard and compliant with existing equipment. We do not claim this to be the best and final solution. We simply present it as a possible means for an operator to implement service publishing in 802.11 networks while waiting for a standardized mechanism.

It is worth noting that this solution has also allowed us to present a quantitative evaluation of the benefits perceived by users in terms of service discovery times. To achieve this goal, we set up a multi-service/access 802.11 environment and exploited the capabilities of the TWS tool. This software is able to assist the user in performing a correct network selection

depending on service-related information coded within the SSID field. In addition, it is able to interact with both DHCP and SLP protocols to acquire configuration information. Results show that a beacon-based solution for service publishing definitely outperforms the legacy "try-and-see" approach. To the best of our knowledge, this is the first paper which presents a quantitative analysis on this topic, and we hope that this paper will help discussion within the scientific community towards the final solution(s) concerning service publishing in 802.11 networks.

Future works will address the definition of an autonomic mechanism to engineer the service offer (publishing and provisioning) in a single operator scenario with multiple (V)APs, so as to control network resources and improve service performance. In addition, another objective is to measure the cost of a beacon-based service publishing procedure in terms of wireless bandwidth consumption when the amount of coded information within beacons varies.

ACKNOWLEDGMENT

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

REFERENCES

[1] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE Standard 802.11, 1999.
[2] Y.W. Lee, S.C. Miller, Network Selection and Discovery of Service Information in Public WLAN Hotspots', 2nd ACM international workshop on Wireless mobile applications and services on WLAN hotspots, Philadelphia, USA, 2004.
[3] E. Guttman et al., Service Location Protocol, version 2, IETF RFC 2608, 1999.
[4] Virtual AP Technology Multiplies WLAN Services, Whitepaper, Colubris Networks, March, 2004, available at http://www.colubris.com/downloads/whitepapers/wp_vap.pdf.
[5] The Colubris Web Site, MultiService Controllers, http://www.colubris.com/downloads/datasheets/DS_MSC_3000.pdf.
[6] The OpenSLP Project, http://www.openslp.org/.
[7] The Squid Web Proxy Cache, http://www.squid-cache.org/.
[8] Draft IEEE Standard for Local and Metropolitan Area Networks: Media Independent Handover Services, July 2005.
[9] http://grouper.ieee.org/groups/802/11/Reports/tgu_update.htm, IEEE P802.11, TASK GROUP U.
[10] M. Moreton, Suggested TGu Functional Requirements, IEEE Contribution, doc.: IEEE 802.11-05/0279r15, June 2005.
[11] M. Moreton, Requirement Motions, IEEE Contribution, doc.: IEEE 802.11-05/0643r2, July 2005.
[12] A. McDonald, E. Hepworth, 802.11 TGu Initial Network Selection Concept, IEEE Contribution, doc.: IEEE 802.11-05/0870r0, September 2005.
[13] B. Aboba, M. Beadles, The Network Access Identifier, IETF RFC 2486, January 1999.
[14] R. Mahy, Network Characteristics for AP Selection, IEEE Contribution, doc.: IEEE 802.11-05/1595r0, January 2005.
[15] M. Liebsch, A. Singh, H. Chaskar, D. Funato, E. Shim, Candidate Access Router Discovery (CARD), IETF RFC 4066, July 2005.
[16] B. O'Hara, P. Calhoun, J. Kempf , Configuration and Provisioning for Wireless Access Points (CAPWAP) Problem Statement, IETF RFC 3990, February 2005.
[17] L. Yang, P. Zerfos, E. Sadot, Architecture Taxonomy for Control and Provisioning of Wireless Access Points (CAPWAP), IETF RFC 4118, June 2005.
[18] S. Alexander, R. Droms, DHCP Options and BOOTP Vendor Extensions, IETF RFC 2132, March 1997.
[19] R. Ahmed, R. Boutaba, F. Cuervo, Y. Iraqi, T. Li, N. Limam, J. Xiao, and J. Ziembicki, Service Discovery Protocols: a Comparative Study, IFIP/IEEE International Symposium on Integrated Network Management (IM'2005) Application Sessions, Nice, France, May 2005.
[20] IEEE 802.1Q Standard, Virtual Bridged Local Area Networks, May 2003.
[21] B. Anton, B. Bullock, J. Short, Best current practices for Wireless Internet Service Providers (WISP) Roaming, v1.0, Wi-Fi Alliance, February 2002.
[22] IEEE 802.1X Standard, Port-based Network Access Control, 2001.
[23] F. Zhu, M.W. Mutka, L.M. Ni, Service Discovery in Pervasive Computing Environments, IEEE Pervasive Computing, 4(4), October-December 2005.

APPENDIX

Let us compute the values of $T_w(i)$, i.e., the time needed to select the correct network access at the i-th attempt, weighted by the probability that this event occurs.

It is straightforward to verify that

$$T_w(1) = T_{FULL} \cdot \frac{Y}{N}. \quad (8)$$

The computation of the values of $T_w(i)$ with $i>1$ is easy but cumbersome. In more detail, we can express $T_w(i)$ as follows

$$T_w(i) = \sum_{(j,k,l)\in A(i)} \left(T_A(j,k,l)\cdot P_A(j,k,l)\right) = \sum_{(j,k,l)\in A(i)} \left(T_A(j,k,l)\cdot \frac{Y}{N-(i-1)} \cdot Q(j,k,l) \right). \quad (9)$$

$T_A(j,k,l)$ is the time needed to find the desired network at the i-th attempt after the occurrence of a number of j α–events, k β–events, and l γ–events. The relevant equation is the following:

$$T_A(j,k,l) = j\cdot T_{DENIED} + k\cdot T_{NO\_SERV} + l\cdot T_{FULL} + T_{FULL} = j\cdot T_{DENIED} + k\cdot T_{NO\_SERV} + (l+1)T_{FULL}. \quad (10)$$

$A(i)$ represents the set of integer solutions of the equation

$$j + k + l = i - 1 \quad (11)$$

with the constraints:

$$\begin{cases} 0 \le j \le N - M \\ 0 \le k \le M - X \\ 0 \le l \le X - Y \end{cases}. \quad (12)$$

$P_A(j,k,l)$ is the probability of achieving a successful network access selection at the i-th attempt after the occurrence of a number of j α–events, k β–events, and l γ–events. $P_A(j,k,l)$ is given by multiplying the probability of success at the i-th attempt (equal to $\frac{Y}{N-(i-1)}$) by the probability, $Q(j,k,l)$ of achieving (i-1) failure attempts with j α–events, k β–events, and l γ–events.

At this stage, we need to compute the expression of $Q(j,k,l)$. To this end, note that, once the value of the triple (j,k,l) has been fixed, there are a number of sequences of events compliant with that triple. Thus, $Q(j,k,l)$ is equal to the

probability $P_F(j,k,l)$ of having one of the sequences of events of types α, β, and γ compliant with (j,k,l) multiplied by the number of these sequences (equal to $(i\text{-}1)!$, i.e., the permutations of $(i\text{-}1)$ objects). It is easy to verify that

$$P_F(j,k,l) = \frac{P^{j}_{(N-M)} P^{k}_{(M-X)} P^{l}_{(X-Y)}}{P^{i-1}_{N}}, \tag{13}$$

where P^{L}_{H} represents the number of permutations of size L taken from H objects, i.e.,

$$P^{L}_{H} = \prod_{i=1}^{L} \left(H - (i-1)\right) = \frac{H!}{(H-L)!}. \tag{14}$$

Finally, the expression of $Q(j,k,l)$ is equal to:

$$Q(j,k,l) = \frac{P^{j}_{(N-M)} P^{k}_{(M-X)} P^{l}_{(X-Y)}}{C^{i-1}_{N}}, \tag{15}$$

where C^{L}_{H} is the number of combinations of size L of H objects without repetitions, i.e.,

$$C^{L}_{H} = \frac{H!}{L!(H-L)!}. \tag{16}$$