A Proxy-Enabled Service Discovery Architecture to Find Proximity-Based Services in 6LoWPAN

Shafique Ahmad Chaudhry, Won Do Jung, Chaudhary Sajjad Hussain, Ali Hammad Akbar, and Ki-Hyung Kim[1]

Graduate School of Information and Communication
Ajou University, Suwon, Korea
{shafique, yarang, sajjad, hammad, kkim86} @ ajou.ac.kr

Abstract. Recent advances in wireless communication and sensor and actuator technologies have played an essential role to realize the envisioned ubiquitous world. Wireless sensor networks (WSNs) have emerged as a great catalyst for morphing personal area networks (PANs) into low power personal area networks (LoWPANs) which itself is a channel to achieve higher degrees of ubiquity and pervasiveness. These LoWPANs need to be connected with other wireless and wired networks in order to maximize the utilization of information and other resources which are mainly associated with the IP networks. Interworking of LoWPANs with IP networks brings in many challenges for service discovery and network selection. A great problem in this scenario is to find and use services in the closest proximity of the user. In this paper we propose novel service discovery architecture and algorithms that help proximity based service discovery and network selection within an IP network and LoWPAN interworked environment. The results show that our architecture helps finding and using the closest services from inside as well as outside the LoWPAN. It also reduces the traffic overhead for service discovery considerably as compared to other protocols.

1 Introduction

Low power wireless personal area networks (LoWPANs) conform to the IEEE 802.15.4-2003 standard [1]. The IEEE 802.15.4 devices are characterized by low power, low bandwidth, short range, and low cost. While IEEE 802.15.4 provides specifications for physical and link layers, other alliances like ZigBee [2] are striving for defining the upper layers over IEEE 802.15.4. The 6LoWPAN [3], a working group of the Internet Engineering Task Force (IETF) [4], standardizes the use of IPv6 over IEEE 802.15.4. The internet draft [5] describes the overview of 6LoWPAN. It portrays the problems, assumptions and the goals for transmitting IPv6 over IEEE 802.15.4 networks. A frame format for IPv6 transmission over IEEE 802.15.4 networks is presented in [6].

[1] Corresponding author

6LoWPANs need to be connected with each other and with other wired networks in order to maximize the utilization of information and other resources which are mainly associated with IP networks. This integration will help realizing ubiquity by allow users to access the services across LoWPANs and IP networks. Interworking of 6LoWPANs with IP networks brings in many challenges for service discovery and network selection. In an environment where IEEE 802.15.4 networks, which generally have a large number of lower power nodes, will help rendering numerous services across the 6LoWPAN and IP networks, the manual configuration for each service is a burdensome rather an impractical solution. In a ubiquitous and pervasive environment, a service must be searched and configured autonomously with the least user intervention. To provide such a self-configuration environment in 6LoWPANs an intuitive service discovery and selection mechanism is needed.

Both the technologies have drastically big differences in terms of devices' processing power, memory resources and data rate etc, therefore, there are several issues to be resolved to apply the IP based service discovery schemes to IEEE 802.15.4 networks. The limited packet size of 6LoWPANs is one of them; given that in the worst case the maximum size available for transmitting the IP packets over the IEEE 802.15.4 frame is 81 octets, and that the IPv6 header is 40 octets, (without optional headers), this leaves only 41 octets for the upper-layer protocols, like UDP and TCP. UDP uses 8 octets in the header, thereby leaving 33 octets for data, like service discovery mechanism, over UDP. An IP based service discovery mechanism, like simple service discovery (SLP) [7], message could easily be greater than these remaining octets, and it should be transmitted as multiple packets, causing traffic overheads to 6LoWPAN. These limitations require a light-weight service discovery protocol to discover, control, and maintain services provided by devices in 6LoWPAN.

In a 6LoWPAN environment it is better, in most situations, to find and use the nearest service as similar services may be scattered throughout the network. Finding and utilizing the nearest service is not supported by existing service discovery protocols. While using SLP or Simple Service Location Protocol (SSLP) [8] a user may get the list of available services (and their IP addresses) and can communicate with them. The IP address does not mention anything about the physical location. Two consecutive IP addresses in the network may be physically far away from each other. Summing up, an IP address provides no information about the closest service with respect to the user. The proximity of the service is an essential attribute and it must be integral part of service discovery mechanism.

In this paper we present a service discovery architecture that is based on directory proxy agent (DPA) that acts as a proxy to the directory agent (DA) in SSLP. We exploit on the fact that the IEEE 802.15.4 devices are inexpensive and put multiple DPAs within a 6LoWPAN. These DPAs are responsible to maintain and provide proximity-based service information within the 6LoWPAN. The users and service nodes are connected to the nearest DPA that help users find closest services. The connectivity between 6LoWPAN and IPv6 makes the users to find and use local LoWPAN services as well as the services available in external IP networks. Our simulation results show that proposed architecture not only helps finding and using the closest services in inter-network environment but also considerably reduces the traffic overhead, as compared to other protocols, for service discovery.

The rest of the paper is as follows. In Section 2, we review existing service discovery protocols and related work with an emphasis on 6LoWPANs. Different service discovery scenarios for 6LoWPAN have been conversed in Section 3. We describe our proposed service discovery architecture in Section 4. In Section 5 we present performance and evaluation of our scheme and section 6 concludes the paper.

2 Related Work

There are many proven approaches and technologies for IP networks but the service discovery in 6LoWPANs is in its infancy. In this section, firstly, we shall describe the service discovery mechanisms for IP networks and their limitations for 6LoWPAN. Secondly, we shall mention the available service discovery mechanisms for 6LoWPAN.

Jini [9], a product of Sun Microsystems, is an extension of the Java programming language. Jini allows platform independence through Java Virtual Machine environment. This feature limits Jini's applicability in smart environments, which are characterized by heterogeneous device capabilities; and lack of support for Java technology. Universal Plug and Play (UPnP) [10] uses simple Service Discovery Protocol SSDP [11] and extends Microsoft's Plug and Play® technology to the scenarios where devices are reachable through a TCP/IP network. There is a technological conflict between the address-centric nature of UPnP and address-agnostic feature of smart spaces. A framework that mitigates such a limitation of UPnP is still awaited. Bluetooth [12], a joint effort of Microsoft and Intel, is also poised for discovery, but does not elaborate device or service accessibility and usage procedures. The SLP supports a framework by which client applications are modeled as User Agents (UA) and services are advertised by Service Agents (SA). A third entity, called a Directory Agent (DA) provides scalability to the protocol by providing directory services for the network. The UA issues a service request on behalf of the client application, specifying the characteristics of the service which the client requires. The UA receives a service reply from DA specifying the available services in the network which satisfy the request. SLP is used in IP networks for access to information about the existence, location, and configuration of networked services. SLP is well operating in IP networks, but it cannot be applied directly to 6LoWPAN because of the limited resources at 6LoWPAN. The modifications for SLP, to be used over IPv6, have been presented in [13]. This standard describes the use of SLPv2 over IPv6. But there is no provision for use of SLP over LoWPANs.

After describing all these works, we state that currently there is no considerable service discovery architecture for 6LoWPANs except SSLP. SSLP supports the same framework as SLP, i.e., based on UAs and SAs. The DA functions as a database of the services. Besides, SSLP introduces Translation Agent (TA) which performs the translation of messages for the interoperability with SLPv2. The TA must work on a machine, called a gateway, which reaches both an IP network and a 6LoWPAN. When a TA receives either an SLP message from an IP network or an SSLP message from a 6LoWPAN, it performs the translation operation to make the messages recognizable to the agents in the other network. This operation is essential for SSLP to be interoperable with SLP and vice versa. With the help of the TA, a UA can discover and control services in 6LoWPAN regardless of whether they are located inside the 6LoWPAN or outside. The figure 1 depicts a scenario where a 6LoWPAN is connected to an external IP network through a gateway. Both the networks have their DAs which communicate using the TA.

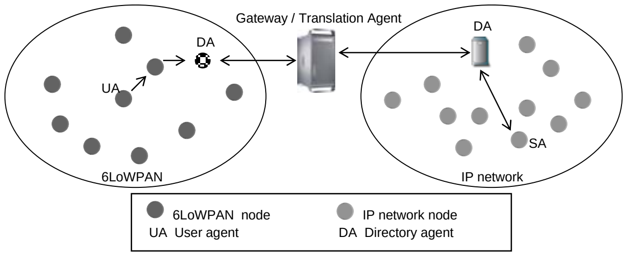

Figure 1. Integrating 6LWPAN with IP networks

3 Service Discovery Scenarios For 6LoWPANs

In this section we shall describe the possible scenarios for service discovery and network selection for a UA in a 6LoWPAN, while integrating the LoWPANs and IPv6 networks. We assume 6LoWPAN supports SSLP for service discovery whereas IP network supports SLPv2.

First, we consider that both the networks are working without DAs. A UA, in 6LoWPAN, that wants to use a service, will broadcast a service request *SREQ* within its local network to find the service. In case the service is available in the local network the respective SA will reply with an *SREP* to the UA. If the service is not available within the local network, service request may be forwarded to other network through gateway after translating the message from SSLP to SLPv2. The service request will be broadcasted in the IP network and if the required service is available, respective SA will respond to the UA, using the TA. As the whole mechanism is based on broadcast, huge amount of traffic is generated that puts heavy overhead on the network. This overhead is definitely not tolerable in 6LoWPANs, which already have very limited data rates.

Second, only one of the 6LoWPAN and IP network has a DA. When there is a DA in IP Network, and not in 6LoWPAN, the UA broadcasts the *SREQ* and gets a *SREP* from the SA if the service is available within the local network. Otherwise, the request is sent to the DA in IP network through the gateway. The service discovery process within 6LoWPAN still has a broadcast and must be mitigated. In case, only 6LoWPAN has a DA, the situation becomes cumbersome. If a UA from 6LoWPAN needs to discover a service it will send a unicast *SREQ* to the DA, which will reply with the address for the service, if the required service is available within 6LoWPAN. In case the service is not available, the service request can be sent to the IP network, directly by the UA or by the DA, through gateway. This request will then be broadcast though IP network to find the service. The SA in the IP network needs to send the reply to the UA in 6LoWPAN and it’s a bit complex. This situation is quite complex as there is no direct route setup between the UA and SA which are in different networks and an overhead of going through gateway is always involved.

Third possible way is to put the DA in both of the networks. In such case whenever a UA in the 6LoWPAN needs a service it sends a unicast service request to the DA, which replies with the address of the required SA. In case the required service is not available in the 6LoWPAN the request can be sent to the DA in the IP network through the gateway. The whole communication between two nodes in different networks can be done through gateway. This approach needs a dedicated node with sufficient resources to act as a DA for the 6LoWPAN. Unfortunately, 6LoWPAN nodes are characterized with limited resources, thus, lack the capability to work as a dedicated DA.

As we have discussed all the apparent scenarios with certain advantages and disadvantages, we insist there is a need of a better architecture for service discovery and selection in 6LoWPANs. There is a need for an architecture that can work as a trade-off between flooding and putting dedicated DAs within a 6LoWPAN.

We propose an architecture that mitigates the traffic overhead for service discovery and network selection and at the same time eliminates the need of a dedicated DA within 6LoWPAN. We introduce Directory Proxy Agent (DPA) to be deployed in 6LoWPAN rather than placing a DA. As the name suggests, DPA is a node working as a proxy for the DA. It gets services information, from DA and peer DPAs, and caches it to use in the local PAN. Exchange of directory information between DPAs in respective networks makes the approach proactive.

We exploit the fact that 6LoWPAN nodes are inexpensive: we suggest putting multiple DPAs, each responsible for a certain area, within a 6LoWPAN. These DPAs cache the information for the services within 6LoWPAN as well as the services within the IP networks accessible through gateway. These DPAs are connected with each other in a hierarchical manner. This approach reduces the traffic overhead for the whole system and does not require dedicated nodes for the role of local DA as the role of DPA can be rotated among the nodes.

4 Detailed Architecture

Before we discuss the detailed architecture, consider a scenario for a building with multiple floors, with many rooms at each floor. There are many available services that are distributed within the building. The whole building can be considered as a big network. We distribute multiple DPAs within the building; putting one DPA in a certain area, e.g. in each room. Each DPA acts, within its limited proximity, as a proxy for the DA. As mentioned before, it is feasible to have many DPAs within a network. Each DPA gets the services information from the DA and maintains a cache in order to provide directory service to the nodes within its proximity. Each DPA periodically advertises itself and the SAs within this DPA’s proximity register their services with it. These DPAs could be arranged in a hierarchical way i.e. they can communicate with their peer DPAs as well as central DAs which might be the part of external IP network. The architecture is depicted in figure 2.

Our architecture is independent of the underlying routing algorithms and can be implemented on any routing algorithm with minimal changes. We have evaluated its performance with AODV, however, we believe that availability of a hierarchical routing mechanism e.g. HiLoW [15] as underlying routing algorithm, will further improve its performance. The major strength of using HiLoW is that if 16-bit address of the service or destination node is known, it can be reached without using a routing table. Once a UA knows the 16-bit address of the DPA or SA, it can start communicating with that, without finding a route to the DPA or SA.

DPAs share their caching information with each other periodically. This sharing of information allows knowing about the services registered wit the neighbor DPAs. This periodic sharing of information reduces flooding which, otherwise, will be required to find services from the neighboring networks. The connectivity of a DPA with the DA of IP network through the gateway facilitates to find services from IP networks. DPAs may exchange the services information with central DA as well, in order to maintain the information consistency.

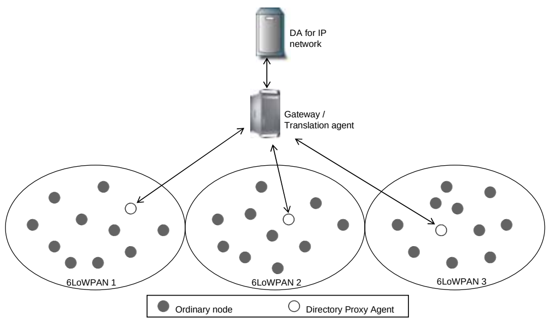

Figure 2. Scenario for proposed architecture

4.1 Proximating the Services

To access the closest services we make it essential that a UA is connected to the nearest DPA. This condition is required to ensure that the service request is sent to the closest DPA, which then replies with the nearest service's information. To realize this condition we propose neighbor assisted DPA discovery protocol that uses the following messages.

- *DDREQ*: The request message is used to ask the neighbors about their respective DPAs. Initiated as a one hop-broadcast by the UA that needs to find the closest DPA.
- *DDREP*: This is the reply message in response to a *DDREQ***.** It contains the address of the DPA as well as distance to the DPA in terms of hop count.

The protocol works as follows. Whenever a UA needs to send a request to DPA, it checks with its single hop neighbors, by broadcasting *DDREQ* in one hop, the closest DPA in terms of hop count. The neighbors reply with *DDREP* that contains the address of closest DPA and distance to it in hop count. The nearest DPA, in terms of hop count, is considered as the closest DPA. Once the address of DPA is known, hierarchical routing makes it possible to send unicast messages between the UA and the DPA. Whenever a UA needs a service, it sends a unicast *SREQ* to the DPA. If the required service is registered with DPA, DPA responds with an *SREP*. UA then starts communicating with SA to use the service. Neighbor assisted DPA discovery algorithm helps in handling mobility of the UA as well. This protocol makes sure that the UA stays connected with the existing DPA as long as it is the nearest one, even when UA is moving. Figure 3 illustrates the algorithm.

Whenever a node wants to join a 6LoWPAN, it first tries to discover an existing 6LoWPAN. IEEE 802.15.4 specifies active and passive scanning procedures for this discovery operation. By following either one of the scanning procedures, the new device determines whether there is a 6LoWPAN in its personal operating space. Once a PAN is found, next step is to connect with the DPA. After getting the address of the DPA, the UA must find a route to DPA if an on-demand routing algorithm like AODV is being used. In case a hierarchical routing algorithm being used, knowing the address of DPA makes this UA capable of communicating with DPA. As the hierarchical routing is available, there is no need to explicitly find and maintain routes between the communicating nodes. If 16-bit short address of a node within 6LoWPAN is known, the path can be traversed by underlying routing mechanism. Though hierarchical routing algorithms eliminate the route finding process, they do not provide optimal routing path.

```
Legend:  DDREQ      : DPA discovery request message
         DDREP(i)   : DPA discovery reply message from the node i
         HC(i, d_i) : Hop count for the DPA from node i
         AD(d_i)    : Address of the DPA for node i


begin procedure
    broadcast DDREQ message in one hop
    closest ⟵ -1
    dp ⟵ -1
    for each neighbor node i
        on receiving DDREP (i)
        if (HC (i, d_i) < closest) then
            closest ⟵ HC (i, d_i)
            dp ⟵ AD(d_i)
        end if
    next
end procedure
```

Figure 3. Neighbor assisted discovery protocol

5 Performance Evaluation

We have implemented our routing protocol in network simulator-2 [NS-2] by modifying the AODV implementation. We have used AODV to evaluate our architecture. Table 1 shows the list and values of parameters for the simulation setup.

The results show that our architecture improves the service discovery time, mitigates the broadcasting overhead, saving the nodes' energy.

Table 1. List of simulation parameters

Parameter	Measurements
Area	380 * 60 Meters
Number of nodes	3hop 15 nodes 30hop 150 nodes
Total time of simulation	100s
Node's transmission range	15m
Protocol	AODV
Traffic type	CBR
Inter-packet transmission delay	0.05 ~ 0.5s
Node transmission power	0.28J

We evaluated our architecture's working under different scenarios by varying the DPA's advertisement interval, which is the time between two consecutive advertisement broadcasts by the DPA. In the same way nodes try to discover a service after certain time, we call it service discovery interval. Originally AODV does not provide a service discovery mechanism but we have used it to find a specific node that may host an SA.

5.1 Number of Generated Control Packets

We define control packets as sum of total number of route request (*RREQ*), route reply (*RREP*) and route error (*RERR*) messages. We varied the service discovery interval to examine its effect on control traffic. The results show that AODV generated more control traffic as compared to our architecture. As depicted in Figure 4, increase in service discovery interval increases then number of control packets in case of AODV. This is mainly because of the fact that when service discovery interval is increased, the existing routing table entries for the SAs remain no more valid. This causes new routes to be discovered and during the process considerable control traffic is generated.

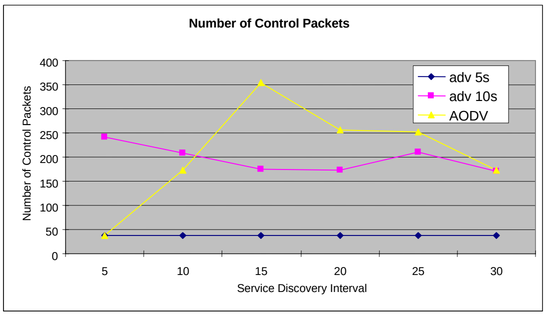

Figure 4. Number of control packets

5.2 Service Discovery Time

We define service discovery as the time interval from making SREQ to sending first data packet in order to invoke the service. It includes the time to receive a SREP from DPA and finding a route to the SA. Figure 5 shows the service discovery time for our protocol when used with AODV. The difference between service discovery times is significant because our architecture is working as an overlay on AODV.

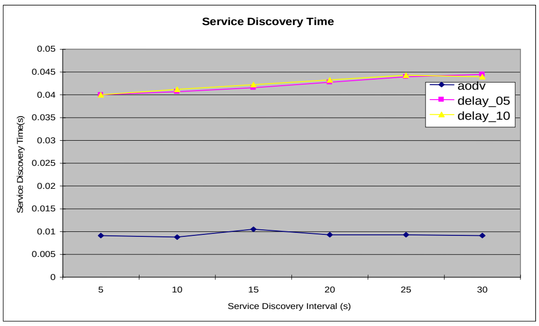

Figure 5. Service discovery time

5.3 Energy Consumption

We have also analyzed the energy consumption of the nodes and the Figure 6 shows the results. All the nodes started with similar energy levels and we checked the remaining energy at the end of the simulation time. For this analysis, we examined the energy level with DPA advertisement interval being 5 seconds and 10 seconds. For this simulation the service discovery interval is 15 seconds. The results show that AODV causes more energy consumption as compared to DPA-based architecture. This fact can be explained with the fact that AODV generated more control traffic.

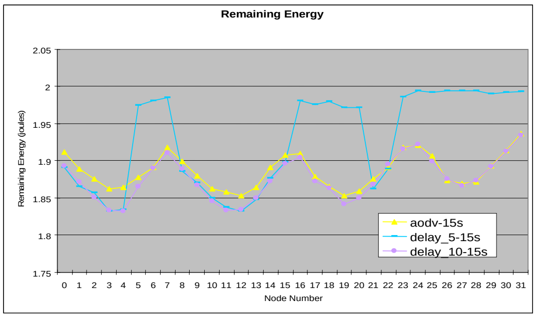

Figure 6. Remaining node energy

6 Conclusion

With the rapid emergence of LoWPANs, it is needed to interwork them with other LoWPANs and with IP network. This integration requires robust and novel architectures for efficient service discovery that can find the services within the close proximity of the user. Existing architectures either don't provide such information or cannot be applied directly to 6LoWPANs. We propose service discovery architecture, based on Directory Proxy Agent (DPA). Our architecture not only finds inter-network services in closest proximity but also relaxes the need of a dedicated DA for LoWPAN. The results show that our architecture reduces the traffic overhead for service discovery considerably, helps finding and using the closest service, and enables users to discover and use inter-networks services.

References

1. IEEE LoWPAN Standard 802.15.4.
2. ZigBee Alliance, http://www.zigbee.org
3. IPv6 over Low Power WPAN Working Group, http://www.ietf.org/html-.charters/6lowpan-charter.html
4. Internet Engineering Task Force, http://www.ietf.org/
5. Kushalnagar, N., Montenegro, G.: 6LoWPAN: Overview, assumptions, Problem Statement and Goals. draft-ietf-6lowpan-problem-02 (2006)
6. Montenegro, G., Kushalnagar, N.: Transmission of IPv6 Packets over IEEE 802.15.4 Networks, draft-ietf-6lowpan-format-02 (2006)
7. Guttman, E., Perkins, C., Veizades, J., Day, M.: SLPv2: Service Location Protocol Version 2, RFC 2608 (1999)
8. Kim, K., Yoo, S., Kim, H., Park, S.D., Lee J.: Simple Service location Protocol for LoWPAN, draft-daniel-6lowpan-sslp-00 (2005)
9. http://www.jini.org/
10. Universal Plug and Play, http://www.upnp.org/
11. Yaron, Y., Goland, T.C., Paul, L., Ye, G., Shivaun A.: Simple Service Discovery Protocol, draft-cai-ssdp-v1-03 (1999)
12. http://www.bluetooth.com
13. Guttman, E.: Service Location Protocol Modifications for IPv6, RFC 3111 (2001)
14. Kim, K., Yoo, S., Park, J., Park, S.D., Lee, J.: Hierarchical Routing over 6LoWPAN, draft-daniel-6lowpan-hilow-hierarchical-routing-00 (2005)