View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by INRIA a CCSD electronic archive server

Energy-efficient middleware-layer multi-radio networking: An assessment in the area of service discovery

Damien Charlet, Valérie Issarny, Rafik Chibout

To cite this version:

Damien Charlet, Valérie Issarny, Rafik Chibout. Energy-efficient middleware-layer multi-radio networking: An assessment in the area of service discovery. Computer Networks, Elsevier, 2008. inria-00567860

HAL Id: inria-00567860
https://hal.inria.fr/inria-00567860

Submitted on 22 Feb 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Energy-efficient Middleware-layer Multi-radio Networking: An Assessment in the Area of Service Discovery

Damien Charlet *

LIFC – University of Franche-Comté

Valérie Issarny, Rafik Chibout

ARLES project-team – INRIA Rocquencourt

Abstract

The massive deployment of mobile networks together with the emergence of powerful portable devices has given rise to pervasive computing environments. In such environments, mobile users may discover and access services offered in surrounding networks using Service Discovery Protocols (SDPs). However, several SDPs are currently in use, each one designed for a specific target network architecture and setting. As a result, in today's multi-radio networking environment, SDPs do not equally suit each radio interface. In order to provide effective service discovery in multi-radio networks, the most resource-efficient radio interface should be chosen with respect to two main criteria: the adequacy of the interface against the SDP to be used, and energy saving, which is crucial for battery-powered devices. Toward this goal, this article assesses how to effectively take advantage of the multiple radio interfaces now embedded within wireless devices with respect to energy consumption, from the standpoint of service discovery and access. It further investigates the adequacy of legacy SDPs with the various networks, so as to classify the most appropriate networks for each SDP. Exploitation of these results is finally investigated with the description of an energy-efficient algorithm for SDP-based context sensing in a multi-radio pervasive environment.

Key words: B3G networks, context awareness, energy consumption, service discovery

[*] This work was done while the author was at INRIA

Email addresses: damien.charlet@pu-pm.univ-fcomte.fr (Damien Charlet), valerie.issarny@inria.fr (Valérie Issarny, Rafik Chibout).

1 Introduction

While wireless networking technologies were developed autonomously, and sometimes in direct concurrence with each other, recent evolution of mobile networks outlines a new trend: several radio technologies are to be used concurrently and complementarily. This new trend, called convergence of mobile networks, is more particularly addressed in Beyond 3G (B3G) networks. B3G-capable devices then hold several radio interfaces, and allow switching from one radio interface to another, e.g., upon disconnection (vertical hand over). Thus, effective deployment of distributed applications over several heterogeneous radio networks appears as a key challenge for distributed systems.

In this article, we more particularly focus on the concept of smart spaces [1] surrounded by B3G networks. That is, users are evolving in an environment where many networked objects (e.g., PCs, PDAs, CE devices, smartphones, sensors) provide services and information via one or more of the available wireless networks. Users further own handheld devices with several radio interfaces, and are thus able to move around and access the services offered in all the networks. When services may be accessed via several radio interfaces, B3G-capable applications should communicate through the most appropriate interface according to criteria such as energy consumption, which is a crucial constraint for embedded devices [2] [3]. Still, in order to be accessed, services must first be discovered. Over the years, many academic and industry supported Service Discovery Protocols (SDPs) have been proposed for specific networks (e.g., Jini [4] for intranets, UPnP [5] for home networks). While efficient for the targeted environment, existing SDPs prove to be inefficient (e.g., communication cost overhead) or not applicable (e.g., filtered multicast) in different network settings. Furthermore, when an application may use several radio interfaces, the least power consuming interface may not be the most effective choice, if the SDP to be used does not suit the interface's properties and settings.

Within the above framework, we investigate support for context sensing in B3G networks, which is a base requirement of the smart space concept [1]. Specifically, context sensing in the multi-radio networking environment is realized through energy-efficient service/resource discovery. Our solution then exploits the various network interfaces embedded in mobile, wireless devices to discover networked services, which make up the context. Further, our solution overcomes the heterogeneity of legacy SDPs run in the environment in a way that both minimizes resource consumption and offers response time comparable to that of legacy SDPs.

This article is structured as follows. The next section is dedicated to the assessment of the various radio interfaces in use today with respect to energy consumption for typical service discovery and access. Then, Section 3 inspects the adequacy of the key features of legacy SDPs with those of the radio interfaces in order to rate the efficiency of each SDP on the different radio networks. Building upon the above assessments, Section 4 introduces an energy-efficient algorithm for SDP-based context sensing in a multi-radio environment, which is a comprehensive scheduling of legacy SDPs over embedded radio interfaces. Finally, Section 5 concludes with a summary of our contribution.

2 Assessing Multi-radio based Service Discovery with respect to Energy Consumption

It is a known fact that all radio interfaces do not consume the same amount of energy during transfer. In particular, several studies of the power consumption of the Bluetooth and WiFi interfaces may be found in the literature [6] [7]. However, all these studies concern measurements done while transferring large files at a sustained rate, whereas SDPs are not based on this type of sustained exchange. SDPs rather use short and cyclic transfers for active discovery, and sniffing for passive discovery.

The goal of this study is to measure the power consumed on handheld devices during passive (§ 2.1) and active (§ 2.2) service discovery, and service access (§ 2.3) through their radio interfaces, i.e., WiFi, Bluetooth and GPRS. In our test bed, two different handheld devices are used: HP Ipaq HX4700 and HP Ipaq H6340. Both are running the same operating system (namely, Microsoft Pocket PC 2003) and have been configured with similar system settings. In particular, for power management, stock settings have been used for all the features except those altered by external factors, such as automatic backlight adjustment, which have been set to a fixed value or disabled to allow reproducibility of the experiment. The two PDAs are powered by the same type of Li-ion batteries, having a capacity of 1800 mAh. The processor of the HX4700 is an Intel PXA 270 running at 624 MHz, while the H6340 embeds a Texas Instrument OMAP 1510 running at 168 MHz. These two models hold the same integrated 802.11b WiFi network interface and the same integrated Bluetooth network interface, while the H6340 model also embeds a GSM/GPRS interface. Measurements with WiFi and Bluetooth have been done with the two PDAs, while the GPRS test bed only includes the Ipaq H6340. In the experiments where data is exchanged, the PDAs are communicating with a laptop computer having several networking interfaces (i.e., WiFi, Bluetooth and wired ethernet). The WiFi and Bluetooth interfaces are studied according to their various operating modes: the WiFi interface is divided into the ad hoc and infrastructure modes, and the Bluetooth interface into the Bluetooth piconet and Bluetooth NAP modes. We simulate passive discovery by putting the interfaces of the three devices in a listening state during a defined period of time.

Table 1
Base consumption due to peripherals other than radio interfaces

Device	Consumption (%)
Ipaq H6340	7.5
Ipaq HX4700	6

Active discovery is simulated by executing several short transfers between one of the PDAs and the laptop, cyclically during a defined time period. This experiment is reiterated several times with each PDA, one after the other. The same *modus operandi* is used with sustained transfers to simulate services access. We measure power consumption during sustained transfers between each PDA, one after the other, and the laptop. We then infer from these measurements a model of consumption for each interface, in order to compare our measures with the ones that may be found in the literature. Finally, these results are interpreted and discussed in the framework of energy-efficient service discovery and access in multi-radio networks (§ 2.4).

In all the tests, we measure the relative power consumption after half an hour of operation. The curves of discharge of the devices not being linear, all measurements are started when the battery is fully loaded, and the level of battery remaining is read from the device at the end of the operation. This cycle of measurements is reiterated several times and the mean value is given as the final result, as we obtained coefficients of variation lower than 0.04 at the 95 % confidence level in all the different configurations we studied. We first perform a base measurement consisting in leaving the Ipaq switched on for half an hour with all the radio interfaces down, in order to be able to discriminate the consumption of the radio interfaces with that of the other peripheral elements. This base result is given in Table 1. Our measurements have shown that when taking into account the relative consumption of the Ipaqs (i.e., measured consumption minus that of the peripherals elements other than the studied radio interface), we obtain the same values for both the Ipaq H6340 and the Ipaq HX4700 in all the different configurations. Therefore, results provided in the following sections gather only one value per radio interface: the percentage of battery consumed in addition to that of the above base consumption, whichever the Ipaq is.

2.1 Measurements for passive discovery

The first test bed consists in simulating service discovery under the passive model as with, e.g., UPnP. In this mode, clients never send requests but are continuously listening on the network interface for service advertisements. To simulate this operating mode, both Ipaqs have one radio interface at a time forced in an ”always on” state. In the WiFi ad hoc mode, the laptop computer

Table 2
Consumption of each radio interface for passive discovery

Interface		Consumption (%)
WiFi	Ad-hoc	5
	Infrastructure	1.5
Bluetooth	Piconet	2.5
	PAN	3
GPRS		4.5

is used as a peer, and we make sure all the devices are bonded to the group. In the WiFi infrastructure mode, the devices are associated with an access point and in the Bluetooth PAN Mode, the Ipaqs are linked with the laptop, which takes the role of the Bluetooth network access point. For these configurations, we also make sure all the devices obtain an IP address and are fully network-operational before starting the measurements. In the Bluetooth piconet mode, the laptop takes the role of the master of the piconet, and we make sure both Ipaqs belong to the piconet. In all the configurations, no specific traffic is generated between the devices. Thus, devices are only listening to a typical background-noise type traffic. We measure the power consumed after half an hour of listening by reading the battery level on the Ipaqs. The consumption of each radio interface after half an hour of passive discovery is given in Table 2.

For passive discovery, the most consuming radio interface is WiFi in the ad hoc mode with 5% of the battery consumed by the interface after half an hour. This result is explained by the amount of signaling messages exchanged between the devices to manage the group, forbidding hardware optimization mechanisms used in the infrastructure mode such as power save polling [8]. Indeed, power save polling allows putting the WiFi interface in an energy saving mode, at the hardware level, until the client receives a beacon frame from the access point indicating it should wake up to receive data. Thanks to the these hardware level optimizations, WiFi infrastructure is the least consuming interface with only 1.5% of the battery consumed. The second most consuming interface is GPRS with a consumption of 4.5%. The Bluetooth interface is commonly known as being more energy efficient than WiFi. This assertion may be verified when comparing the Bluetooth consumption with the measurements in the WiFi ad hoc mode. In its two modes, Bluetooth has a better efficiency with 2.5% of the energy used in the piconet mode and 3% in the PAN mode. Nevertheless, this assertion is no longer true when comparing Bluetooth with WiFi in the infrastructure mode. Indeed, the power saving of WiFi in the infrastructure mode when idled is more efficient than that of Bluetooth.

As a conclusion, the most energy-efficient radio interface for passive discovery

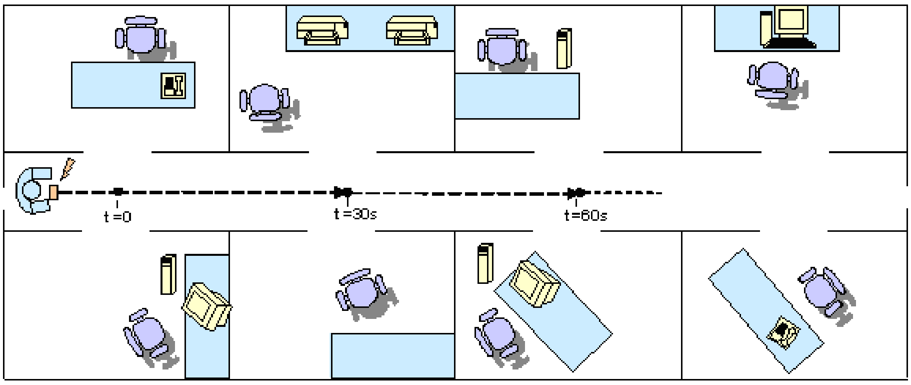


Fig. 1. Scenario for active service discovery

is WiFi in the infrastructure mode. On the other hand, switching this interface to the ad hoc mode renders passive discovery very power consuming and should thus be avoided. When Bluetooth is used for passive discovery, both of its modes consume almost the same amount of energy.

2.2 Measurements for active discovery

We now measure the energy consumption of the different radio interfaces when used to cyclically perform active discovery in order to maintain knowledge of service offers over time as with, e.g., the SLP [9] discovery protocol. We simulate a typical real life scenario where a user carrying a handheld device is walking slowly along a corridor. The device cyclically emits packets, called discovery requests, in order to discover services offered by providers in its surrounding. When a device providing a service receives a relevant discovery request, it replies with a service announcement describing the features and interfaces of the service. Thus, the requester has at its disposal all the required information for service access. The scenario is depicted in Figure 1. In order to simulate this behavior, request/response sequences are processed cyclically between one of the handheld devices and the laptop computer. Each Service Discovery Protocol describes its own message format for service discovery and response [10]. Furthermore, the volume of data exchanged is in the order of the mean volume of data transferred by leading SDPs (i.e., 1.5 kilo bytes, see § 3). Indeed, requests sent by the PDAs are 500 bytes and response messages from the laptop are 1 kilo bytes long. The period between two sequences lasts 30s, which allows effectively taking into account the dynamicity of the service offers in an evolving environment, and also represents, for a human walking slowly, about 16 meters traveled, which corresponds to the emission range of Bluetooth emitters placed in offices as in, e.g., smart spaces. Like the previous measurements, we let the simulation run for half an hour. Results are given in Table 3.

Table 3
Consumption of each radio interface for active discovery

Interface		Consumption (%)
WiFi	Ad-hoc	6.5
	Infrastructure	6.5
Bluetooth	Piconet	5.5
	PAN	5.5
GPRS		28.8

We first notice that when small transfers occur periodically, the energy consumption of WiFi and Bluetooth is not influenced by the mode in which they operate. Therefore, even if the volume of data transferred is small, the hardware optimizations of the WiFi infrastructure mode are not able to take place and the measured consumption of the infrastructure mode catches up with the one of the ad hoc mode. Indeed, both modes of the WiFi interface are more consuming than the ones of the Bluetooth interface, with respectively 6.5% and 5.5% of the battery consumed after half an hour. Finally, a huge increase in the power consumption of GPRS is measured. This interface becomes the most power consuming with a score of 28.8%, which is about five times greater than the consumption of the WiFi and Bluetooth interfaces. According to these results, active service discovery should be performed via the Bluetooth interface, in order to optimize energy consumption.

2.3 Measurements for services access with sustained transfers

The aim of the last simulation is twofold: to compare consumption of the network interfaces when accessing services where sustained data flows are exchanged, and to verify that our results are congruent with those found in the literature [6,7]. In this simulation, we make one of the handheld devices exchange large sets of data back and forth with the laptop computer, fully filling its bandwidth during half an hour. More precisely, One of the handheld devices sends a 10 MB file, and then receives back a 10 MB file from the laptop. The cycle reiterates without delay during half an hour. The laptop is monitoring the communication during all the operation and logs the volume of data transferred at the end, while the energy consumption value is measured on the PDA at the end. The two PDAs and all the networks interfaces are assessed one after the other. The measured consumption values for the PDAs are reported in Table 4, along with the volume of data transferred on each interface. We further recall that consumption measures for the 2 iPaqs are equivalent and are thus not distinguished in the table.

Table 4
Power consumption and data transfer.

Interface		Consumption	Volume
WiFi	Ad-hoc	8.5 %	270 MB
	Infrastructure	8.5 %	250 MB
Bluetooth	Piconet	4.8 %	54 MB
	PAN	5.5 %	80 MB
GPRS		29 %	10 MB

Table 5
Power consumption per 100 megabytes

Interface		Consumption (%/100 MB)
WiFi	Ad-hoc	3.1
	Infrastructure	3.4
Bluetooth	Piconet	8.8
	PAN	6.9
GPRS		290

After 30 minutes of data exchange at full rate, GPRS is still the most power consuming interface with 29% of energy consumption. The WiFi interface has consumed 8.5% of the battery power, the Bluetooth interface in the PAN mode has consumed 5.5% and the piconet mode is the most frugal with a consumption of 4.8%. Nevertheless, when looking at the different volumes of data transferred across each interface, it is clear that these measurements cannot be compared as is. As GPRS, Bluetooth and WiFi have very different bandwidth capacities, the volume of data exchanged during a defined time period is disproportioned: the WiFi ad hoc bandwidth is respectively 5 times and 27 times larger than the Bluetooth piconet bandwidth and the GPRS bandwidth. It is thus not fair to compare the interfaces by comparing the measured consumptions at full rate. This is a well known problem in the literature. Work has been undertaken to model the energy consumption of the Bluetooth and WiFi interfaces [6], and also face the need to define comparison equivalencies with respect to the asymmetrical bandwidths of the radio interfaces. Nevertheless, no well recognized solution has emerged.

A basic comparison equivalency consists in converting the consumptions of each interface reported in Table 4 into a 100 MB base by applying a golden rule. These values are gathered in Table 5. It may first be seen that the consumption of the GPRS interface exceeds 100%. That is, battery power would not be sufficient to transmit 100 megabytes of data. It may also be seen that on the golden rule basis, the cost per megabyte is at least twice lower for the

Table 6
Base consumption for each interface

Interface		Consumption (%)
WiFi	Ad-hoc	5
	Infrastructure	5
Bluetooth	Piconet	2.5
	PAN	3
GPRS		4.5

WiFi interface than for Bluetooth. The WiFi interface seems thus much more attractive than Bluetooth for data-intensive applications. Nevertheless, this type of comparison implies that the only source of energy consumption that must be taken into account comes from the emission and reception of packets. The results outlined in the previous section about measurements done after passive discovery show that even when almost no communication occurs on the interfaces, their power consumption is not negligible because of, e.g., signaling. We call this value base consumption. Therefore, the energy consumption of an interface is made up of the constant base consumption and of the per-packet consumption relating to the throughput of the transmission. When comparing the measured consumptions when the interfaces are idle (Table 2) with the measured consumptions at full rate (Table 4), we can see that base consumption represents a large part of the interface's total consumption. Moreover, the golden rule applied on the consumption at full rate to obtain the power consumptions per 100 MB (Table 5) does not take into account the fact that the global consumption also depends on the base consumption, which is not dependant on the volume of data transmitted. Therefore, in order to refine the approximation operated in Table 5, we must differentiate base consumption and per-packet consumption from the consumptions given in Table 4 before applying the golden rule.

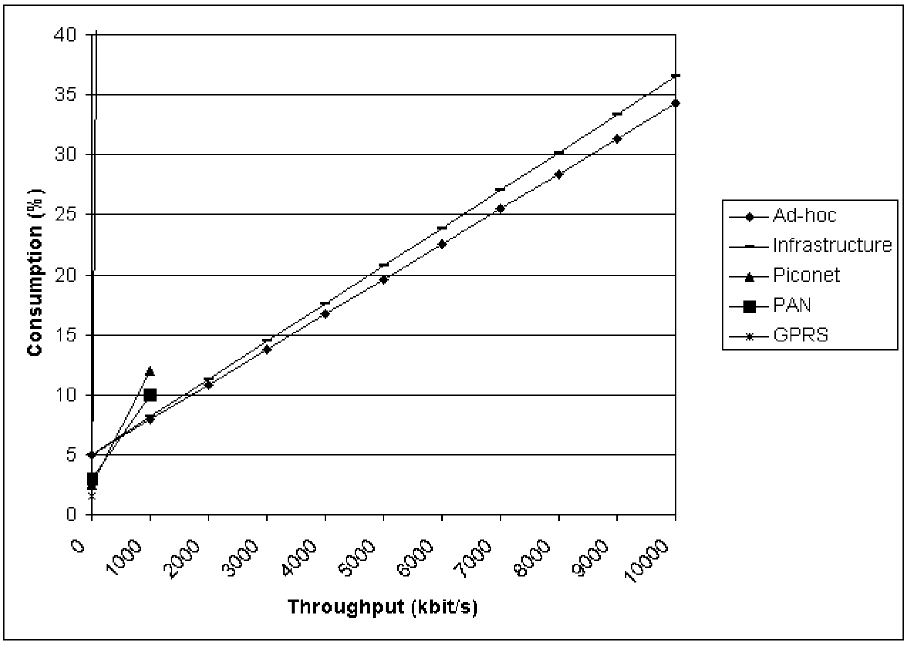
Because of energy saving optimizations of the WiFi infrastructure mode (i.e., power save polling), the values measured in Table 2 cannot be used as is as base consumption values. As outlined in Table 3, a small amount of data transmitted on the WiFi interface in the infrastructure mode is sufficient to significantly increase the power consumption of the interface. The base consumption should be measured by transmitting a sufficient volume of data not to let the radio interfaces idle, in order to overcome hardware optimizations, but small enough to negligibly alter the measured consumption with per-packet consumption. In order to operate such measurements, we modified the test bed used in the active discovery simulation (see § 2.2) by only sending one ping packet every 30 seconds. A ping packet being 84 bytes long, 5 kilo bytes are sent after half an hour. This is 18 times smaller than the volume of data exchanged for active discovery, and negligible compared to the amount

Table 7
Transmission and global consumptions per 100 MBytes for each interface

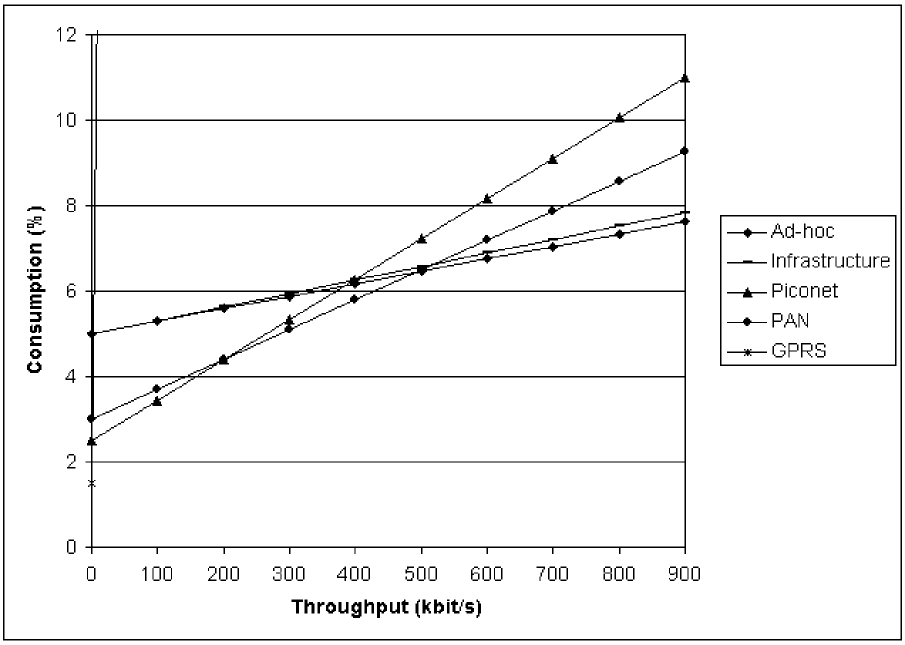
Interface		Transmission overhead (%)	Per-packet consumption (%/100MB)	Global consumption (%/100MB)
WiFi	Ad-hoc	3.5	1.3	6.3
	Infrastructure	3.5	1.4	6.4
Bluetooth	Piconet	2.3	4.2	6.7
	PAN	2.5	3.1	6.1
GPRS		24.5	245	269.5

of data exchanged at full rate. The results of this experiment are given in Table 6. The small amount of data periodically exchanged is sufficient to prevent the use of the energy optimizations of the WiFi infrastructure, and this value increases from 1.5% in Table 2 to 5% in Table 6. The alteration of the consumption of the other interfaces is so negligible that it does not affect the other (rounded) values reported in Table 6. This hints that in this simulation, the per-packet consumption is negligible; hence, the measured consumptions reported in Table 6 are quite precisely the base consumption values of each interface. Once base consumption is known, we may substract it from the consumption measured at full rate in order to obtain the transmission cost. This remaining value may be correlated with the volume of data transmitted from Table 4 to approximate the per-packet consumption.

Finally, by adding the base consumption value from Table 6 with the estimated per-packet consumption, we obtain the global energy consumption given in Table 7. When 100 MBytes are transmitted in 30 minutes, all the estimated global consumption values, except the one for the GPRS interface which is much higher than 100%, lie between 6.1% and 6.7% (Table 7, right most column). When taking into account the error rate due to the approximation of the formula, we can conclude that those values are almost identical. The value chosen for comparison, i.e., 100 Mbytes/30 min, that converts into 450 kbit/s, is the throughput around which WiFi becomes more energy efficient than Bluetooth. More precisely, the results gathered in Tables 6 and 7 characterize the consumption models of the radio interfaces. The base consumption values in Table 6 show that the Bluetooth interface, in the two modes, has a lower base consumption than the WiFi interface, while the per-packet consumption values in the second column of Table 7 show that the energy consumption of Bluetooth increases faster than the one of WiFi, when the transmission rate increases. According to this model, the power consumption of each interface according to the throughput may be plotted by a line starting at the base consumption value and whose slope is the per-packet cost.



(a) Power consumption of the radio interfaces after 30 min. according to throughput (scale : WiFi bandwidth)



(b) Power consumption of the radio interfaces zoomed around Bluetooth bandwidth

Fig. 2. Power consumption of Bluetooth and WiFi according to throughput

The power consumption curves for each radio interface are depicted in Figure 2. Figure 2(a) shows that the Bluetooth interface is more efficient than the WiFi interface for low throughputs, as outlined by the base consumption values. Nevertheless, as the per-packet cost of the two modes of Bluetooth is more important than the ones of the two modes of WiFi, the consumption lines of the Bluetooth interface abruptly increase and thus the power consumption of the Bluetooth interface quickly overtakes the consumption of the WiFi interface. Moreover, the plot of the Bluetooth curves has been stopped at 1 Mbit/s, as this is the theoretical maximal bandwidth of the Bluetooth interface, whereas WiFi curves go much further, as 802.11b's theoretical limit goes up to 10 Mbit/s. The same graphic, zoomed around the cross points, may be found in Figure 2(b). Figure 2(b) clearly shows that the Bluetooth interface is only preferable over the WiFi interface for throughputs up to 380 kbits/s in the piconet mode and 520 kbits/s in the PAN mode. For larger throughputs, the WiFi interface is less consuming than the Bluetooth interface. Obviously, the WiFi interface also becomes the sole practical solution when the Bluetooth interface cannot cope with the required bandwidth any more. Validity of these results is reinforced by observing that congruent models of energy consumption have been described for WiFi and Bluetooth interfaces in [6] and [7]. Finally, both curves show that the GPRS interface, apart from being the interface with the most limited bandwidth, is also the most consuming interface, whatever the throughput is.

Globally, even if for very small throughputs the Bluetooth interface is less consuming than the WiFi interface, its high per-packet cost quickly renders the WiFi interface more attractive. Therefore, the WiFi interface should be preferred for services access, as the required bandwidth will most of the time be larger than this cross point. Moreover, if while using the Bluetooth interface, the required throughput increases over time, one may quickly face hardware non feasibility whereas WiFi could sustain a much larger bandwidth.

2.4 Energy-efficient service discovery and access in multi-radio networks

Globally, the most energy efficient operating mode for service discovery and access consists in using the Bluetooth interface for discovery, and the WiFi interface for service access. This scenario requires that gateways between the Bluetooth network and the WiFi network be available in the environment, or that all the devices use both interfaces concurrently. These usages are not currently widespread, but are conceivable in specific cases such as enterprise deployment. Nevertheless, such a scheme assumes that every SDP can be used adequately with any radio interface. However, SDPs are designed with a specific networking environment in mind, and thus do not behave the same way on different radio networks. According to its intrinsic specifics, such as its discovery model or networking protocol, a SDP may be technically unusable on a specific radio interface. The next section assesses the features of existing

SDPs with respect to the wireless network interfaces composing the multi-radio network, in order to rate the adequacy of each SDP against each radio interface.

3 Assessing Existing SDPs in Multi-radio Networking Environments

Service Discovery Protocols (SDPs) enable finding and using networked services without any previous knowledge of the services' specific location. To provision exhaustive knowledge of the services offered on the reachable wireless networks, a client may need to use several SDPs, which are not identically suited for every radio interface. Some combinations are even technically unsound. Beyond these incompatibilities, a device may offer connectivity to the same part of the network through two different interfaces. SDPs can thus use one or the other to discover available services. A discovery protocol optimized for multi-radio networking should be able to choose the most suited interface according to the SDP to be used. Therefore, it is crucial to rate the adequacy of each SDP against each radio interface.

As sketched in the previous section, the WiFi technology has been designed to operate in two different modes: ad hoc and infrastructure. The infrastructure mode may also be split into a "pure" infrastructure mode (access to the local area network) and a hybrid infrastructure + gateway mode (access to the LAN and beyond). Bluetooth may also be configured into two different operating modes: the "master-slave" piconet mode (the default Bluetooth configuration belongs to this mode), and the IP-based PAN (Personal Area Network) mode. Each of these two modes may also be divided into two. In the piconet mode, a device may provide the "LAN Access" profile, in which case it allows the other devices of the piconet to access the local area network. A node of a PAN can offer the same function, called NAP (Network Access Point). Our analysis thus treats 8 cases of radio interfaces: WiFi infrastructure, WiFi gateway, WiFi ad hoc, Bluetooth piconet, Bluetooth Lan-ap, Bluetooth GN, Bluetooth NAP and GPRS.

Many academic and industry-supported SDPs are available, each with its own features and use case. Among these SDPs, a few are more widely deployed and adopted: the second version of IETF's SLP has been standardized in RFC 2608 [9]; UDDI (Universal Description, Discovery, and Integration) [11] is one of the core Web services standards [1], while Sun Microsystems develops

[1] WS-discovery also emerges as a major standard for Web services. However, its behavior is close to that of UPnP-SSDP and SLP, and is thus not studied in the remainder.

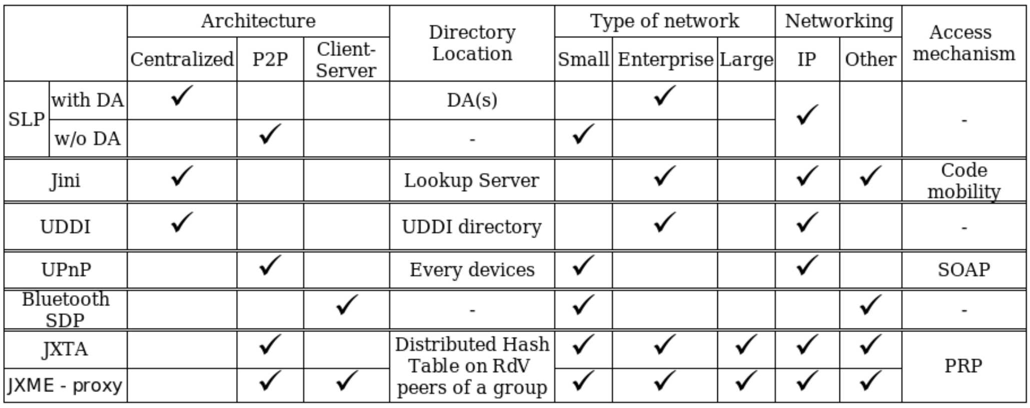
and maintains Jini [4], and Microsoft bases its strategy on UPnP [5]. In an open environment, these SDPs might be often met and should therefore be studied in the context of multi-radio networking. SLP having two different operating modes/architectures, is divided into two distinct protocols: SLP with Directory Agent (called SLP with DA) and SLP without DA. Bluetooth being one of the studied radio interfaces, the service discovery protocol defined in the Bluetooth standard (Bluetooth-SDP [12]) is also studied. It is the only SDP that cannot function over IP at all. Finally, even if JXTA [13] is not just a SDP, but rather a set of open protocols that allow any connected device on the network to communicate and collaborate in a P2P manner, its resource discovery support needs to be studied as it brings different discovery paradigms and possibilities. We also study the version of JXTA for lightweight devices, called JXME [14], whose discovery architecture slightly differs from JXTA's: a JXME peer only addresses one single peer (its proxy), which carries out the discovery and forwards the results. Our analysis thus takes into account 8 SDPs: SLP with or without DA, Jini, UDDI, UPnP, Bluetooth SDP, JXTA and JXME.

In the following, we first itemize and describe the SDP features that relate to the adequacy of SDPs against the different radio networks (§ 3.1) and outline the relevant properties of the radio interfaces (§ 3.2). We then present our study, which results in a matrix of values rating the adequacy of each SDP against each radio interface. We carry out this analysis by checking the adequacy of each SDP feature with the relevant properties of each radio interface (§ 3.3). The aggregation of these results concludes this analysis by providing for each SDP a global adequacy against each radio interface, so that a technical incompatibility with only one of the features gives a null value of adequacy (§ 3.4).

3.1 SDP features

The major architectural difference between SDPs concerns the existence of a central repository, and the way service offers are handled. When a central repository exists, the offers are stored and retrieved from this repository, which may possibly be distributed. The architecture is then known as (logically) "centralized". When no central repository exists, two cases arise: requests and offers may be directly exchanged from one to all in a peer-to-peer scheme, or discovery is done solely via unicast communications between two devices. In the former scheme, each peer is in charge of caching the received results. We call this architecture peer-to-peer (or P2P). In the latter case, no broadcasting of requests or announcement takes place. Service discovery consists in a single requester directly requesting a specific provider for its service offers. Therefore, this architecture is called client/server. SDPs are also aimed at a

Table 8
SDPs features



		Architecture			Directory Location	Type of network			Networking		Access mechanism
		Centralized	P2P	Client-Server		Small	Enterprise	Large	IP	Other	
SLP	with DA	✓			DA(s)		✓		✓		-
	w/o DA		✓		-	✓					
Jini		✓			Lookup Server		✓		✓	✓	Code mobility
UDDI		✓			UDDI directory		✓		✓		-
UPnP			✓		Every devices	✓			✓		SOAP
Bluetooth SDP				✓	-	✓				✓	-
JXTA			✓		Distributed Hash Table on RdV peers of a group	✓	✓	✓	✓	✓	PRP
JXME - proxy			✓	✓		✓	✓	✓	✓	✓	

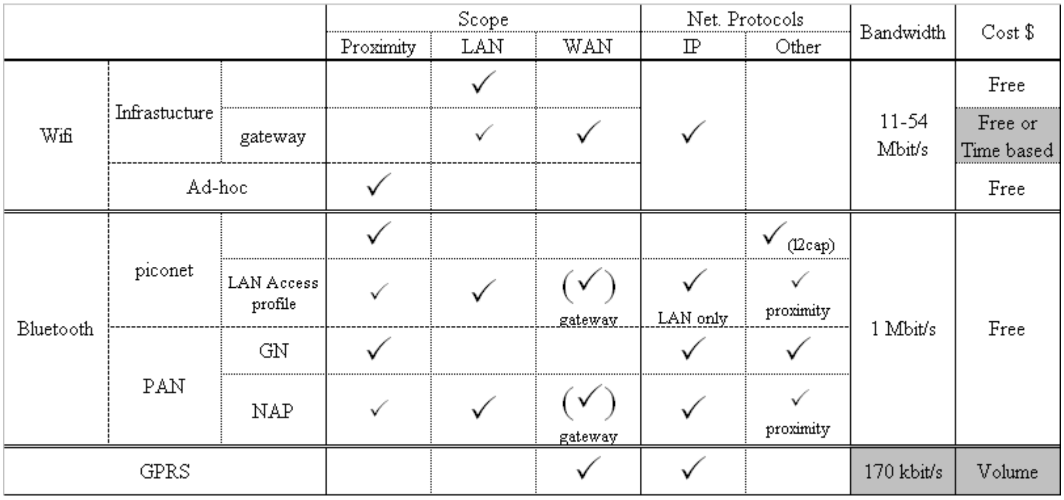
specific networking environment defined by the network size (i.e., small, enterprise or large-size network) and the network protocol used. SDPs may or may not allow scalability in terms of number of clients and network size. SDPs are further often strongly coupled with some middleware solution forbidding service discovery and access with other middleware.

The features of the studied SDPs are characterized in Table 8. It is noticeable that, even if JXTA was designed with the aim of being executed without any dependency on a particular network technology, the porting of the JXTA framework on the Bluetooth stack not being yet operational, the network protocol independence is still theoretical. The SDPs features we have just elicited are the variables rendering the use of each SDP more or less effective with respect to each radio interface. In order to evaluate this level of adequacy, the relevant features of the radio interfaces are discussed in the next section.

3.2 Properties of the radio interfaces

Wireless networks appeared in the late 90s, where each radio technology had been designed for a specific usage. Bluetooth was created with a ”personal area network” in mind, in order to avoid wires between peripherals, WiFi networks were designed in order to replace wired local area networks, and GPRS technology to offer dial-up wide area network access over the GSM network. In the course of time, these technologies have evolved, providing different possibilities than what they were formerly designed for. Nowadays, radio networks even advertise overlapping usages. This characteristic is addressed by beyond 3G networks, where convergence of radio networks should take place so that users become unaware of the actual radio technology in use. Nevertheless, radio interfaces still have their own specifics, mostly inherited from their previous usage. Table 9 gathers the specifics of the WiFi, Bluetooth and GPRS interfaces that affect the behavior of SDPs: network scope (vicinity, local area

Table 9
Radio interfaces properties



			Scope			Net. Protocols		Bandwidth	Cost $
			Proximity	LAN	WAN	IP	Other		
Wifi	Infrastucture			✓		✓		11-54 Mbit/s	Free
		gateway		✓	✓				Free or Time based
	Ad-hoc		✓						Free
Bluetooth	piconet		✓				✓ (l2cap)	1 Mbit/s	Free
		LAN Access profile	✓	✓	(✓) gateway	✓ LAN only	✓ proximity		
	PAN	GN	✓			✓	✓		
		NAP	✓	✓	(✓) gateway	✓	✓ proximity		
GPRS					✓	✓		170 kbit/s	Volume

network or beyond), standard network protocol, bandwidth, and financial cost when their use is not free of charge.

As presented in Table 9, GPRS is the interface that has the least evolved from its original function, as it still gives access to WAN solely, by using IP. Its bandwidth is still very limited (170 kbit/s) and the user must pay while using it according to the volume of data transferred. The WiFi bandwidth continuously increases, ranging nowadays between 11 and 54 Mbit/s. It uses the IP protocol in all its operating modes. In the ad hoc mode, the device has access to the devices in its vicinity, while in the infrastructure mode the interface gives access to the local area network. In these usages, the communications are free of charge. The addition of a gateway allows Internet access (WAN) in the infrastructure mode, but may render the communications lucrative (case of hotspots where the user is charged according to time). Bluetooth usage is always free of charge and offers a bandwidth of 1 Mbit/s. In the piconet mode, devices in the vicinity are the only reachable ones, using a particular protocol. If one of these devices offers the LAN access profile, it can be used as a gateway in order to reach the machines on the local area network using IP. Although the piconet mode is the default Bluetooth mode, it is however possible to create an IP network between Bluetooth devices in order to create a "personal area network" (PAN). A Network Access Point (NAP) may reside in the PAN network to thus offer to Bluetooth clients an access to the LAN/WAN.

3.3 Assessing the adequacy of SDPs against radio interfaces

In order to obtain a global level of adequacy for each combination of SDPs with radio interfaces, the correlation of the aforementioned SDPs features with radio interfaces properties must be studied. These observations are translated

Table 10
SDPs vs. Network scope

		WiFi			Bluetooth				GPRS
		Infrastructure		Ad hoc	piconet		PAN		
			gw			LAN-AP	GN	NAP	
SLP	with DA	2	2	1	1	2	1	2	1
	w/o DA	1	0	2	2	1	2	1	0
Jini		2	2	1	1	2	1	2	1
UDDI		2	2	1	1	2	1	2	1
UPnP		1	0	2	2	1	2	1	0
Bluetooth SDP		0	0	2	2	1	2	1	0
JXTA		2	2	2	2	2	2	2	2
JXME		2	2	2	2	2	2	2	2

into numerical values, gathered in matrices, one for each SDP feature, where one cell represents the level of adequacy of one SDP against one radio interface with respect to this feature. These matrices are then aggregated in order to provide each SDP with a global level of adequacy against each radio interface, so that a technical incompatibility with only one of the features gives indeed a null value of adequacy.

Table 10 assesses for each SDP, its adequacy with the radio networks of interest, according to the type of network and scope assumed for the underlying network. For each pair (protocol, interface) a note of feasibility is given: 2 if the binding is adequate, 1 if operation is possible but imperfect and 0 if it is impossible. JXTA and JXME being designed to function on networks of any size, all the interfaces are adequate. SLP without DA, UPnP and Bluetooth-SDP functioning on small networks, are suited to WiFi in the ad hoc mode and to Bluetooth piconet and PAN. They can also function in a degraded manner on LAN-type networks (WiFi infrastructure, Bluetooth LAN AP and NAP without gateway) but cannot be used on wide-area networks (GPRS, WiFi with gateway). SLP with DA and Jini function perfectly on average sized networks like LAN (WiFi infrastructure, Bluetooth LAN AP and NAP) and can scale if these networks are extended (with a gateway). However, they badly function with interfaces offering only WAN access (GPRS). In the same way, it could be possible to use these protocols on proximity interfaces, even if this case is not adequate (WiFi ad hoc, piconet, PAN).

Table 11 assesses the adequacy of SDPs with radio interfaces according to the supported network protocols. All the SDPs are IP-based, except Bluetooth-SDP, which is based on L2CAP. Even if JXTA, JXME and Jini claim network

Table 11
SDPs vs. Network protocol

		WiFi			Bluetooth				GPRS
		Infrastructure		Ad hoc	piconet		PAN		
			gw			LAN-AP	GN	NAP	
SLP	with DA	2	2	2	0	1	2	2	2
	w/o DA	2	2	2	0	1	2	2	2
Jini		2	2	2	0	1,5	2	2	2
UDDI		2	2	2	0	1	2	2	2
UPnP		2	2	2	0	1	2	2	2
Bluetooth SDP		0	0	0	2	1	2	1	0
JXTA		2	2	2	0,5	1,5	2	2	2
JXME		2	2	2	0,5	1,5	2	2	2

protocol independence, they currently do not have any usable implementations apart from IP-based. Therefore, all the SDPs except Bluetooth-SDP can be used with GPRS and WiFi in all its operating modes, as they offer IP connectivity. Bluetooth in the PAN mode also offers IP connectivity, while in the standard piconet mode it can only use Bluetooth-SDP. Finally, when Bluetooth is switched in the LAN-AP mode, L2CAP being used to reach the devices in the vicinity and IP to reach the local area network, all the SDPs can be used even if none makes it possible by itself to traverse the whole set of reachable devices.

As mentioned in Section 3.1, we can classify the studied SDPs into three distinct families according to their architecture: centralized directory based (SLP with DA, Jini, UDDI, JXME), peer-to-peer directory based (SLP w/o DA, UPnP, JXTA) and directory-less client/server (Bluetooth SDP). This classification may be used to assess SDPs versus the topology of the radio network. The architecture of centralized directory-based SDPs is particularly suited to asymmetrical networks (i.e., all the links of these networks do not share the same characteristics, in particular considering bandwidth limitation) such as GPRS, WiFi infrastructure with gateway, BT-LAN AP and BT-NAP. On these networks, the wireless edge link on the device side typically has a lower bandwidth than the network to which it is inter-connected (GPRS/Internet, Bluetooth/LAN). The centralized architecture limits the volume of communication on this link, as only messages relevant for the client are exchanged between the device and the repository. Moreover, there exists a ”logical” optimal location for the directory: on the gateway (offering the LAN-AP service or NAP, on the GPRS proxy or on the WiFi gateway). Operating centralized SDPs on peer-to-peer type networks (WiFi ad hoc, BT piconet, BT GN) is

Table 12
SDPs vs. Architecture

		WiFi			Bluetooth				GPRS
		Infrastructure		Ad hoc	piconet		PAN		
			gw			LAN-AP	GN	NAP	
SLP	with DA	2	2	1	1,5	2	1,5	2	2
	w/o DA	2	1	2	1	1,5	1	1,5	1
Jini		2	2	1	1,5	2	1,5	2	2
UDDI		2	2	1	1,5	2	1,5	2	2
UPnP		2	1	2	1	1,5	1	1,5	1
Bluetooth SDP		2	2	2	2	2	2	2	2
JXTA		2	1	2	1	1,5	1	1,5	1
JXME		2	2	1	1,5	2	1,5	2	2

possible but less suitable. There is indeed no logical location for the directory, since by definition all peers are equal and no device should have a particular role. Bluetooth SDP, which belongs to the client/server family, is a case apart since this protocol neither stores the offers in a directory nor performs peer-to-peer exchanges of the advertisements. The only authorized discovery is done by unicast communication between one client and one provider. This simplistic discovery model, particularly suited to the mode of communication used by Bluetooth, does not induce nor is based on any architecture or particular network topology. The only need is that the two devices can reach each other. This protocol is thus appropriate to all the cases. Finally, the WiFi interface in the infrastructure mode is also apart. At first sight, the infrastructure mode seems asymmetrical, since an access point acts as a gateway between wired and wireless networks. Thus, SDPs with a centralized directory find there a logical location for the directory. Nevertheless, nowadays, the characteristics of the wired and WiFi networks are generally rather close. Moreover, the aim of this operating mode is to combine wireless sub-networks with the local area network. From this point of view, the network appearing homogeneous, SDPs with peer-to-peer directory find there a logical operation land. The WiFi infrastructure interface is thus adapted, from the architectural point of view, to both centralized and peer-to-peer SDPs.

It is necessary to refine the above assessment by also taking into account the broadcasting models and protocols of the SDPs and radio interfaces. If a packet emitted on an ad hoc WiFi network is inevitably received by all the other devices in range, the Bluetooth standards only allow one-to-one communication. Therefore, a ”one-to-all” communication must be simulated by several successive unicast transmissions, multiplying the cost of communica-

Table 13
SDPs vs. bandwidth usage

		WiFi			Bluetooth				GPRS
		Infrastructure		Ad hoc	piconet		PAN		
			gw			LAN-AP	GN	NAP	
SLP	with DA	1	1	1	1	1	1	1	1
	w/o DA	1	1	1	1	1	1	1	1
Jini		1	1	1	0,75	0,75	0,75	0,75	0,5
UDDI		1	1	1	1	1	1	1	1
UPnP		1	1	1	1	1	1	1	1
Bluetooth SDP		1	1	1	1	1	1	1	1
JXTA		1	1	1	1	1	1	1	1
JXME		1	1	1	1	1	1	1	1

tion. This reveals that directory-less SDPs are better suited to ad hoc WiFi networks, while directory-based SDPs are preferable for the Bluetooth ”proximity” scope (piconet and GN). In the case of Bluetooth LAN-AP and NAP, multicast and broadcast cannot be realized between Bluetooth devices, but can be initiated between the gateway and its connected networks. Table 12 presents the assessment with respect to the broadcasting models.

Bandwidth is not a de facto discriminating factor in the field of services discovery, as discovery only requires low bandwidth. Bandwidth may however be determinant for service access. All the protocols do not propose a particular access mechanism, and thus cannot be compared from that standpoint. Nevertheless, a particular case arises: Jini. Indeed, as Jini replies to discovery requests by using a code mobility mechanism, service discovery may become expensive in terms of volume of data to be transferred. Jini is thus not indicated when the user has to pay according to the volume of data transferred (GPRS) or when the bandwidth is restricted (GPRS, to a lesser extent Bluetooth). Therefore, the bandwidth factor does not really gauge its adequacy versus access mechanism, but rather aims to operate a decrease in the global rating, reflecting the preceding remarks. Table 13 reflects this observation, as default adequacy values according to bandwidth are set to 1, and a small decrease is used for values of Jini against GPRS and Bluetooth values.

Table 14
Global SDPs / radio interfaces adequacy

		WiFi			Bluetooth				GPRS
		Infrastructure		Ad hoc	piconet		PAN		
			gw			LAN-AP	GN	NAP	
SLP	with DA	8	8	2	0	4	3	8	4
	w/o DA	4	0	8	0	1,5	4	3	0
Jini		8	8	2	0	4,5	2,25	6	2
UDDI		8	8	2	0	4	3	8	4
UPnP		4	0	8	0	1,5	4	3	0
Bluetooth SDP		0	0	0	8	2	8	2	0
JXTA		8	4	8	1	4,5	4	6	4
JXME		8	8	4	1,5	6	6	8	8

3.4 Matching SDPs against radio interfaces

In order to obtain a global quantitative rating, providing for each SDP its adequacy against each radio interface, we aggregate all the adequacy values from Tables 10 to 13 by calculating the scalar product of these matrices, so that a technical incompatibility with only one of the features gives indeed a null value of adequacy. We thus obtain a matrix of global adequacy represented in Table 14. Each value indicates the level of suitability between a SDP and a radio interface. The higher the value is, the more relevant the use is. The pairs evaluated with the maximum rating (maximum is 8) have a perfect adequacy and should be preferred when their use is possible. Finally, the intermediate values greater than 0 indicate that the use is technically possible, although not being perfectly appropriate.

Among the 64 (SDP, radio interface) pairs, 18 obtain the maximal adequacy value and 13 are totally incompatible. The SDP obtaining the best average adequacy is JXME, since it has a perfect adequacy with half of the interfaces. JXME operating mode being based on JXTA with added optimizations for embedded devices, JXME supersedes JXTA in the context of this study. JXTA logically ranks in second position. JXTA and JXME are the only two SDPs that do not present incompatibility with any radio interface at all. Among the other SDPs, SLP with DA obtains the best adequacy rating, but is incompatible with Bluetooth in the piconet mode. Bluetooth-SDP is logically found as the least adequate, since it is not IP-based and is thus incompatible with the WiFi and GPRS interfaces. It is nonetheless notable that if SLP obtains a very good adequacy when used in the directory mode, this protocol used without directory decreases its rank to the penultimate, with the same rating as UPnP.

The matrix of Table 14 giving the adequacy of SDPs against radio interfaces, may also be read the other way around, in order to estimate the adequacy of the radio interfaces when they are to be used with several SDPs. The interface having at the same time the strongest global adequacy and the largest count of maximal adequacy cases is WiFi in the infrastructure mode. Note that this alters the global energy-efficiency assessment of Section 2. WiFi infrastructure is closely followed by Bluetooth NAP, which has less SDPs with perfect adequacy (3 against 5) but more homogeneous results: on this interface, no SDP is rated with a null adequacy. Bluetooth piconet has at the same time the worst average adequacy and the greatest number of total incompatibilities. It is followed by GPRS for which three SDPs are incompatible, and only one SDP (JXME) offers a perfect adequacy. When taking into account the physical radio interface by combining the operating modes, WiFi is the interface with the greatest number of perfect adequacy cases. This interface is thus the least expensive to carry out discovery with a large set of SDPs. However, we can also notice that WiFi does not authorize discovery with Bluetooth-SDP at all. From this point of view, Bluetooth is the only radio interface that allows discovery with all the SDPs, since 3 out of its 4 configuration modes are compatible with the full set of SDPs, even if the adequacy is not perfect. However, the default Bluetooth mode (piconet) is the interface with the greatest number of incompatibilities since only Bluetooth SDP and JXME (with a weak adequacy for the latter) function. Moreover, piconet is the default Bluetooth mode and switching the Bluetooth interface to one of the 3 other modes rather depends on the availability and capabilities of the other devices in the vicinity than on the client device itself. Finally, it is noticeable that results outlined in this section do not always match our previous assessment related to the energy-efficiency of the radio interface with respect to service discovery. Thus, globally energy-efficient service discovery in multi-radio networking environments must take into account and balance both the energy efficiency and the adequacy of the SDPs with respect to the radio interfaces to be used for running the protocols. The next section introduces such an energy-efficient service discovery, which is specifically aimed at supporting context awareness in multi-radio pervasive environments.

4 Energy-efficient SDP-based Context Sensing in Multi-radio Networks

Context sensing is a key requirement for pervasive computing environments, as this allows adapting service provisioning according to both the physical environment and available computing resources. Context sensing specifically lies in the sensing of context information, which is further aggregated to enable reasoning according to the application specifics [15]. An effective approach to the sensing/discovery of context then lies in discovering the networked resources available in the environment where services get provisioned, as these resources form the physical and computing system context. In a multi-radio environment, discovery of the networked resources must take place on all the available networks. Thus, the configuration of a multi-radio device for context sensing must be worked out with great care, to not waste energy with unusable, energy-inefficient or inadequate combination of SDPs with radio interfaces.

In the following, we introduce a middleware solution to SDP-based energy-optimized context sensing in multi-radio networks, together with its prototype implementation. Building upon our assessment of the energy requirement of multi-radio based service discovery (§ 2) and our analysis of the adequacy of legacy SDPs against B3G networks (§ 3), the proposed middleware in particular implements an algorithm that realizes effective service discovery in multi-radio networks by accounting for both network availability and resource consumption in the discovery process. Specifically, our algorithm supports the exhaustive discovery of networked services on multi-radio devices, by exploiting the various embedded network interfaces and interfacing with the various SDPs that are now available at a location. The algorithm further minimizes resource-consumption on the device and offers response time comparable to that of the underlying SDPs.

4.1 SDP-based context sensing over the multi-radio network

Figure 3 depicts the main components of the proposed middleware solution for energy-efficient SDP-based context sensing in multi-radio networks. Components are deployed on multi-radio devices and are layered on top of some legacy networked operating system, while our current prototype is more specifically implemented upon the compact .Net framework.

The lowest layer component is the *Multi Radio Network Manager (MR_Manager)*, which manages multi-radio networking on the device. MR_Manager implements relevant network-related functionalities, such as switching and configuring radio interfaces. MR_Manager basically offers the upper components a common API, aggregating the underlying APIs that manage the embedded radio interfaces. MR_Manager further deals with overlapping coverage of the radio interfaces. Indeed, one interface may give access to part of the network coverage of another interface (e.g., a WiFi interface in the infrastructure mode with gateway gives access to both LAN and WAN, whereas the GPRS interface only gives access to the WAN). When the network coverage of two radio interfaces is identical (i.e., both interfaces have the same network *scope*), run-

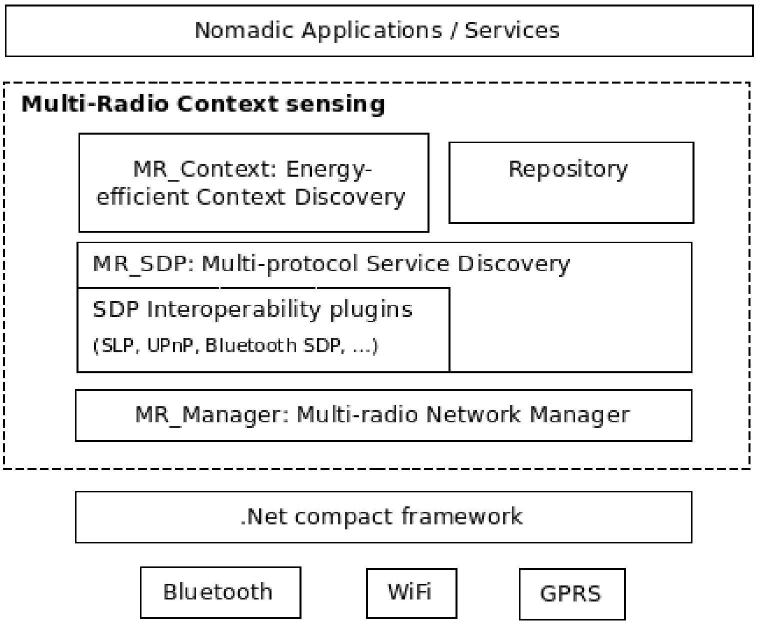

Nomadic Applications / Services
Multi-Radio Context sensing
MR_Context: Energy-efficient Context Discovery
Repository
MR_SDP: Multi-protocol Service Discovery
SDP Interoperability plugins
(SLP, UPnP, Bluetooth SDP, ...)
MR_Manager: Multi-radio Network Manager
.Net compact framework
Bluetooth
WiFi
GPRS

Fig. 3. SDP-based Context Sensing

ning the same SDP over the two interfaces leads to discover the very same set of services. It is then useless and resource inefficient to run a given SDP via both interfaces. Therefore, an application aware of multi-radio networking shall have the ability to choose, for a given interaction, which interface to use among relevant/eligible ones. The role of the MR_Manager component is specifically to find such intersections in network coverage and group network interfaces that have identical *scopes*. Thus, MR_Manager provides the list of every reachable network *scope* and, for each scope, the eligible network interfaces and configurations.

Given multi-radio networking, context sensing lies in the discovery of all the networked resources/services that are advertised using some (legacy) SDP run in the various reachable network scopes. This calls for *Multi-protocol Service Discovery* that bridges with legacy SDPs, as offered by the MR_SDP component. MR_SDP specifically integrates relevant interoperability plug-ins, each implementing a legacy service discovery protocol. Such an approach to multi-protocol service discovery has indeed been proven successful, as in particular detailed in [16,17].

The algorithm for effective context sensing in multi-radio networks is implemented in the *Energy-efficient Context Sensing* component (MR_Context). Building upon assessment of service discovery protocols in multi-radio networks, discussed in the previous sections, MR_Context is in charge of the orchestration of the MR_Manager and MR_SDP components to efficiently process context sensing, and stores service offers that have been discovered during the process in the *Repository* component. More specifically, MR_Context gets network-related information (e.g., availability and configuration of network interfaces and network scopes) from MR_Manager; and the list of plugins available for service discovery from MR_SDP. Given this information, MR_Context embeds the algorithmic to detect all the SDPs in use in the environment (§ 4.2). The lists of detected SDPs and of network scopes are then exploited to choose the configuration of radio interfaces to use for running the service discovery process in surrounding scopes (§ 4.3). Finally, MR_Context dynamically computes a resource-efficient configuration of the SDPs that are concurrently running on the same network interface (§ 4.4) during the service discovery process by, e.g., changing the discovery model used by the SDPs plugins or correlating their discovery cycles.

4.2 SDP monitoring

By identifying the network scopes reachable from the various radio interfaces within MR_Manager, the number of interfaces via which service discovery must take place to provide exhaustive context sensing may be reduced. Specifically, the SDPs running within one scope can be known via any of the radio interfaces giving access to that scope. Therefore, the detection of active SDPs implemented by MR_SDP takes as input the list of network scopes and associated radio interfaces, and operates dynamic SDP detection on one interface of each scope to output the list of SDPs running in each network scope.

Dynamic SDP detection at one interface uses the solution presented in [10], which basically works as follows. All the SDPs use a multicast group address and a UDP/TCP port that must and have been assigned by the Internet Assigned Numbers Authority (IANA). Thus, assigned ports and multicast group addresses are reserved, without any ambiguity, to only one type of use. These two properties form unique pairs, which may be interpreted as a permanent SDP identification tag. As an entity may subscribe to, and thus be a member of, several multicast groups simultaneously, we passively discover the SDPs in use in the environment by listening to the well-known SDP multicast groups. This is sufficient to provide simple but efficient environmental SDP detection, without additional traffic. Moreover, as we learn the SDPs that are currently used from both services' multicast announcements and clients' multicast service requests, the specific protocol of either the passive or active service discovery may be determined.

Figure 4 depicts the mechanism used to detect active and passive SDPs in a repository-less context. The monitor component embedded in MR_SDP, located at either the service client side or service provider side, joins both the SDP1 and SDP2 multicast groups and listens to the corresponding registered UDP/TCP ports. SDP1 and SDP2 are identified by their respective identification tag. However, SDP1 is based on an active discovery model. Hence, clients

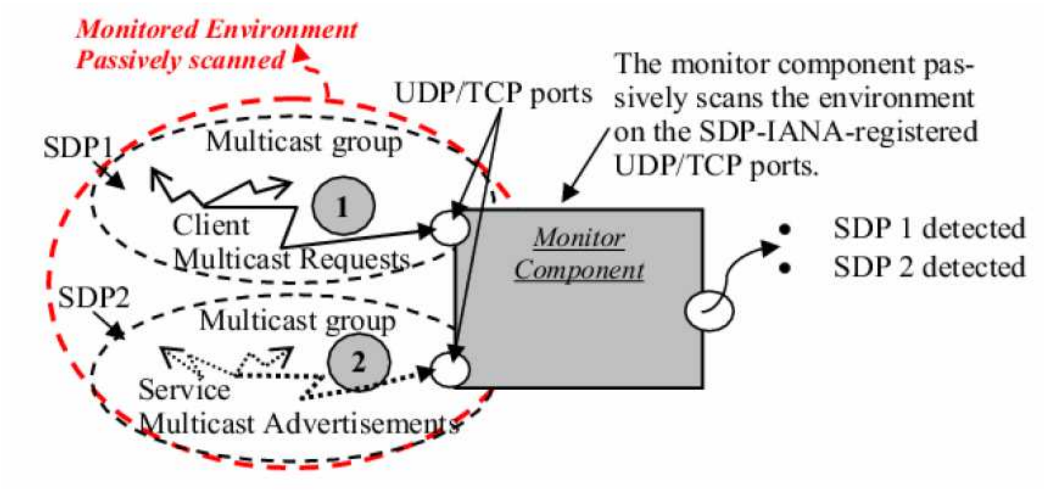
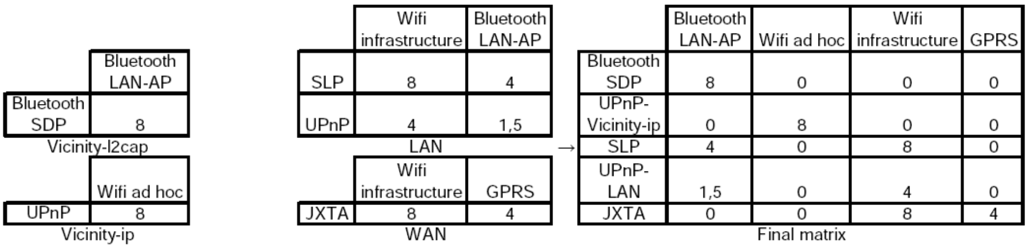


Fig. 4. Detection of active and passive SDPs

perform multicast requests to the SDP1 multicast group to discover services in their vicinity. The monitor component, as a member of the SDP1 multicast group, receives client requests and thus is able to detect the existence of SDP1 in the environment as data arrival on the SDP1-dedicated UDP/TCP port identifies the discovery protocol. Still, in Figure 4, SDP2 is based on a passive discovery model. So, services advertise themselves to the SDP2 multicast group to announce their existence to their vicinity. Once again, similarly to SDP1, as soon as data arrive at the SDP2-dedicated UDP/TCP port, the monitor component detects the SDP2 protocol. The monitor component is able to determine the current SDP(s) that is (are) used in the environment upon the arrival of the data at the monitored ports without doing any computation, data interpretation or data transformation. It does not matter which SDP model is used (i.e., active or passive) as the detection is not based on the data content but on the data existence at the specified UDP/TCP ports inside the corresponding groups.

The subscription and listening operations of all the SDPs we take into account (see § 3), except Bluetooth-SDP, are solely IP features. Thus, implementing SDP monitoring for these SDPs is easy. Obviously, a simple static correspondence table between the IANA-registered permanent ports and their associated SDP must be maintained. Hence, the SDP detection only depends on which port where raw data arrived. Therefore, the SDP detection cost is reduced to a minimum. The only non-IP based SDP of this study, Bluetooth-SDP, does not need to be detected, as it is a mandatory part of the Bluetooth standards and must thus be implemented by any Bluetooth stack. Therefore, the existence of devices in the Bluetooth piconet vicinity implies the use of Bluetooth-SDP.

During the initialization of the SDP-based context sensing algorithm, a monitoring stage precedes the discovery phase. This gives the MR_Context com-



	Bluetooth LAN-AP
Bluetooth SDP	8

Vicinity-l2cap

	Wifi ad hoc
UPnP	8

Vicinity-ip

	Wifi infrastructure	Bluetooth LAN-AP
SLP	8	4
UPnP	4	1,5

LAN

	Wifi infrastructure	GPRS
JXTA	8	4

WAN

→

	Bluetooth LAN-AP	Wifi ad hoc	Wifi infrastructure	GPRS
Bluetooth SDP	8	0	0	0
UPnP-Vicinity-ip	0	8	0	0
SLP	4	0	8	0
UPnP-LAN	1,5	0	4	0
JXTA	0	0	8	4

Final matrix

Fig. 5. Example of per-scope adequacy matrices, and resulting global matrix

ponent an accurate view of the SDPs run in the environment. Due to the dynamic nature of the network environment, the environment is continuously monitored for active SDPs while an interface is active. Meanwhile, in specific situations (e.g., when a network interface is never used in the discovery process), MR_Context may force the activation of a network interface from time to time in order to operate SDP monitoring. This SDP monitoring scheme allows MR_Context to maintain the list of SDPs running in each network scope.

4.3 Selection of a globally-efficient service discovery configuration

Exhaustive context sensing requires operating service discovery with all the detected SDPs at all the reachable network scopes, in order to discover all the available services. The aforementioned MR_Manager and SDP monitoring components provide a list of relevant network scopes, and a list of SDPs detected inside each scope. By construction, as stated in § 4.1, the same services will be discovered in a given network scope, whichever relevant interface used is. Therefore, in order to obtain efficient and exhaustive context sensing, the most suited radio interface for each detected SDP may be chosen in each network scope, according to the outcome of our analysis presented in the previous section. Nevertheless, a simple concurrent use of the optimal interface for each active SDP may result in a non-optimal global discovery process, with respect to optimization criteria such as energy consumption. As an example, operating a specific SDP on the radio interface having the best adequacy may lead to switch this interface on, whereas choosing another radio interface, already in use by another SDP, would greatly optimize the global energy consumption in spite of a small decrease in the global adequacy. The algorithm we introduce below selects the most globally-efficient set of SDPs to be run for service discovery over the various embedded radio interfaces, with respect to both exhaustive context sensing and tradeoffs between adequacy and energy consumption.

The SDP/interface adequacy matrix defined in the previous section (see Table 14) shall be refined by removing the SDPs that have not been detected by the SDP monitoring component, and the radio interfaces that the given

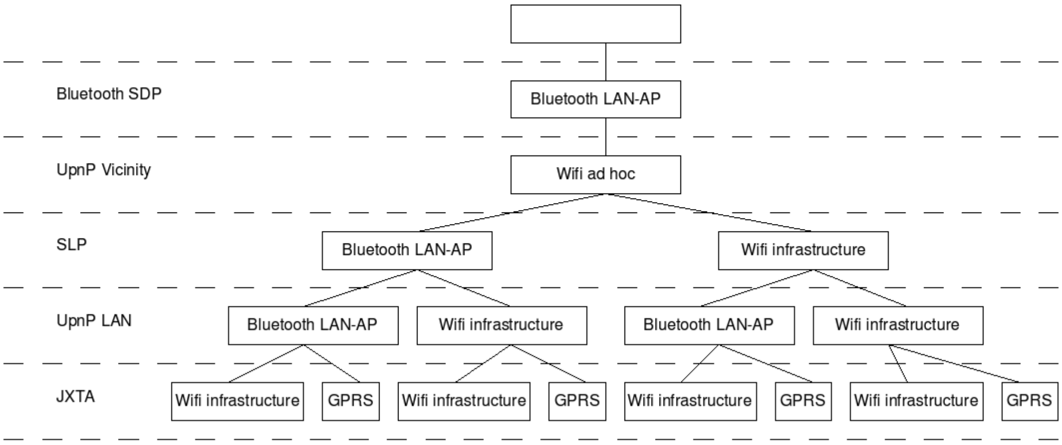


Fig. 6. Example tree representation of SDPs/radio interfaces configurations

device does not embed. Further, some of the SDP/radio interface combinations of the matrix are irrelevant and shall be removed (e.g., when the given radio interface gives access to network scopes where the given SDP has not been detected). Thus, a partial matrix is generated for each detected network scope. Lines are limited to the detected SDPs in the scope, and columns only refer to radio interfaces belonging to the network scope. These matrices are then grouped into one global matrix by aggregating the radio interfaces and concatenating the SDPs (see Figure 5). That is, radio interfaces are referenced once even if they belong to several network scopes, whereas SDPs detected in several network scopes are seen as two different *scoped* SDPs. Combination values are reported as is, and void combinations are assigned a value of 0. In favor of the method used to build the matrix, choosing one and only one non-null value for each scoped SDP (i.e., each line of the matrix) ensures obtaining an exhaustive and redundancy-less configuration for context sensing.

The next step aims at taking into account the energy consumption of the radio interfaces. As power consumption is dependent of the global configuration of the device, it cannot be directly referenced in the matrix. Indeed, using a radio interface only increases the global consumption if the interface is not already in use. We use a tree representation where each path, from the root node to a leaf, represents an effective configuration. Each line of the matrix represents one level of the tree, and all the non-null values of the matrix are the children of each of the non-null values of the line above. An example of a tree representation for the matrix of Figure 5 is sketched in Figure 6. Edges are further weighed with values taking into account our criteria: adequacy of the SDP against the relevant radio interface, as assessed in Section 3, and a consumption overhead value. This overhead value is defined as the energy consumption of the interface to be used (see § 2) if the interface has never been used along this path of the tree, and 0 otherwise. Finally, leaf nodes are scored with the global evaluation of the distribution defined by the mobile mean of the sum of the consumption values and the mean of the adequacy values, i.e.:

$$f(configuration) = \alpha \times \overline{adequacy} - (1 - \alpha) \times \sum consumption$$

with $\alpha \in [0..1]$

The above variable α represents the relative weight of the adequacy against the energy consumption. Therefore, the bigger α is, the more discriminating adequacy is. This formula also allows both extremes, as when $\alpha = 0$ adequacy values are discarded and when $\alpha = 1$ consumption values are useless. Henceforth, tree sorting algorithms have all the necessary elements to find the most efficient distribution of SDPs on the radio interfaces. In our implementation, the α value may be easily changed as it should be tweaked according to the environment in order to get efficient operation.

4.4 SDP combination and scheduling

Once the optimal global configuration has been identified, specific optimizations of the configuration of the SDPs may be realized by taking into account each radio interface separately, and combining the SDPs to be processed on the same interface energy-efficiently. Specifically, it is possible to maintain context sensing over time by only using active discovery with every SDP. This technique is the most energy efficient when the lengths of the periods of all the SDPs are correlated. That is, radio interfaces remain switched off all the time, except during short time laps when discovery requests are processed all at one time. Nevertheless, this energy optimization scheme may only be applied when no other application operates on the radio interfaces, as when in use the interfaces cannot be switched off. When an interface has to be kept on, the SDP should be switched to the passive model, as it does not induce any overhead in terms of power consumption, offers a precise knowledge of the context and even decreases the volume of discovery communication.

The SDP combination algorithm implemented in MR_Context is sketched in Figure 7. This algorithm takes care of the choice of the discovery model according to the other SDPs in use, eventually computes the correlated value of the period, and manages these parameters over time. Whatever the discovery model chosen will be, all the sequences begin with a discovery request in order to get a snapshot of the context at time 0. Therefore, when a new SDP is to be used to provision context, the first step is to send an active discovery request. The algorithm then verifies if the passive discovery model has been chosen for another SDP already in use, else inquires the radio manager component to know if the radio interface serves another application. The radio manager measures the amount of data transferred on the interface while the algorithm does not access it, and tests if it is allowed to switch off the inter-

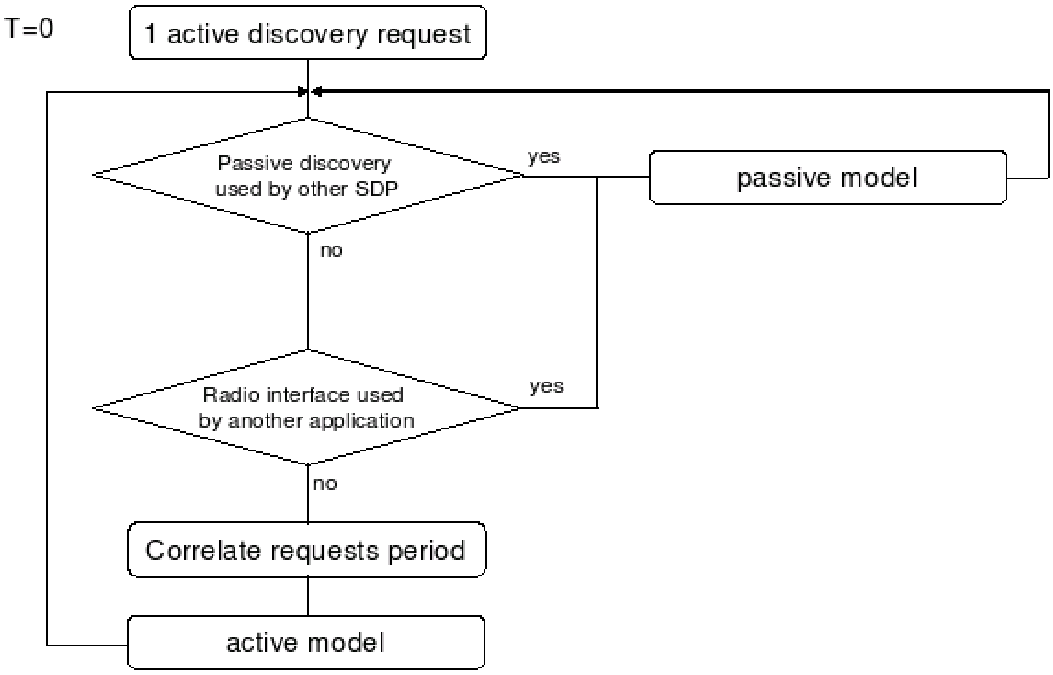

T=0
1 active discovery request
Passive discovery used by other SDP
yes
passive model
no
Radio interface used by another application
yes
no
Correlate requests period
active model

Fig. 7. Flowchart of the SDPs management for context sensing algorithm

face. If the passive discovery model is already used, or the interface is used by another application, the interface cannot be switched off and thus a passive discovery model is chosen. Otherwise, the active discovery model is preferred and, in order to optimize the energy consumption, the algorithm computes the correlated interval length with respect to the other SDPs in use. The upper and lower time limits parameters are hard coded by the SDP plugin developer, and the SDP plugin interface offers mandatory methods to get and set the current interval. This algorithm reiterates over time in order to take care of the changes of external factors such as usage of radio interfaces by other applications.

5 Conclusions

B3G networks combine multiple wireless networking technologies in order to benefit from their respective advantages and specificities. The increase in computing and communication capacities of portable devices, as well as their mass marketing, allows envisaging the widespread deployment of multi-radio network pervasive environments. The emergence of such ambient networks opens new challenges and issues in the development and deployment of distributed systems. A user having a multi-radio capable device benefits from such an ambient network by increasing the perimeter of reachable service providers, at the expense of a higher network management complexity. This complexity, induced by the heterogeneity of the wireless technologies, should be hidden from the user and, to be effective, from the application (e.g., by a middleware solution). In a pervasive environment, services must first be discovered using SDPs. Several SDPs are currently in use, and each one has been designed with specific target network architecture and mode of operation. This leads to another important heterogeneity, which must be taken into account. This article presents two studies needed for multi-radio service discovery in order to exploit the various networks while overcoming SDPs heterogeneity. These two studies assess the two main criteria to be taken into account in order to optimize multi-radio service discovery: (i) energy saving, which is a major issue for battery-powered devices, and (ii) the adequacy of SDPs against multi-radio networks. To the best of our knowledge, no other work about energy consumption of radio interfaces has taken into account the specific case of service discovery, and SDPs have never been assessed with multi-radio networking in mind.

Our first study, which measures the power consumption of wireless interfaces during service discovery and access, has shown that the most appropriate case would consist in discovering services via the Bluetooth interface, and accessing them via the WiFi interface. However, this perfect case is not always feasible as: (i) both connectivities are not always available, (ii) when they exist, they do not always reach the same networks and devices, (iii) most of the SDPs have been designed with one networking technology or architecture in mind and may be very inefficient on another one. Thus, the second study takes into account the specifics of the main SDPs and discusses how they behave when operated on each wireless interface. In order to transpose these qualitative observations into parameters usable by an adaptive algorithm, we inferred quantitative values reflecting the adequacy of use of each SDP against each radio interface. These results have further been exploited for the development of an energy-efficient algorithm for context sensing in multi-radio networking environments. This algorithm has been implemented into a middleware prototype solution for multi-radio handheld devices [18], which integrates an adaptive service discovery effectively exploiting the various networks and SDPs found in the pervasive environment, while hiding the network complexity from the application and user. The proposed solution continuously provisions services and stores a list of the retrieved offers on the client in order to minimize the response time of the client's discovery requests, and also takes care of updating the local knowledge over time to reflect the dynamics in the services offer.

This article has specifically focused on context sensing in service-oriented multi-radio networks, as we consider service-orientation to be a prime enabler of the pervasive computing vision, due to the openness of the computing environment that it enables. Our future work is aimed at extending our middleware-layer solution to also work out energy-efficient network configurations for service access in multi-radio networking environments. The trivial approach to making the network energy-efficient (i.e., choosing the most energy-efficient radio link for each service session) is not optimal as it does not account for the global energy-consumption performance of the network, considering both concurrent network access on the terminal and network usage on peers. We have shown in [19] that this problem is NP-hard and thus requires a careful approximation.

Acknowledgement

This work was supported by Alcatel-Lucent Reasearch & Innovation France

References

[1] B. A. Akyol, M. Fredette, A. W. Jackson, R. Krishnan, D. Mankins, C. Partridge, N. Shectman, G. D. Troxel, Smart office spaces, in: Proceedings of the Embedded Systems Workshop, 1999, pp. 23–28.

[2] T. Pering, V. Raghunathan, R. Want, Exploiting radio hierarchies for power-efficient wireless device discovery and connection setup, in: IEEE International Conference on VLSI Design (VLSID), 2005, pp. 774–779.

[3] P. Bahl, A. Adya, J. Padhye, A. Walman, Reconsidering wireless systems with multiple radios, SIGCOMM Comput. Commun. Rev. 34 (5) (2004) 39–46.

[4] J. Waldo, The Jini architecture for network-centric computing, Communications of the ACM 42 (7) (1999) 76–82.

[5] Universal Plug and Play Forum, Universal Plug and Play Device Architecture, http://www.upnp.org/download/UPnPDA10_20000613.htm (Mar. 2000).

[6] J. Lorchat, T. Noel, Power performance comparison of heterogeneous wireless network interfaces, in: Vehicular Technology Conference, 2003.

[7] W. Qadeer, T. S. Rosing, J. Ankcorn, V. Krishnan, G. D. Micheli, Heterogeneous wireless network management, in: PACS 2003, Vol. 3164, Lecture Notes in Computer Science, 2004, pp. 86–100.

[8] WI-FI Alliance, WMM power save for mobile and portable Wi-Fi certified devices, Tech. rep., WI-FI Alliance (dec 2005).

[9] E. Guttman, C. Perkins, J. Veizades, M. Day, Service Location Protocol, Version 2 , RFC 2608 (Jun. 1999).

[10] Y.-D. Bromberg, V. Issarny, INDISS: Interoperable discovery system for networked services, in: ACM/IFIP/USENIX 6th International Middleware Conference (Middleware'2005), 2005.

[11] L. Clement, A. Hately, C. von Riegen, T. Rogers, UDDI Version 3.0.2, http://uddi.org/pubs/uddi_v3.htm (2004).

[12] Bluetooth Specification Part E. Service Discovery Protocol (SDP), http://www.bluetooth.com (1999).

[13] T. Dengler, JXTA v2.0 protocols specification, Tech. rep., Sun Microsystems, http://spec.jxta.org/nonav/v1.0/docbook/JXTAProtocols.html (2001, last changed 2004).

[14] C. Arora, K. Pabla, JXTA J2ME Implementation Project, http://jxme.jxta.org (Feb. 2003).

[15] D. Fournier, S. B. Mokhtar, N. Georgantas, V. Issarny, Towards ad hoc contextual services for pervasive computing, in: International Workshop on Middleware for Service Oriented Computing (MW4SOC), 2006.

[16] P.-G. Raverdy, V. Issarny, R. Chibout, A. de La Chapelle, A multi-protocol approach to service discovery and access in pervasive environments, in: MOBIQUITOUS - The 3rd Annual International Conference on Mobile and Ubiquitous Systems: Networks and Services, 2006.

[17] P. Grace, G. S. Blair, S. Samuel, A reflective framework for discovery and interaction in heterogeneous mobile environments, ACM SIGMOBILE Mobile Computing and Communications Review 9 (1) (2005) 2–14, special section on Discovery and Interaction of Mobile Services.

[18] D. Charlet, R. Chibout, V. Issarny, Service discovery for multi-radio networks: Implementation, Tech. rep., INRIA, internal report available upon request (2006).

[19] M. Caporuscio, D. Charlet, V. Issarny, A. Navarra, Energetic performance of service-oriented multi-radio networks: Issues and perspectives, in: Sixth International Workshop on Software and Performance (WOSP 2007), 2007.