IEEE 802.11-based Wireless Mesh Network Testbed

Heecheol Song, Bong Chan Kim, Jae Young Lee, and Hwang Soo Lee
Division of Electrical Engineering, School of EECS, KAIST
373-1, Guseong-dong, Yuseong-gu, Daejeon, 305-701, Republic of Korea
Tel: +82-42-869-5428, Fax: +82-42-869-8670
E-mail: practice00@mcl.kaist.ac.kr

Abstract—In this paper, an IEEE 802.11-based wireless mesh network testbed is developed to evaluate the performance of wireless mesh networks in real environment. The proposed testbed consists of mesh routers and mesh clients, where the mesh routers have minimal mobility and form the backbone of the wireless mesh network. In order to build the proposed 802.11-based wireless mesh network testbed, both an ARM-based embedded Linux development board and an Intel x86-based laptop PC were used to implement the mesh routers and mesh clients. Multimedia services such as voice, video and text were also used in order to confirm the functions of the testbed components, the mesh routers and mesh clients.

Keywords—IEEE 802.11, wireless mesh networks, mesh router, mesh client

I. INTRODUCTION

In the last few years, much of the research concerning multi-hop wireless networks has been focused on mobile ad hoc networks. In spite of this enormous effort in this area, it remains to be deployed in the marketplace. Thus, in order to activate mobile ad hoc networks, many of the wireless network researchers are interested in wireless mesh networks, which are dynamically self-organized and self-configured networks with the nodes in the network automatically establishing and maintaining mesh connectivity among themselves [1].

Wireless mesh networks are a type of the wireless multi-hop network consisting of mesh routers and mesh clients, where the mesh routers have minimal mobility and form the backbone of the wireless mesh network. Recently, numerous companies have provided a large variety of products related to wireless mesh networks and researchers everywhere continue to develop these networks.

However, many of the wireless mesh networks have undergone research and development in computer simulations. The limited fidelity of simulators has prompted researchers to build wireless network testbeds to facilitate realistic testing. Consequently the need for wireless mesh network testbeds is growing.

There are two objectives in this paper. The first ultimate objective is to design and implement mesh routers and mesh clients for wireless mesh networks.

The second objective is to develop an IEEE 802.11-based wireless mesh network testbed in order to evaluate the performance of wireless mesh networks in a real environment. The proposed testbed consists of mesh routers and mesh clients.

In order to build this testbed, the proper functions necessary for mesh routers and mesh clients were used and modified. An ARM-based embedded Linux development board and an x86-based laptop PC were used to implement mesh routers and mesh clients. Also used were multimedia services such as voice, video and text to confirm the functions of testbed components.

The remainder of this paper is organized as follows. In the following section, related works are presented. In Section 3, the components of IEEE 802.11-based wireless mesh networks are described. The development procedure of the mesh routers and mesh clients is presented in Section 4 and 5, respectively. In Section 6, the process of building the proposed IEEE 802.11-based wireless mesh network testbed using the developed mesh routers and mesh clients is introduced. In addition, the multimedia service test used to verify if the necessary functions are sufficiently provided and implemented is shown. This paper concludes with Section 7.

II. RELATED WORKS

Recently, proprietary solutions are made by several manufactures for wireless mesh networks. Some vendors (e.g., Tropos, BelAir, Firetide, LocustWorld and Strix) initially focused on IEEE 802.11 for the radio technology, but used proprietary software solutions [1]. In the case of other vendors such as Radiant and MeshNetwork, proprietary radio technologies are used instead of IEEE 802.11 radio technology. Therefore these different approaches and protocols make their systems incompatible [1].

Several wireless mesh network approaches have been suggested and used in the research community. The MIT Roofnet [2] and Microsoft Research [3] are the most suitable examples [4]. Roofnet is an experimental 802.11b/g mesh network in development at MIT CSAIL, which provides broadband Internet access to users in Cambridge, MA [2]. There are approximately 20 active nodes on this network. Microsoft Research makes a similar testbed with Roofnet which consists of IEEE 802.11 interfaces in ad hoc mode and ad-hoc routing and link quality measurement in a module that we call the Mesh Connectivity Layer (MCL) [3]. However, these two projects only consider the backbone network, and do not consider mobility management [4].

Recently, the IEEE is developing a set of standards to define an architecture and protocol for ESS Mesh Networking. The
P802.11s ESS Mesh Networking Task Group (TG) actively defines specifications for wireless mesh networks [1, 5].

III. COMPONENTS OF IEEE 802.11-BASED WIRELESS MESH NETWORKS

As stated above, wireless mesh networks consist of two types of nodes, a mesh router and a mesh client [6, 7].

A. Mesh Router

A mesh router functions as a wireless router and contains additional functions such as multi-hop forwarding of packets and mobility support other than the basic routing capability [6]. It forms the backbone of wireless mesh networks. To build wireless mesh networks, a mesh router is equipped with at least two wireless interfaces. One interface is for communication with backbone networks and the other is for communications with mesh clients.

For a mesh router to communicate with mesh clients in an IEEE 802.11 wireless LAN, it should be seen as an access point for mesh clients. Thus, a mesh router should support the function of an IEEE 802.11 access point. It should also provide an appropriate routing protocol for multi-hop communications of wireless backbone networks. The last function of a mesh router is related to mobility management to support the mobility of mesh clients.

B. Mesh Client

A mesh client needs necessary capabilities in order to build a wireless mesh network. In contrast to a mesh router, however, a routing function does not exist due to a lack of support for gateway or bridge functions. A mesh client usually has one wireless interface; therefore, the hardware platform as well as the software for a mesh client can be much simpler than those for a mesh router [7].

As can be seen, the structure of a mesh client is simpler than that of a mesh client; however, it should be able to play a role as a mobile node when it moves. Therefore, it is necessary to support the functions of both a low-delay handover in the data-link layer and the mobility management in the network layer.

IV. DESIGN AND DEVELOPMENT OF THE MESH ROUTERS

In this section, the development procedure of the mesh routers, as well as how each necessary function is implemented, is presented. Fig. 4.1 shows the overall design and development process of the mesh routers.

A. Hardware and Software Components of Mesh Routers

Intel x86-based laptop PCs and FALINUX EX-X5 + EXPCCMIA boards were used to implement mesh routers. The EZ-X5 board is an ARM-based embedded Linux development board based on an Intel 400 MHz PXA255 processor with 64 MB of SDRAM. The EZ-PCMCIA is a sub-board that supports PCMCIA and CF slots.

A CF slot was used as a wireless interface for a wireless backbone network and the PCMCIA slot as the wireless interface for the IEEE 802.11 access point. For a mesh router to act as an access point, Host AP was utilized, which is a Linux driver for wireless LAN cards based on Intersil’s Prism2/2.5/3 chipset. Hence, when a PCMCIA wireless LAN card is selected, it is necessary to consider whether or not the card supports Host AP. For the PCMCIA wireless card, Linksys WPC11 Ver. 3, which is based on Intersil’s Prism chipset, was used. An AcerLAN AWL-1100CF CF card was used to construct the wireless backbone networks. It is also based on Intersil’s Prism chipset.

Fig. 4.2 shows at which software layer each function was implemented in the mesh routers. A Samsung x86-based laptop PC running Redhat Fedora Core 4 distribution with a 2.6.11-1.1369_FC4 kernel and FALINUX EZ-X5 board running Redhat Linux 9 distribution with 2.4.19-x5-v07 kernel in which FALINUX supplies were used. As can be seen in Fig. 4.2, IP Forwarding and Proxy ARP functions are located in the TCP/IP stack. Host AP is configured in the Linux device driver and OLSR for the multi-hop routing of a wireless backbone network is run as a type of daemon.

B. Proxy ARP and IP Forwarding

For organizing wireless mesh networks, mesh routers essentially need two wireless interfaces, one for communications with a wireless backbone network and the other for communications with mesh clients. For two different networks in these two interfaces to be used to communicate with each other via mesh routers, mesh routers should support two functions, Proxy ARP and IP forwarding.

The mesh client connected to the mesh routers broadcast ARP request messages to reach the destination in a wireless backbone network. However, the mesh client’s network and the destination’s network are different, thus the mesh router between them will not pass the mesh client’s broadcast to the destination’s part of the network, as a mesh router does not pass hardware-layer broadcasts[8].

The destination will never receive the request; thus the mesh
client will not receive a reply containing the destination’s hardware address. This issue is resolved by Proxy ARP. As they are not on the same network, the host cannot send directly to the destination in any case. Instead, the mesh router sends the mesh client its own hardware address. The mesh client then sends the message to the mesh router, which forwards the message to the destination on the other network. Evidently, the mesh router repeats the same action on the mesh client’s behalf for the destination, and for every other device on both networks when a broadcast is sent that targets a device not on the same actual physical network as the resolution initiator. Accordingly, the mesh router that receives an ARP request message sends its own MAC address instead of the destination’s MAC address by carrying it to the ARP reply message. This is possible as the mesh router that receives an ARP request message has the information encompassing how it can reach the destination. As mentioned before, mesh routers have two interfaces and make two different subnets. Therefore, Proxy ARP should be supported when constructing mesh routers.

The second capability needed by mesh routers is IP forwarding. The successful implementation of IP forwarding enables one computer to sit on two networks and to act as a gateway forwarding IP packets from one network to another [9].

C. Functions of IEEE 802.11 Access Point

For mesh clients to communicate with mesh routers, mesh routers function as access points for mesh clients to connect to the networks. For this, Host AP should be used [10].

Host AP is a Linux driver for wireless LAN cards based on Intersil’s Prism2/2.5/3 chipset. The driver supports what is termed the Host AP mode, i.e., it handles the IEEE 802.11 management functions in the host computer and acts as an access point. This does not require any special firmware for the wireless LAN card [10].

Thus, by using Host AP, the function of an IEEE 802.11 access point is supported.

D. Mobility Management in Network Layer

When one mesh client moves from an attached network to another network, there is no means to guarantee the macro-motion of that mesh client in an IEEE 802.11-based wireless LAN. As a result, the management of mobility in network layers is needed in order to support the mesh client’s mobility. To develop the function of mobility management, a mobile IP is necessary as a solution, and this grants great freedom to mesh clients [11].

Mobile IP provides mesh clients the freedom to roam beyond their home subnet while consistently maintaining their home IP addresses [12, 13]. This enables transparent routing of IP datagrams to mobile users during their movement, so that data sessions can be initiated to them while they roam. It also enables sessions to be maintained in spite of any physical movement between points of attachment to the Internet or other networks.

As a method of mobility management, Dynamics Mobile IP [14] was used. The Dynamics Mobile IP system, originally developed at the Helsinki University of Technology (HUT), is scalable, dynamical, and hierarchical Mobile IP software for the Linux operating system [14]. As mesh routers function both as home agents and as foreign agents, configuring the role of mesh routers should be straightforward. Dynamics Mobile IP supports this feature. Only a modification of the configuration files for FAs and HAs was needed.

Thus, by using Dynamics Mobile IP, the mobility management in network layer is supported.

E. Routing Protocol

To construct backbone networks in wireless mesh networks, it is necessary to provide the appropriate routing protocol for the multi-hop communications of backbone networks.

Table-driven routing protocols are recommended, as a periodic routing table update reduces the route setup time and is more appropriate for delay sensitive services such as Voice over IP [15]. Among table-driven routing protocols, Optimized Link State Routing (OLSR) was select as a routing protocol for the backbone networks in the proposed testbed [16].

For implementation olsr.org OLSR daemon (Olsrd) was used [17]. The Olsrd is an implementation of the Optimized Link State Routing protocol [17]. OLSR is a routing protocol for mobile ad-hoc networks. The protocol is pro-active, table driven and utilizes a technique called mulepoint relaying for message flooding. Olsrd also implements a popular optional link quality extension. Currently, the implementation compiles on GNU/Linux, Windows, OS X, FreeBSD, OpenBSD and NetBSD systems [17].

Olsrd is meant to be a well-structured and well-coded implementation that is easy to maintain, expand and port to other platforms. The implementation is RFC3626-compliant with respect to both core and auxiliary functioning [17].

As was mentioned above, Olsrd is a routing protocol for mobile ad-hoc networks; consequently, pure Olsrd is not suitable for the proposed testbed, as the mesh clients are not running a routing protocol. Therefore, Olsrd was modified such that when a specific address is updated in the routing table, the network address which includes that address is also updated. The modified Olsrd is presented in Fig. 4.3.
V. DESIGN AND DEVELOPMENT OF THE MESH CLIENTS

In this section, the development procedure of mesh clients as well as the method for implementing each necessary function is presented. Fig. 5.1 shows the overall design and development process of the mesh clients.

A. Hardware and Software Components of Mesh Client

An Intel x86-based laptop PCs was utilized to implement the mesh clients, and a PCMCIA wireless LAN card was used as the wireless interface of the IEEE 802.11 managed mode.

Fig. 5.2 shows at which software layer each function was implemented in the mesh clients. Samsung x86-based laptop PCs running Redhat Fedora Core 4 distribution with the 2.6.11-1.1369_FC4 kernel were used. As can be seen in Fig. 5.2, the selective scanning algorithm for fast handoff is configured in the IEEE 802.11 Linux device driver, and mobile IP for mobility management in the network layer is run as a type of daemon.

B. Functions of Handoff

As can be seen in reference [18], the probe delay accounts for more than 90% of the overall handoff delay. In this paper, selective scanning is used instead of full scanning to reduce the handoff delay time.

Fig. 5.3 shows the principle of the selective scanning. The principle of the scanning algorithm used here is similar to that in reference [19], except the cache is not used, and the fact that most of the users in actuality use the non-overlapped channels 1, 6 and 11 is applied.

C. Mobility Management in Network Layer

Similar to mesh routers, Dynamics Mobile IP is used as a method of mobility management [14]. As mesh clients function as mobile nodes, it should be straightforward to configure the role of mesh clients. Dynamics Mobile IP supports this feature. Only a modification of the configuration files for MN is needed.

Therefore, by using Dynamics Mobile IP, mobility management in the network layer is supported.

VI. IEEE 802.11-BASED WIRELESS MESH NETWORK TESTBED USING MESH ROUTERS AND MESH CLIENTS

The process of building the proposed IEEE 802.11-based wireless mesh network testbed is introduced in this section. Also shown is a multimedia service test to verify if the necessary functions are sufficiently provided and implemented.

A. IEEE 802.11-based Wireless Mesh Network Testbed

As seen in Fig. 6.1, the IEEE 802.11-based wireless mesh network testbed consists of three mesh routers and two mesh clients. MR(HA) acts as a HA of MC(MN). MR(FA1) and MR(FA2) act as FAs of MC(MN). MC(MN) and MC(CN) maintain the connection with each other and support multimedia services while they change networks.

An x86-based laptop for MR(FA1), MR(FA2), MC(MN) and MC(CN) and ARM-based embedded Linux development board was used for the WMR(HA). The actual topology of the IEEE 802.11-based wireless mesh network testbed is shown in Fig. 6.2.

B. Multimedia Service Test

To verify if the necessary functions are sufficiently provided and implemented in the proposed testbed, a certain type of test should be performed. In this paper, a multimedia service test using a multimedia application is performed.
As an application for the multimedia service test, Gnomemeeting was used. Gnomemeeting is an open source VoIP and video conferencing application. It supports many audio and video codecs, and is interoperable with other SIP compliant software and also with Microsoft NetMeeting [20].

Fig. 6.3 (a) shows the testbed when MC(MN) is communicating with MC(CN) through MR(HA) and MR(AF1). MC(MN) and MC(CN) communicate with each other, and it is possible to test the communication using Gnomemeeting. It is clear that the necessary functions of the mesh routers and mesh clients, i.e. Proxy ARP, IP forwarding, mobility management and AP functionality, are operating correctly in this test. Fig. 6.3 (b) shows the testbed after MC(MN) changes the network from Foreign Network 1 to Foreign Network 2. It was also observed that the connection is not maintained by using a mobile IP.

VII. CONCLUSION

In this paper, an IEEE 802.11-based wireless mesh network testbed was developed in order to evaluate the performance of wireless mesh networks in a real environment. In order to develop the 802.11-based wireless mesh network testbed, mesh routers and mesh clients were needed.

In order to develop mesh routers, several necessary functions should be supported. These are multiple wireless interfaces, IEEE 802.11 access points, mobility management and a routing protocol. For this reason, each function was implemented with appropriate open source software and was then modified until they were suitable for the proposed testbed.

For a mesh client, a low-delay handoff in data link layer and mobility management are necessary capabilities for a mesh client to work as desired. Therefore, a selective-scanning method instead of a full-scanning method was implemented into the network device driver. Mobile IPv4 was used for mobility management.

In order to build the proposed IEEE 802.11-based wireless mesh network testbed, an ARM-based embedded Linux development board and x86-based laptop were used to implement mesh routers and mesh clients. Multimedia services such as voice, video and text were also utilized to confirm the functions of the testbed components, mesh routers and mesh clients.

ACKNOWLEDGMENT

This research was supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Assessment).

REFERENCES