Regular Paper

On Providing Secure and Portable Wireless Data Networking Services: Architecture and Data Forwarding Mechanisms

Lusheng Ji,† Jonathan Agre,† Tadashige Iwao†† and Nobutsugu Fujino††

In this paper we address the need for secure and portable wireless data networking. Our solution, named the Secure Nomadic Wireless Network (SNOWNET) is a hierarchical network consisting of a dynamic, multi-hop, wireless backbone network interconnecting a number of local access service areas. SNOWNET provides a secure, quickly deployable, modular networking infrastructure for many networking applications such as extending existing networks to environments with no existing trusted infrastructure as in battle field situations, disaster relief operations, or temporary events (e.g., conventions, parades, fairs, etc.). Design and implementation aspects of SNOWNET nodes, which use IEEE 802.11 WLAN technology to form the backbone network as well as provide local access services, are discussed. The overall architecture of the SNOWNET, the data forwarding protocols executed by SNOWNET nodes, and the functions of several types of SNOWNET nodes are described.

1. Background

Wireless Local Area Networks (WLANs) have been enjoying growing popularity since they were first introduced in the 90's. Its deployment scale, cost, data rate, and timing are ideal for many applications.

Among different WLAN communication specifications, the IEEE 802.11 standard [1] is currently the most popular. Most 802.11 WLANs are deployed in "infrastructure" mode. In this mode, WLAN Access Points (APs) are connected to a backbone network known as the Distribution System (DS). Mobile clients then connect wirelessly to these APs to acquire connectivity to the DS. APs are typically configured as bridging devices to forward data frames between the DS and mobile clients. In most deployments, the DS of the WLAN is a wired LAN (e.g., Ethernet) which connects to the rest of the organizational network or the Internet.

Thanks to a number of dynamic configuration protocols such as the Dynamic Host Configuration Protocol (DHCP) [2] and various operating system-level supports, mobile clients can easily join a network via WLAN links with little or no user configuration effort. A client may also move seamlessly across the boundaries of the coverage areas of different APs. WLANs have gone beyond the simple convenience of relieving users from the restrictions of wall jacks and network cables and have become the foundation of the new ubiquitous computing era.

However, WLANs also have certain problems that need to be properly addressed and resolved or else the potential of WLANs will be seriously limited.

First, it is not a trivial task to deploy an organizational WLAN from scratch. Wires and wall jacks need to be installed so that APs can be connected to the wired LAN. The total cost of the wiring related installation and maintenance often exceeds the cost of WLAN hardware. Locations without nearby wired network infrastructure often become WLAN service "dark spots". The channel and transmission power settings for each AP also need to be planned and configured. Limited by these requirements, WLAN deployment is usually static and long term, and only as direct (one-hop) extensions of the wired network. Its configuration can not be easily adjusted to satisfy changing usage patterns. WLAN also lacks the capability of being rapidly deployed on demand.

Secondly, another serious and well known problem with the 802.11 WLAN lies in its current security mechanism. Due to its broadcast nature, 802.11 must apply encryption to sensitive communication sessions so that only the intended recipients can reconstruct and comprehend the data. The security protocol is known as the Wired Equivalency Privacy (WEP) protocol. However, as many researchers have pointed out, WEP fails to deliver the security protection it promises. Various attacks such

† Fujitsu Laboratories of America, USA
†† Fujitsu Laboratories LTD, Japan

as [3] have been published to defeat WEP's authentication, access control, data integrity, and replay prevention mechanisms.

Given the above pros and cons of current WLAN systems, our goal was to design a 802.11 wireless network device that is capable of providing rapidly deployed, self-configured, portable, and secure wireless network access service to mobile users. Our network system is named the Secure Nomadic Wireless Network (SNOWNET) and is formed by SNOWNET nodes equipped with multiple wireless interfaces. These nodes are capable of providing multiple communication services (i.e., backbone service and local client access service) at the same time.

The SNOWNET installation process can be reduced to the placing of SNOWNET nodes in the field of operation, powering them up, and optionally orienting the external antenna attached to these nodes to connect to other SNOWNET nodes. Configuration parameters, such as the identity of neighboring devices or address assignments, will be determined autonomously by the collaborative operations of a set of such SNOWNET nodes. A multi-hop routing topology for data forwarding supporting both inter- and intra-SNOWNET messaging is automatically determined. One can easily move the SNOWNET nodes to create differing topologies in response to changing needs.

Another feature is that the communication between SNOWNET nodes as well as between SNOWNET nodes and mobile clients is highly secure. Only authorized devices (both SNOWNET nodes and mobile clients) are allowed to access and be served by the SNOWNET. The security mechanism of SNOWNET is an extension of the IEEE 802.1X specification [4]. Our extension supports dynamic keying for both client and for multi-hop backbone communications and is compatible with the relevant components of the Wi-Fi Protected Access (WPA) standard.

Due to the size limitation of this paper, we will only focus on the system architecture, data forwarding mechanisms, and the design and implementation of the SNOWNET system in this paper. Security related designs are addressed in detail by [5].

2. Related Works

The more typical early examples of SNOWNET related technologies are WLAN wireless extension devices, which are often labeled as WLAN repeaters or WLAN bridges. Their basic function is to retransmit a link layer data frame, which is received over WLAN interface, over either the same WLAN interface or a wired LAN interface. The most typical application was connecting separate LAN segments together over 802.11 forwarding links. Examples of this type of devices include the outdoor router products (COR/ROR series) by Orinoco [13] and Cisco's Aironet1400 Wireless Bridges [14]. Later, low cost Small Office Home Office (SOHO) WLAN products began to enter this market as well. However, these low-end products typically lack the capability of chaining more than one wireless forwarding link to form a wireless multi-hop forwarding path. A popular example of this period was the WET11 Wireless Ethernet Bridge by Linksys [15].

A troublesome factor in the use of these WLAN extension products is that the forwarding links are statically configured and therefore the topology of the network may not be dynamic. Another issue is that oftentimes devices made by different manufacturers are not interoperable. The basic 802.11 standard did not support bridging of data frames because its data frame format can not preserve the MAC address of the data frame's original source during forwarding. Manufacturers use different proprietary solutions, resulting in interoperatibility issues. In addition, in many WLAN extension solutions, data frames are simply retransmitted. As a result, data frames are flooded (broadcast) to the whole network. Hence this solution does not scale well.

A more recent development of WLAN extension products came from the firmware support of Wireless Distribution System (WDS) functions. WDS, previously not implemented by firmware vendors, is the part of the 802.11 standard specifying the method for APs to directly communicate with each other over wireless communication. As a side effect, WDS frame format has the additional space to hold a data frame's original source MAC address. Thus it may be used to implement 802.11 bridging. Still, in these products the WDS links need to be configure by hand and are not dynamic

An alternate form of multi-hop wireless network in which client devices communicate with each other directly and form spontaneous peer-to-peer links is beginning to emerge from the research community and into commercial prod-

ucts. In this type of networks, clients may move about at will, hence the name of Mobile Ad hoc Network (MANET)[16]. Client devices will store-and-forward data packets for each other under the guidance of a network-wide routing algorithm so that data packets can be delivered to their final destinations over multiple wireless hops. Some commercial examples include MeshNetworks[17], GreenPacket[18], and Moteran[19].

This type of networks is very flexible, but they have some common drawbacks. First, they require clients to install special software packages (and sometimes hardware) to participate in data forwarding. Second, client devices may need to spend valuable battery energy on forwarding data packets for other devices. Third, all client devices need to be put on the same communication channel so that the radio spectrum can be efficiently used. Lastly, very little security options (other than the static WEP method) are available for this type of networks.

SNOWNET combines the benefits of MANETs and infrastructure mode WLANs. However, unlike WLAN extension networks, SNOWNET's multi-hop wireless backbone is dynamic and data forwarding is efficiently and automatically configured. Also, unlike MANETs, SNOWNET avoids the issues of requiring special client software/hardware, altruistic client energy consumption, and forced channel sharing by separating the backbone links from the AP-to-client links.

3. SNOWNET Architecture

Figure 1 shows an example architecture of SNOWNET. SNOWNET forms a two-layer hierarchical network, whose design addresses the issues discussed in the previous section and maintains the compatibility with the large number of existing WLAN client devices. In the bottom layer, SNOWNET nodes provide standard WLAN access services to regular WLAN clients with standard WLAN interface cards. Hence, normal clients may connect to SNOWNET in the same fashion as they connect to any other standard WLAN. No additional hardware or software modification is required by the client. In the top layer, SNOWNET nodes form a wireless backbone network among themselves. Thus the deployment topology of the wireless network is no longer constrained by the wired network infrastructure, permitting changes in SNOWNET node locations. Links between

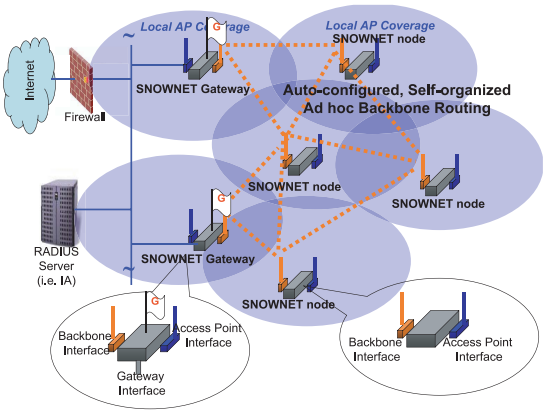

Fig. 1 Sample SNOWNET architecture.

nodes are dynamically established subject to the communication parameters and range constraints of the physical environment. The SNOWNET data forwarding protocol is able to dynamically adjust and self-organize the data forwarding routes based on the current topology of the wireless backbone and the current attachment distribution of clients. Thus, SNOWNET nodes are easily portable and the configuration of the network can be changed to adapt to changing usage patterns by adding, deleting or moving of nodes. We use the term portable, as opposed to mobile, to indicate that additional special features are needed if SNOWNET will be deployed in rapidly moving vehicles.

There are three major functions provided by the SNOWNET nodes: backbone service, access service and gateway service. Every node is equipped with at least one WLAN interface providing the backbone communications between peer nodes. Optional external antennas may be used to extend the communication range of the backbone interfaces.

For maximum flexibility the backbone interfaces of the node run under peer-to-peer mode (also known as IBSS mode or ad hoc mode). Although we have focused on a single technology (IEEE 802.11) link-type for SNOWNET, it is not inherently necessary for all links of the backbone network to use the same link technology. Multiple link technologies may co-exist in the backbone network. In this case, nodes with backbone interfaces of the same technology form sub-backbones. Sub-backbones are connected together to form the overall backbone network by nodes with multiple backbone interfaces of different technologies that are simultaneously on multiple sub-backbones.

In addition to backbone interface(s), a node may be equipped with additional interfaces to provide local WLAN AP access service to clients. In Fig. 1 several typical SNOWNET nodes are shown which have at least two wireless interfaces, one for backbone communications, and the other for providing the local access service in their service coverage area. The same architecture is applicable to other local access technologies (e.g., Ethernet or Bluetooth) too. It is also possible for a single node to have a combination of local access service interfaces.

When SNOWNET is not deployed as a standalone network, some nodes will need to have connectivity to an external organizational network or the Internet. These nodes provide the gateway service and thus become the "gateway nodes" for the other nodes to reach the external network or Internet. In a gateway node there is an additional external network interface that may be of various link technologies as well, e.g., a wired Ethernet interface, a wireless LAN interface to an AP attached to the organizational network, a Point-to-Point Protocol (PPP) connection, or a cellular wide-area wireless communication interface, etc.

Some interfaces of the node may be virtual interfaces. For instance, it is possible to timeshare a physical interface to create multiple virtual interfaces that can be used for different services. This approach can be used to permit the same IEEE 802.11 interface to run in ad hoc mode in some time slots to act as the backbone interface and in AP mode in other time slots to act as the local access service interface.

Communications in a SNOWNET are similarly organized into two levels: backbone communication and local access communication. SNOWNET nodes relay communication between these two levels. Therefore a typical intra-SNOWNET communication path includes the link between the source mobile client and the SNOWNET node serving the source client, a number of backbone links, and finally the link between the destination client and its access service node.

The security of SNOWNET is also managed in two levels. Between the clients and their service SNOWNET nodes, the standard IEEE802.1x authentication and WPA style protection is supported. In the backbone, SNOWNET supports a locally developed extension to the 802.1x standard and WPA-compatible security specifically designed for multi-hop wireless networks. These protection mechanisms ensure that only trusted clients will be served by SNOWNET and only trusted SNOWNET backbone nodes may participate in routing algorithm and data forwarding.

The architectural design of SNOWNET does not pose any hard limits on network size. Delay is usually not a serious issue either when the network is not congested. Our experiences indicate that with our prototype nodes the per-hop delay under normal traffic condition is about 2 ms. However, the throughput of the system can be limited by the multi-hop backbone network. Since the backbone uses a shared channel, bandwidth can not be linearly increased by simply adding more nodes, even though these nodes may increase coverage area and provide frequency reuse opportunities; as opposed to laying more cables to increase the capacity of a wired network. For applications which produce low rate data traffic, a SNOWNET with up to four to six hops is generally adequate. In addition, distribution of the frequency channels among the backbone, the access points and neighboring "interfering" networks can adversely affect the performance. A more comprehensive understanding of the SNOWNET capacity and its channel selection mechanism are under study.

4. SNOWNET Data Forwarding

4.1 SNOWNET Routing

SNOWNET backbone's dynamic and self-organized nature fits well into the profile of MANET. In MANETs, special routing protocols will configure routes within the network to communicate between nodes in the same MANET over one or more hops over MANET links. Many routing protocols, such as the Ad hoc On-demand Distance Vector protocol [6], Dynamic Source Routing protocol [7], Global State Routing/Fisheye State Routing (GSR/FSR) protocol [8),9)], and Optimized Link State Routing protocol [10)], have been proposed by the research community for MANETs for particular situations. The SNOWNET's routing protocol is designed based on MANET routing algorithms and in particular, is a modification to the GSR/FSR protocol. We choose a GSR/FSR-based scheme for two reasons: 1) the GSR/FSR is a relatively simple protocol that can be easily modified for SNOWNET; and, 2) the GSR/FSR is a table-driven MANET routing protocol. Table-driven protocols maintain

all network routes so that clients do not need to wait for route construction (as needed in on-demand type of protocols). In addition, if the destination does not exist in the network, this type of protocols is able to discover the absence immediately by examining the routing table. For on-demand routing protocols, such a decision can only be made after all route construction attempts have failed. The software implementation of SNOWNET routing is a routing daemon named *snowd*.

Currently the supported native data forwarding method within the SNOWNET backbone network is Layer-3 IP style routing. Layer-2 data frame forwarding over is provided via Ethernet-in-IP tunnels. SNOWNET utilizes an IP address space organized so that all backbone interfaces share this address space and form a flat routing space. The backbone interface addresses are automatically assigned using a function implemented as part of the *snowd*. SNOWNET supports two mechanisms for backbone address auto-assignment. In the centralized mechanism, a DHCP-like protocol is supported where a dedicated server manages the address pool and all nodes will request this server for new backbone address assignments. In the distributed mechanism, each node selects its own address then verifies its selection with other nodes in the network. After the backbone interface addresses are assigned, the SNOWNET routing protocol will automatically build the routes and SNOWNET nodes will learn how to forward data packets to reach their destinations.

In routing mode, each SNOWNET local access service interface is allocated a mask-able address space segment from which addresses are dynamically assigned to this access service interface's local clients. The service interface itself is also allocated an address from the same space. In the case that a SNOWNET node has multiple service interfaces, these service interfaces may be bridged together to share the same address space. On the clients, their default routes all point to their SNOWNET access service interfaces. To enable client-side auto-configuration, DHCP server software is installed on each SNOWNET node. This permits the local access service to automatically manage IP addresses and to configure routing and other IP communication parameters for local clients.

In the GSR routing protocol, each node periodically transmits its view of the network topology in a special topology-exchange message called a routing control packet. After a node hears a view transmission from a neighboring node, it will combine the neighbor's view of the network topology with its current knowledge to construct a new view of the network topology. When it is time for this node to transmit its view, this new combined view is sent out in the routing control packet. Initially, every node's view of the network only contains the node itself. Gradually through routing control packet exchanges, each node constructs the knowledge of the full network topology. The FSR scheme reduces the amount of protocol messages by exchanging topology knowledge less frequently for remote part of the network.

In SNOWNET routing algorithm, since clients do not participate in routing algorithm execution, in order to include route construction for the client subnets, SNOWNET nodes act as "proxies" for the clients they serve. Certain modifications to the GSR/FSR routing protocol are needed to include the local access service address spaces together with their own addresses in the routing control packets so that the learned network topology, and subsequently the routing tables, will include the client subnets as well.

4.2 Roaming Support

Since it is common for a mobile user to move from the local coverage area of one SNOWNET node to the coverage area of another SNOWNET node, roaming support is very important. A popular solution for supporting mobile client mobility is the Mobile IP [11]. However, Mobile IP handoff is recognized as not very smooth and traffic forwarding to mobile clients not very efficient. Hence, SNOWNET employs a client roaming mechanism which compliments Mobile IP and provides smoother handoff of the mobile clients between SNOWNET nodes.

In order to support client roaming, in addition to the local service subnets, each SNOWNET node providing local access service also provides routing proxy service for a number of "foreign mobile clients". A foreign mobile client is a client currently attached to a node but with an address outside of the current node's local address spaces. In a similar modification, the addresses of foreign mobile clients are also included in the routing control packet.

Thus, in each routing table maintained by SNOWNET nodes, there are two types of route

entries: subnet routes and host routes. The former are aggregated route entries where each entry describes routes for all the hosts within the corresponding address space, expressed in traditional format as a combination of network address and network mask. These subnet route entries are for the local access (client) subnets. The latter are for routes towards specific nodes, either backbone nodes or foreign mobile clients. A longest match rule [12] is applied during route lookups for data forwarding. On the other hand, the SNOWNET node serving the foreign node needs to perform as the ARP proxy for the foreign client's home subnet.

4.3 Tunneling Mode

When the total size of the SNOWNET is not large, it may be advantageous to use the tunneling mode. Using tunneling mode, it is possible to connect some or all of the local service areas into one single broadcast domain, further simplifying client management. When local access clients are in the same broadcast domain, there is no need for separately managing IP addresses and routes for each local access service area. A single DHCP server may serve all clients in the broadcast domain. When clients change their access service node, no network layer configuration (IP address, routes, etc) changes to the clients are needed either. With tunneling mode, it is also possible for SNOWNET to carry non-IP traffic.

In tunneling mode, after a client sends out a link layer data frame, its access service SNOWNET node's local access service interface will capture the date frame and, based on the MAC address of the destination of the data frame, the access service SNOWNET node of the destination client is determined. The captured data frame is then encapsulated in an IP data packet addressed to that SNOWNET node and the IP packet is routed through the backbone network in normal routing mode. The destination SNOWNET node un-encapsulates the IP data packet to reveal the original link layer data frame sent by its original source client. The data frame is then sent from the SNOWNET node's local access interface to the destination client as the same data frame as the original source client sent out.

For the tunneling mode to work, just as a spanning-tree bridging device needs to know where all the hosts (their MAC addresses) are with respect to the location of the bridging device on the spanning tree, each node also needs to know where all the clients (their MAC addresses) are in the SNOWNET topology. This can be done by each node including the MAC addresses of all its local clients in its routing control packet. Such a MAC address list is readily obtained from the association list of the local access service interface.

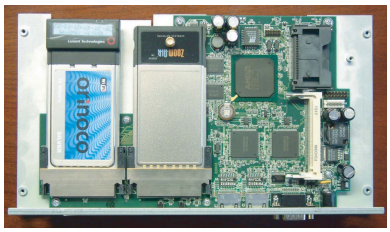

Fig. 2 Prototype SNOWNET node.

5. Prototype Design

5.1 System Design

The SNOWNET node has been implemented as an embedded system in which each node has a manageable and portable form factor. Optionally, each node may be equipped with external omni- or directional antennas to extend the communication ranges of its wireless interfaces.

Due to special considerations for power consumption, cost, and configuration flexibility, we selected the net4521 embedded development board manufactured by Soekris Engineering [20] as our starting point (shown in **Fig. 2**). The Soekris net4521 board is a diskless, PCI architecture embedded system with an AMD ELAN SC520 system-on-chip processor (Intel 486 class) running at 133 MHz and 64 M of RAM as main memory. It also has a rich set of interfaces for communication and peripherals, including 2 10/100 BaseT Ethernet ports, 2 PCMCIA/ Cardbus adapters, and a mini-PCI type III socket. The two PCMCIA slots are typically used for the WLAN interfaces (e.g., backbone and local access) and can be standard commercial WLAN cards, as shown in Fig. 2. The board also has a CompactFlash interface as storage for software and data.

If the SNOWNET nodes are intended to be deployable in potentially hostile environment, additional hardware and operating system features may be provided for enhancing security. The file system used by SNOWNET nodes should be protected using a (partially) encrypted file system for critical node certificate information and hardware booting codes.

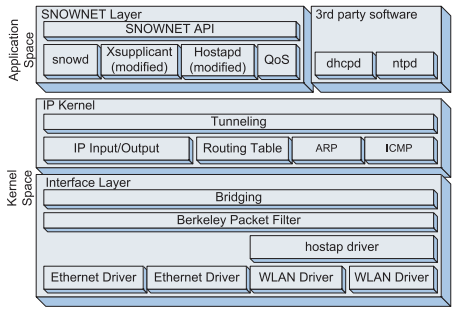

Fig. 3 Software architecture.

When the node is booted up, the operator needs to provide a decryption key and only then can the file system be accessed and system files be decrypted and loaded into system main memory to be executed. When a node is disconnected from the authentication and key management server of the SNOWNET, (i.e., not receiving key management messages from the server), for a certain period of time, an automatic power shut down is performed. Other tampering with the SNOWNET node is further prevented by physical security methods on the device.

5.2 Software Design

The current version of SNOWNET is implemented on FreeBSD platform [21]. The major software components related to SNOWNET are shown in **Fig. 3**. On the bottom is the network interface layer, which consists of various network device drivers. We modified the "hostap" software [22] for supporting the local access service WLAN AP functions and 802.1X authenticator functions for SNOWNET. The software contains both kernel space module (hostap driver) and user space module (hostapd). Above the interface layer is the IP kernel, consisting of standard FreeBSD modules such as IP forwarding, tunneling, routing table, etc. Both the interface layer and IP kernel are in kernel space.

In application space, we implement SNOWNET data forwarding methods. It consists of a routing daemon *snowd* and associated components to support the routing and tunneling methods. Atop of the SNOWNET network layer is an API which offers an application development interface so other network application and service developers can access SNOWNET specific features. The SNOWNET network layer is implemented as a network service to other future middleware components such as a Quality of Service (QoS) module in a SNOWNET node.

In addition to all the modules mentioned above, security related functions are spreaded across *snowd*, a modified version of *xsupplicant* which implements the IEEE 802.1x client functions as part of the Open1X project [23], and hostap. These modules together handle IEEE 802.1x based authentication and dynamic key updating. They also provide the authentication, key management, and other security related supports for the SNOWNET backbone network.

6. Conclusion

In this document, we have described a wireless data network solution which is able to provide secure and portable wireless data networking services to mobile users with WLAN interfaces in many difficult environments. The solution, named the Secure Nomadic Wireless Network or SNOWNET, follows a hierarchical approach: local service areas connected via a wireless backbone. SNOWNET nodes are deployed in areas where networking service is needed and the SNOWNET nodes are capable of communicating with each other by dynamically establishing the links between the SNOWNET nodes to form the backbone communication network. While maintaining backbone connectivity, SNOWNET nodes are also capable of providing local access service to regular clients via WLAN infrastructure mode connectivity and other LAN connectivity. Using a MANET-like routing protocol over the backbone network, SNOWNET is able to support both network-layer routed and link-layer switched (over tunnels across backbone network) data forwarding services for SNOWNET clients.

SNOWNET can be quickly setup as a secure standalone networking infrastructure to provide instant networking services in an area where there is no existing trusted networking environment. Possible scenarios include disaster relief operations, scientific exploration tasks, battlefields and robotics applications. SNOWNET can also be installed as a cost-efficient wireless LAN to provide on-demand extendable wireless networking coverage for an organization when cabling is not feasible. With its flexible, multihop, self-organized, and self-configured wireless backbone network, SNOWNET saves costs for cabling, installation and maintenance. SNOWNET may also be used as a stub network to connect isolated LANs to an organizational net-

work. For instance, it can "glue" a LAN installed in a remote building to the primary network. Several additional aspects of SNOWNET such as security, auto-configuration and performance are the subject of ongoing research and will be described in upcoming papers.

Acknowledgments The authors would like to thank two student interns from the University of Maryland, Arunesh Mishira and Sohil Thakar, for their involvements in the early works of the SNOWNET project.

References

1) IEEE LAN/MAN Standards Committee: Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications (ANS/IEEE Std 802.11-1999), IEEE (1999).
2) Droms, R.: Dynamic Host Configuration Protocol, RFC 2131 (Mar. 1997).
3) Fluhrer, S., et al.: Weakness in the key scheduling algorithm of RC4, *Eighth Annual Workshop on Selected Areas in Cryptography* (2001).
4) IEEE LAN/MAN Standards Committee: Port-Based Network Access Control (IEEE Std 802.1X-2001), IEEE (2001).
5) Ji, L., Feldman, B. and Agre, J.: Self-Organizing Security Scheme for Multi-hop Wireless Access Networks, *Proc. 2004 IEEE Aerospace Conference*, Bigsky, MT, USA (Mar. 2004).
6) Perkins, C., Belding-Royer, E. and Das, S.: Ad hoc On-Demand Distance Vector (AODV) Routing, IETF RFC3561 (July 2003).
7) Johnson, D., Maltz, D. and Hu, Y.: The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR), IETF Internet Draft draft-ietf-manet-dsr-09.txt, work in Progress (Apr. 2003).
8) Chen, T.-W. and Gerla, M.: Global State Routing: A New Routing Scheme for Ad-hoc Wireless Networks, *Proc. IEEE ICC'98*, Atlanta, GA, pp.171–175 (Jun. 1998).
9) Iwata, A., Chiang, C.-C., Pei, G., Gerla, M. and Chen, T.-W.: Scalable Routing Strategies for Ad Hoc Wireless Networks, *JSAC99, IEEE Journal on Selected Areas in Communications*, Vol.17, No.8, pp.1369–1379 (Aug. 1999).
10) Clausen, T. and Jacquet, P.: Optimized Link State Routing Protocol (OLSR), RFC3626 (Oct. 2003).
11) Perkins, C.: IP Mobility Support for IPv4, RFC3344 (Aug. 2002).
12) Doeringer, W., Karjoth, G. and Nassehi, M.: Routing on Longest-Matching Prefixes, *IEEE/ACM Transactions on Networking* (*TON*), Vol.4, Issue 1 (Feb. 1996).
13) Orinoco outdoor routers, www.proxim.com/products/all/orinoco/outdoor/index.html
14) Cisco Aironet1400, www.cisco.com/warp/public/cc/pd/witc/ps5279/ps5285/prodlit/
15) Linksys WET11, ftp.linksys.com/datasheet/wet11_ds.pdf.
16) IETF Mobile Ad-hoc Networks (MANET) Working Group, www.ietf.org/html.charters/manet-charter.html.
17) Mesh Networks, www.meshnetworks.com.
18) Greenpacket, www.greenpacket.com.
19) Moteran, www.moteran.com.
20) Soekris Engineering, www.soekris.com.
21) FreeBSD, www.freebsd.org.
22) HostAP, hostap.epitest.fi.
23) Open1X, www.open1x.org.

(Received April 10, 2004)
(Accepted July 1, 2004)

Lusheng Ji is a Researcher at the Fujitsu Laboratories of America in College Park, Maryland, USA. His research interests include ad hoc networks, routing protocols, m-commerce, and wireless security. He has a Ph.D. in Computer Science from the University of Maryland where he developed routing protocols for ad hoc networks.

Jonathan Agre is the Director of the Pervasive Computing Department at the Fujitsu Laboratories of America in College Park, Maryland, USA. His research interests include wireless protocols, sensor networks, m-commerce, and performance analysis. He has been involved with various aspects of distributed systems at Jet Propulsion Laboratory, Rockwell Science Center and ARINC Research. He has a BS, MS and Ph.D. in Computer Science from the University of Maryland.

Tadashige Iwao is a Researcher at the Fujitsu Laboratories in Japan. His research interests include ad hoc networks and multi-agent systems. He has a Ph.D. in Computer Scienece from Kyushu University.

Nobutsugu Fujino received the B.S. and M.S. degrees in Electronics Engineering from University of Osaka Prefecture in 1984 and 1986, respectively. He joined Fujitsu Laboratories Ltd. in 1986 and has been engaged in research and development of radio communication systems and mobile computing technology. His research interests include ubiquitous network environment and mobile computing architecture. He is a member of IPSJ.