

A Network Management System for u-City

Jin Goo Kang*, Ju Wook Jang*, Chang Ho Yun**, Yong Woo LEE (Corresponding Author)**

*Department of Electronics Engineering, Sogang University, Korea

**School of Electrical & Computer Engineering, University of Seoul, Korea

zilkang@eeca1.sogang.ac.kr, jjang@sogang.ac.kr, {touch011, ywlee}@uos.ac.kr

Abstract— We propose a network management system which is suitable to disaster situation in U-city environment. U-city is a smart city with the intelligent convergence system that integrates IT technology and urban engineering technology such as environment, road traffic, safe and GIS. The network used in U-city should provide services without transmission failure and support functionality which enlarges the role of sensors used to collect information. Thus, a network management system which manages user's information is necessary. SNMP is the most widely used management protocol standard on the internet. But it manages a network has a static topology and the number of devices which are managed is small. To overcome SNMP problems, we propose an ubiquitous network management system (uNMS) that mobile devices in it is suitable in mobile network which go in and out U-city frequently and can manage lots of agents. And uNMS minimizes a usage of bandwidth. As shown in simulation results overheads are decreased about 40% and the connectivity is increased about 14%. When the mobility of mobile agents increases, the connectivity is 30% higher than SNMP.

Keywords-Network Management System, U-city, Ad-hoc

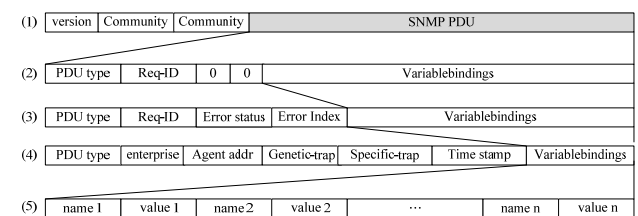
I. INTRODUCTION

A lot of attention has recently been paid to the establishment of the services such as intelligent disaster prevention, intelligent building management, intelligent health care, intelligent traffic control, and so on in metropolitan cities [1]. The services can be converged more efficiently and can be accessed more easily via ubiquitous networks [2]. The city which can offer the services based on ubiquitous networks is generally called u-City. The u-City consortium [3] is consisted of projects which are Ubiquitous Sensor Network, Convergence Network, Information Convergence, etc. And when a disaster occurs in U-city, it is changed into ad-hoc mode to keep providing services to users. User(mobile agent) in the u-City should be served information, anytime anywhere. the u-City is set as specific region so mobile node go in and out u-city frequently. Thus, network management system should serve mobile agents with proper quality information as monitoring information and status of mobile nodes in u-city. For the reason, a NMS which can manage these information is necessary.

Managing disaster networks is a significantly harder task than managing networks using conventional SNMP for several reasons.

First, the topology of an ad hoc network may be quite dynamic. Thus, automated “on-the-move” reconfiguration is necessary. [9]

Second, the SNMP server considers that the mobile agent is broken down or there exists a network connection problem when the server doesn't get a response from the mobile agent. In u-City environment, the mobility of mobile agents can cause high load to the server. [11] Thus, when the mobility of mobile agent is increased, the connectivity of mobile agent to the network management server is reduced.



(1) SNMP message (2) GetReq PDU, GetNextReq PDU, SetReq PDU (3) GetResponse PDU (4) Variablebindings

Figure 1. SNMP Format

Third, SNMP is difficult to manage many devices because of high bandwidth usage and information of agents is collected by a management station so the management station gets a high load. A get-request command can only acquire an MIB object value. The network performance index is normally obtained by combining many simple MIB objects. The collection of network variables is costly. It takes 351 bytes to acquire a single MIB object using system group in MIB-II, for example. If there are 100 network hosts in the centralized network management architecture and the MIB object is updated every five seconds, the bandwidth will be occupied by about 606MB per day (351 bytes X 100 hosts X 17,280 samples) [12].

When an AP is broken down due to a disaster, every mobile agent which can't connect to the AP turns on ad hoc mode and constructs ad hoc network [4][5][6][7]. Then uNMS(Ubiquitous Network Management System) can send disaster information to mobile agents. But the topology as well as the channel state of a network may be quite dynamic, uNMS should provide a suitable service to mobile agents. So we need a network management system which can monitor information of mobile nodes (e.g., bandwidth of a mobile agents, IP address, etc.)

II. PROPOSED SYSTEM

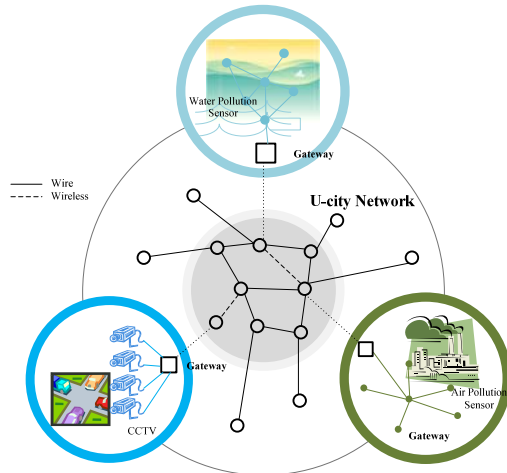


Figure 2. uNMS Structure in u-City

As shown in Figure 2, u-City is a system which manages a specific area automatically based on information such as water, air pollution and road traffic collected by sensors.

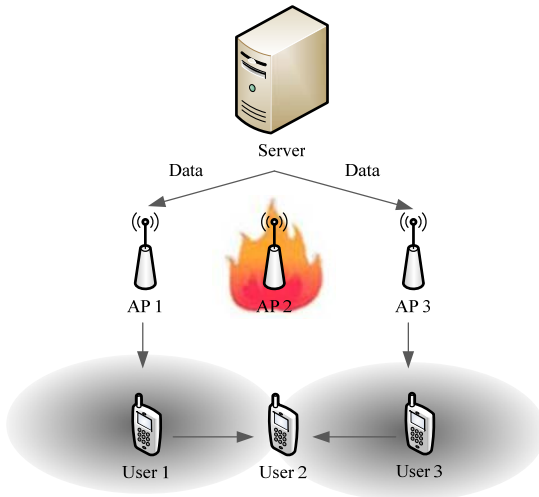


Figure 3. Transmission in disaster situation

In Figure 3, when a mobile node loses a connection to AP and can't find any available AP, mobile node turns on ad hoc mode to communicate with other mobile nodes. If any of the ad hoc network is connected to infrastructure network via AP, the stray mobile recovers connectivity to infrastructure network.

And server can send disaster information to mobile agents through ad hoc network. Though server and mobile agent connect to each other, because the bandwidth of ad hoc network is limited and channel state of each user changes dynamically, server can't send a high quality data to mobile

agents. So a server should provide a suitable service considering such constraints and need a management protocol which can manage mobile agents. We propose an uNMS which satisfies problems mentioned above. Conventional network management protocols (SNMP and ANMP) have many constraints such as loads of a server, limitation of a resource, so have a difficulty to use in a disaster situation. Messages which include information of mobile agents should be reduced when the number of mobile agents which are managed by uNMS gets larger, messages between an uNMS and mobile agents gets also increased.

And the SNMP considers a mobile agent as dead when the mobile agent doesn't respond to a message from the server. But in u-City environment that the mobility of mobile nodes is high, mobile nodes go in and out u-City frequently. For this reason, the uNMS tries to connect to a mobile agent which doesn't respond to a message of the uNMS for several times. When a mobile agent enters u-City again while the uNMS tries to connect to the disconnected mobile agent, the uNMS can save resources which may be used to get a status of the mobile agent. So uNMS can manage more effective than the conventional SNMP.

A. uNMS packet format

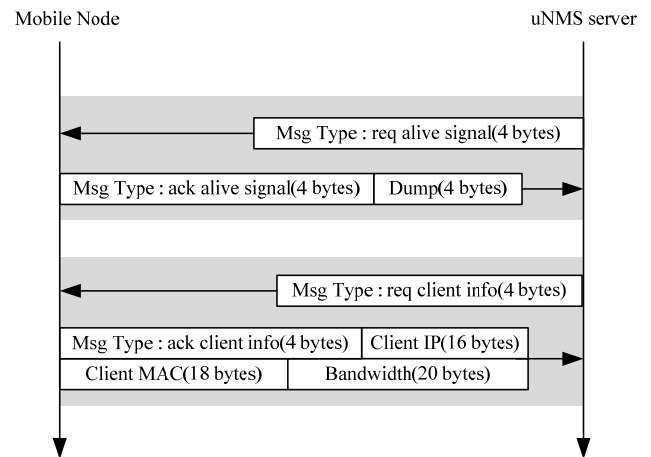


Figure 4. uNMS packet format

Figure 4 shows a data format and a process which is registration and exchanging messages between server and mobile agent. We implement the uNMS that can minimize a usage of bandwidth and it uses two packets. One is a packet which size is 4bytes and shows a status of a mobile node (alive or death). The other is a packet which size is 58bytes and includes information of a client (IP address, MAC address, and bandwidth). So, uNMS can make provide suitable service to each mobile agent.

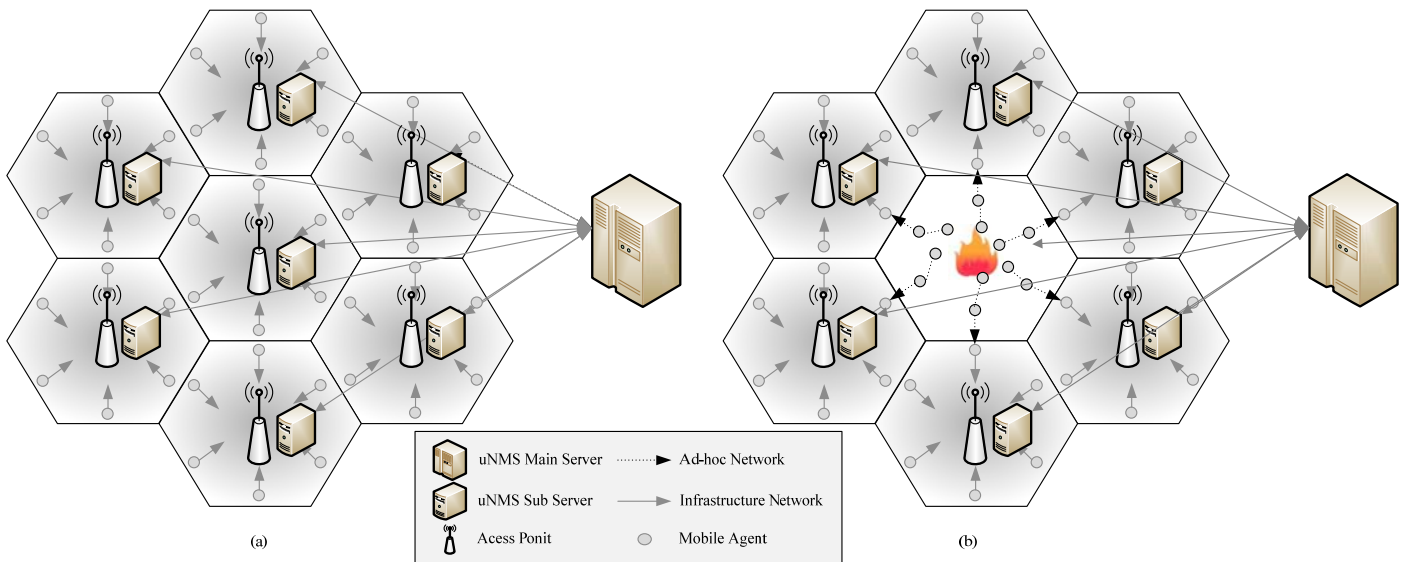


Figure 5. uNMS Structure

B. Alternative route selecting method using uNMS

In general, Figure 5-(a) shows the structure of mobile agent in u-City. The AP provides wireless communication services to the mobile agent under AP managing area and each mobile transmits message to uNMS sub-server attached AP. The sub-server collects data collected by mobile node and transmits it to uNMS server. However, as shown in Figure 5-(b) if a disaster situation arises in specific area, mobile agents under disaster area changed into ad-hoc mode. And they connect with surrounding mobile agents so that they can recover network. In this situation, the mobile node that has changed into ad-hoc mode transmits status information to uNMS main server via mobile node that receiving general network service.

C. uNMS diagram

Figure 6 shows a diagram of mobile node, uNMS sub server, uNMS main Server and disaster msg server. First, in general situation, mobile agents communicate with server through corresponding AP using installed uNMS client application and when disaster occurs, mobile agents which lose connection to AP turns on ad hoc mode. uNMS client application responds to alive message and information message and information message includes IP, MAC, bandwidth of mobile agent. uNMS Sub Server collects information of mobile agents and makes a data which is to be sent to uNMS server.

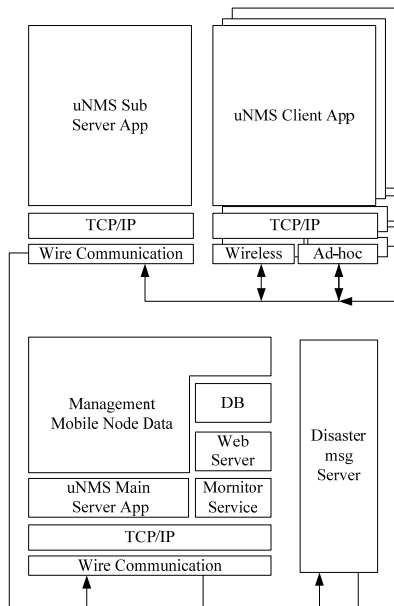


Figure 6. uNMS Diagram

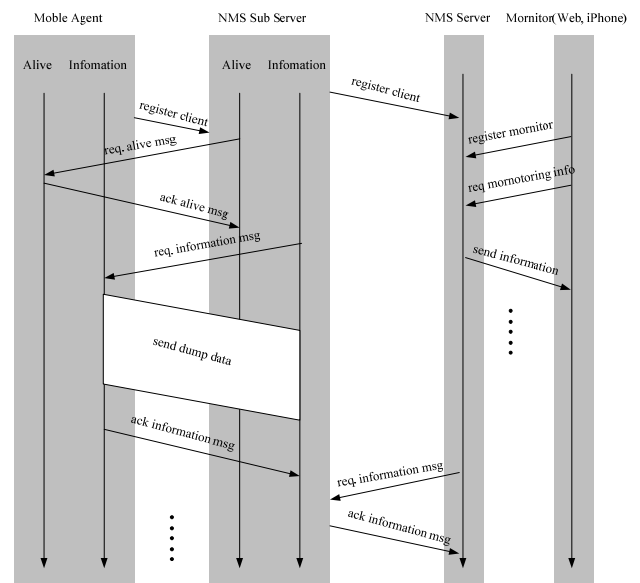


Figure 7. uNMS Packet flow

D. uNMS packet flow

Figure 7 shows packet flow diagram between a mobile agent, a sub-server, an uNMS server, and Monitoring devices. The mobile agent registers itself as sub-server that locates specific area. And sub-server transmits an alive-message to registered mobile agent and requests the status of mobile agent such as alive or death. The mobile agent transmits an alive message to the sub-server corresponding to transmission format. Then the sub-server transmits a message for requesting information to a mobile agent and it measures bandwidth using a dump message. It transmits bandwidth information such as IP, MAC information to the sub-server. Sub-server collects data and transmits to uNMS server. In addition, administrator can check the performance of a mobile agent in the u-City using Web-server or DB to provide monitoring service which is to see collected information.

III. SIMULATION

In this section, we simulate and verify the practicality of the network management system of the uNMS.

A. Simulation environment

We used LINUX system and C language for implementing uNMS system. The version of LINUX system is Ubuntu 9.04. We constructed main server and DB using apache and my-sql. Also, the application development OS is MAC OS X 10.5.7 and development tool is iPhone SDK 3.0. The programs used between mobile node and sub server, sub server and main server are employed both a client application and a server application. uNMS server application runs on daemon and is seen in Figure 8. This application acts in Interactive mode and makes a session if a client application tries to connect to a server. Then, the server application sends alive message and information message through client application to mobile agents periodically. When the mobile agents receive messages, the client application responses messages. This process is shown in Figure 8. Figure 9 shows the process which mobile agents turn on ad hoc mode when lose connection to an AP.

```
1005.12 000352::### Server start...
1005.12 000352::## IP Address(IPv4) of this systems is [IP:192.168.123.100] [Port:8989]
1005.12 000352::### Packet handler thread starting
1005.12 000353::SYMSG: Communication socket has successfully opened
1005.12 000353::SYMSG: Heartbeat loop started. Timer is now running
1005.12 000401::SYMSG: New client has connected
1005.12 000401::SYMSG: IP address of new client is "192.168.123.171"
1005.12 000412::SYMSG: Terminal information updated [IPv4:192.168.123.171] [MAC:00:11:11:BB:AA:E5]
[Signal:3.64 Gbytes ][Ad-hoc:none]
1005.12 000413::SYMSG: New client has connected
1005.12 000414::SYMSG: IP address of new client is "192.168.123.181"
1005.12 000426::SYMSG: Terminal information updated [IPv4:192.168.123.181] [MAC:00:11:1F:D8:E2:B1]
[Signal:3.42 Gbytes ][Ad-hoc:none]
```

Figure 8. Server log

```
1005.12 000401::### Client start...
1005.12 000401::## Client start...
1005.12 000401::SYMSG: Trying to connect to [IP:192.168.123.100] [Port:8989]
1005.12 000401::SYMSG: Connected to server [IP:192.168.123.100] [Port:8989]
1005.12 000401::SYMSG: Packet handler loop start
1005.12 000401::SYMSG: Received the connected signal. Connection OK!
1005.12 000412::SYMSG: Sent terminal information to server
1005.12 000412::SYMSG: Sent alive signal to server
```

Figure 9. Client log

```
1005.12 000722::SYMSG: Terminal information updated [IPv4:192.168.123.171] [MAC:00:11:11:BB:AA:E5]
[Signal:3.82 Gbytes ][Ad-hoc:none]
1005.12 000734::SYMSG: Terminal information updated [IPv4:192.168.123.251] [MAC:00:11:1F:D8:E2:B1]
[Signal:2.45 Gbytes ][Ad-hoc:true]
```

Figure 10. Mobile agent turn on ad hoc mode

B. Performance of the uNMS

In this section, we compare the size of data using uNMS and SNMPv2 as increasing the number of mobile agents. In the case of SNMP, 10 packets and 824 bytes of data [10] are needed but in the case of uNMS, 4 packets and 505 bytes of data are needed. Thus, when the number of mobile agents increases, uNMS can reduce overhead about 40% than SNMPv2.

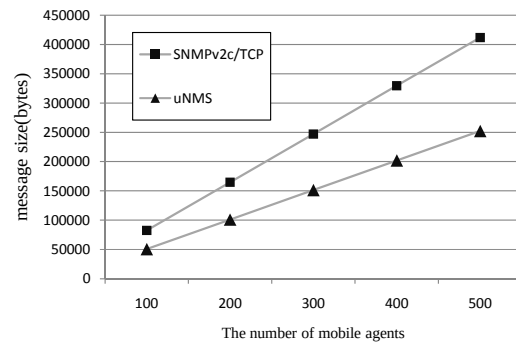


Figure 11. Message size between uNMS and SNMPv2/TCP

Figure 12. shows that the performance comparison between the uNMS and SNMPv2/TCP when the number of retrying is set to 3. There are 9 APs and 4000 mobile agents. When 5 APs are broken down due to the disaster, the connectivity of mobile agents is 14% higher than the SNMP.

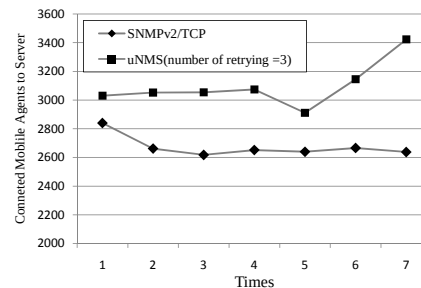


Figure 12. Connectivity of mobile agents

Figure 13. shows the number of connections when changing the mobility of mobile agents in both SNMPv2/TCP and uNMS. In the case of SNMP, because it removes a mobile agent which is disconnected in a management list, the number of connections decreases when the mobility of mobile agents

increases. But the uNMS tries several times to connect to a mobile agent which is disconnected and for this reason, the connectivity is 30% higher than the SNMP.

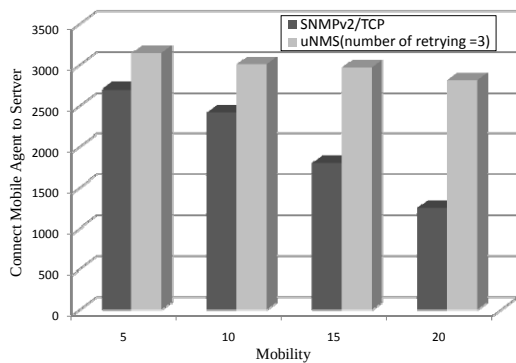


Figure 13 Mobility of mobile agents

IV. CONCLUSION

We propose the uNMS which manages status and bandwidth of mobile agents to send disaster information through ad hoc network when mobile nodes can't connect to APs due to disaster. We implement our proposed uNMS in LINUX system using C language and show that our uNMS is effective through simulation results in our testbed. As shown in simulation results, we reduce the load of main server using uNMS sub-server and develop client application in ad hoc mode. If uNMS requires additional information of mobile agents, we will improve uNMS which can dynamically collect information.

ACKNOWLEDGMENT

This study was supported by the Seoul Research and Business Development Program (10561), Smart (Ubiquitous) City Consortium and Seoul Grid Center.

REFERENCES

- [1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, Wireless Sensor Networks: A Survey. *Journal of Computer Networks* 38 (2002) 393–422.
- [2] N. S. Correia, and N. Patwari, Wireless Sensor Networks: Challenges and Opportunities. *Proc. of Virginia Tech Symp. Wireless Personal Comm.* (2001) 1–9.
- [3] SMART (UBIQUITOUS) CITY CONSORTIUM - Introduction , "http://ucity.uos.ac.kr".
- [4] Weiquan Lu, Winston K.G. Seah, Edwin W.C. Peh, Yu Ge, "Communications Support for Disaster Recovery Operations using Hybrid Mobile Ad-Hoc Networks," *lcnn*, pp.763-770, 32nd IEEE Conference on Local Computer Networks (LCN 2007), 2007.
- [5] Jungchae Kim, Doyoon Kim, Seok Myung Jung, ChungK Lee, Dongkyu Lim, SungHye Hong, Yoo, S.K., "Development of Mobile Ad Hoc Network for Emergency Telemedicine Service in Disaster Areas," *New Trends in Information and Service Science*, 2009. NISS '09. International Conference on, pp.1291-1296, June. 2009.
- [6] J. Chang, J. Jang, "BECT: A Balanced Energy Consumption Algorithm by Threshold-Tuning for Mobile Ad Hoc Networks", 13th Joint Conference on Communication and Information 2003 (JCCI 2003), Apr. 2003.
- [7] Kanae SAKAMOTO, Misako URAKAMI, Tetsuya SHIGEYASU, Hiroshi MATSUNO, "Disaster Information Service System for Relief Activities using Ad-Hoc Network -A Network Construction Algorithm and Results of Field Experiments-, " 22nd International Conference on Advanced Information Networking and Applications, March. 2008.
- [8] Stallings, W. *SNMP, SNMPv2, SNMPv3, and RMON 1 and 2*, 3rd edition; Addison-Wesley: Reading, MA, 1999.
- [9] Wenli Chen, Nitin Jain, "ANMP: Ad Hoc Network Management Protocol," *IEEE Journal on Selected Areas of Communications*, vol. 17, no. 8, PP.1506-1531, August 1999.
- [10] Jurgon Schonwalder, "Performance of SNMP over SSH/TLS/DTLS,"
- [11] Deuk-Whee Kwak, HyunYong Lee, JongWon Kim, "SNMP-based Management for Mobile Network Devices" *JCCI*, vol. 33, no. 7 2008-2007
- [12] T.C. Du, E.Y. Li and A.P. Chang, "Mobile agents in distributed network management." *Communications of the ACM* 46 7 (2003), pp. 127–132.