A PROTOTYPE IMPLEMENTATION OF VPN ENABLING USER-BASED MULTIPLE ASSOCIATION

O. Honda, H. Ohsaki and M. Imase
Graduate School of Information Science and Technology
Osaka University
Osaka, Japan email: {o-honda,oosaki,imase}@ist.osaka-u.ac.jp

Y. Hara
Dai Nippon Printing Co.,Ltd
Tokyo, Japan
email: y-hara@dev.cio.dnp.co.jp

M. Maruyoshi and K. Matsuda
NTT Information Sharing Platform Laboratories NTT Corporation
Tokyo, Japan
email: {maruyoshi.masahiro,matsuda.kazuhiro}@lab.ntt.co.jp

ABSTRACT
In our previous work, we have proposed a new VPN architecture for enabling user-based multiply associated VPNs. In this paper, we implement a prototype system of a VPN that enables users to be associated with multiple VPNs using existing network technologies for demonstrating the feasibility of our architecture and for clarifying the service image of a multiple association service. Our prototype system enables hosts to be simultaneously associated with multiple VPNs where each VPN is constructed using existing VLAN technology.

KEY WORDS
VPN(Virtual Private Network), prototype implementation, VLAN(Virtual Local AreaNetwork), multiple association, access control

1 Introduction

In recent years, along with the rapid development of network technologies and Web-based services [1], various networking services have emerged, including network-based procurement and distribution, network-based governmental operations [2], the development of network-based working environments such as telework and SOHO activities [3], and so on. As a result of networking, various types of social activities are now free from geographical restrictions. As such, it is expected that our social structure will become increasingly diversified and dispersed. Within such a social framework, the emergence of network-based virtual organizations is highly probable in the near future. We collectively refer to such virtual organizations as "Cybersociety." People who work within this Cybersociety will have to establish secure communication channels with multiple virtual organizations; that is, "multiple association" relationships will have to be established. We belive that such virtual organizations will be realized based on the effective use of virtual private networks (VPNs).

There are several VPN technologies, such as PPVPN (Provider-Provisioned VPN) [4, 5] and extranet [6, 7] already in existence. However, these are insufficient in that individual users cannot be multiply associated with a number of different VPNs at the same time on an individual basis.

In view of the above-described circumstances and in order to realize multiple associations of individuals with virtual organizations based on network services, we are studying a new VPN technology by which users can be associated with a number of different VPNs [8, 9]. Here, we refer to this new VPN as "Multiply-Associated VPN (MAVPN)."

In this paper, we aim to demonstrate the feasibility of our proposed MAVPN system by implementing a prototype system using existing network technologies in order to obtain a clearer view of multiple association services.

In Section 2, we describe service perspectives for the planned implementation of the target MAVPN system. In Section 3, the basic strategies to be applied in implementing the prototype system are discussed. In Section 4, each element of the prototype design is discussed. Section 5 describes the actual implementation methods and the resulting operational functions. In Section 6, we summarize the results of this study and discuss future challenges.

2 MAVPN Service Perspectives

This section describes the concept of MAVPN services from a user perspective. The user's procedures for using a MAVPN system are assumed as follows:

1. The user connects his/her terminal (host) to a nearby network device.
2. The user starts up the MAVPN terminal software on his/her host.
3. From a list displayed on the terminal of VPNs to which he/she may be granted association, the user selects the multiple VPNs that are desired for connection. If not, predefined VPNs may be automatically connected upon booting the host.
4. The user or any application of the user can communicate with any target host by specifying the desired host name.
5. The user selects and disconnects any VPN to be disconnected from the terminal software. If not, exiting the terminal software could automatically disconnect all VPN connections.

In this context, it is desirable for users to only be required to establish a network connection, start up the terminal software, and enter the user ID and password to enable connection to multiple VPNs. It is also desirable for any VPN to be able to be connected or disconnected at any time by specifying the VPN name on the VPN list shown on the terminal display.

3 Implementation Strategy

Towards the realization of the services outlined in Section 2, an experimental prototype system is to be implemented. Here, the aim is to implement only some of the service requirements specified in the above section. To be more specific, the following requirements are to be met:

- Users can communicate with their target hosts without taking care of multiple VPNs
- VPNs can be specified by logical names
- User terminal addresses can be automatically configured
- Access to VPNs can be controlled on an individual user basis

Any requirements other than those listed above remain for future consideration.

In the prototype design, we are to use commonly available standard technologies that can be implemented easily. In implementing a prototype MAVPN system using existing standard technologies, we have shown in [9] that it is best to have an architectural framework to use layer 2 technologies in realizing a basic networking framework and a VPN in it and to use layer 3 technologies to control user accesses to multiple VPNs.

The prototype implementation described in this paper uses such an architectural framework. To be more specific, the Ethernet is used for the base network, IEEE 802.1Q VLAN is used for establishing VPNs, and IP is used for controlling user access to VPNs.

This prototype is implemented by referring to an existing technology called authenticated VLAN [10]. Authenticated VLAN technology provides a mechanism to add authentication functions to existing VLAN systems in order to realize access control on an individual user basis. The major target of this prototype is to realize an authenticated VLAN system where the user can utilize multiple VLAN systems simultaneously. Another objective is to achieve a security level comparable with that of authenticated VLAN systems. To be more specific, the basic assumption is that terminals (hosts) within a site (within the same LAN connection) are trusted with each other. Therefore, the possibility of cracking by a host located in the same site or the misrepresentation of MAC addresses is not considered.

4 Prototype Design

4.1 Network Configuration

This section first describes the network configuration of the prototype system. The logical network configuration of the prototype is as shown in Figure 1.

The VPNs to be offered to users ("User VPN" as shown in Figure 1) will be implemented by IEEE 802.1QVLAN [11]. IEEE 802.1Q technology uses a tag containing a 12-bit VLAN ID attached to an Ethernet frame to virtually divide Ethernet connections. In order to simplify this prototype implementation, each VLAN is configured so as to have only one IP subnet so that no duplication can occur between IP address spaces.

The user host is connected to the VLAN via a device called an "MAVPN gateway," as shown in Figure 1. The

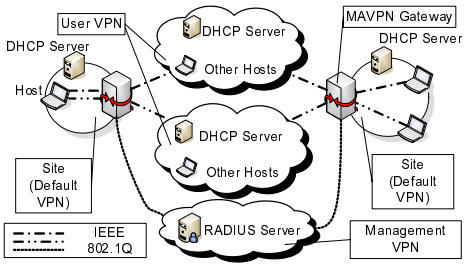

Figure 1. *Logical network configuration of the prototype system*

access lines between the hosts and the MAVPN gateway are multiplexed using the IEEE 802.1Q, i.e., the MAVPN gateway bridges the access circuits to VPNs that are implemented with IEEE 802.1Q VLAN. In this way, the hosts can use multiple layer 2 VPNs. The MAVPN gateway also has a function to authenticate users; only the access lines from the user host that may be granted for association with the target VPNs can be connected. In the initial condition where the host does not yet belong to any VPNs, communication is allowed only within the site, as shown in Figure 1. To represent this initial condition, such an initial site is referred to as "Default VPN," as shown in Figure 1.

The IP address of the user terminal is assigned by the DHCP server installed within each VLAN from the IP address spaces of respective VLANs. Note that IP addresses for servers are statically assigned from the respective VLAN IP address spaces.

User account information to be used for authentication purposes will be provided to every MAVPN gateway from the RADIUS server, as shown in Figure 1. The RADIUS server will be installed within a special "Management VPN" system, as shown in Figure 1. User account information to be stored in the RADIUS server will include user IDs, VPN names for which association is granted, and passwords. As such, a user can have as many accounts as the number of VPNs for which he/she is granted association.

4.2 Communication Methods

This section describes the communication methods used in this prototype design. When the host is first connected to the site, it will create an interface port for connection to the default VPN and receives an IP address for the default VPN from the DHCP server within the site.

4.2.1 Association Procedures

The association of a user to a VPN is processed according to the following procedures (refer to Figures 2 and 3):

1. As shown in Figure 2, the user at the host performs the "Request" operation by entering his/her user ID, the name of the VPN to be associated with (e.g., a logical name such as "Company VPN"), and a password to the MAVPN gateway through the default VPN.

2. The MAVPN gateway then sends a query to the RADIUS server regarding the user ID, VPN name, and password entered by the user ("Proxy Authentication" in Figure 2).

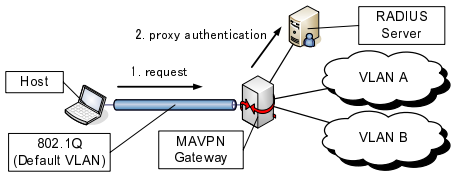

Figure 2. *User VPN association procedures (1. request and 2. proxy authentication)*

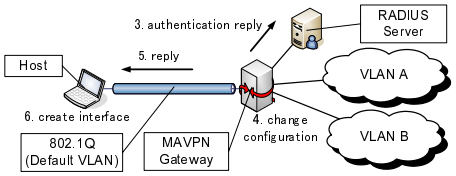

Figure 3. *User VPN association procedures (3. authentication reply, 4. change configuration, 5. reply and 6. create intgerface)*

3. Receiving the "Authentication Reply" response from the RADIUS server as shown in Figure 3, the MAVPN gateway checks the result to see whether or not the user is accepted for access to the target VPN.

4. After confirming that the user is in fact accepted by the target VPN, the MAVPN gateway modifies the filtering rule so that the data frames from the terminal can be bridged to the target VPN. At this instant, the VPN name will be changed to the VLAN ID for use in the actual IEEE 802.1Q communication framework as indicated by "Change Configuration" in Figure 3.

5. The MAVPN gateway notifies the host that the association request to the target VPN has been accepted ("Reply" in Figure 3).

6. The host creates an IEEE 802.1Q interface port in order to work with the associated VPN and acquires an IP address from the DHCP server in the VPN ("Create Interface" in Figure 3).

The MAVPN gateway and the host are to be implemented in such a way as to allow the above-described transactions.

4.2.2 Disconnection Process

The disconnection of users from VPNs is processed according to the following procedures (refer to Figures 4 and 5):

1. As shown in Figure 4, the user performs the "Request" operation by entering his/her user ID, the name of the VPN to be disconnected from, and a password to the MAVPN gateway using the default VPN.

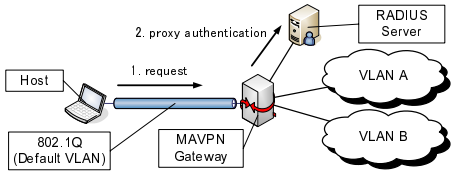

Figure 4. *User VPN disconnection procedures (1. request and 2. proxy authentication)*

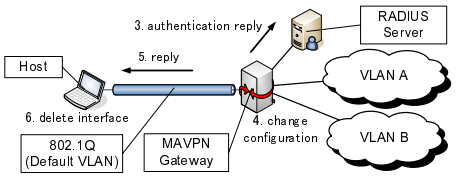

Figure 5. *User VPN disconnection procedures (3. authentication reply, 4. change configuration, 5. reply and 6. delete intgerface)*

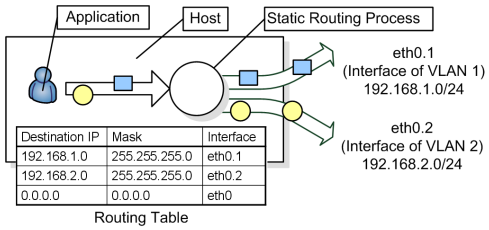

Figure 6. *VLAN interface selection at the host*

2. The MAVPN gateway then sends a query to the RADIUS server regarding the user ID, VPN name, and password entered by the user ("Proxy Authentication" in Figure 4).

3. Receiving the "Authentication Reply" response from the RADIUS server as shown in Figure 5, the gateway checks the result to see whether or not the user is accepted for access to the target VPN.

4. After confirming that the user is in fact accepted by the target VPN, the MAVPN gateway deletes the appropriate filtering rule that has allowed bridging between the host and the target VPN. At this instant, the VPN name will be changed to the VLAN ID for use in the actual IEEE 802.1Q communication framework as indicated by "Change Configuration" in Figure 5.

5. The MAVPN gateway notifies the host that the connection to the target VPN has been removed ("Reply" in Figure 5).

6. The host deletes the IEEE 802.1Q interface that has been in place for the disconnected VPN ("Delete Interface" in Figure 5).

4.3 Host System Design

This section describes the host system design concepts. The terminal (host) is required to issue an authentication request to the MAVPN gateway in order to be associated with the target VLAN. Therefore, some sort of terminal software will be required to perform authentication requests. The following three functions are required for the terminal software:

- Transmit the user ID, the name of the VPN to be associated with, and a password to the MAVPN gateway;
- After the association to the VPN is accepted, create an IEEE802.1Q interface to enable access to the VPN and issue a DHCP request.

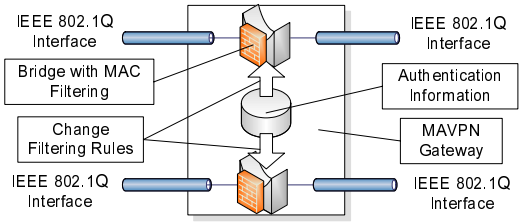

Figure 7. *Host access control of the MAVPN gateway*

- After the host is disconnected from the VPN, delete the IEEE802.1Q interface function that has been in place for the VPN access.

In this prototype configuration, it is assumed that each VPN has only one IP subnet and that no duplication is allowed between IP address spaces. Therefore, it is the static routing process of the host that determines which IEEE 802.1Q interface port of VPNs is to be used in passing the packets generated from an application program running on the host. This situation is depicted in Figure 6. For this, the static routing process of the host searches in the "Routing Table" using the packet "Destination IP" address to find an appropriate IEEE 802.1Q interface port (e.g., eth0.1 or eth0.2) to output the packet, as shown in Figure 6.

4.4 MAVPN Gateway Design

This section describes the design concept for implementing the MAVPN gateway. The MAVPN gateway controls access from the host to the VLAN by enforcing user authentication. The following two major functions are required for the MAVPN gateway:

- Perform user authentication by receiving a connection request from the user.

- Control connection of the user terminal (host) according to the authentication results.

The function to perform user authentication by receiving a connection request from the user requires an interface facility to accept the user ID, VPN name, and password from the user. In addition, a process to verify the information received from the user by consulting with the RADIUS server in order to determine whether or not the user is granted association to the target VLAN is necessary.

This situation is depicted in Figure 7, showing connection control for the user terminal (host) performed according to the authentication results. After the user is authenticated and granted association to the VPN, the MAVPN gateway implements a bridge to connect IEEE 802.1Q frames from the user host to the VPN, as shown in "Bridge with MAC filtering" in Figure 7. Any frames from a user host not granted association to the VPN will be discarded based on MAC address filtering.

5 Prototype Implementation

Based on the design features described above, a prototype system was implemented.

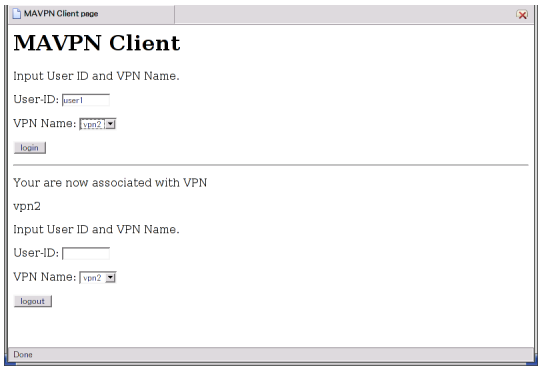

Figure 8. *Association VPN selection window of terminal software*

5.1 Implementation of the Host System

This section describes the implementation for the terminal (host) part. This implementation uses Linux kernel functions for IEEE 802.1Q [12] because the host is required to be compliant with IEEE 802.1Q functions.

In this design, the host is implemented with a WWW server; the host's terminal software is implemented using PHP scripts. The terminal software window looks like the one shown in Figure 8. The user enters his/her user ID, selects the desired VPN name, and clicks on the "Login" button. Then, the user ID, the VPN name, and the MAC address of the host are sent to the MAVPN gateway, resulting in a password prompt window to be displayed. When a password is entered, an authentication process is initiated. The terminal software uses CGI in sending information to the gateway. In this particular implementation, it is assumed that each user has a list of VPNs that he/she can use.

If use of the VPN is granted as a result of the authentication, the terminal software uses the Linux `vconfig` command to create an IEEE 802.1Q interface port for connection to the VPN and issues a DHCP request to receive an IP address. The VLAN ID to be used in the IEEE 802.1Q interface will be notified from the MAVPN gateway.

In the window shown in Figure 8, pressing the "Logout" button after entering the user ID and the VPN name with which the user is currently associated will initiate the disconnection process in the MAVPN gateway. Then, the terminal software will remove the IEEE 802.1Q interface connection using the `vconfig` command of Linux. The MAVPN gateway will notify VLAN ID of the IEEE 802.1Q interface connection to be deleted.

5.2 MAVPN Gateway Implementation

This section describes the implementation of the MAVPN gateway. With this prototype design, the MAVPN gateway is realized by a computer system that is implemented with two physical network interface ports. The gateway has to implement the following three functions:

Interface Function An interface function to exchange information with the terminal software is implemented. This function is implemented by a gateway-processing program written in PHP scripts, which can run on the WWW server in the MAVPN gateway. The gateway-processing program communicates with the terminal software via CGI.

To be more specific, the gateway-processing program receives user ID, VPN name, password, and terminal MAC address information from the terminal software. This function also informs the relevant VLAN ID and instructs the terminal software to either create or delete an IEEE 802.1Q interface port using the ID supplied. In this, all VPN names are to be converted to the corresponding VLAN ID for use in IEEE 802.1Q communication. For this purpose, the MAVPN gateway is also implemented with a relevant mapping table.

User Authentication Function Next, a user authentication function is to be implemented. Authentication is performed based on the account information including the user ID, VPN name, and password to be received from the terminal software. Authentication is performed upon association of the user to the VPN as well as upon disconnection from the network. In this particular implementation, it is assumed that any account information is stored in MAVPN gateways and that data synchronization is maintained between multiple MAVPN gateways. Therefore, authentication is made by simply comparing the account information received with the data stored in the gateway.

Bridge Function Lastly, a function to bridge VLAN frames from the authenticated users to the target VPNs is implemented. With this prototype, the "Ethernet bridge" function [13] of Linux is used. After authentication is completed and the user can be associated with the VPN, IEEE 802.1Q interface ports are created for both the terminal side and the VPN connection side. Then, an Ethernet bridge is created to connect these two IEEE 802.1Q interface ports.

In this way, a bridge is created for each of the VPNs and MAC address filtering is applied to every Ethernet frame that passes through each bridge. The MAC address filtering tool, `ebtables` [14], is used for the filtering function. The following two basic filtering rules are applied:

- All of the frames from hosts to VPNs should be discarded.
- All of the frames from VPNs to hosts should be discarded.

After user authentication is completed and the user can be associated with the VPN, the following rules are added:

- All of the frames from the MAC address of the authenticated host to the VPN should be passed.
- All of the frames from the VPN to the MAC address of the authenticated host should be passed.

The two additional rules should be deleted upon disconnection of the user from the VPN.

5.3 Operation Verification of the Prototype System

The hosts and MAVPN gateways were implemented as described in the above paragraphs and tested for operation. Shown in Figure 9 is the physical configuration of the network subjected to the operational tests. There are two sites in the prototype network. Site 1 is composed of MAVPN Gateway 1, Host 1, and Host 2. Site 2 is composed of

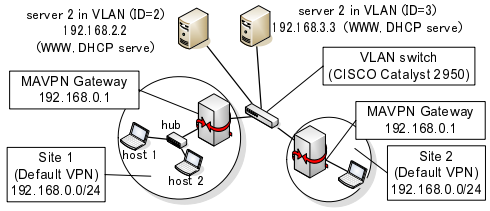

Figure 9. *Physical network configuration for operational tests*

Table 1. Account information for prototype tests

User ID	PN of valid association	Password
`user1`	`vpn2`	xxx
`user1`	`vpn3`	xxx
`user2`	`vpn3`	yyy
`user2`	`vpn4`	yyy
`user3`	`vpn4`	zzz

MAVPN Gateway 2 and Host 3. The VPN is an IEEE 802.1Q VLAN.

Server 1 and Server 2 are always associated with the VPNs named `vpn2` (VLAN ID=2) and `vpn3` (VLAN ID=3), respectively. It is assumed that non-overlapping network addresses are used in each VPN to avoid any duplication; `vpn2` and `vpn3` are assigned the network addresses of 192.168.2.0 and 192.168.3.0, respectively. Also, a DHCP server and a WWW server are running on each of Servers 1 and 2.

Under this network configuration, the account information for users to use Hosts 1, 2, and 3 is specified as in Table 1. Here, it is assumed that `users 1`, `2`, and `3` are to use Hosts 1, 2, and 3, respectively.

Based on the account information given in Table 1, the hosts are multiply associated with the respective VPNs, as shown in the logical network configuration presented in Figure 10. With this network configuration, we can test and verify the following functionalities of the prototype system:

- A host can be associated with multiple VPNs.
- Different hosts within the same site can be associated with different VPNs.
- VPNs can be dynamically implemented on an individual host basis.
- Data frames from unauthenticated hosts are filtered out.

First, it is verified that a host can be associated with multiple VPNs. When `user1` from Host 1 is multiply associated with `vpn2` and `vpn3` using the terminal software,

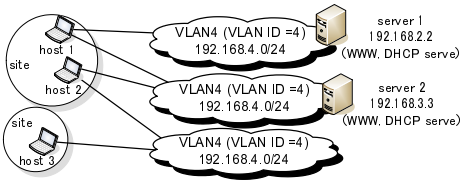

Figure 10. *Logical network configuration for operation verification*

Table 2. Static routing table of Host 1 with multiple association

Destination	Gateway	Genmask	Iface
192.168.3.0	*	255.255.255.0	eth0.3
192.168.2.0	*	255.255.255.0	eth0.2
192.168.0.0	*	255.255.255.0	eth0
default	192.168.3.1	0.0.0.0	eth0.3
default	192.168.2.1	0.0.0.0	eth0.2

the resulting static routing table of user1 is as shown in Table 2. In this table, it is shown that packets to be sent to 192.168.3.0/24 are output to the eth0.3 port.

Since the eth0.3 port is the IEEE802.1Q interface port of VLAN ID 3, the connection is made to `vpn3`. Similarly, it is shown that packets to be sent to 192.168.2.0/24 are routed to the eth0.2 port, for connection to `vpn2`. It was confirmed using ping, ssh, and http that access from Host 1 to Server 1 and Server 2 was successful.

Next, it was verified that different hosts within the same site can be associated with different VPNs. The user `user1` of Host 1 is associated only to `vpn2` and the user `user2` of Host 2 is associated only to `vpn3` establishing communication. At this time, it was confirmed that communication between Host 1 and `vpn2` as well as between Host 3 and `vpn3` was satisfactory. At the same time, it was also confirmed that communication between Host 1 and `vpn3` as well as Host 2 and `vpn2` failed. Therefore, it is verified that Hosts 1 and 2 are associated with different VPNs.

The next test relates to the dynamic establishment of VPNs on an individual host basis. The user `user2` of Host 2 and the user `user3` of Host 3 are both associated with the `vpn4` network for communication. Because there is no DHCP server in the network `vpn4`, the network address of `vpn4` is set to 192.168.4.0/24 and the address is set to Hosts 2 and 3 manually. In this test, successful access was verified from Host 2 to Host 3.

Lastly, it was also confirmed that data frames from unauthenticated hosts are filtered out. To be more specific, with `user1` of Host 1 associated with `vpn2` (VLAN ID=2), `user2` of Host 2 created an unauthorized IEEE 802.1Q interface port for connection to VLAN ID 2. It was confirmed that Host 2 was unable to communicate with the network `vpn2`.

6 Summary and Future Works

In view of the coming Cybersociety, this paper have presented an experimental VPN prototype implementation that verified that network users can be multiply associated with different VPN networks simultaneously. By implementing a prototype system using commonly available network technologies, we have successfully provided a clearer view on the feasibility and service perspective of multiple VPN systems in which users can be multiply associated with different VPNs at the same time. To be more specific, we have implemented a prototype system that enables a host to be multiply associated with different VPNs that are implemented using VLAN technology. Using the prototype system, it was confirmed that network hosts can use transparently multiple VPNs system on an individual user basis. It was also confirmed that host access can be controlled individually by way of an authentication mechanism.

Included in future challenges are an improved level of security for the prototype system and a higher level of scalability. It will also be a future challenge to implement a prototype system that can allow the duplication of address spaces between different VPNs in addition to improved methods of user disconnection from VPN. Further evaluation of the prototype in terms of its scalability regarding the maximum number of VPNs that can be connected as well as the maximum bandwidth to be utilized is another area of future study.

Acknowledgements

This study was performed through Special Coordination Funds for Promoting Science and Technology from the Ministry of Education, Culture, Sports, Science and Technology of the Japanese Government.

References

[1] S. J. Vaughan-Nichols, "Web Services: Beyond the Hype," *IEEE Computer*, vol. 35, no. 2, pp. 18–21, Feb. 2002.

[2] N. Ohyama, "Progress of e-Government in Japan," *IPSJ Magazine*, vol. 44, no. 5, pp. 455–460, May 2003. (in Japanese).

[3] Ministry of Public Management, Home Affairs, Posts and Telecommunications, "Infomation and communications in japane," May 2004. available at `http://www.johotsusintokei.soumu.go.jp/whitepaper/eng/WP2004/Chapter2-%7.pdf`.

[4] A. Nagarajan, "Generic requirement for provider provisioned virtual private networks (PPVPN)," *Request for Comments (RFC) 3809*, June 2004.

[5] M. Carugi and D. McDysan, "Service requirements for layer 3 provider provisioned virtual private networks (PPVPNs)," *Request for Comments (RFC) 4031*, Apr. 2005.

[6] H. Hara, J. Murayama, K. Isagai, and I. Imaida, "IP-VPN Architecture for Policy-Based Networking," *IEICE Technical Report* IN2000-101, vol. 100, pp. 39–46, Oct. 2000. (in Japanese).

[7] J. Miyoshi, I. Imaida, K. Isagai, J. Murayama, and S. Kuribayashi, "A Mechanism of Policy-Based Service Control in Communication between VPNs," *IEICE Technical Report* SSE99-171, vol. 99, pp. 61–66, Mar. 2000. (in Japanese).

[8] Y. Hara, H. Ohsaki, M. Imase, Y. Tajima, M. Maruyoshi, J. Murayama, and K. Matsuda, "VPN architecture enabling users to be associated with multiple VPNs," in *Proceedings of the 5th Asia-Pacific Symposium on Information and Telecommunication Technologies (APSITT 2003)*, pp. 195–200, Nov. 2003.

[9] Y. Hara, H. Ohsaki, M. Imase, Y. Tajima, M. Maruyoshi, and J. Murayama, "On layered VPN architecture for enabling user-based multiply associated VPNs," in *Proceedings of the International Conference on Information Networking (ICOIN) 2004*, Feb. 2004.

[10] Alcatel, "Authenticated VLANs, Secure Network Access at Layer 2," *An Alcatel White Paper*, Nov. 2002.

[11] IEEE standards for local and metropolitan area networks, "Virtual bridged local area networks," *IEEE Standard 802.1Q-1998*, Dec. 1998.

[12] "802.1Q VLAN implementation for Linux." `http://www.candelatech.com/~greear/vlan.html`.

[13] "Linux Ethernet bridging." `http://bridge.sourceforge.net/`.

[14] "ebtables." `http://ebtables.sourceforge.net/`.