Engineering

Industrial & Management Engineering fields

Okayama University *Year* 2005

A Method of Dynamic Interconnection of VLANs for Large Scale VLAN Environment

Kiyohiko Okayama[*] Nariyoshi Yamai[†] Takuya Miyashita[‡]
Keita Kawano[**] Takuji Okamoto[††]

[*] Okayama University
[†] Okayama University
[‡] Tsuyama National College of Technology
[**] Okayama University
[††] Okayama University of Science

A Method of Dynamic Interconnection of VLANs for Large Scale VLAN Environment

Kiyohiko Okayama[†] Nariyoshi Yamai[†] Takuya Miyashita[‡]
Keita Kawano[†] Takuji Okamoto[††]

[†]Information Technology Center, Okayama University
3-1-1, Tsushima-naka, Okayama 700-8530, Japan
{okayama,yamai,keita}@cc.okayama-u.ac.jp
[‡]Department of Electronics and Computer Engineering,
Tsuyama National College of Technology
624-1, Numa, Tsuyama 700-8509, Japan
miyasita@tsuyama-ct.ac.jp
[††]Faculty of Engineering, Okayama University of Science
1-1 Ridai-cho, Okayama 700-0005, Japan
okamoto@ee.ous.ac.jp

Abstract

VLAN (Virtual LAN) is a technology which can configure logical networks independent of the physical network structure. With VLAN, users in common spaces (such as meeting rooms) can access to their department networks temporarily because changing of logical network structure is achieved only by configuration of VLAN switches.

However, in the general configuration method, because VLANs are managed statically by administrators, various problems such as high administrative cost and conflict or insufficiency of VLAN-IDs may arise especially in large scale organizations where VLANs are managed by each department.

To solve these problems, we propose a method which provides an interconnection between a temporary configured VLAN in a common space and a VLAN of a user's department. In the proposed method, a user in a common space can access to his/her department network seamlessly by converting a temporary VLAN-ID in the common space and a VLAN-ID used in his/her department each other automatically. The effectiveness of the proposed method is confirmed by the experiment on the actual network using VLAN managers, VLAN-ID converters and authentication servers based on the proposed method.

1. Introduction

VLAN (Virtual LAN) is a technology which can configure logical networks independent of the physical network structure. With VLAN, because changing of logical network structure is achieved only by configuration of VLAN switches, users can temporarily access to their department networks from a common space (such as meeting room) using a temporarily configured VLAN.

In the general configuration method of VLANs, however, the following problems will arise: (1) since VLANs are managed statically by administrators, administrative cost for managing temporary VLANs is considerably high, (2) since the VLAN identifier (VLAN-ID) which is defined by IEEE802.1Q[1] has only 12bits space, insufficiency of VLAN-IDs may occur, (3) especially in a large scale organization where VLANs are managed by each department independently, conflict of VLAN-IDs allocated by each department may occur.

As a solution of the third problem, a method of encapsulating IEEE 802.1Q VLAN tags within 802.1Q[2] (it is also called "nested VLAN" or "double-tagged VLAN") has been proposed. With this method, different sites of a department can be connected with same VLANs via a single VLAN of another department. However, since this method is based on encapsulation, it is impossible to apply to common spaces described

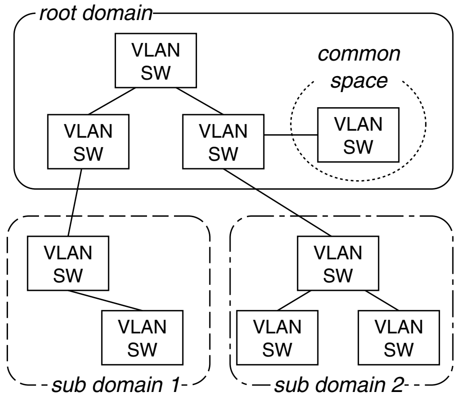

Figure 1. An example of target network structure.

above where users access to different departments.

In order to solve these problems, we propose a method which provides dynamic interconnections between temporary configured VLANs in a common space and users' VLAN of their departments. In the proposed method, a user in a common space can access to his/her department network seamlessly (that is data-link level communication) by converting a temporary VLAN-ID in the common space and a VLAN-ID used in his/her department each other. Furthermore, The proposed method provides dynamic VLAN-ID allocation and automatic remote VLAN switch configuration.

The rest of the paper is the followings. Section 2 discusses some assumptions. In Section 3, we describe the VLANs interconnection method proposed. Then, in section 4, we evaluate the performance of the proposed method. Finally, in section 5, we describe the conclusion and the further work.

2. Target Network Environment

In this paper, it is assumed that a network of an organization consists of layers shown in the Figure 1. Upper layer (root domain) corresponds to a backbone network of an organization managed by a department such as a network center, lower layers (sub domains) corresponds to other departments' networks. All domains consists of VLAN switches and VLANs are managed independently by each domain. In principle, although the proposed method can be applied to a network which has three or more layers, we will consider a network which has only two layers in the subsequent sections for simplicity.

Furthermore, although common spaces can be located on any domains, we will consider that only the root domain has common spaces by the same reason. A common space consists of at least one VLAN switch to which users of any departments may connect their PCs.

3. A Method of Dynamic Interconnection of VLANs

3.1. Dynamic VLAN-ID Allocation and Conversion

In order to solve the first two problems described in the Section 1, two features are required: (1) automatic configuration of VLAN switches and (2) dynamic VLAN-ID allocation for users in common spaces. The former can be achieved by using remote configuration function of VLAN switches, since most VLAN switches provide remote configuration with TELNET, HTTP or SNMP. The latter means that one of the unused VLAN-IDs preserved for temporary use in the root domain is allocated to a user by request.

However, since an allocated VLAN-ID in the root domain may be different from the VLAN-ID used in the user's sub domain, the user in the common space cannot access to his/her sub domain. Consequently, a feature which converts between the VLAN-ID of the root domain and that of the sub domain at the boundary of two domains is also required. With this feature, a user in a common space can make a data-link level connection to his/her sub domain.

3.2. Authentication and Destination Determination

Since various users (including outsiders) may attempt to connect to VLAN switches in common spaces, some kind of authentication feature is required to prevent invalid access. Especially in the proposed method, a data-link level authentication method which needs no IP addresses is required because users' PCs in common spaces can not obtain their sub domains' IP addresses at the authentication time.

On the other hand, in the root domain, destination sub domains must be determined according to users automatically at the authentication time. Therefore, we introduce domain names to user IDs which are used for authentication.

In the proposed method, user IDs are represented in the form of mail addresses, *username@domainname*. Using *domainname* of a user

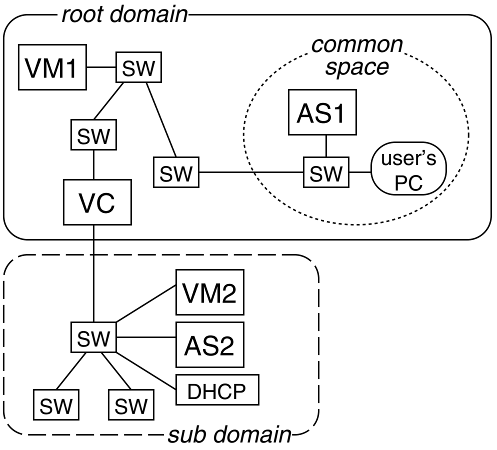

Figure 2. An example of system structure.

ID, not only the destination sub domain can be determined, but also account information of users (user IDs and passwords) can be managed by each sub domain.

3.3. System Structure

A system structure of the proposed method is illustrated in the Figure 2. Features described in the Section 3.1 and 3.2 are realized by the three kinds of components, which are added to existing VLAN network: VLAN Managers, VLAN-ID Converters and Authentication Servers.

- VLAN Manager (VM)

  VMs are located to every domain and manages VLAN-IDs and VLAN switches in its domain. A VM in the root domain (VM1) allocates one of the unused VLAN-IDs for temporary use to a user in a common space dynamically, and sets up a VLAN from the common space to the boundary of the user's sub domain by configuring VLAN switches on the path. VM1 has two databases: a VLAN-ID Database to manage VLAN-IDs for temporary use, and a Switch Information Database which keeps physical connection information of VLAN switches in its domain to set up a temporary VLAN.

  On the other hand, a VM in the sub domain (VM2) determines the VLAN-ID which corresponds to the VLAN which are normally used by the user, and extends the VLAN to the boundary of the root domain by configuring VLAN switches on the path. VM2 also has two databases: a User Mapping Database which keeps user names and corresponding VLAN-IDs, and a Switch Information DB of its domain.

- VLAN-ID Converter (VC)

  VCs, which are located to every boundaries of the root domain and sub domains, convert VLAN-IDs included in Ethernet frames carried across the domains. To convert VLAN-IDs between the root domain and its sub domain, each VC has a VLAN-ID Conversion Database which keeps tuples of VLAN-ID of the root domain and that of its sub domain.

- Authentication Server (AS)

  ASs are located to every common spaces and sub domains. An AS of a common space (AS1) works as proxy between a user's PC in a common space and the AS of the user's sub domain. An AS of a sub domain (AS2) has account information of users of its domain.

Note that these servers are assumed to communicate each other at any time.

3.4. Sequence of temporary VLAN establishment

In the Figure 2, following steps are performed when the user of the sub domain connects his/her PC to the VLAN switch in the common space.

1. After the user's PC is connected to the VLAN switch in the common space, it sends the user ID described in the Section 3.2 to AS1. Note that the user's PC can access only to AS1 at this time.

2. According to the domain name of the user ID, AS1 relays the communication data for authentication from the user's PC to AS2, vice versa.

3. If the authentication between AS2 and the user's PC is succeeded, AS1 sends a request message to VS1. This message includes the user ID.

4. VS1 searches its VLAN-ID DB and gets one of the unused VLAN-IDs for the temporary use. Then, VS1 configures the VLAN switches on the path from the common space to the boundary of the user's sub domain according to its Switch Information DB and set up a VLAN based on the temporary VLAN-ID.

   At the same time, VS1 sends a request message (including the user ID) to VS2.

5. VS2 searches its User Mapping DB with the user name of the user ID and gets the VLAN-ID for the user. Then, according to its Switch Information DB, VS2 extends the user's VLAN to the boundary of the root domain by configuring VLAN switches on the path.

   After the extension of the VLAN is completed, VS2 sends a reply message to VS1. This message includes the VLAN-ID for the user.

6. VS1 sends a request message to VC. This message includes the VLAN-ID sent from VS2 and the temporary VLAN-ID of the root domain.

7. After VC registers these VLAN-IDs to its VLAN-ID Conversion DB, it starts conversion of VLAN-ID of the frame. Then, VC sends a reply message to VS1.

8. After VS1 sends a reply message to AS1, AS1 sends a reply message (means "authentication completed") to the user's PC.

After the final step is done, the user get a data-link level connection to his/her sub domain's network. At this time, the user's PC can obtain an IP address from the DHCP server of the sub domain.

4. Implementation and Evaluation

4.1. Implementation

In order to evaluate the effectiveness of the proposed method, we implemented the servers described in the Section 3.3. All severs were written by C language, and ran on FreeBSD 4.X PCs. The detail of the implementation of each server is described below.

- VLAN Manager (VM)

  The VMs reads the physical connection information of the VLAN switches from the configuration file and constructs its Switch Information DB at the start time. The VM in the root domain also reads the information of the reserved VLAN-IDs to construct its VLAN-ID DB, while the VMs in the sub domains read the mapping information between users and VLAN-IDs to construct its User Mapping DB. In the current implementation, these informations must be specified in the configuration file by the administrators in advance.

  To achieve remote configuration of VLAN switches. we employed Expect[3] to the VMs. Since Expect has a capability of automating interactive applications such as TELNET, the VMs can configure VLAN switches which have remote configuration feature via TELNET.

- VLAN-ID Converter (VC)

  In the normal operation, incoming Ethernet frames are passed to the kernel and stripped its protocol headers. Since we implemented VCs as the user program, the VCs are required to handle Ethernet frames directly to rewrite VLAN-ID field of an Ethernet header.

  To satisfy this requirement, we employed the Berkeley Packet Filter (BPF)[4] to the VCs. With the BPF, a user program can read/write Ethernet frames from/to its Ethernet interface directly.

- Authentication Server (AS)

  As described in the Section 3.2, the ASs in the proposed methods needs an data-link level authentication function. Among the existing authentication methods, we introduced IEEE 802.1X[5] since it is commonly used in the Internet (for example, Windows2000 and WindowsXP natively support it).

  As the implementation of the authentication server based on 802.1X, we employed FreeRADIUS[6]. A FreeRADIUS server works not only normal 802.1X authentication server, but also proxy server. We also modified the program of FreeRADIUS to communicate with the VMs of the proposed method.

4.2. Evaluation

In the actual environment, the setup time of a temporary VLAN and the throughput of the VLAN-ID Converter are important for users. In this section, we describe each experiment and its result.

4.2.1. Setup Time of Temporary VLANs

To evaluate the setup time of temporary VLANs, we constructed a experimental network illustrated in the Figure 3. In our experiment, we employed Cisco Systems Catalyst3550s as the VLAN switches and all the PCs and VLAN switches was connected by 100Mbps Ethernet.

When a user connects his/her PC (WindowsXP) to the VLAN switch in the common space, the user's PC automatically communicates with the AS1 and the login dialog is displayed. After the user input his/her user ID and password to the dialog, the sequence of temporary VLAN establishment described in the Section 3.4 is started.

In this experiment, temporary VLAN establishment was repeated 10 times, and we measured the

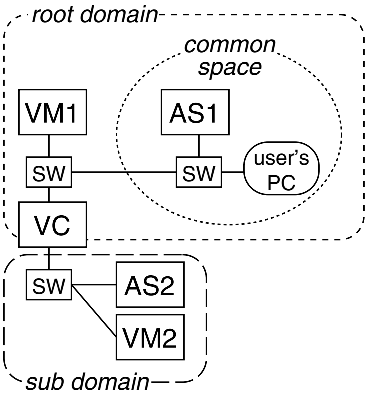

Figure 3. The experiment network #1.

	Time(sec)
authentication time	0.01
configuration time of VLAN switches	3.61

Table 1. The result of the experiment #1.

average time required for setup of the temporary VLAN (the waiting time for the login dialog is excluded).

The result of the experiment is shown in the Table 1. Although most of the time is occupied by the configuration of the VLAN switches, this result seems to be acceptable for many users. In the current implementation, since VMs configure VLAN switches sequentially, the time for the configuration can be reduced by introducing the parallel configuration.

4.2.2. Throughput of the VLAN-ID Converter

In order to evaluate the performance of the VLAN-ID Converter, we constructed another experiment network illustrated in the Figure 4. VLAN switches and speed of the links are the same with the previous experiment.

Furthermore, to make a comparison, we employed VPN software (OpenVPN[7]) which is commonly used for remote access on the Internet. When we use the OpenVPN, VC is stripped from the experiment network and two VLAN switches are connected directly. Since the OpenVPN provides the packet encryption function and the packet authentication function (checking integrity of the packet), we tried three cases about the OpenVPN: (1) both of the two functions are enabled (OpenVPN1), (2) only the packet authentication is enabled (OpenVPN2), (3) both of the two functions are disabled (OpenVPN3).

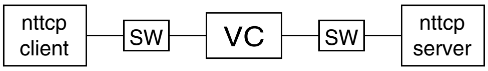

Figure 4. The experiment network #2.

	Throughput (Mbps)
OpenVPN1	65.46
OpenVPN2	83.53
OpenVPN3	85.37
VC	88.39
direct access	89.74

Table 2. The result of the experiment #2.

To measure the throughput, we employed nttcp[8] client and server. In the experiment, 500MB data is transmitted from the nttcp client to the nttcp server. We repeated this transmission 100 times in each case, and measured the average throughput.

The result of the experiment is shown in the Table 2. “direct access” means that the nttcp client accessed to the nttcp server without VC and OpenVPN, it is considered as the maximum throughput of the experiment network. In comparison with direct access, since the overhead of the VC was only about 1.5%, the impact of the VC is considerably small.

On the other hand, the overhead of the OpenVPN varies from 5.1% (OpenVPN3) to 37.1% (OpenVPN1). Since the overhead of the packet encryption is large, it may be disabled for an access within an organization. However, if a VLAN switches on the path encounters MAC address flooding attacks, communication data may leak from all of the ports of the attacked VLAN switch. Therefore, the packet encryption is required even if the communication is limited within the organization. In contrast, in the proposed method, a temporary VLAN-ID is allocated only to the minimum ports of the VLAN switches which are on the path of the temporary VLAN. Consequently, the proposed method have a tolerance for MAC address flooding attacks.

We can conclude based on these results that the proposed method provides faster and secure communication than VPN software.

5. Conclusions

In this paper, we proposed a method of dynamic interconnection of VLANs. Our experiment shows that our method has enough performance while our VLAN converter is implemented as a software program.

Future work includes consideration of a method of automatic search for VLAN switches. In the current implementation, since administrators have to specify the structure information of VLAN switches manually to VLAN managers, administrative cost may become fairly large if a domain has many VLAN switches.

Acknowledgment

This research was partially supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Young Scientists (B), 16700071, 2004–2005, and the Ministry of Internal Affairs and Communications, Strategic Information and Communications R&D Promotion Programme for Top-Priority Research and Development in Specific Fields, 041108001, 2004–2006.

References

[1] IEEE. 802.1Q-1998 IEEE Standards for Local and Metropolitan Area Networks: Virtual Bridge Local Area Networks, *IEEE*, 1998.

[2] Cisco Systems. IEEE 802.1Q-in-Q VLAN Tag Termination,*http://www.cisco.com/en/US/products/sw/iosswrel/ps5207/products_feature_guide09186a00801f0f4a.html*.

[3] D. Libes. Expect Home Page, *http://expect.nist.gov*.

[4] S. McCanne and V. Jacobson. The BSD packet filter: A New Architecture for User-level Packet Capture, *Proc. of The 1993 Winter USENIX Conference*, pp.259–269, 1993.

[5] IEEE. 802.1X-2001 IEEE Standards for Local and Metropolitan Area Networks: Port-Based Network Access Control, *IEEE*, 2001.

[6] The FreeRADIUS Project. FreeRADIUS – building the perfect RADIUS server, *http://www.freeradius.org/*.

[7] J. Yonan. OpenVPN, *http://openvpn.sourceforge.net/index.html*.

[8] E. Bartel. nttcp: New TTCP Program, *http://www.leo.org/˜elmar/nttcp*.