SIN – Service-Based Interconnected Networks

Filipe Costa and Rui M. Rocha

Instituto de Telecomunicações, Instituto Superior Técnico – Technical University of Lisbon,
Av. Prof. Dr. Cavaco Silva, 2744-016 Porto Salvo, Lisbon, Portugal
filipe.costa@ist.utl.pt, rui.rocha@lx.it.pt

Abstract. In an ideal world, service discovery protocols would be available across different wireless networks ensuring that most of the services would be searchable and accessible, anytime, everywhere. Yet, typical service discovery protocols were designed for specific scenarios and not conceived with user mobility in mind, where it would be possible to search for services through whatever access network might be available. To help increase the users' mobility, service-based interconnected networks (SIN) aims to develop an interoperability system between service discovery protocols in a wireless heterogeneous framework for existing protocols and devices. SIN provides the possibility to transparently search and find services across neighbour networks and through several protocols, resulting in gathering all services features. SIN was experimentally evaluated in a test-bed built to exercise a dynamic and pervasive service environment and used to prove the concept of service discovery interworking.

Keywords: Service discovery protocols, Wireless networks, Interoperability, Mobility.

1 Introduction

Through the last decade, a handful of wireless network technologies have raised much interest from research and standardization forums. Either at personal, local or metropolitan area levels there was a remarkable development of network architectures suitable for a multitude of scenarios thanks to their freedom, flexibility, mobility and ease of installation characteristics. However, the lack of resource and service interoperability between such networks, have doomed devices with access to only one wireless technology to be isolated from the rest, meaning that users are being restricted to resources and services available in their own networks.

Nowadays, the proliferation of devices combining Bluetooth and Wi-Fi technologies issues a challenge of how to be *always connected*. To overcome the divergences between different wireless technologies the concept of "Always Best Connected" (ABC [1]) has been proposed.

Aside from pure connectivity, what matters to users most is the availability of services they need across multiple access networks. Through the use of network services, users were able to improve their experience of all these technologies, since

K. Pentikousis et al. (Eds.): MONAMI 2011, LNICST 97, pp. 304–315, 2012.

different services please different needs. Therefore, a general access network would promote a better user experience throughout the use of network services spread across different networks.

Typical service discovery protocols (SDP) are dedicated to specific scenarios and adaptations to different ones are possible, although entailing typically a large effort. Even so, users are increasingly familiarized with providing and using services across different networks, being already conformed to the different environments where they run. Services are supposed to be helpful for users on a daily basis; yet with so many different scenarios and services available, this has become counter-productive for any one who has to invest countless hours in setting up services or searching for the correct services for his/her needs.

Being a known problem for quite some time now, manufactures and telco operators alliances have tried to come up with standards that will bring some order to this area, in the future. One example of such effort is DLNA [2], a standard proposed and accepted by its non-profit collaborative trade organisation and aggregating all available and future services with the same objectives as this work, although focused on the home networking arena. Nevertheless, legacy systems and equipment must not remain unconnected until they are discontinued and, finally, disappear. Besides not being exclusively targeting home networking scenarios, our focus is on a solution for legacy and existing service discovery protocols systems on IP networks, which are today the most common in the market.

Having realized the inadequacy of legacy and existing service discovery protocols, there was a need to develop a Service-based Interconnected Networks (SIN), a transparent service discovery protocol that successfully discovers services across wireless networks.

To do so, a general and *meta* service discovery protocol was designed, implemented and evaluated, in order to bridge all the relevant service discovery protocols. the Service-based Interconnected Networks Service Discovery Protocol (SIN-SDP) will allow every user to make use of any available service in all wireless neighbour networks. This innovation ensures that SIN-SDP will be useful for a wide range of scenarios. Indeed, not only in home networking can SIN-SDP prove its usefulness but also small business or public areas involving personal and local area networking such as malls, hotels, railway stations, convention centers, etc. may benefit from this service discovery interworking.

The reminder of this article is organized into four main sections. The following section gives an overview of which protocols are actually used, and discuss some proposals of protocol integration. Section three provides a detailed view of the proposed architecture. Section four delivers an objective validation of SIN-SDP's functionality, while section five draws some final conclusions and future work.

2 Related Work

Presently there are several service discovery protocols that provide users the chance to discover several services for a great range of applications. From the bucket of existent protocols, the ones that arise as important are the Bluetooth SDP (BSDP [3]), Bonjour through Zeroconf [4], UPnP [5] and SLP [6].

BSDP is considered the most dynamic of them all and its search engine is based on paging, where any request will be replied with all available services so the required one can be chosen; SLP was considered the more complete and flexible where its request result in service browsing or specific reply; Bonjour and UPnP are quite useful for their simplicity and market share, their search engine is based on device advertisement or specific queries.

The literature shows that some protocol integrations were already attempted. A new and minor service discovery protocol, Jini [7], was used to seek its integration with UPnP [8] and Bluetooth [9], respectively. These two approaches were significant to some design choices made for the SIN architecture.

The first one [8] uses proxies between the two separated service discovery protocol networks. These proxies would be used as protocol clients from one side and virtual services from the other. Yet, since the Jini protocol is quite rigid, the services must be coded in the proxies previously to its utilisation, making this approach as least flexible as it can possible be.

The second [9] integrates the Jini protocol into the Bluetooth stack as a new profile. This assures a smooth integration of this protocol in the Bluetooth network. However, the authors were not capable to create an integration as smooth in the opposite direction. This approach denotes the necessity of looking into protocol integration from both access network perspectives, simultaneously.

3 SIN Design

SIN proposes a new service discovery protocol layer that aggregates all language information about the others SDP's. This new layer would not interact directly with the client requester application, which will guarantee that it is fully transparent and do not have any impact on client applications. SIN-SDP is easily described as a protocol translator, trading messages with every other service discovery protocols, and making sure that service requests and replies could be listened and understood in many protocols at the same time.

The state-of-the-art on service discovery protocols revealed the following protocols as the relevant ones to be considered in this work. Bluetooth SDP for its wide use in Personal Area Network (PAN) scenarios being present in a massive majority of terminals. Service Location Protocol (SLP) for its flexibility and complete vision of service discovery. Apple's Bonjour and Microsoft's Universal Plug and Play (UPnP) for their complete vision in restrict environments; furthermore for their rising and upcoming use on numerous devices in homes and Small office and Home office (SoHo) environments.

System's transparency is one of the most important goals that can be guaranteed by using most of the common service discovery protocols and by the allocation of the system inside each network router to work as a proxy - a SIN Proxy. The strategy of placing the system within a proxy server in common network access routers will ensure full access to its network and the surrounding ones, which will allow each proxy to connect with services across neighbour networks.

In addition to transparency, the system must also be efficient. In fact, a system is not transparent if it takes too long to reply. Therefore, a balanced trade-off between network efficiency and network bandwidth performance should be accomplished in order to efficiently reply to users without wasting network resources that jeopardize the narrow bandwidth performance.

Since all successfully service searches in neighbour networks will maintain state in the request proxy, it is necessary to have a keep-alive system that manages all active proxy services. Periodically, a service request per foreign service in the proxy will be issued towards the original service network. If the same response is heard within the expected timeout it means the service is still alive and running smoothly; otherwise the service is down or fading, which means it will no longer be available through the proxy. The system will check all its foreign services availability every x seconds.

Along with the just mentioned key design goals the system will also have to intermediate protocol messages between one-to-many (*1-to-n*) SDP's. Clearly, when one client protocol application makes a search, the n neighbour networks will also have to make the same service search. Therefore, SIN may be thought as a set of language translators in multi-cultural communities, where each of them speaks only its native language. When someone inside its community addresses a question to it one translator will listen the question and then conveys it to all the other communities resident translators through a common language between them. Each and every community will be addressed with the same question but in a different language, thanks to a set of language translators communicating through a language common to all.

3.1 Architecture

The proposed system is based on a functional specification with 3 macro blocks: the listener, the translator and the adapter.

All these blocks organise themselves as Fig. 1 shows. In any iteration there are always 3 service discovery protocol languages. The first is the language of the protocol used in the original request, then the SIN-SDP's general language and, at last, the neighbours' protocol language.

The system's listener is responsible for handling the messages and corresponding local protocol language from the requester network. After apprehending a service request message, the translation block will translate it to a general service discovery language. Finally, the adapter will use this translated message to adapt its service description information to other neighbour protocols.

Each listening block will sniff for protocol multicasted messages in order to grasp all service query messages so every service request, in whatever network, can be translated to its neighbours. With the multicast capability, all local protocol messages will be heard and evaluated if they are relevant or not to neighbour networks. All protocol messages are considered irrelevant except for service request messages. This listener is also responsible for sending messages originated in SIN-SDP through the same multicast channel as the one being listened.

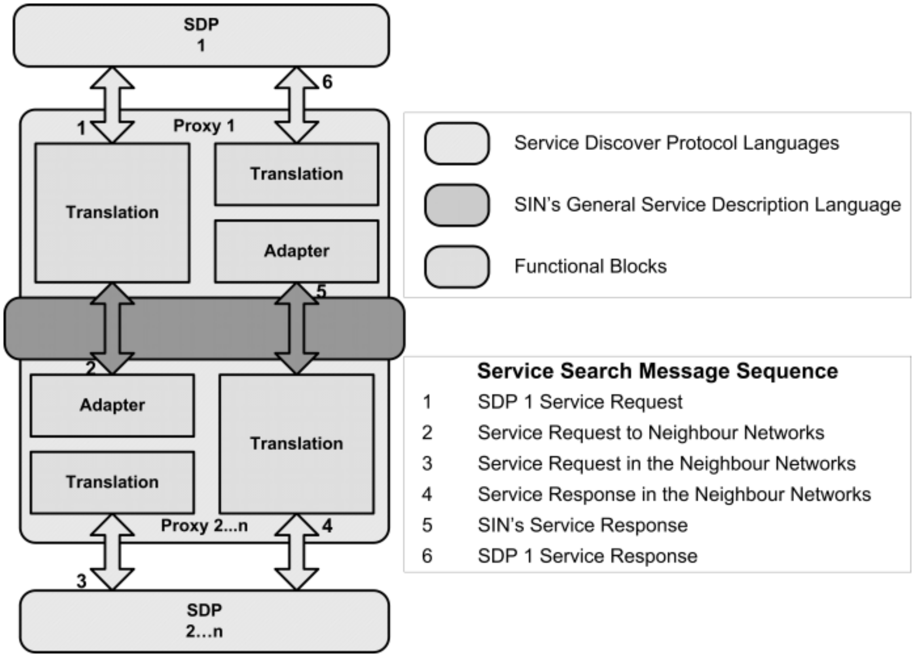
SDP
1
Proxy 1
1
6
Translation
Translation
Adapter
5
2
Adapter
Translation
Translation
Proxy 2...n
3
4
SDP
2...n
Service Discover Protocol Languages
SIN's General Service Description Language
Functional Blocks
Service Search Message Sequence
1 SDP 1 Service Request
2 Service Request to Neighbour Networks
3 Service Request in the Neighbour Networks
4 Service Response in the Neighbour Networks
5 SIN's Service Response
6 SDP 1 Service Response

Fig. 1. SIN-SDP Functional Architecture

The translation block will translate all local service discovery protocols messages into a general service description language. To do so, it is imperative that the system has full knowledge of the messages of each protocol and its content. If this is the case, all message attributes will be successfully translated to the SIN-SDP's language in order to be useful in other SDP's. Throughout these translations, the system should try to gather as much information about the description of the service as possible. This information set will have further use in specifying the service in a neighbour network, besides being used in the adaptation block.

The adapter is responsible for managing what is relevant for translation and what it is not. This block will change some of the attributes' names in order to use them as general service description information to enable neighbour proxies to provide them. It is mandatory that the vital information of the service like service name, type and location, is translated. As a last resort, the rest of the information in the original protocol, even if obsolete in foreign protocols, should be provided, unmodified for eventual future use. An example may be a service with 6 attributes: *friendlyName*, *deviceType*, *location*, *URL*, *modelName* and *modelDescription*. In this example, the first three are automatically translated to *serviceName*, *serviceType* and *IP:Port*, since they are vital. Then the *URL* and *modelDescription* will be adapted to *serviceURL* and *serviceDescription*, at last, the *modelName* will not be translated since it does not have any correspondence in other protocols.

In order to understand the dynamics of SIN-SDP, the behaviour of the system is exemplified by showing the message flow involved in a service discovery request, which is depicted in the message sequence chart shown in Fig. 2.

This message flow starts with a service request message from the requester application and followed by two actions from the proxy. The first one is to reply with the actual existing foreign services for that service type, and secondly it inquires the rest of the neighbour proxies for the requested type. Hence, the neighbour proxies will inquire their networks for that type of service. If a service is compliant with the requested service type, it will reply with a detailed message about itself. Therefore, a SIN's reply message is traded between proxies, the local service proxy and the original requester. At last, the first proxy will reply only when it has all the responses from all the neighbour proxies or when it reaches the timeout; in this case only the received proxy responses will be forwarded to the requester client.

It is believed that this system's architecture is the one that yields the best results considering the problem at hand and the objectives initially proposed.

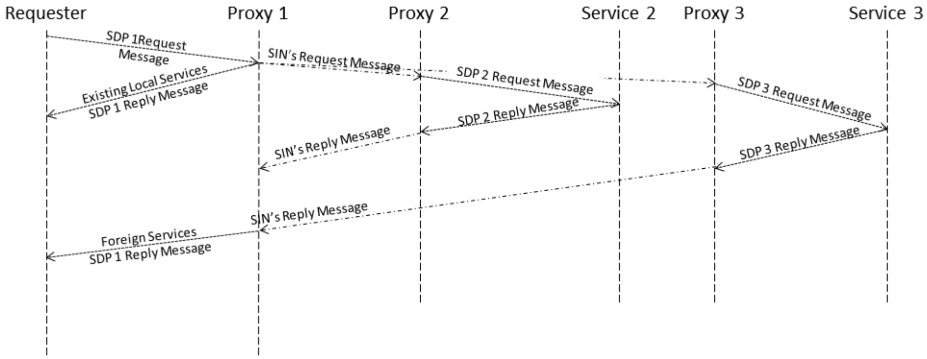

Fig. 2. Service Request and Reply MSC

4 SIN Test and Evaluation

As a proof-of-concept thus validating the viability of using this architecture in service discovery interworking, and simultaneously to assess its limits, SIN-SDP was deployed in a test-bed and exercised in a series of test scenarios. These tests intend mainly to evaluate the system's effectiveness and efficiency but also to assess its behaviour when facing dynamic service patterns that are typical of real-world pervasive environments. The test-bed was deployed as 4 different networks, one for each service discovery protocol. Since each network has its own access router, the SIN Proxy was deployed in these routers. Fig. 3 illustrates how the test-bed was structured and how it was able for each SIN Proxy to connect to any neighbour services.

The three test scenarios performed will be briefly discussed:

- *Effectiveness*: to show if the system can correctly find every neighbour network service as the one requested in any network. In this case and as a benchmark, it has been considered acceptable a system that retrieves correctly 9 out of 10 possible services. So, for the system to be effective it should find more than 90% of the available neighbour services.
- *Efficiency*: to assess if the system spends an acceptable amount of time to discover neighbour services or if it takes too long to discover them. If this is the case, it may not be acceptable for most users. As an empiric benchmark, the maximum amount of time acceptable by users when waiting for service responses was considered to be 30 seconds.
- *Pervasive service environment*: the idea was to evaluate the system under dynamic service conditions. This is supposed to be the final test of a system like the one being studied, since it is as challenging as it can be in a laboratory environment. This last test will be as close as possible to a real environment, such as a home or a SoHo setting. To do so, several services in different networks were deployed. The total number of available services reaches 46, spread throughout the aforementioned 4 networks. We have deployed a range of network services allowing the access to headsets, printers, media servers, cameras, ftp and ssh servers, etc. being 25 under Bluetooth's, 7 in UPnP's, 9 in Bonjour's and 5 in SLP's..

For all this to be possible, the system was modified in order to retrieve the necessary times of the request and reply messages.

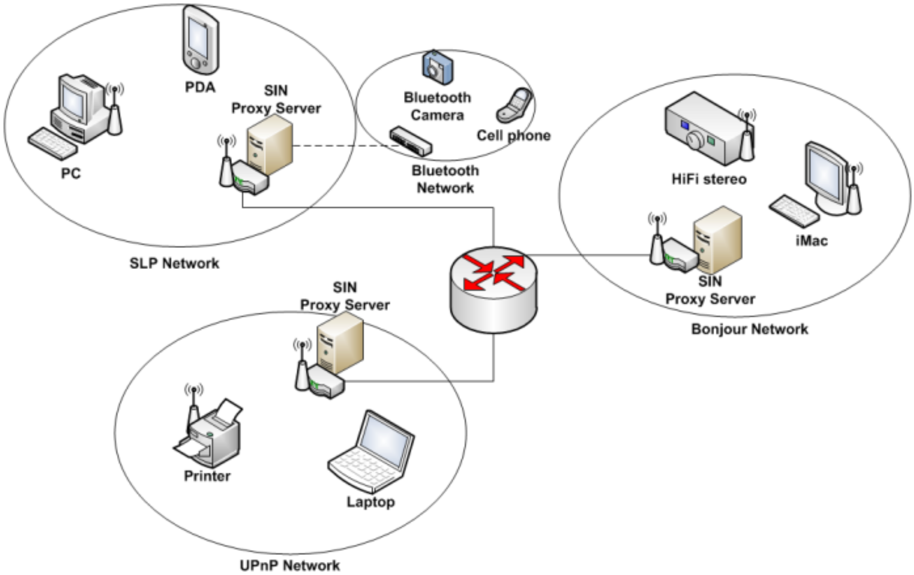

Fig. 3. Test-bed Network Environment

The first two test scenarios were deployed in the test-bed with only printer servers as services throughout all the networks rendering a homogeneous service pattern thus avoiding collateral effects due to the different nature of services that could influence the results. The Bluetooth network had 4 findable printer services in the rest of the networks, while SLP's and UPnP's were responsible for 3, and Bonjour had only 2 neighbour printers to be found.

4.1 Results

The results from the first set of tests indicate that the system is effective. The SIN's average effectiveness was 94.1%, being the most effective SIN Proxy the one located in the Bluetooth network. This Proxy found 95.2% of the available neighbour services. Moreover, average effectiveness for the Proxy in the SLP network was 93.6% while the Proxy in the Bonjour network and the one in the UPnP network featured 94.7% and 93.0%, respectively.

Since this test scenario had the purpose of evaluating the system's effectiveness, several printer services were located across all networks, as mentioned before. To attain the systems effectiveness, a user would make the same printer service request in all the networks in order to assess the effectiveness in several points of the test-bed.

Fig. 4 depicts the found services pattern for each network proxy throughout all the test iterations. It can be seen that the Bluetooth proxy is the one with less oscillation between service requests iterations, due to the paging search engine BSDP which is rather slow and faulty, affecting the effectiveness on the rest of the proxies that searches into its network.

It can be concluded that the system has passed this test successfully, since the average percentage of found available services was above the benchmark defined.

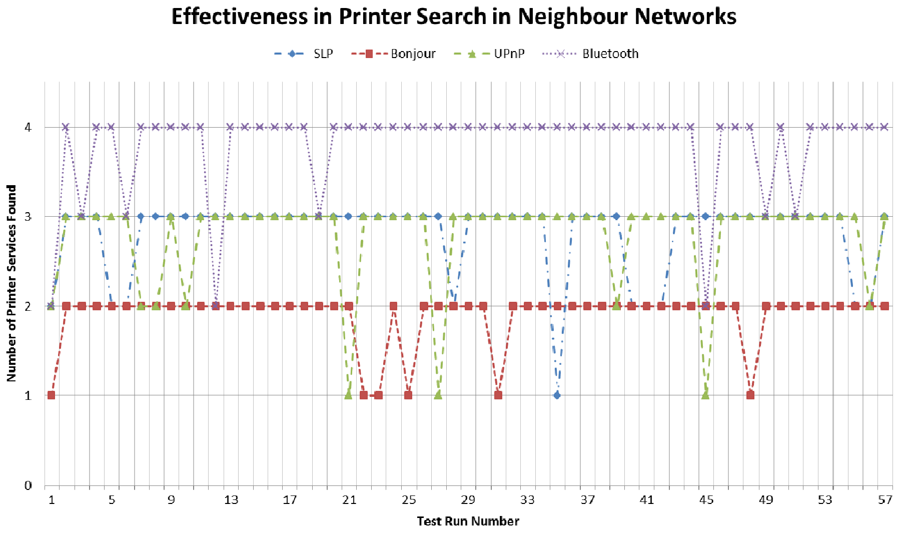

Fig. 4. Effectiveness Search Test Results

The results of the second set of tests were not as good as the first ones. The SIN's average efficiency was 18.63 seconds. This result was mainly supported on the excellent performance when searching from the Bluetooth's SIN Proxy. It had an average efficiency of 12.16 seconds, while the SLP's Proxy featured as much as 21.65 seconds for the same metric. The rest of the proxies were found to have an intermediate efficiency, being the Bonjour's and UPnP's Proxies characterized by 21.65 and 20.81 seconds, respectively.

As illustrated in Fig. 5, it can be seen that the first search attempt made by SLP, Bonjour and UPnP are rather slow than the following and well within the range of the highest search delays observed. Another obvious fact is the high variation on the delays observed for all Proxies but the one connected to the Bluetooth network. These results led to the following conclusions:

- The first search attempt consumes more time than consecutive ones. This was expected and happens because, initially, the Proxies have no knowledge about the neighbour services, as their internal caches are empty. This does not happen in the Bluetooth proxy, because the others SDP's are on average faster responders, rendering the impact of the Bluetooth's Proxy cache much less important than of the other networks.
- Searching from other proxies into the Bluetooth network does imply an additional delay. Bluetooth is slower to respond because of its searching mechanism; it queries devices for all of their services and, only then, the required one can be chosen. This particularity slows down service requests for all the Bluetooth neighbour proxies, being also an important contribution to the overall performance instability of the other SDP.
- The efficiency has been considered poor since the average is not as fast as expected, and it is clear that in several times the delay was superior to 30 seconds. This would invalidate those test iterations since they are superior to the established benchmark.

The last and final set of tests gave the possibility to draw several new conclusions. The pervasive service environment determines a challenging real scenario, where in few minutes several services enter and exit the networks. This pervasiveness creates a situation where in any given time period a different number of findable services can be observed in each of the four network islands of the test-bed. As such, the following question will automatically arise: will the system behave as effective and, at least, as efficient as in the previous tests?

The result of the average search effectiveness is presented in Fig. 6, and as it can be perceived the average effectiveness of the system has dropped from 94.1% down to 81.4%. This is caused by the numerous temporary services that made the system to stutter and, consequently, leaving some service search requests and replies unanswered.

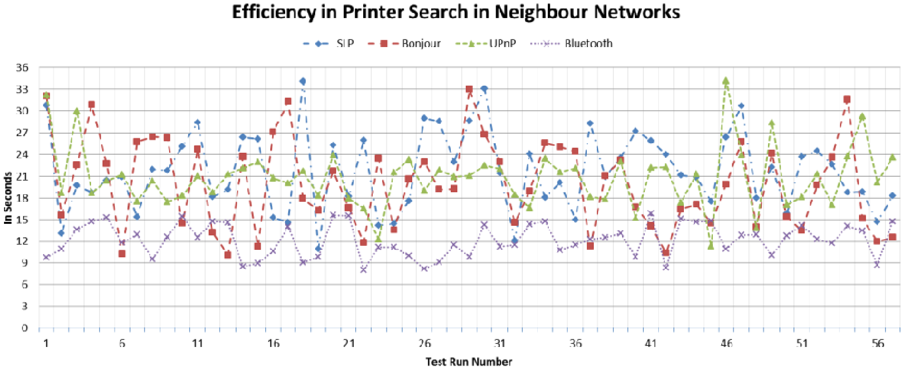

Fig. 5. Efficiency Search Test Results

The results of the efficiency of the system in this harsh environment are illustrated in Fig. 6. As it can be seen, the system performance has also dropped to an overall efficiency of 22.17 seconds. A closer look into the results (not shown) demonstrated that even with a lower effectiveness, there were some cases where the proxies found all available neighbour services. On the other hand, there were some few cases of effectiveness lower than 50%. In the first service request, the Bonjour's, UPnP's and SLP's proxies had a rate lower than 50% due to the empty cache problem.

Considering all the results above, it is clear that the system works for stationary and pervasive service settings. However, as it was expected, it has a better performance in the first than in the second case.

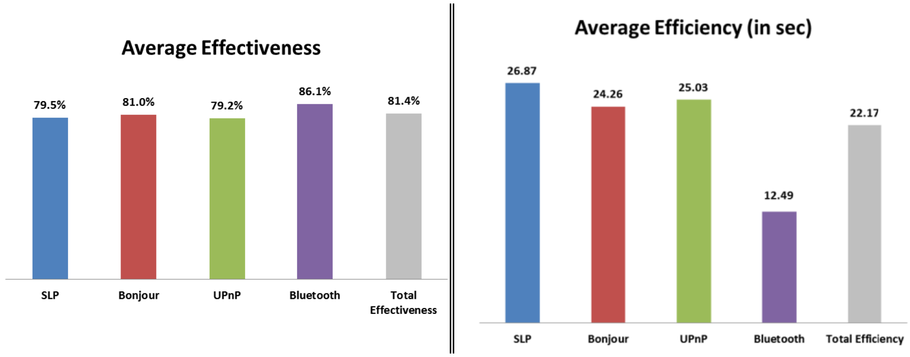

Fig. 6. Average Effectiveness and Efficiency in a Pervasive Service Environment

As a general conclusion of this last set of tests one can infer that SIN-SDP is ready to behave effectively and efficiently considering the existing constrains, such as Bluetooth service reply slowlyness. However, the reference implementation has not been fully prepared for some demanding environments, i.e., the ones that require more than just a functional solution.

5 Conclusions

In the last few years there has been some standardization efforts in order to propose protocols being able of seamlessly supporting different kinds of user equipment in a common framework capable of integrating access networks, services and devices. Such equipment has started to come to the market but still there is a long way to go.

As for the legacy equipment, users are not wiling to get ride of them overnight and as such, solutions to ease users' lives through interworking of existing heterogeneous technology is also a path worthwhile to follow.

In line with this vision, SIN-SDP has been proposed with such a system's architecture capable of supporting networked devices running most of all the major standard SDP's, namely Bluetooth, UPnP, Bonjour and SLP. As simple as it may seem, SIN-SDP has a translation and adaptation roles between each protocol language.

The SIN's architecture was validated through a given number of test scenarios set up over a specific test-bed. This allowed not only to prove the concept of service discovery interworking but also to assess the limits of such reference implementation.

From all the performed tests, the following conclusions were drawn: i) the system was able to find nearly all of the neighbour network services since the average effectiveness, in a stationary service environment with only one available service type in several devices, was 94.1%; ii) the system efficiency was found to be enough satisfactory for some users as the average delay was 18.63 seconds, while for some applications it may be considered too high; iii) even in a highly pervasive network the system was able to assure the correct service discovery, although not reaching the marks observed in the more favourable environment as the system prevailed in asserting 81.4% and 22.17 seconds of average service request effectiveness and efficiency, respectively.

Overall, SIN proved to be a valid approach to integrate service discovery protocols in some real network scenarios. Bringing auto-configuration into play, one can think of use cases where PAN and LAN-based services, independently from where they are being offered, can be used by inexperienced people effortlessly.

Anyhow, the system is at its toddler age as only a reference implementation has been achieved. Hence, it is natural that some future work is envisaged. The following topics are being considered as possible development paths for the future: i) to improve the service request response time, a proxy should dispatch, as soon as possible, any service reply that arrives to it, instead of aggregating all the service replies from all neighbours; ii) the proxy should work with any protocol, providing the same transparency between protocols in the same LAN, since in the SIN's reference implementation each proxy would deal with the local SDP and with the meta protocol, which is clearly not flexible; iii) it should be developed permission restrictions for information access, since it may be a problem when SIN Proxies from different owners try to federate and communicate freely with each other.

Acknowledgements. We would like to thank our colleagues at Group of Embedded networked Systems and Heterogeneous Networks (GEMS), Laboratory of Excellence in Mobility (LEMe) and Instituto Superior Técnico for all the help during the development and evaluation of this work. The blind reviewers of this paper also deserve our gratitude, for the many suggestions for improvement they provided us with.

References

[1] Gustafsson, E., Jonsson, A.: Ericsson Research, "Always Best Connected" (2003)

[2] Digital Living Network Alliance, DLNA Overview and Vision Whitepaper (2007)

[3] Bluetooth Special Interest Group, "Part E, Service Discovery Protocol". Specifications of the Bluetooth System, Version 1.1, pp. 331–392 (2001)

[4] IETF, Zero Configuration Networking, Zeroconf (November 2009)

[5] UPnP^{TM} Forum, "UPnP^{TM} Device Architecture 1.0". ISO/IEC 29341 (2008)

[6] Guttman, E.: Service Location Protocol: Automatic Discovery of IP Network Services. In: IEEE Internet Computing, ch. 4, vol. 3, pp. 71–80 (1999)

[7] Sun Mycrosystems, Inc., "Jini^{TM} Architectural Overview," Technical white paper (1999)

[8] Allard, J., Chinta, V., Gundala, S., Richard III, G.G.: Jini Meets UPnP: An Architecture for Jini/UPnP Interoperability. In: Proceedings of the 2003 Symposium on Applications and the Internet, SAINT 2003 (2003)

[9] Chau, O.S., Hui, P., Li, V.O.K.: An Architecture enabling Bluetooth/Jini Interoperability (2004)