SIMPLIFYING HOME NETWORK MANAGEMENT

Using Instant Messaging as Unified Interface and Wiki as Knowledge Retention Repository

KEE-LEONG TAN, SEE-LENG NG, PANG-LEANG HIEW
Malaysian Research Centre, British Telecom
{keeleong.tan, seeleng.ng, pangleang.hiew} @bt.com

AND

MOHD REZA SYAHZAN BIN ONN
Faculty of Information Technology,
Multimedia University, Persiaran Multimedia,
63100 Cyberjaya, Selangor, Malaysia
mohd.reza.syahzan.04@mmu.edu.my

Abstract. This paper describes a proposed system architecture which offers a novel interface to simplify home network management, with knowledge retention capability. The architecture described here joins the concepts of network management, presence, Wiki and log services, and uses instant messaging system as the service delivery and communication platform. This architecture allows the status of network device and services (such as wireless routers and firewalls) to be represented in an innovative and compelling graphical display, and thereby improve distributed group problem-solving and enhance the user experience. The architecture also introduces simple deployment architecture and user-friendly interface that facilitate rapid service introduction and makes the service deployment and maintenance cost and time effective. Home or small business users can easily integrate this into their network system and use a well-familiarised IM interface to manage the network locally or remotely, regardless of the heterogeneity of network components. It allows minimum fuss and provides maximum availability for home or small business users in network management, which is not their expertise.

1. Introduction

Network management has become an increasingly challenging task as broadband penetration increases. One approach is to employ the help of

propriety and/or open source network management tools to aid with the management of a growing and increasingly complex network of heterogeneous devices probably from different equipment vendors. However, this approach suffers from a number of key setbacks when it comes to smaller networks like home or small office. Firstly, the network administrator is assumed to have sufficient technical knowledge and competencies to use the tools to administer the various devices or troubleshoot any problem. Secondly, most modern network devices are equipped with administration interface, such as web, SNMP or CLI. However, the layout, attributes, command, etc. are specific to the device, and might not be supported by the network management tool. Thirdly, this approach is also not suitable for your stereotypical homes or even small to medium offices, where the network setup is a lot less complex and range of possible devices in the network are predictable. Introducing another network management tool into this environment is overkill, as it requires the user to familiarize with some new interface, command and behavior.

From a broadband network provider perspective, in order to improve customer experience and to address the challenge of network management tool limitations described earlier, we argue the following three requirements are important in managing a home or small network: (i) A unified and easy-to-use interface to manage network devices; (ii) The ability to automate problem identification at the system level; and (iii) Knowledge retention and sharing capability, so that users could leverage on community knowledge to help each others.

In this paper, we present a prototype which was developed as a proof-of-concept system. Section 1 describes the problem statements from home or small business perspective and how network and service provider can solve it by offering a new service. In Section 2, the proposed architecture is described. We further elaborate the different capabilities of the system and describe scenarios where these capabilities are targeted at in Section 3. Section 4 discusses the detail implementation as well as the experiments that we have carried out in an actual live environment. Section 5 concludes the paper by highlighting how the system can improve human-computer interface when dealing with telecommunication system. The focus will specifically be on how the proposed architecture helps to improve customer's experience, reducing costs and identifying new services for a telecommunication service provider.

2. Architecture

The system architecture can be further divided into three layers based on locality, namely home network layer, gateway layer and remote network layer.

2.1. HOME NETWORK - UI LAYER

This is where user will interface with the system, and we further define two main components within this layer, namely: *IM Client* and *Managed Device*.

2.1.1. IM Client

The IM Client could be any instant messaging software client which supports asynchronous communication via Jabber/XMPP protocol (Jabber, 2008; XMPP, 2008). It retrieves status and message from *XML Messaging Server* and provides a unified interface to visualise all managed devices. The sending of remote commands to managed devices is also handled by this component.

2.1.2. Managed Device

Network equipments such as network printer, wireless router and computers which could communicate with the system are called Managed Devices. The architecture supports two types of managed devices: (i) Device which has built-in log capability like Syslog or Syslog-ng (Syslog-ng, 2008), with capability to publish its logs remotely to *Logging System*; (ii) Devices without log capability, where its status will be acquired from periodically checking using protocol such as ICMP.

2.2. HOME NETWORK - GATEWAY LAYER

The intelligent and core features of the proposed architecture resides in this layer. Figure 1 shows a high-level description of the components which manage registration of *IM Client* and *Managed Device*, handle connections, formatting and routing of messages; and transform device's status into XML, with additional semantic description.

The monitoring of *Managed Device*(s) health is handled by the following processes. Referring to Figure 1, *Log and Status Parser* component will continuously aggregate/check for anomaly event from all statuses and logs data. During event of alarm, it will parse the event data, which will be transformed into XML description object, with additional non-functional information. Next, a query is performed via the *Wiki API* to check for the existence of similar alarm in the knowledgebase. The XML alarm description

and the query results are sent to *XML Messaging Server* and subsequently forwarded to respective *IM Client*(s).

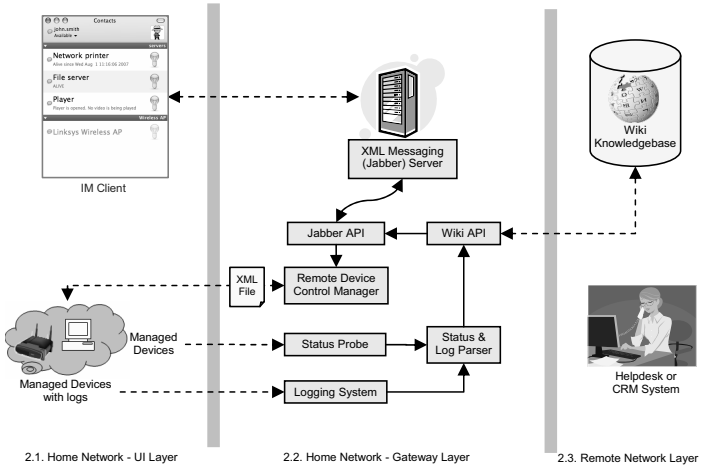

Figure 1. System Architecture.

2.3. REMOTE NETWORK LAYER

This is where network provider like BT can make use of the communication infrastructure it owns to enhance user experience by extending a remote knowledgebase service beyond the home network – gateway layer. One example is to host a *Wiki* server which functions as knowledgebase accessible by different home users. It provides a repository where troubleshooting steps can be centrally archived for future reference. This will also encourage the participation of user community to help each others. Another usage of the *Wiki Knowledgebase* is to complement helpdesk or Customer Relationship Management (CRM) system.

3. Usage Scenarios

Due to its layered, open standard and scalable features, the architecture described in section 2 is able to support new services creation and many deployment scenarios. Here, we present three scenarios focusing on enhancing home customer experience, from the perspective of a broadband network provider.

3.1. SCENARIO 1 – NETWORK MANAGEMENT DO-IT-YOURSELF (DIY)

The simplest implementation of the architecture can be done for residential or Small Office Home Office (SOHO) network. The use of *IM Client* allows device status (i.e. availability, alarms, warnings) to be brought to user's attention automatically and immediately. This is possible as the user may use a single *IM client* to manage his/her personal contacts as well as his managed devices. During troubleshooting, identification on the problem is handled at the system level by *Status and Log Parser* component, and should the similar problem is found in the *Wiki Knowledgebase*, it will provide the user with a link to the Wiki page. Users with moderate IT knowledge or no prior diagnosis experience will find this time and effort saving when dealing with unfamiliar network problems.

3.2. SCENARIO 2 – REMOTE HELPDESK

The aggregation of network status to a centralised knowledgebase allows remote monitoring and administration of managed devices. Over time, troubleshooting steps archived in central knowledgebase will be valuable information for operation and system support, and troubleshooting reference for customer helpdesk as decision support tool. For example, a third party service provider can offer different remote monitoring service packages with different Service Level Agreement (SLA) to home users. The architecture allows service provider to perform remote troubleshooting in subscriber's home network with minimum user interaction as the problem diagnostic and identification are handled by the system described in Section 2.

3.3. SCENARIO 3 – INTELLIGENT AUTOMATIC REPAIR FEATURES EMBEDDED AT GATEWAY SETUP BOX.

Assuming a scenario where the knowledgebase has matured and contains comprehensive problem solving procedures. By moving the knowledgebase to home network – gateway layer (Section 2.2), or embedding it into *XML Messaging Server*, certain form of autonomic computing capabilities (Autonomic, 2004) can be introduced at the gateway setup box (e.g. broadband router). Procedures in the knowledgebase can be semantically enhanced and converted into reusable tasks, towards achieving the autonomic computing vision of a network which has self-configure, self-healing and self-optimised capabilities.

4. Implementation of Prototype

The layout of our testbed to run experiments is presented in Figure 2. The proposed architecture implementation framework, which includes the *gateway layer* and *Wiki Knowledgebase*, operates in the gateway machine. A set of managed devices (with and without logs built in the devices) are connected to the hub. For demo purposes, we assume a jabber account for John Smith has been created, and he uses Adium (Adium, 2008) as IM client to monitor and manage network devices as in Figure 2.

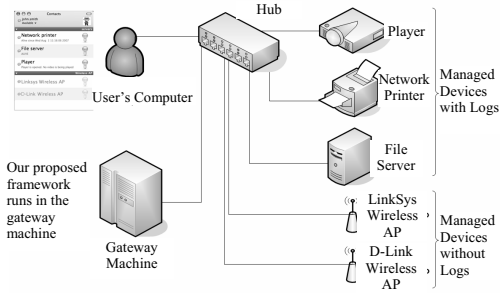

Figure 2. Test bed setup for our prototype.

The following subsections discuss three features which are demonstrated in our prototype in the testbed environment.

4.1 UPDATES OF DEVICES' PRESENCE STATUSES

IM client, which is used for device monitoring, is shown in Figure 3. The managed devices are displayed as buddies in the user's IM client. The presence status of each device is presented through the IM client.

4.1.1 Managed device without logs

Status probe component (as shown in Figure 1) sends periodic probing commands to the managed devices without logs to check the availability. In the case where probing command gets reply from the managed device, it means the device is available in the network. In other words, if the probing command times out, the managed device will appear offline in the IM client.

4.1.2 Managed device with logs

For devices with logs, logging system extracts the log data from the devices and sends the extracted log data to log parser module. Log parser module reads the data and interprets the data to get the availability of the device. The

data is sent to XML Messaging Server. The server will then update the IM client the presence status of the managed devices.

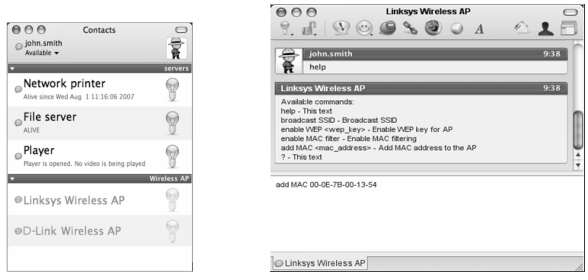

Figure 3. Instant Messaging (IM) client and message box to send remote command.

4.2 REMOTE CONTROL OF DEVICES USING SIMPLE COMMAND

Figure 3 shows how user can send simple command through instant messaging to remotely control a managed device. For example, user types *add_mac 00-0E-7B-00-13-54* in instant message box to add a MAC address to a wireless access point. Authorization is done by sending user name and password. Once user is authorized, command is entered and the server processes the command matching it to the appropriate function for the device. If the command is matched, a web scraping of devices' web administration page is done to get values stored in the devices and the server will execute the appropriate command such as sending a string of HTTP Post or Get method to the associated device. The device receives the string and processes it just like it processes from a form submitted from its web administration page. If the command is successful, the server will respond to the user by sending a message saying that the process is a success.

4.3 IMMEDIATE AND AUTOMATIC ALERT NOTIFICATION

An intrusion attempt is simulated, where an unauthorised user logs into the file server with an invalid name. The log parser detects an event of high severity, it generates a XML descriptor for the event. It performs a query to check the existence of similar event in the knowledgebase. The XML alarm description and query results are sent to XML messaging server and then being forwarded to the IM Client. The user will get notification of the event through instant messaging. File server appears as the sender of the instant message, which simply means that the event occurred in the file server

(Figure 5). By clicking on the hyperlink provided in the message, user is directed to the wiki page that shows troubleshooting steps for the event. The user can choose to provide better troubleshooting suggestion if he has any.

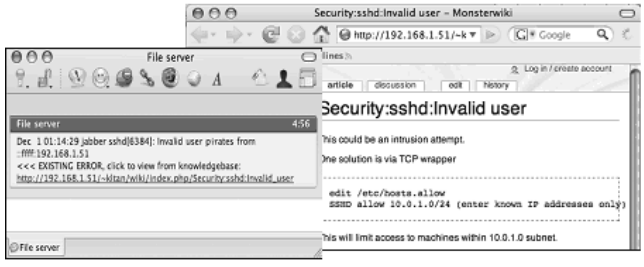

Figure 5. Instant message received with link to wiki knowledgebase.

5. Conclusion

In this paper, we have demonstrated that the proposed architecture helps to simplify home network management. Having the ability to monitor and manage the network built onto the instant messaging will not only make the interaction with network equipments natural, it also simplifies the actual task of managing the network and keeps the users informed "to-the-minute" about the status. With Wiki and log services used as a knowledge retention tool, the system can be used to reduce cost of network management within business units it allows novice users to self help in terms of managing the network. Because of instantaneous fault reporting, network/equipment faults and problems are brought to the attention to the users and the solution to address these problems are also at their fingertips in the form of a log/Wiki page of similar cases resolved in the past. In conclusion, we believe the proposed architecture will greatly simplify home network management..

References

Jabber: 2008, Jabber Software Foundation,[online] http://www.jabber.org/

XMPP: 2008, XMPP Standard Foundation, [online] http://www.xmpp.org/

Syslog-ng: 2008, Syslog-ng Logging System, [online] http://www.balabit.com/network-security/syslog-ng/

Autonomic: 2004, Autonomic computing, [online] http://www.ibm.com/developerworks/autonomic/newto/

Adium: 2008, Adium, [online] http://www.adiumx.com/