Accepted to WTS 2008, Cal Poly Pomona, Pomona, California, April 24-26, 2008

Service Organization and Discovery for Facilitating RFID Network Manageability and Usability via WinRFID Middleware

Xiaoyong Su, Chi-Cheng Chu, B.S. Prabhu, Rajit Gadh
University of California, Los Angeles
Department of Mechanical and Aerospace Engineering
420 Westwood Plaza, UCLA, Los Angeles, CA 90095

Abstract

A Radio Frequency Identification (RFID) network is a heterogeneous environment in which various hardware devices (RFID readers, sensors, and computers etc) and software components co-exist. Achieving resource manageability and usability in an RFID network is a challenge because of the diversity and complexity of the resources within the network. Service organization and discovery is an effective way to manage resources in a network. However, current research efforts have not addressed the characteristics of RFID network and are mostly focusing on either hardware device or software components discovery. In this paper, we present a service discovery architecture extended from existing service discovery protocols. Service organization, service discovery, service description and exchange format are discussed.

1. Introduction

RFID Technology has being widely used in Supply Chain Management [1] [3], manufacturing, healthcare, automotive, agriculture, disaster recovery [4] and homeland security etc [2]. Typically, an enterprise RFID system consists of following components which is shown in Figure 1: RFID tags, transponders or labels; RFID readers or interrogators; RFID printers; Sensors; Reader controller or edge servers; Enterprise applications and application enabling services; and the network which connects all components together [5] [6] [13]. RFID data carried by RFID tags is captured by RFID readers and then is processed by the edge server before it is delivered to the end applications through the network. In most cases, the pure RFID data does not provide business information of the tagged object. RFID data related business information and events are generated by the RFID application enabling services.

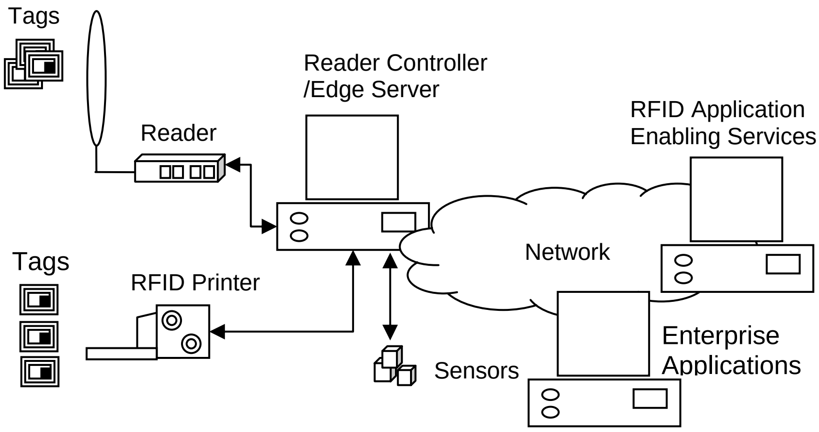

Fig. 1 A typical RFID system [13]

Based on the functionalities, resources in a RFID can be categorized into:

- Device. It captures RFID data. The devices include RFID readers, RFID printers, and Sensors etc.
- Software Component. It process and RFID data and application requests. These software components are used for enabling RFID data for business operation.
- Enterprise Application. The enterprise application utilizes the RFID data for business decision and operation.

To simplify, devices and software components together are called services in this paper.

It is general knowledge that the services need to be organized so that they can be located and consumed by other components when needed. For example, if a reader is attached to an RFID network, the network must provide a way (either by client request or by server broadcast) to notify other components that a new reader is being added and can be accessed. However, the organization and discovery of the services in a RFID network is challengeable. The challenges are:

- The diversity of the devices. In a RFID network, various devices are involved. Usually, each device vendor has its proprietary system to run and control the device. The connection between the device and the network varies from one device to another. For example, RFID reader with Ethernet port can be directly attached to the network while RFID reader with serial port has to be attached to a computer. More over, mobile readers are usually nomad in the network.
- The diversity of the software components. For example, in a RFID network, Device Management Service, RFID Data Services, Application Integration Services, and Business Information Association Services etc. may exist. Each software component may runs as windows service, daemon, application or Web Service etc.
- The hosting platforms are various. In enterprise computing environment, Windows Operating System and UNIX Operating System are most likely co-existing. The software components may run on .Net framework or J2EE platform, both or either.

To solve the challenges, we propose a service organization and discovery architecture for RFID network. The proposed architecture is to increase the RFID resources visibility, usability, and manageability of RFID network.

The rest of this paper is organized as follows. Section 2 discusses existing service discovery technologies. Section 3 explains the proposed services organization and discovery architecture in RFID network. Section 4 describes the proposed service message format. Section 5 demonstrates the implementation of the service organization and discovery architecture in WinRFID middleware. A summary of current research is presented in section 6.

2. Literature Review

Service Location Protocol (SLP) [7] is one of the most widely studied service discovery protocols. It has been defined as a standard track document (RFC2608) by IETF. SLP allows computers and other devices to announce, discover, and select services in a Local Area Network (LAN). It uses URLs for identifying resources. A set of attributes used for describing the service are also defined. There are three roles/agents in SLP: 1) User Agents which are devices that lookup services; 2) Service Agents which are devices that announce services and 3) Directory Agent which is a server that caches services, accepts service announcements, and looks up requests. A client device can have both User Agents and Service Agents. A URL that is used by SLP consists of a service type and a service address. The Service type can be divided into the abstract service type and the service registration. A service template that describes service attributes may also associate with a service type [8]. SLP defines a simple scheme for locating services. however, the interaction between clients and services is not yet defined.

JINI [9] is yet another example of a service discovery implementation. It is a software infrastructure for platform-independent service exchange. JINI relies on Java Virtual Machine (JVM). There are three major components within a JINI system: 1) the Service Provider which allows clients to make use of this service; 2) the Lookup service which has interfaces for service registration and lookup to provide clients access to the service directly or through a proxy; and 3) Client-Service interactions which use Java's Remote Method Invocation (RMI), CORBA or SOAP. Unlike SLP, the Service provider publishes a java based object to a lookup service. When the client finds the service object, the required codes are downloaded for communication with the service.

Universal Plug and Play (UPnP) [10] is a device oriented service discovery protocol derived from Simple Service Discovery Protocol (SSDP) and is released by UPnP Forum. UPnP allows connections to devices and simplifies the implementation of unmanaged or zero-configuration networks for both home and enterprise environment. In UPnP framework, there are two roles. One is the Control Point which is similar to the Directory Agent in SLP and Lookup service in JINI. The other is the Root Device that publishes the service. Unlike other service discovery protocols which mostly define two processes: service announcement and lookup, UPnP involves addressing, discovery, description, control, events, and presentation. Addressing means allocating an IP address to the device. The discovery process contains the service advertisement and search. In this process, an essential amount of information such as the UPnP type, Universal Unique Identifier and URL of UPnP description are exchanged. The description process will exchange the description of a device and its capabilities by using a XML based UPnP Template Language. As long as the control device understands the device description, the control device can control other devices and subscribe to the services. Furthermore, UPnP can represent device services depending on its capability.

Universal Description, Discovery, and Integration (UDDI) [11] [12] is an open industry initiative developed by the Organization for the Advancement of Structured Information Standards (OASIS) consortium to classify, catalog, and manage Web services. UDDI registry is a server which manages a set of Web services and programmatic interfaces for publishing, retrieving, and managing information about services. Information about a Web service is represented as UDDI Data which contains a description of a service's business function (businessService), information about the organization that published the service (businessEntity), service technical details (bindingTemplate), and various attributes or metadata. UDDI Data is represented as XML, and thus can be easily understood by other applications. Each entry has a unique key so that it can be searched. It also allows the publisher to update their registrations. By definition, multiple registries can form an affiliate registry to support UDDI sharing based on policies.

The existing service discovery protocols discussed above are based on a single type of service such as hardware devices, programming interfaces, and reusable software components etc. The RFID enterprise computing environment is a heterogeneous environment where various hardware devices and software components (services) co-exist. Because of the diversity of the resources and their functionality, the existing service discovery protocols can not be directly adopted in an RFID network. Based on current research and investigation, we have developed a service organization and discovery mechanism to facilitate the manageability of an RFID network which allows RFID resource usability and system extensibility.

3. Service Organization and Discovery Architecture

3.1 Architecture Overview

In our approach, a centralized service organization and discovery architecture is built for accommodating various resources (hardware, software services, data services and web services) within a large scale RFID infrastructure. This centralized architecture is adopted in the current research due to the fact that the RFID network is, most of the time, a static rather than a dynamic system, i.e., the topology of an RFID network remains unchanged most of the time. To facilitate server-client information exchange, a service oriented message definition for registering services is also proposed in the current research. In this paper, a "service" is defined as a component which can provide data or controllable actions, and a "client" is defined as a component that consumes service. Figure 2 shows the service organization and discovery architecture. There are five major components in this architecture: Registration Server (RS), Registration Agent (RA), Service Provider (SP), Service Consumer (SC) and Discovery Agent (DA).

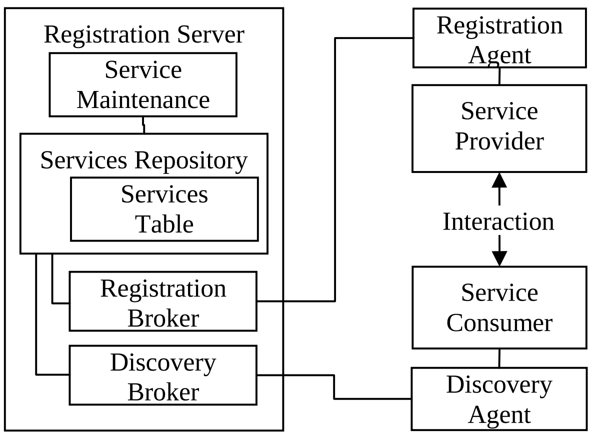

Fig. 2 Architecture of RFID Service Organization and Discovery

The RS is responsible for accepting registration requests, maintaining services tables, managing sessions and providing services lookup. It consists of a Registration Broker, a Discovery Broker, a Service Maintenance module, and a Service Repository which host a services table. The two brokers are responsible for processing request messages and sending results back. Service Maintenance module updates the TTL of each service entry in the Services Table, and deletes the service entry with expired TTL. The SP is a component that provides one or more services. The SP registers itself to RS via a RA. Two types of RA exist. One is internal RA, which hosted by SP. The other one is external RA. It acts as proxy to help some services who can not register by themselves. For example, a reader has to be registered through an external RA because it does not support any service registration protocols. Similarly, services such as database and message queue are independent services which may be used to store RFID information. They have to be registered through an external RA. There are two types of DA: internal DA or external DA. The internal DA is embedded in the service consumer. The external DA is an individual component that acts as a proxy to process requests from service consumer. The advantage of separating RA and DA from SP and SC respectively is that the SP and the SC are not required to implement service discovery mechanism. Meanwhile, the manageability of the network can be improved by hierarchically organizing the services through the RA. Another advantage of using RA and DA is that it can reduce the complexity of the registration service. A SC is any software component that requests a service. It locates a service by using a DA.

All services in an RFID network have a limited lifetime. The service lifetime is maintained by the service management agent and is periodically updated. Authentication is not considered in this architecture because service authentication should be integrated with system authentication to simplify discovery process.

3.2 Service Organization Process

As implied in the architecture overview, there are two processes in the architecture: service organization process and service discovery process. The service organization process is the most important process in a service discovery protocol. Various components in an RFID network are organized through the service organization process. As shown in Figure 3, the service organization process has three procedures: service registration, service maintenance, and service de-registration.

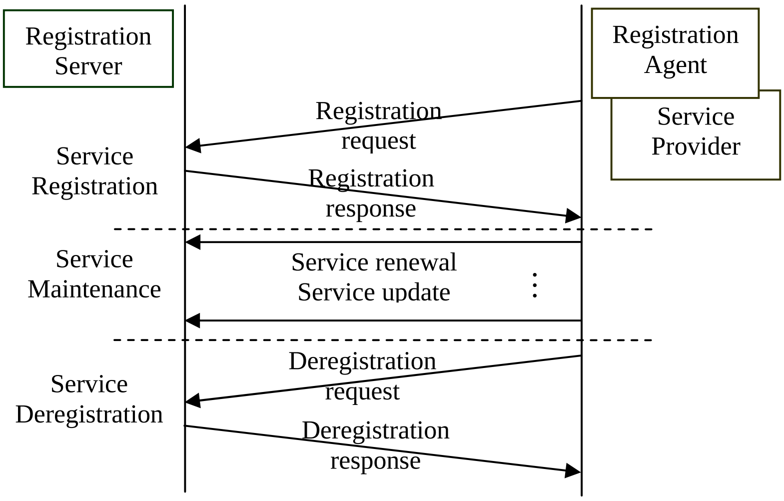

Fig. 3 Service Registration Procedures

Service registration is the first step of service organization process. The registration process starts with a SP sending a service registration request message to the RS. If the registration broker recognizes the message, it will try to match this service with the existing service entries in the services repository. A new service entry will be added into services table if it does not exist. Otherwise an error response message will be returned to SP. The service entry contains all the information defined in the service registration message that defined in section 4 except message type. To prevent unauthorized registration, only authenticated RA or SP can register. The authentication process is implemented though the operating system or upper layer application.

The functions of service maintenance module are shown in Figure 4. The service maintenance module is used to keep the service table up-to-date, preventing isolated service or dead service calls. Each service entry in the service repository contains a field called Time-To-Live

(TTL). A service management module on the registration server will check the TTL of a service entry periodically. If within a time period, a service entry didn't update its TTL, either the service represented by this service entry is no longer available or there is a network problem between the service provider and the registration server. This service entry will be deleted from the service table to prevent dead service calls. The TTL will be reset by a service renewal message. SP sends a renewal request periodically to RS to refresh the service entry. The interval of sending a renewal message must be shorter than TTL contained in the registration request message. If the service changes, an update message is sent to the RS. This process is called service update. The registration broker in the RS processes the updated message and updates the existing service entry.

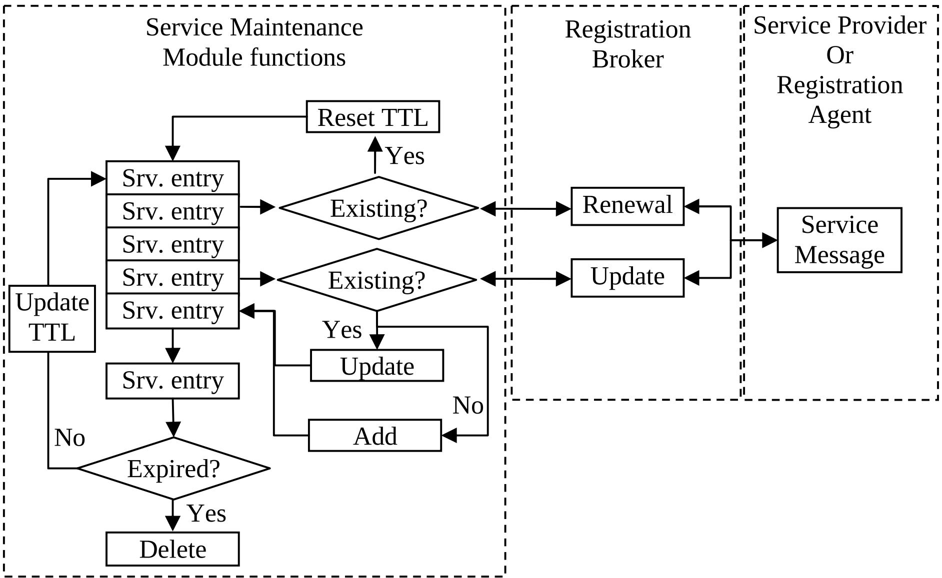

Fig. 4 Service Maintenance Procedures

Service de-registration process de-register a service from an RFID network. SP or RA sends a de-register message to RS, the corresponding service entry is removed from the service entry table immediately.

3.3 Service Discovery and Service Consumption Processes

To provide information about the resources managed by the registration service to service consumers, a service discovery mechanism is

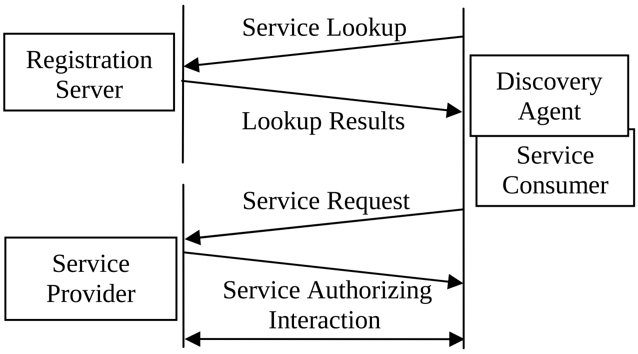

Fig. 5 Service lookup and consumption

defined. The service discovery process starts when a SC tries to find a service by sending a service request message to the RS. The discovery broker accepts the request message if the SC is authenticated (security and authentication related issues forms another research topic and are not addressed in this paper). Then, the service request message is parsed and a lookup of the service in the services table is performed. If a service entry matches the search criteria, the discovery broker returns a service entry message or a service entry message list to the SC. Once the service is found, the SC communicates with the SP or RA for negotiating information exchange. The service discovery process is shown in Figure 5.

4. Message Definitions

To support multiple types of services, a platform neutral service description methodology is required to represent a service. Extensible Markup Language (XML) is a widely used format for platform neutral information changes. However, an XML parsing engine requires significant computational resources which may not be available on certain components such as sensors. Because of the variation in resources, the service message format should be concise, machine readable. In essence, the message that is used in an RFID network environment should be simple and efficient. Binary format is the simplest and most concise formats. It is easy to be processed and thus can be supported by a simple device such as a sensor. Compared to XML representation format, the computing requirement of binary format is far less. Therefore, in our approach, a binary format based description message is defined. Two categories of messages are defined: service organization messages and service discovery messages. Each category has two types of messages: request message and response message. All the fields defined as variable length in the message structure are ASCII codes converted from string. Each type of message has request message and response message respectively.

To ensure Quality of Service (QoS), Bandwidth Index, Network Index the request message for service organization and the response message for service discovery. However, they are only used as reference because they are not representing the actual network bandwidth and status between the SP and SC. In a RFID network, the client usually request a service because of the requirement of fulfill a particular task instead of looking for better service quality. The detail message format is not discussed in this paper.

5. Implementation

This service organization and discovery architecture has been implemented in WinRFID middleware research whose aim is to build an RFID infrastructure that facilitates RFID devices, sensors, and software components management and improves the RFID resources usability. In the WinRFID network, all the resources such as devices and software components are defined as service and are registered into RS. The client application can find services by sending the service discovery message to the RS.

Figure 6 shows the tree structure of services in WinRFID network. Some of these services are software components, while some are hardware (with software supports). Each service has to be registered to the WinRFID network either by itself or through a proxy it can be consumed by other applications. The WinRFID registration service provides three interfaces: Socket, Web Service and .Net Remoting object to accept service organization and discovery messages.

Two levels of services are identified in the WinRFID network. Currently, the first level of services in WinRFID network includes Reader Coordinator, Local Object Name Service (ONS), EPC service, Data Collector, WinRFID Identification Information Service (IIS). The first level services are sophisticate services that register itself to the RS. It also acts as RA to register its child service to the RS. The second level of services does not has the capability to register itself to RS. They have to rely on the first level services thus are managed by the first level services.

The Reader Coordinator is responsible for managing and controlling readers and sensors or other devices in WinRFID network. Each Reader Coordinator has readers, sensors, printers, web service, and message queue associated with it. The EPC service manages a web service that provides EPC functions such as EPC information preparation and inquiry. Data Collector manages a web service that provides interfaces for trading partners to register EPC information and

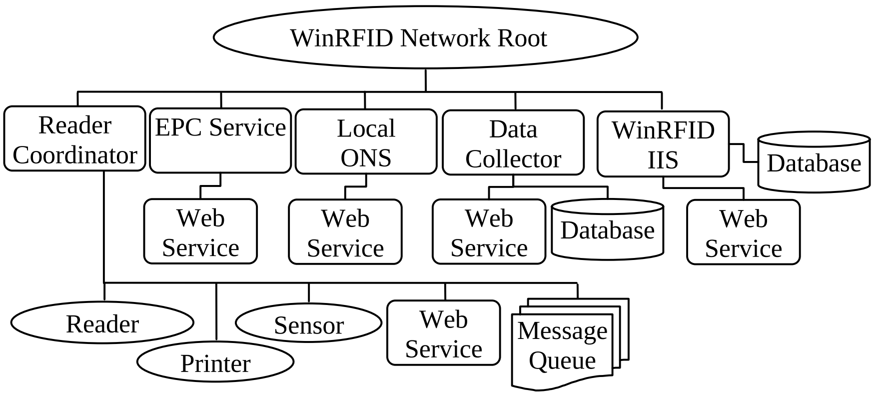

Fig. 6 WinRFID Service Components Tree Structure

subscribe EPC information. It also has a local database associate with it. WinRFID IIS service manages a web service and database that can be directly accessed by SC. Local ONS manages a web service that provides the function for looking up the WinRFID IIS service.

6. Experiments on The Service Organization and Discovery Architecture Implementation

The performance of the registration service directly affects the usability of the RFID resources. We identify that the following factors that represents the performance of RS. These factors are: Network Utilization, CPU Usage, and Client Callback Round-Trip Time (CCRTT). Since all the RFID services are registered to the registration service and renew themselves periodically, all the activities related to the registration process generate network traffic. Too many traffic may cause network congestion that leads to service quality reduction or denial of service. At meanwhile, the CPU Usage reflects the capability of processing registration request of the services. For example, if it takes too much CPU computing power to process a registration request, the total supported RFID services is limited. It is important to know how many identification services can be support by the registration service. It is helpful to decide if additional registration services are required in a large scale enterprise environment. The Client Callback Round-Trip Time is the total time between the client sending request and receiving the response. It is an important measurement that represents the response capability of the registration service.

The experiment environment consists of one Dell Precision 850 workstation, three HP X4000 workstations and one Dell Latitude D600 laptop. The Precision 850 workstation equipped with an Intel Xeon 2.0GHz Processor and 1.0GB RAM. The X4000 workstation equipped with an Intel Xeon 2.2GHz Processor and 512MB RAM. The D600 laptop has 1.73GHz Centrino processor and 512MB RAM. All five computers are resided in a local area network with 100MB fast Ethernet connection. The registration service is running on the Dell Precision 850 workstation. Multi-threading RA running on X4000 workstation is used to simulate the registration process of multitude RFID services. Each X4000 runs one RA. The D600 laptop runs the application used to measure the CCRTT. For every test, the number of services in each RA is equal. We conducted measurement at 200, 400, 600, 800, 1000, and 1200 services on each RA. Respectively, we have 600, 1200, 1800, 2400, 3000, and 3600 services registered to RS. The renewal rate for each service is 10 second.

Figure 7 shows the peak network utilization of the Precision 850 that runs RS. As indicated by the figure, the network utilization is proximately proportional to the number of registered services. However, we did not observe significant network traffic increasing that could cause the network congestion.

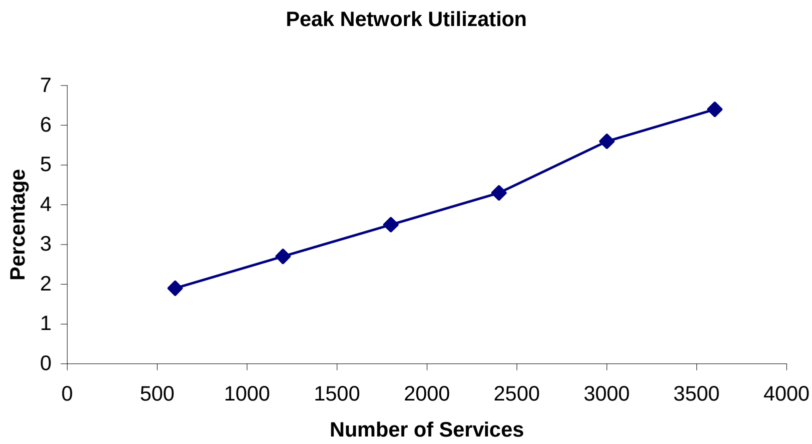

Fig. 7 Network Utilization

Figure 8 shows the peak CPU usage of the Precision 850. Similar to the peak network utilization test, the peak CPU usage is proportional to the number of services. However, the significance of the CPU usage is much higher than the network utilization. When 3600 services registered, the peak CPU usage is about 38%. We did not perform full load test of the registration service because large number of services in the RA used up memory resource of the hosting workstation.

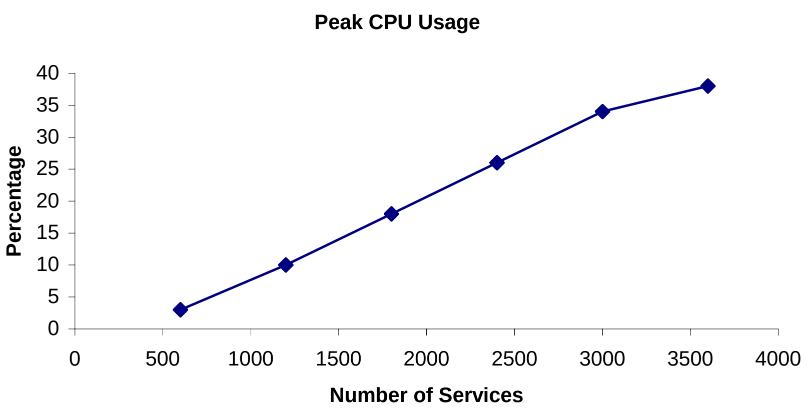

Fig. 8 Peak CPU Usage

Figure 9 shows the Client Callback Round-Trip Time. Similar to the network utilization and CPU usage, the CCRTT of listing all the services registered on the registration service is proportional to the number of the services. However, the register processes and de-register process are not affected by the number of the services.

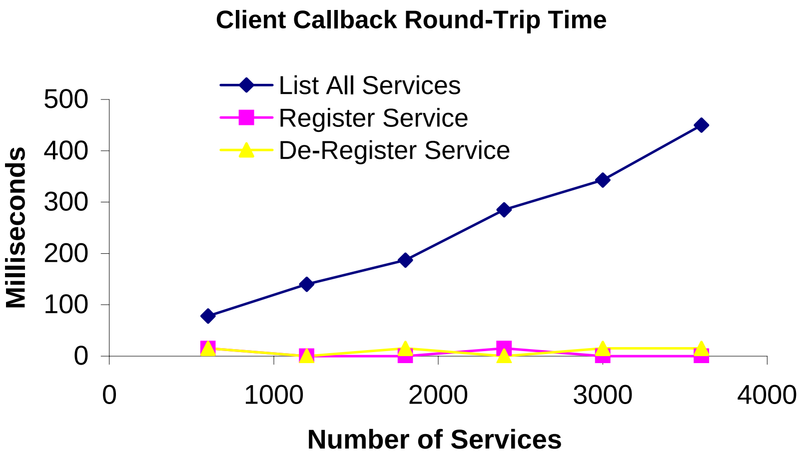

Fig. 9 Client Callback Round-Trip Time

7. Discussion and Summary

Although there have been intensive research efforts on service discovery protocols, there is little research on solving service discovery for RFID networks where both hardware devices and software components are existing simultaneously in the network. Most service discovery protocols have been developed for discovering useable resources in a network. They are typically developed for dealing with a single type of resource: either hardware components or software components.

Based on current research work on service discovery and the characteristics of RFID resources, we have implemented a unique service discovery architecture to facilitate RFID network manageability and to extend resource usability. In this research, we define a novel architecture which can effectively manage different types of components - from hardware to software, from devices with high computational capability to devices with low computational capability and from data source to controllable devices. To simplify service description and service information exchange between these components, a message format for server-client information exchange has also been defined and discussed. This approach allows manageability of the RFID network as well. However, the underlying concepts can also be extended for any management of heterogeneous computing environment. The proposed architecture is an essential module of WinRFID middleware. Within WinRFID network, various services are organized into a hierarchical structure and are managed via a centralized user interface.

One weakness in our approach of using the centralized architecture is that if the registration service is off-line then the new components cannot be registered and therefore the services cannot be discovered. In addition, the existing service information on the registration server may be lost in case of sudden system shutdown. This will cause a loss of the entire network topology and service entry information. In this case, all the components in RFID network are in an unmanaged status and isolated. Therefore, redundancy of the service information becomes a key factor in building a managed RFID network. For future research, a distributed registration service is proposed to eliminate the possibility of registration service failure. Several registration services are deployed at different location in an RFID network. Each registration service has the same copy of service repository. Service repository is synchronized when the network changes (such as service registered, service de-registered and a new registration service started etc). Thus, once registration service failure does not affect the entire network.

8. Acknowledgment

This project is sponsored by UCLA-WINMEC and UCLA-Wireless Media Lab

Reference:

[1] The History of RFID Technology, http://www.rfidjournal.com/article/articleview/1338/1/129/

[2] Rajit Gadh, "RFID: Getting From Mandates to a Wireless Internet of Artifacts," In Computerworld, October 2004

[3] Rajit Gadh, "RFID moves beyond supply chain mandates," In Computerworld, April 2005

[4] Rajit Gadh and B.S. Prabhu, Radio Frequency Identification of Katrina Hurricane Victims, IEEE Signal Processing, March 2006

[5] B. S. Prabhu, Xiaoyong Su, Harish Ramamurthy, Chi-Cheng Chu, and Rajit Gadh, "WinRFID – A Middleware for the enablement of Radio Frequency Identification (RFID) based Applications", Invited chapter in Mobile, Wireless and Sensor Networks: Technology, Applications and Future Directions, Rajeev Shorey, Chan Mun Choon, Ooi Wei Tsang, A. Ananda (eds.), John Wiley, December 2005.

[6] Harish Ramamurthy, B. S. Prabhu, Rajit Gadh and Asad Madni, "Smart sensor platform for industrial monitoring and control", 4th IEEE SENSORS 2005, October 31st – November 3rd 2005, Irvine, California

[7] Erik Guttman, "Service location protocol: antomatic discovery of IP network services", IEEE Internet Computing, Volume 3, Issue 4, July-Aug, 1999

[8] E. Guttman, C. Perkins, and J. Kempf. "Service templates and service: Schemes." RFC 2609, Internet Engineering Task Force, June 1999

[9] Sun, "Jini Network Technology", http://www.sun.com/software/jini/whitepapers/jini-execoverview.pdf

[10] UPnP Device Architecture, http://www.upnp.org/download/UPnPDA10_20000613.htm

[11] UDDI version 3.0.2, UDDI Spec Technical Committee Draft, Dated 20041019. http://uddi.org/pubs/uddi-v3.0.2-20041019.pdf

[12] Introduction to UDDI: Important Features and Functional Concepts, http://uddi.org/pubs/uddi-tech-wp.pdf

[13] Xiaoyong Su, Chi-Cheng Chu, B.S. Prabhu, Rajit Gadh, "On the creation of Automatic Identification and Data Capture infrastructure via RFID and other technologies", The Internet of Things: from RFID to the Next-Generation Pervasive Networked Systems, Lu Yan, Yan Zhang, Laurence T. Yang, Huansheng Ning (eds.), Auerbach Publications, Taylor & Francis Group, February 2007