Design of a Service Plug-and-Play Middleware Based on SLP and mSLP

Yu-Jun Zhang, Hsu-Yang Kung*, and Yu-Hua Tseng

Department of Management Information Systems
National Pingtung University of Science and Technology, Pingtung, Taiwan.

*Correspondence: kung@mail.npust.edu.tw

Abstract

The middleware can provide the services for mobile computing assists user and service provider to transparently access heterogeneous services. This paper proposes a f*lexible Service Plug-and-Play Middleware based on mobile computing environment* (SP2M). SP2M extends IETF *Service Location Protocol* (SLP) and *Mesh-enhanced Service Location Protocol* (mSLP) to provide scalable mobile services. This paper also proposes a novel *Service Plug-and-Play* (S-PnP) control procedure, which works as hardware Plug-and-Play service for legal users to provide flexible network application services over accessible heterogeneous service domains. The performance evaluation of the proposed S-PnP mechanism reveals that the scalability and successful rate of service discovery effectively are improved.

Key words:

Service-Aware, Flexible and Scalable, Service Location Protocol, Mesh-enhanced Service Location Protocol, Service Plug-and-Play.

1. Introduction

Mobile computing can change the way we use computing devices and broaden the network's applications enormously [1, 2]. For ideal mobile computing environments, heterogeneous devices and applications should integrate to work together to provide many accessible mobile services [3]. However, for such mobile services environments, it is not easy for users to know and access what types of service via heterogeneous networks and computing devices [4, 5]. Furthermore, there are various elements, which form any type of mobile application networks, for examples, the medical treatment network, early warning network, sensor network, home network etc. Although mobile computing environments have many kinds of application network, they still bring some big challenges to users and service providers [6]. Some of problems are as follows. (i) If we don't know address of the specific network service, we cannot use it. (ii) In most time, we don't totally understand properties of the service that we need. So we will spent much time searching and retrying the service. (iii) Because we need to spend a

large amount of time searching suitable for services, it is difficult to reach the instant demand [7]. (iv) Personalization service is not achieved easily due to the high computing complexity of customized service. (v) Users or service providers maybe move their logical or physical location, so we must consider hand-off issues. For example, user used mobile appliances to see the films, used desktop computer to see later. In this example, we have to consider the computing capability of the device to find out the suitable network service in time dorm a large scale network.

To achieve a flexible and scalable network service, this paper proposes a flexible service plug-and-play middleware infrastructure on mobile computing environment (SP2M) to assist users and service provider to easily find and access services transparently. Service Location Protocol (SLP) [8, 9] defines the service discovery procedure, which is a core component of the middleware. However, the SLP RFC does not describe how to invocate and extend services over multiple service domains. This paper proposes a novel Service Plug-and-Play (S-PnP) control procedure, which achieves the service discovery and invocation to make users obtain and use mobile services across multiple service domains on their own computing devices.

The rest of this study is organized as follows. Section 2 describes the research backgrounds and related works of SLP and mSLP. Section 3 describes the system design and implementation of the proposed FSM platform, and the performance evaluation of the proposed S-PnP mechanism. Section 4 gives conclusions.

2. Related Work

Service location protocol is developed by IETF [8]. The goal of SLP is to provide a flexible and scalable service discovery mechanism in network environments. It has three types of agent, they are user agent, service agent and directory agent.

(1) User agent (UA): It requests service discovery messages instead of user or application.

(2) Service agent (SA): It advertises its service information to user agents or directory agents instead of server side.

(3) Directory agent (DA): It provides a space which saves service register information from service agent.

There are two network architecture of SLP, the first is only exists UA and SA. This architecture fits discovering for smaller network size, e.g., a classroom or home. Fig. 1 depicts discovering steps in this architecture.

(1) Step 1. User tells his need service to user agent.

(2) Step 2. When UA gets this message, it will transmits this message in unicast or multicast style to SAs.

(3) Step 3. After SAs process this message, SA will transmit reply message to UA. Step 4. UA marshals those result messages and transforms them in a list to user.

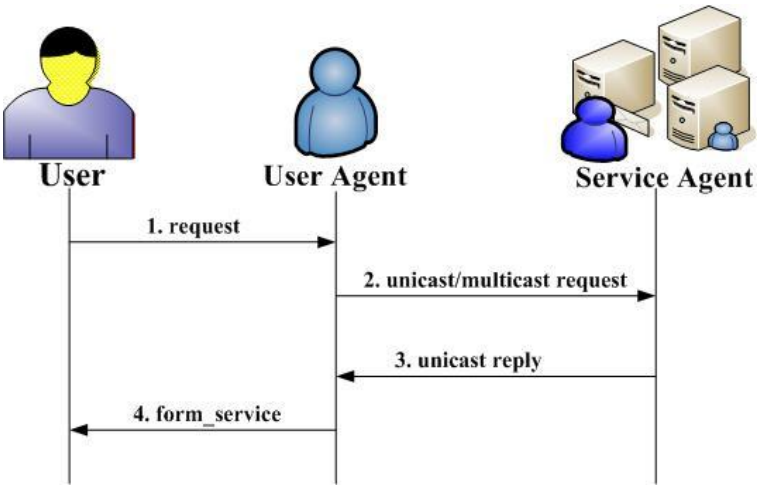

Figure 1. SLP network based on SA and UA.

The second architecture is composed by UA, SA, DA, which is suitable for big network environments, e.g., a campus network. SA registers its service information to DA. User discoveries service through UA, which sends service discovery message to DA. Fig. 2 depicts steps flow in this architecture.

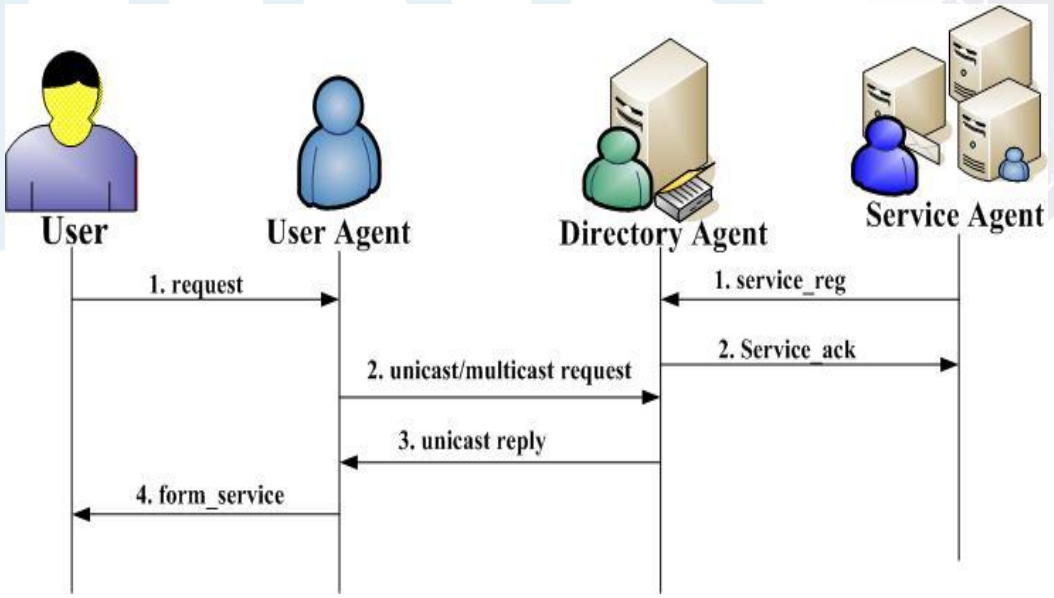

Figure 2. SLP network based on SA, UA, and DA

SLP provides three methods to detect DA exists in network or not. Details are described as follows.

(1) Active discovery. UA and SA sends DA discovery message on specific multicast address. After DA gets this message, it reply its information to them, Fig. 3 depicts the operation procedure.

(2) Passive discovery. DA sends its information periodically on specific multicast address. Fig. 4 depicts the operation procedure.

(3) DHCP server assignment: If any DHCP server exists in network, DHCP can

assign DA's address to UA or SA by the78th option. Fig. 5 depicts the operation procedure.

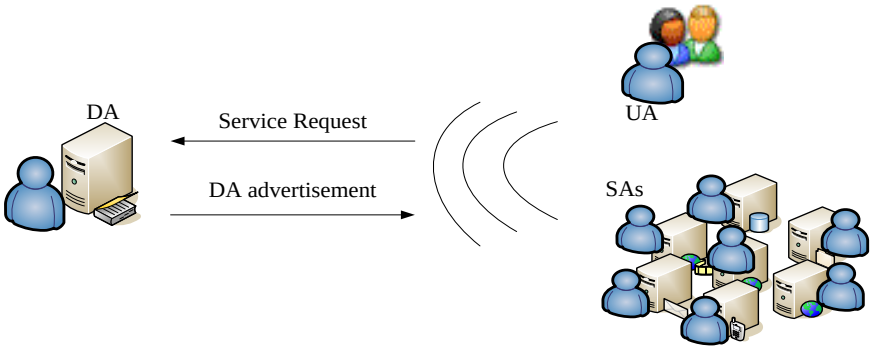

Figure 3. Active discovery for DA.

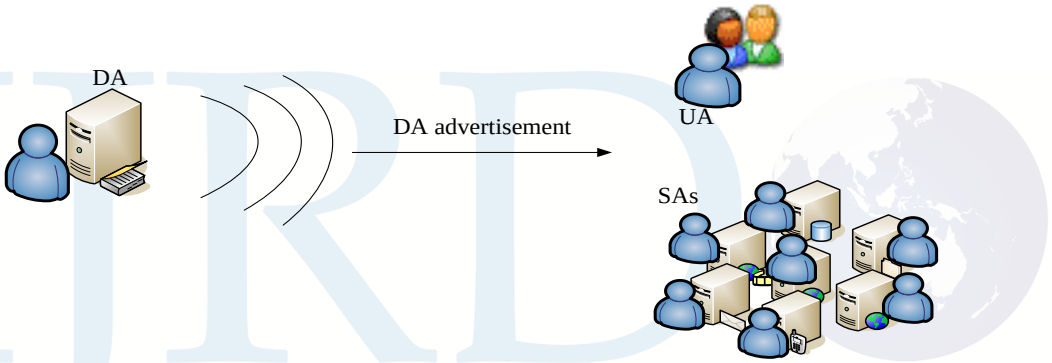

Figure 4. Passive discovery for DA.

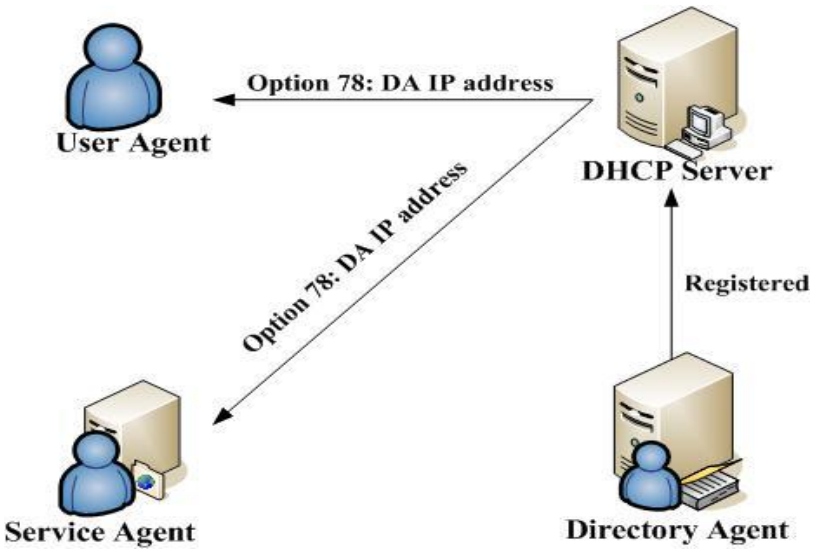

Figure 5. Service discovery via the DHCP 78th option.

The lifetime of every service information in the DA only keeps eighteen hours. SA has to register periodically. SLP also proposes three flexible and scalable methods to handle administration for the SLP network administrator. The three flexible and scalable methods are as follows. (i) Multiple DAs over network share load of each other. (ii) To setup scope attribute, if scope is setup, some services could be used by some department or specific domain in limited. (iii) To adopt the 78th option of DHCP lets some services which are used by some specific user or domain.

Fig. 6 depicts the function of scopes attributes. Users are with scope A attribute. Therefore, the users cannot discovery and access services of scope B. It is s critical problem.

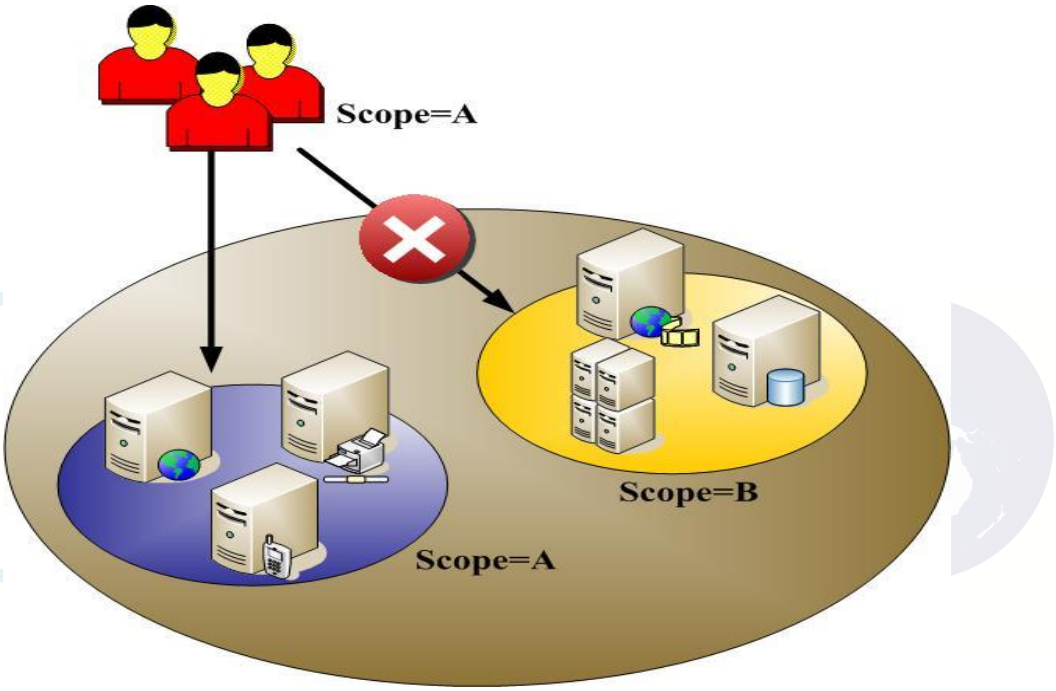

Figure 6. Functions of scope attributes.

The agreement of SLP does not define the communication methods between Das. SLP cannot correct the mistakes and change the service registration information between DAs. Therefore, Mesh-enhanced Service Location Protocol (mSLP) defines the control process of exchanging the service registration information between DAs [10, 11]. mSLP mainly improves drawbacks of SLP. For mSLP, DA cannot communicates and exchanges service information with each other. It defines a relation message which is called peering. If DA establishes peering relation successfully through peer message, DA would exchange service information with each other. Fig. 7 depicts the conceptual operations of mSLP.

DA_1 already establishes peering relation with DA_2. SA send service register message to DA_2. If this message can exchange with DA_2, DA_2 has to add the attribute, which is called mesh-forward. After UA sending service request to DA_1, DA_1 also gets service register information of DA_2. mSLP extends SLP service domain and increase its reliability and scalability.

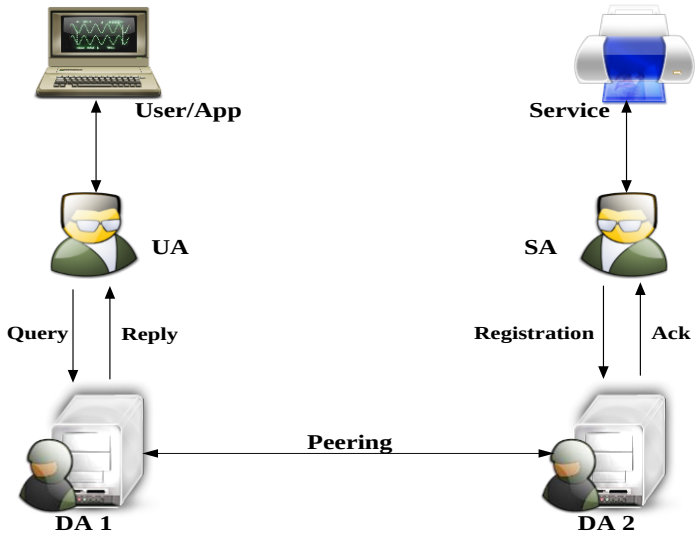

Figure 7. The conceptual operations of mSLP.

3. System Design and Implementation

This paper proposes a flexible Service Plug-and-Play Middleware (SP2M) that directed against the above problems. SP2M network contains three kinds of agents are SP2M directory agent (SP2M DA), SP2M service agent (SP2M SA) and SP2M user agent (SP2M UA) separately. Fig. 8 depicts the SP2M network architecture, which is based on the standard SLP and comprises the user agent (UA), directory agent (DA), and service agent (SA). Users can invoke their discovery services via the SP2M UA. The SP2M DA is responsible for service registration and advertisement. The SP2M SA provides service programs and contents.

3.1 System Design

Fig. 9 depicts the SP2M system components and architecture. SP2M system is divided into three layers.

(1) Service Aware Layer is responses for (i) the personalization control, which achieves the user-oriented tracking and service package management, and (ii) service discovery and plug-and-play functions, which are based on the SLP and Mesh-enhanced Service Location Protocol (mSLP) standards [11].

(2) Service Management Layer is responses for (i) resources adaptation control, which achieves the resource detection, collection, admission and scheduling, and (ii) hand-off control, which achieves the content, location and devices hand-off functions.

(3) Communication Layer achieves the transmission functionality of underline networks.

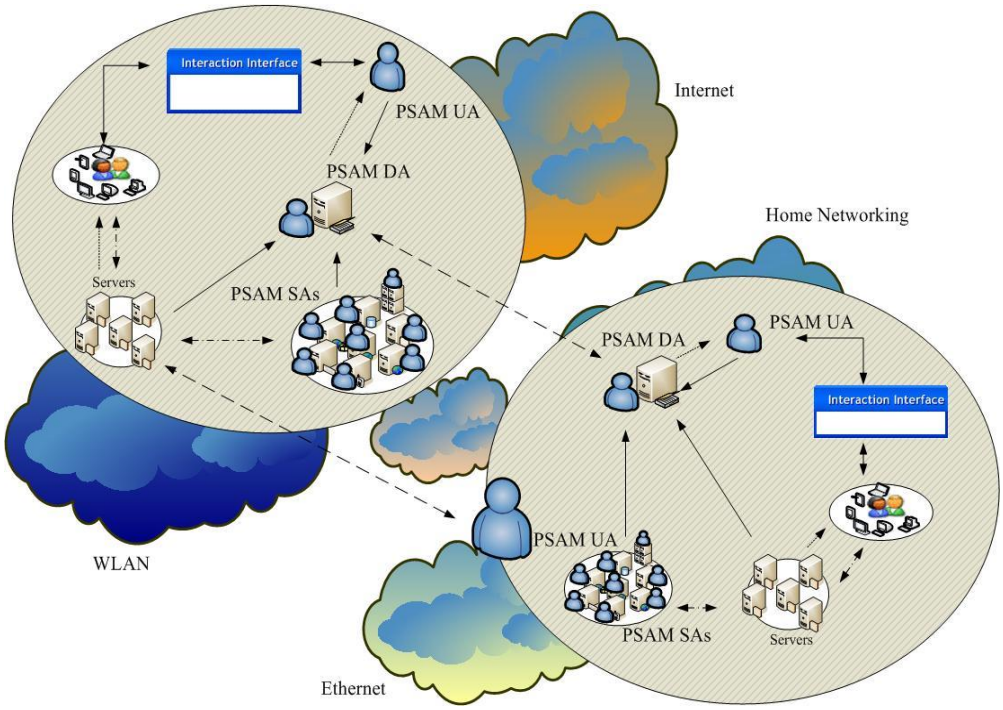

Figure 8. SP2M network architecture.

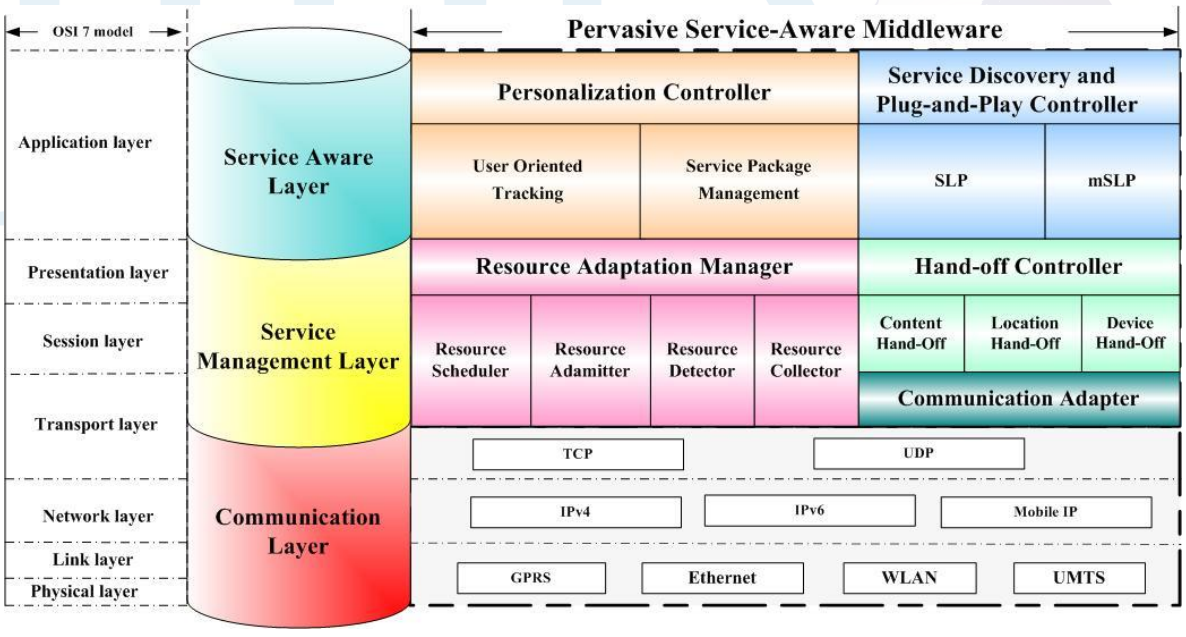

Figure 9. SP2M system components and architecture.

S-PnP mechanism adopts SLP, mSLP and Java Web Start technology [12] to provide flexible service accessibility and execution for heterogeneous users. Fig. 10 depicts the control procedure described as follows. (1) For intra-domain S-PnP: User sends SrvQuery message to the SP2M UA. SP2M UA executes SLP and mSLP procedures to find the requested service. When obtaining the requested service, SP2M UA utilizes the Java Web Start technology to upload service execution environment and execute the service. (2) For inter-domain S-PnP: If SP2M UA1 can't find the requested service from the SP2M DA1 in the same service domain A, the SP2M DA1 sends the peer list to

SP2M UA1 and the user chooses a suitable SP2M DA2 in the other domain B. If DA2 provides the requested service, DA2 sends the service list to the SP2M UA1 and plugs the requested service into SP2M DA1 for users of domain A.

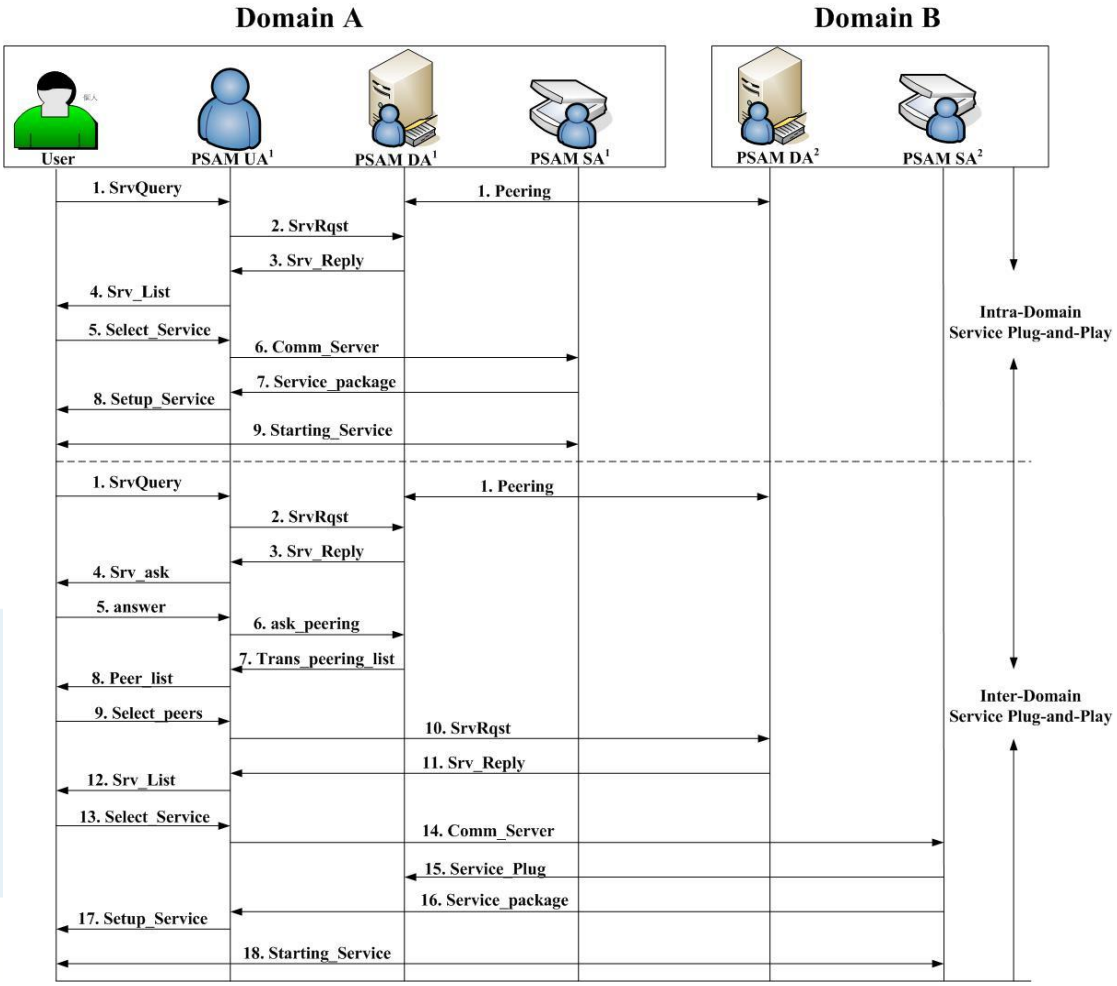

Figure 10. Service plug-and-play control procedure.

Fig. 11 depicts the system components of resource adaptation manager. The service provider specifies the resource policy, which is parsed by Policy parser of SP2M SA to extract the resource elements and policy criteria. Resource Detector monitors the specified resource element value, which includes the CPU, memory and thread utilization of the user client, and the network bandwidth status. Resource Collector transfers the resource elements value to Resource Scheduler. Resource scheduler is responsible for scheduling the requested resources. Resource Admitter achieves the resource admission and resource reservation control. Fig. 12 depicts the control message flow of the resource management control.

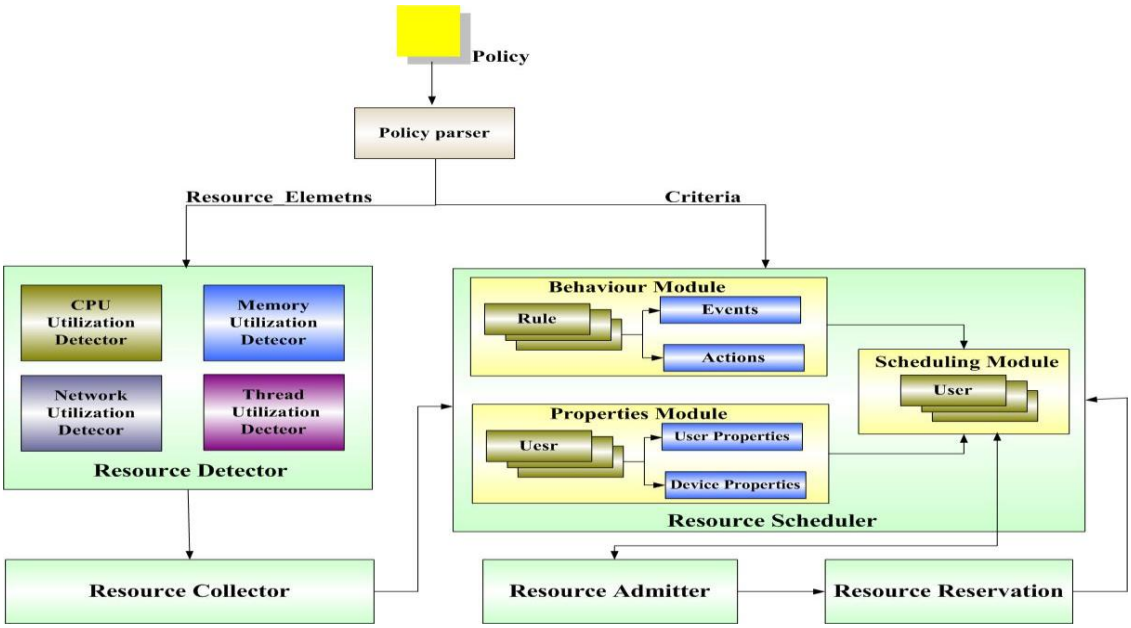

Figure 11. System components of resource adaptation manager.

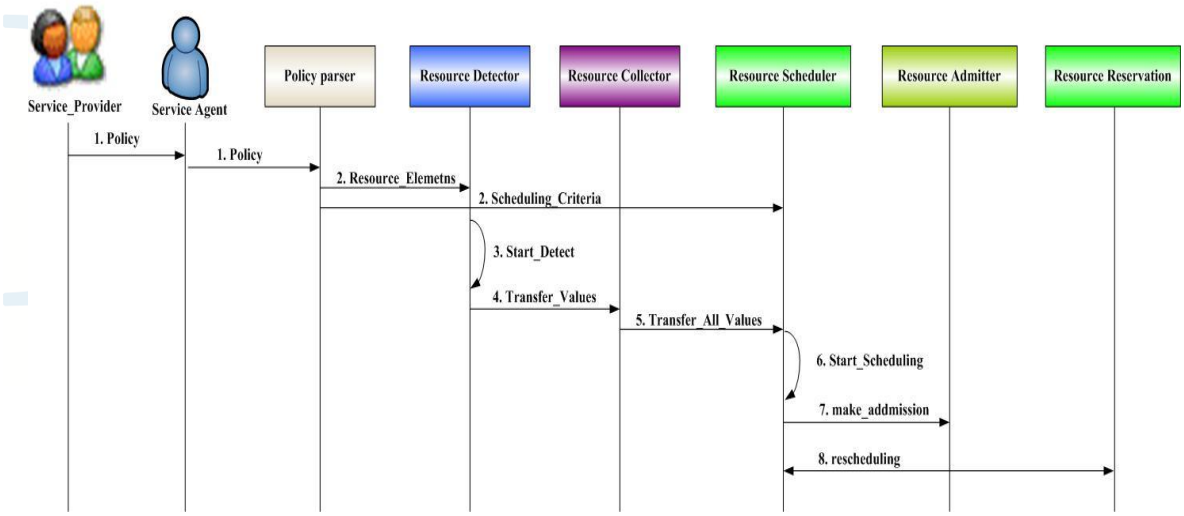

Figure 12. The control message flow of the resource management control.

3.2 System Usage and Performance Evaluation

Fig. 13 depicts the SP2M UA usage that a user requests the javaeditor service and enables the requested service. Fig. 14 depicts the SP2M SA usage that the service provider registers in a SLP DA and monitors resources management condition.

For the performance evaluation of the proposed SP2M, this study constructs a web style experimental environment and execute 300 times service discovery. Fig. 15 depicts the experimental results of S-PnP scalability. The performance of the proposed S-PnP mechanism is better than SLP with mSLP. It is because the time in the second section causes time increase in linear and S-PnP improves this computing complexity in a large and mobile computing environments.

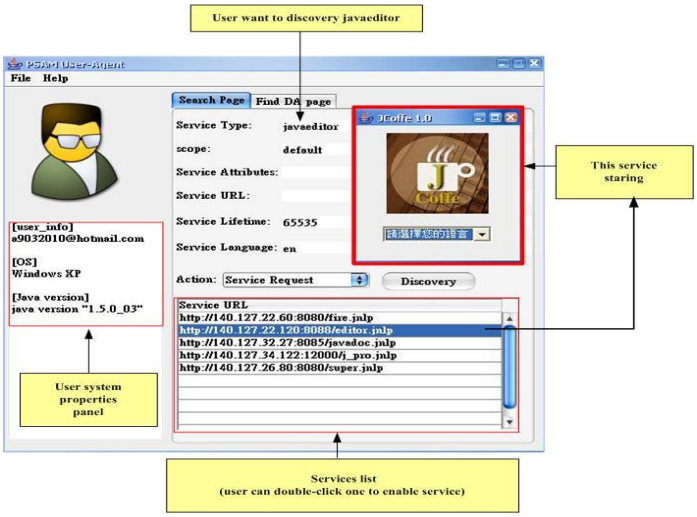

Figure 13. A usage example of SP2M user agent.

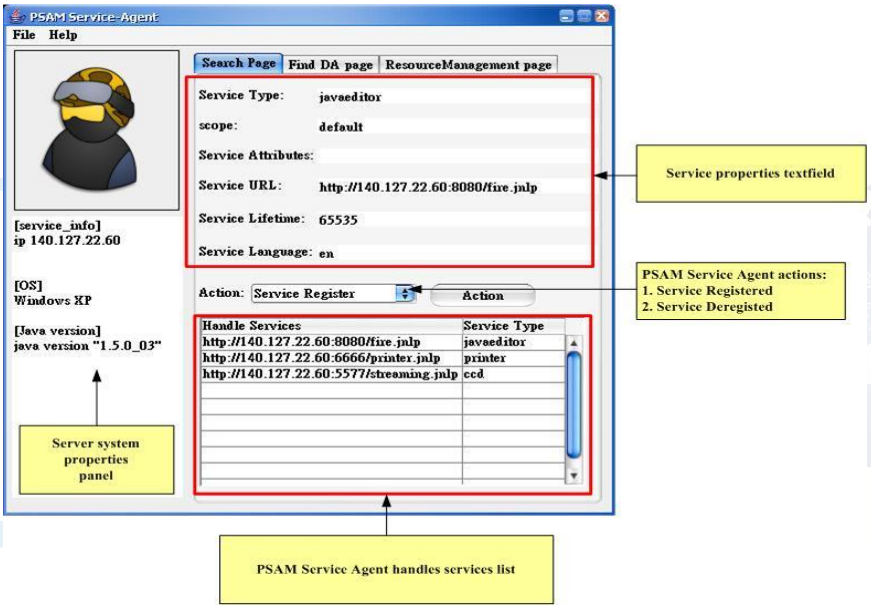

Figure 14. A usage example of SP2M service agent.

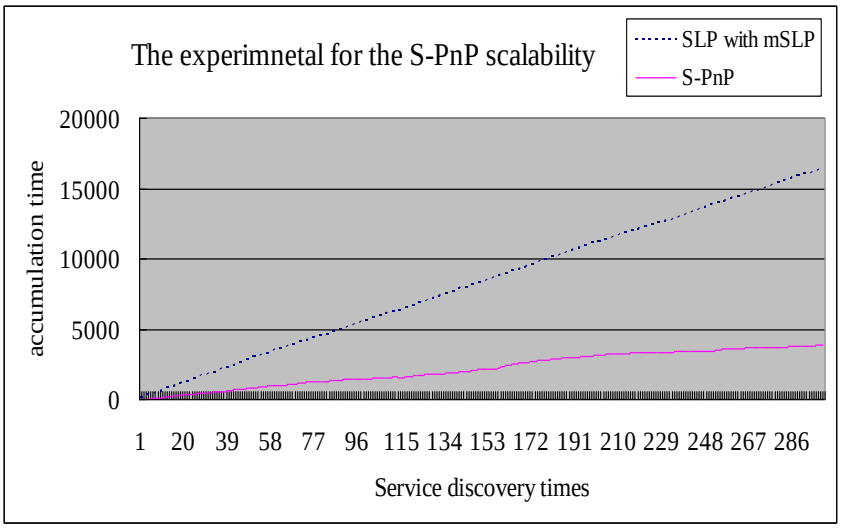

Figure 15. The performance evaluation for the S-PnP scalability.

For the traditional web service, as the number of web service discovery increases, the successful number of web service usually decreases. To evaluate the service successful rate of the proposed S-PnP mechanism. This study evaluates 16 kinds of different services of each SP2M DA. There are additional 4 services never exist in those SP2M DAs. The experimental performance evaluation are100 times in DA. Fig. 16 depicts this performance evaluation results of the propose S-PnP has apparently improvement in service discovery.

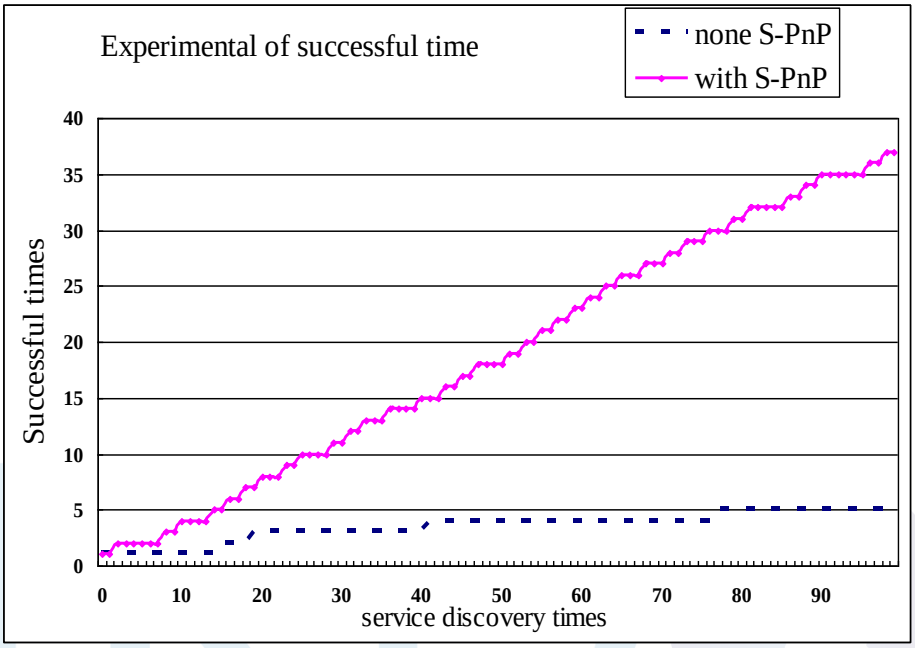

Figure 16. The service discovery evaluation for the S-PnP.

4. Conclusion

This paper proposed a SP2M which provides users with service-aware and personalization functions. The proposed novel S-PnP mechanism works as hardware Plug-and-Play service for legal users to provide flexible network application services over accessible heterogeneous service domains. The S-PnP mechanism increases the scalability and successful rate of service discovery, and effectively solves the performance degradation problems, which are caused by the mesh characteristic of the mSLP.

Acknowledgements

This work is partially supported by the National Science Council, Taiwan, R.O.C., under the grant No. MOST 105-2221-E-020-024.

References

[1] A. Chen, R. R. Muntz, S.Yuen, I. Locher, S. I. Park, and M. B. Srivastava, “A Support Infrastructure for Smart Kindergarten”, IEEE Pervasive Computing, Vol. 1, No. 2, pp. 49–57, 2002.

[2] K. Gai, M. Qiu, L. Tao, and Y. Zhu, "Intrusion detection techniques for mobile cloud computing in heterogeneous 5G", Security and Communication Networks, Vol. 9,

No. 16, pp. 3049-3058, 2016

[3] G. Tao, K.P. Hung, Q. Z. Da, “A Service-Oriented Middleware for Building Context-Aware Services”, Journal of Network and Computer Applications, Vol. 28, pp. 1–18, 2005.

[4] Y. Durmus and E. Onur, "Service Knowledge Discovery in Smart Machine Networks", Wireless Personal Communications, Vol. 81, No. 4, pp 1455–1480, 2015.

[5] Y. Wang, I. R. Chen, and D. C. Wang, "A Survey of Mobile Cloud Computing Applications: Perspectives and Challenges", Wireless Personal Communications, Vol. 80, No. 4, pp 1607–1623, 2015.

[6] L. Tawalbeh, Y. Jararweh, F. Ababneh, and F. Dosari, "Large scale cloudlets deployment for efficient mobile cloud computing", Journal of Networks, Vol. 10, No. 1, pp. 70-91,2015.

[7] N. Islam and Z. A. Shaikh, "Towards a Robust and Scalable Semantic Service Discovery Scheme for Mobile Ad hoc Network", Pakistan Journal of Engineering and Applied Sciences, Vol. 10, pp. 68-88, 2012.

[8] E. Guttman, C. Perkins, J. Veizades, and M. Day, “Service Location Protocol, Version 2”, IETF RFC 2608, 1999.

[9] M. Strasser and S. Albayrak, "Conceptual architecture for self-discovering in fragmented service systems", 7th International Conference on New Technologies, Mobility and Security (NTMS), pp. 27-29, July 2015.

[10] M. X. Chen and F. H. Sung, "Integrated service discovery architecture for heterogeneous networks", International Journal of Communication Systems, Vol. 29, No. 4, pp. 772–786, 2016.

[11] W. Zhao, H. Schulzrinne, and E. Guttman, “Mesh-enhanced Service Location Protocol”, IETF RFC 3528, 2003.

[12] Sun Microsystems, “Java Web Start technology specifications”, available at http://java.sun.com/products/javawebstart/, 2016.