Employing SNMP to Manage Ubiquitous Environments

Syed Shariyar Murtaza*, Syed Obaid Amin*, Choong Seon Hong*, Sang Hyun Choi**
*Dept. Of Computer Engineering Kyung Hee University, South Korea
**National Computerization Agency
{Shariyar,obaid}@networking.khu.ac.kr, cshong@khu.ac.kr, csh@nca.or.kr

Abstract

The vision of ubiquitous computing is becoming a reality now .Service discovery in ubiquitous environment, as well as adding semantics to the data is quite established. But, not many researchers have thought about the management of these devices. We envisage that by integrating SNMP with service discovery protocols, we could leverage the use of management factors (like performance, fault and security management etc) in the business, organizations, and other multi-user environments.

1. Introduction

Ubiquitous computing is envisaged as a computing paradigm with minimal user intervention, emphasizing detection of environmental conditions and user behaviors in order to maximize user experience.

To discover and control services in ubiquitous environments different service discovery mechanisms have been proposed, like UPnP, JINI, Salutation, SLP .Some researchers have also proposed their semantic versions of service discoveries.

Till now, the researchers have mostly focused on making the vision of ubiquitous computing a reality. Service. discovery issues, Context aware issues, wireless connectivity, sensors, RFIDs and etc, have mostly been the focus of the researchers. But, the devices are growing in number, being embedded ubiquitously all around our environment, there is a need to manage these devices. By management we mean, analyzing the performance, fault diagnosis, accountability, security and configuration. A management system would want to be informed about these devices, there activities, and usage.

More specifically, we need a unified management model to manage these devices. Let us consider a complicated ubiquitous environments e.g. office , organization, manufacturing environment etc, where we have to keep track of all the events occurring in the systems ,or if some devices experiences havoc , then we have to keep log of these incidents, the wrecks of these incidents, when they occurred and what is effected?

SNMP based network management is quite popular .It provides the simple management of resources, usually network, but could be extended to applications as well. There are different reasons for using SNMP. It is quite simple and several APIs are available for its implementation. Service discovery protocols like UPnP, JINI etc could be complemented with the security features of SNMP version 3, particularly UPnP. With the MIB we can keep log of the events occurred in the system, which would help in identifying the exact cause of the problems, and would help the administrator in recovery of devices or system. SNMP has the rich feature of extensibility, as MIBs could be defined privately, as well. This is particularly important in ubiquitous environment, where the devices and their interfaces change rapidly.

Let us elaborate on this, with some examples from management perspectives e.g. consider an air conditioner, which can be controlled by service discovery like UPnP, but to monitor its performance whether its cooling performance is appropriate or not, is it heavily used! a management protocol like SNMP could be beneficial. Similarly, in a multi user environment, misuse of shared resources could be taken into account e.g. a person's car usage by other persons, use of shared printers. This will also help in detecting the faults in ubiquitous environments, like malfunctioning of sensors, too much traffic from sensors, leading to improper results.

The rest of the paper is articulated as follows. We'll mention the relevant work, our architecture, MIB, implementation and finally conclusion.

2. Relevant Work

SNMP refers to a set of standards for network management, including a protocol, a database structure specification, and a set of data objects. SNMP encompasses a manager/agent model. The manager and agent interact with each other using Management Information Base (MIB) and a set of commands.

UPnP (Universal Plug and Play Protocol) enables discovery and control of networked devices and services, such as network-attached printers, Internet gateways, and consumer electronics

This work was supported by the Korea Research Foundation Grant (R05-2003-
000-12193-0) funded by the korean Government(MOEHRD) and by NCA.

equipment. Similarly, we have other service discovery protocols like JINI, salutation, SLP etc. Our objective here is to merge these service discovery protocols and SNMP .Use the service discovery to discover and control devices, while employ SNMP to perform the management activities, like accounting, fault tolerance, performance monitoring.

In the past, certain approaches for network management have been proposed. For embedded devices, Hong et al [7], proposed the integration of SNMP and Web servers into the embedded devices. XML based network management is also an important area of research in the realm of network management. The authors [8, 9] mentioned certain advantages of XML over SNMP, like HTTP header compression for large data, use of HTTP over TCP, SSL for reliability and security, and more data types in XML Schema. Taek Ju et al [10] proposed embedded web server with XML based management. A work similar to our work has been carried out in [3].In this paper author has proposed a proxy architecture with SNMP, concerning only the EIB (European Installation Bus). They didn't consider about the ubiquitous environments as a whole. The MIB designed was also static, supporting only EIB systems.

Although, several advantages of XML has been articulated, but still SNMP has been deployed widely. SNMP's ability to measure data, to support fault and performance management is efficient. SNMP v3.0 also provides good security features. While configuration management and remotely controlling and management suffers at SNMP. In our architecture, we'll keep the UPnP to control the devices and SNMP for management.

3. Architecture

Overview of our architecture is shown in the next figure. Here we manage one dedicated UPnP control point with the SNMP agent to monitor the system. This control point subscribe to different events with different devices, as requested by the SNMP manager. .It then receives the information from the devices, which is stored by the SNMP agent. The manager then queries the agent, as per MIB and converts it into a visualized form, showing statistics etc. In other words this SNMP agent acts as a proxy between SNMP manager and UPnP devices via control point It also keeps record of the events or history.

The dedicated control point is used because in multi user environments, multiple users can enter and leave the system, with personal or different control points. By using a dedicated CP with SNMP agent, we can monitor the activities and use of the shared resources of the system.

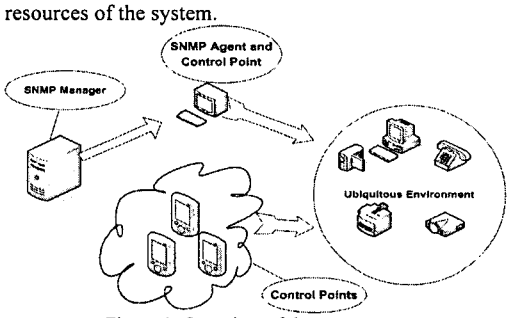

Figure 1: Overview of the system

4. MIB

In SNMP based system MIBs are static, they don't allow to add dynamic contents .But, in ubiquitous environment, appliances enter and leave the system dynamically. Also; the interfaces of the appliances are liable to change, either because of the updates by manufacturer or by the replacement of manufacturer by a user.

Due to this, we have created a generic MIB. This MIB is like a relational database, in which device ID is acting as the primary and foreign key in different MIB tables. These MIB values would be filled by the agent, working together with the UPnP control point. This MIB is shown below in the following diagram.

This MIB constitutes three different tables. 'Device Table' is used to keep the general information about the devices. 'Device Property Table' keeps the record of the properties or methods exposed by devices, their access status and possible values. It also contains the field monitor which could be set by the manager, whether to monitor a particular device property or not. Finally, the 'Log Table' maintains the history of the events which have occurred or the properties of the device which have been accessed.

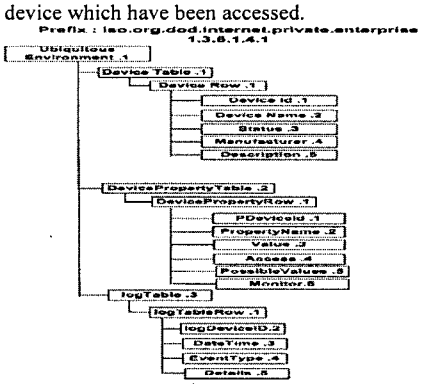

Figure 2. MIB Tree

5. Implementation

Our prototypes of the implementation are shown in the following figures. We have developed the system using C#, .NET and Intel's UPnP SDK. The following figure shows our SNMP manager, which could load MIB files and listen to SNMP traps and perform different SNMP commands and operations.

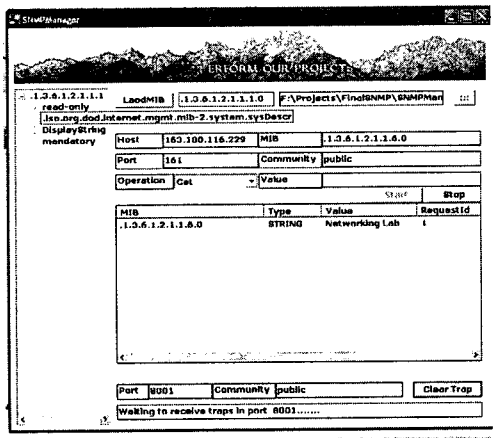

Figure 3: SNMP Manager

We have also developed UPnP based devices and a control point. Control point discovers services in a UPnP based environment and subscribe to the events exposed by those devices. Our SNMP agent converts the information form these devices into the MIB format as shown above. This MIB agent could be queried by any SNMP manager, with its MIB. To avoid the overhead, we have implemented the control Points and SNMP agent as a console based application, along with UPnP devices. Our UPnP Control point + SNMP agent is shown below. We have shown some sample information, just to provide a purview.

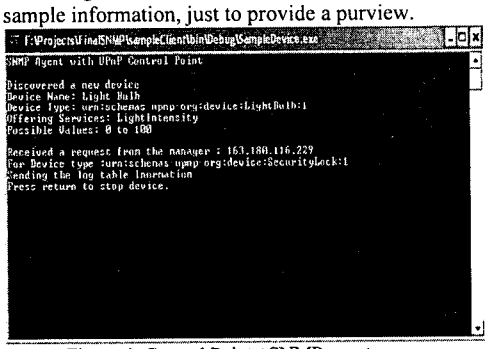

Figure 4: Control Point +SNMP agent

6. Future Work and Conclusion

We have observed that UPnP control point with the SNMP agent works quite well, but it consumes more memory, which is not suitable for small devices. Also; to keep the log of the events, disk size has to be enough. Further, we have just implemented a simple application, to test our idea of managing ubiquitous devices. We are yet to analyze its result in a highly complicated environment, where many services move to and fro, in and out of the system. We also have to stress test the system. Our parameters for stress testing include the number of devices, services, event notifications, queries and traps for the agent.

We envisage that by using SNMP for management of ubiquitous computing paradigm, we could easily manage the devices, particularly in environments with large number of devices. Performance monitoring, accounting or security management could be done quite easily. We are also looking for the XML based management in our next version, with the details of performance, accounting and fault management statistics

6. References

[1]. William Stallings, SNMP, SNMP v 2, SNMP v3, RMON1 and RMON2, third edition,1999.

[2]. UPnP : www.upnp.org

[3]. Ran Gladi, "SNMP for home automation", International Journal of network Management, 2004

[4]. JINI:www.jini.org.

[5]. Michael Jeronimo and Jack Weast, UPnP Design by Examples, A software developer's guide to Universal plug and play,2003

[6] MSDN, http://msdn.microsoft.com

[7] Liu hong, Bai dong, Ding Wei ,The integration of SNMP and Web in embedded devices, Info-tech and Info-net, Proceedings ,IEEE,2001.

[8]Frank Straub, Torsten Klie,"Towards XML oriented Internet Management",___Integrated Network Management, 2003. IFIP/IEEE Eighth International Symposium on

[9]Mi-Jung Choi*, Hong-Taek Ju** and James W. Hong*, Towards XML and SNMP Integrated Network Management, proceedings of Integrated Network Management, IFIP/IEEE ,2003

[10] Hong-Taek Ju; Mi-Jung Choi; Sehee Han; Yunjung Oh; Jeong-Hyuk Yoon; Hyojin Lee; Hong, J.W, "An embedded Web server architecture for XML based network management", proceedings of NOMS, IEEE/IFIP ,2002.