

Provider Provisioned Internet VPN for Personal Communication Environment

Kenji Hori, Kiyohito Yoshihara, and Hiroki Horiuchi

KDDI R&D Laboratories Inc.
2-1-15 Ohara Kamifukuoka-shi Saitama
356-8502, Japan
TEL: +81 49 278 7651 FAX: +81 49 278 7510
{hori, yosshy, hr-horiuchi}@kddilabs.jp

Abstract

This paper proposes a new method for provisioning of Internet VPN (Virtual Private Network), which will be used for ad-hoc personal activity and group work across users' networks over the Internet.

In order to utilize Internet VPN in ad-hoc fashion, the users must configure the VPN routers and hosts in their networks correctly and rapidly. Because these configuration tasks require technical knowledge of the VPN, users often find it difficult and troublesome to accomplish the tasks, even though they are not new to the Internet.

In some method to alleviate this difficulty, part of the configuration of VPN routers is provisioned from the management servers on behalf of the users. Unfortunately, it is not sufficient to enable ad-hoc usage. This is because the complex and time-consuming manual configuration of the VPN routers still remains and is required when joining the VPN. It is also because the hosts' configuration is out of their scope.

In order to enable the ad-hoc usage, we propose a new method for provisioning of Internet VPN as an additional service of an ISP (Internet Service Provider)'s xDSL (x Digital Subscriber Line) or FTTH (Fiber To The Home) service. In our method, the terminal units for such services also serve as a VPN router and DHCP (Dynamic Host Configuration Protocol) relay agent, to minimize the configuration complexity. We also introduce two management servers placed at the ISP: the VPN management server handling VPN routers and the DHCP server providing hosts' configuration. With these servers, users can rapidly join their desired VPN by a simple and fairly small task, using the web-based GUI (Graphical User Interface), through which the VPN router configuration is automatically generated and provisioned from the VPN management server. In addition, users can easily retrieve the hosts' configuration being consistent with the VPN that they are joining, by the coordination of the auto-configuration of the DHCP relay agents and the DHCP server on the terminal units.

We show evaluation results obtained from the testbed including up to 6 VPN routers, a VPN management server, and a DHCP server running on a modest PC. The VPN routers are auto-configured correctly in less than 16 seconds, which is less than the time for the manual configuration. The hosts in at most 500 VPNs are configured within 6 seconds, showing that the proposed hosts' configuration method is adequately scalable in this range of the number of VPNs.

Keywords

VPN, DHCP, configuration management

Contact Person

Kenji Hori (hori@kddilabs.jp)

Introduction



Background

- The Internet has become indispensable for personal activities and group work.
- New requirement for the ad-hoc usage of the communication applications for such activities is rising.
- Internet VPNs are suitable for this requirement, because they are secure and low-cost.

- However, the complex and time-consuming configurations of Internet VPNs obstructs the ad-hoc usage.
- So, the provisioning of the configurations of Internet VPNs from the management servers is desired.

- The existing provisioning methods are not sufficient to enable ad-hoc usage, because:
 - Complex and time-consuming manual configurations of the VPN routers still remain.
 - Configurations of the hosts are out of their scope.

In this paper:

- A new method to automate the provisioning of Internet VPNs is proposed to enable ad-hoc usage.
 - The VPN routers are totally auto-configured by the VPN management server.
 - The hosts' configurations are provided by the auto-configured DHCP server.
- Certain evaluation results obtained from the small network testbed are shown.

Introduction

The Internet has become indispensable to personal activity and group work[1]. This brings new requirements of ad-hoc usage of communication applications, such as desktop sharing and online games between hosts in users' networks over the Internet. In addition, the users want secure communication to exchange private information.

Internet VPNs are suitable for this requirement, because the communication content among the hosts is encrypted by the VPN routers at the edge of the users' networks, and are forwarded through encrypted routes called the VPN tunnels.

In order to use Internet VPNs in ad-hoc fashion, users must correctly and rapidly determine and apply the following configuration, which is unique to the VPN that they are joining: (a) peer IP addresses of the VPN tunnels to be configured for the VPN routers, and (b) an IP address and default gateway address to be configured for the hosts. However, it may be hard for novices and troublesome for most users to perform such complex configuration tasks. Thus, users want provisioning of such configurations on behalf of them. Moreover, provisioning by the management servers rather than operators is desired, because the operators cannot always configure correctly and rapidly the VPN routers and the hosts.

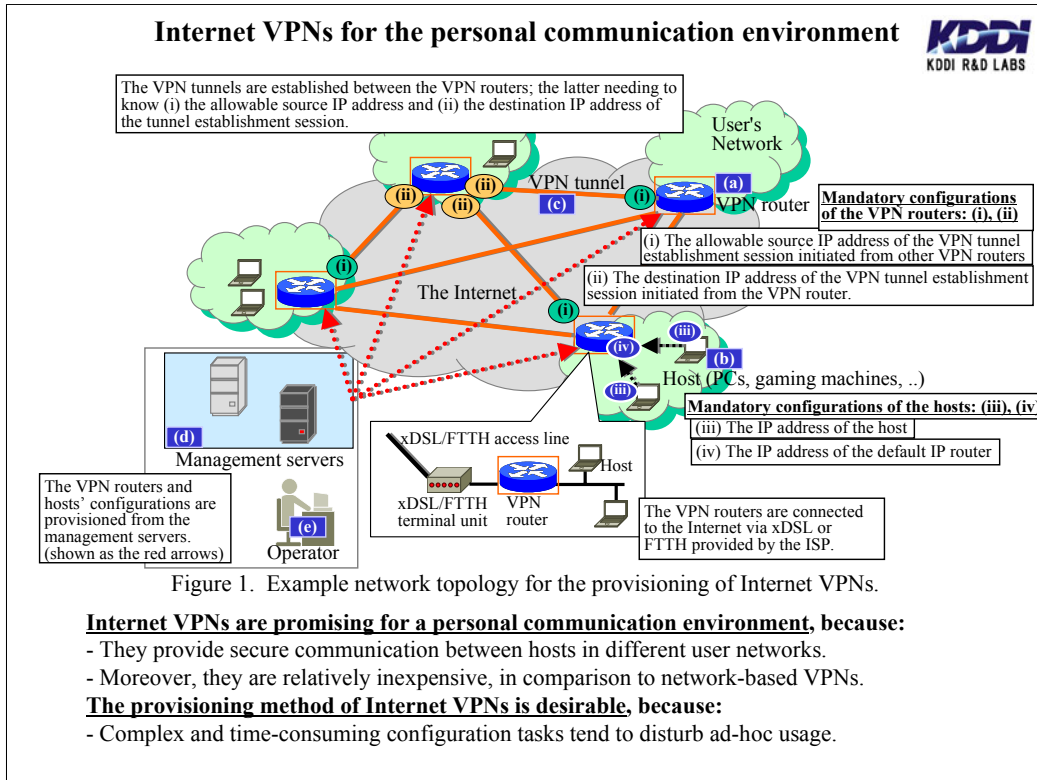
To enable ad-hoc usage, the provisioning method should satisfy the following three requirements: (1) the management servers should be rapidly auto-configured to be able to provision the configurations of the VPN routers and hosts based on users' requests for the creation of a new VPN, (2) the users should be allowed to rapidly join their desired VPN, and (3) the hosts' configuration consistent with a VPN that a host is joining should be provided as well as the VPN routers' configuration. The DHCP (Dynamic Host Configuration Protocol) is desirable as the host's configuration protocol, because it has already been used in common by many hosts.

There exist some provisioning methods. However, to the best of our knowledge, most of them fail to satisfy the requirements for the following reasons: (i) they are unable to auto-configure the management server for a new VPN rapidly to be able to provide the host configuration, (ii) they do not allow users to join their desired VPN rapidly, due to remaining manual tasks such as setting correct IP addresses of hosts for the VPN routers, or due to the disregard to the users' desire in the VPN routers' auto-configuration, and (iii) they cannot provide host configuration, or can provide only through a protocol other than the DHCP.

In this paper, to satisfy all the requirements simultaneously, we propose a new provisioning method for Internet VPN as an additional service of ISP (Internet Service Provider)'s xDSL (x Digital Subscriber Line) or FTTH (Fiber To The Home) access service. In our method, the terminal units of such access services also serve as VPN routers and DHCP relay agents. This relieves users' difficulty in configuring the terminal units when they are joining to a VPN. The unit also provides a simple web GUI (Graphical User Interface), to reduce the complexity and amount of configuration task for creation and/or joining of the users' own Internet VPNs. We also introduce two management servers placed at the ISP: the VPN management server that handles the VPN routers and the DHCP server that provides the host configuration. In our method, (i) users can easily and rapidly auto-configure a DHCP server, and (ii) users are allowed to rapidly join their desired VPN by means of the automatic generation and provisioning of all of the VPN routers' configuration, including the DHCP relay agent's configuration, from the VPN management server. Users accomplish the above (i) and (ii) by simple and fairly small tasks such as entering a brief word into the web GUI. Moreover, (iii) users can retrieve host configuration consistent with a VPN being joined from a DHCP server with support from auto-configured DHCP relay agents.

In order to evaluate the proposed method, we implemented it into a small network testbed, which consists of up to 6 VPN routers, a VPN management server, and a DHCP server. The results show that: (i) auto-configuration of the VPN routers was completed in at most 16 seconds, less than the time for the manual configuration. (ii) hosts in up to 500VPNs are configured within 6 seconds, by a single and fairly modest DHCP server with 512Mbytes of memory and the 2.4GHz CPU. Therefore, the host configuration by the proposed provisioning method is adequately scalable within this number of VPNs.

Finally, we present our conclusions with some future directions.



Internet VPNs for personal communication environment

A VPN (Virtual Private Network) is a group of remotely connected user networks. An Internet VPN is a sort of VPN, in the form of an overlay network over the Internet. It is a promising candidate for a suitable communication environment for personal group work among user members, for the following reasons:

- Communication content, such as data files or digital pictures that probably contain personal information, is secured against other users outside the group.
- They are inherently low-cost compared to network-based VPNs where user networks are interconnected through a costly ISP's closed dedicated network for the VPN. The VPRN (Virtual Private Routed Networks) [3] is an example of network-based VPNs.

An overview of a typical Internet VPN is shown in Fig.1. An IP router connecting a number of user networks is called a "VPN router" (Fig.1 (a)). User hosts (Fig.1 (b)) are connected to the VPN routers. The traffic exchanged among the hosts in different user networks (henceforth VPN traffic) is bypassed into routes specially encrypted by IPSec (Security Architecture for Internet Protocol) or SSL (Secure Socket Layer) tunnels established between the VPN routers, called "VPN tunnels" (Fig.1 (c)). Note that there are two types of Internet VPN in terms of the number of IP subnets exists within: L2 (Layer2) Internet VPNs containing one IP subnet exist within, and L3 (Layer3) Internet VPNs in which multiple IP subnets exist.

However, when users begin to use an Internet VPN, they may face complex and/or time-consuming configurations tasks of the VPN routers and the hosts in front of them. The following configurations, (A) and (B), are mandatory, since they are prerequisites for the correct routing of VPN traffic. Further configuration, such as that required for authentication, may be necessary in practice. [2] and [3] provide one kind of such configuration.

(A) VPN router configuration, such as the allowable source and destination IP addresses for a VPN tunnel establishment session initiated from the local (Fig.1 (i)) and remote VPN routers (Fig.1 (ii)).

(B) Host configuration, such as an IP address of a host (Fig.1 (iii)) and an IP address of a default IP router (Fig.1 (iv)). Typically, the default IP router is identical to the VPN router to which the hosts are connected.

Necessity of the provisioning of Internet VPNs

The configuration of VPN routers and hosts should be provisioned on behalf of the users, to overcome the above-mentioned difficulty of the configuration tasks. Moreover, the auto-configuration from the management servers (Fig.1 (d)) is desired rather than the manual configuration by operators (Fig.1 (e)), because the limited number of operators cannot always configure the large number of VPN routers and the hosts correctly and rapidly.

The requirements for the provisioning method of Internet VPNs



The provisioning method of Internet VPNs should satisfy the following requirements, to enable ad-hoc usage:

(1) Rapid preparation of a new VPN.

A new VPN for newly started user group work should be prepared rapidly upon user request.

- (1a) The preparation for the provisioning of the VPN routers' configurations
- (1b) The preparation for the provisioning of the hosts' configurations

(2) Rapid joining to the user's desired VPN.

The users can rapidly join their desired VPN, whenever a user desires.

(3) Provisioning of the configurations of the hosts in the user's network.

The hosts' configurations should also be provided and must correspond to the VPN being joined. The DHCP is desirable as the host's configuration protocol.

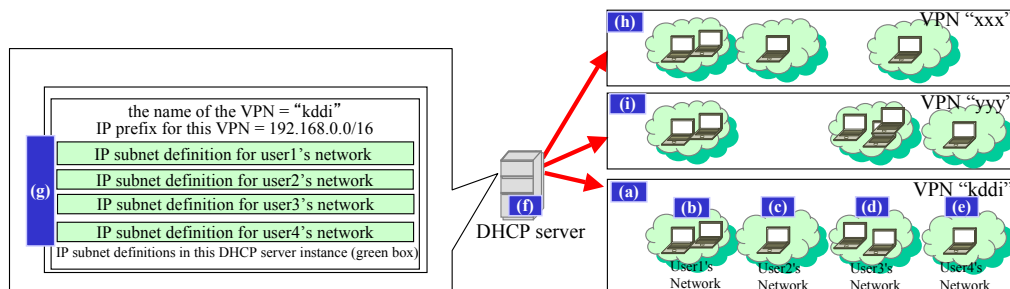


Figure 2. Necessary configurations of the DHCP server for the multiple VPNs and IP subnets.

Requirements for the provisioning method of Internet VPNs

It is desirable that the provisioning method of Internet VPNs (Hereafter "VPN") for personal group work should satisfy the following requirements, to enable ad-hoc usage.

(1) Communication applications for personal group work are likely to be used in ad-hoc fashion, meaning users may wish to use a new VPN impromptu. Hence, a provisioning method should be able to rapidly prepare a new and previously undefined VPN upon user request. More precisely, management servers for VPN routers and hosts should be prepared so that they can provide the configuration of the VPN routers and hosts according to their dynamic joining and leaving. For further discussion, we denote the requirements concerning the management servers for the VPN routers as (1a), and those concerning the management servers for the hosts as (1b).

(2) Because of the ad-hoc nature of personal group work, users will not always participate in the same group work. Therefore, the provisioning method should allow users to join their desired VPN based on their requests. More precisely, the users should be allowed to join the VPN routers to the desiring VPN, without complex and time-consuming configuration of the former.

(3) The hosts in the users' networks should be configured such that they are consistent with the VPN being joined, in order to route the VPN traffic correctly. This means that the configuration should contain not only VPN router configurations, but also the basic IP configurations of the hosts in the users' networks. The IP address of the host and a default IP router address are considered to represent the basic IP configurations of the hosts for correct routing. The configurations must correspond to the VPN being joined. Because most of the Windows/Linux PCs, gaming machines, etc. have the DHCP clients, it is reasonable to assume that users can easily retrieve configurations of such hosts via the DHCP to configure them, if the DHCP server exists and is accessible to the hosts.

In terms of Requirement (3), we note that the difficulty of the configuring the DHCP server increases for the total number of IP subnets defined within, in general. For example, if an L3 VPN (called "kddi", Fig.2 (a)) consists of four user networks as shown in Fig.2 (b), (c), (d) and (e), then the number of IP subnets in this VPN would increase up to four. In this example, four IP subnets must be defined in the DHCP server (Fig.2 (f)) as shown in Fig.2 (g) correctly and rapidly. In addition, if there are multiple VPNs as shown in Fig.2 (h), (i), and (j), then the DHCP server must be configured to be able to serve the hosts in multiple VPNs. These tasks are complex and time-consuming for the users, and even the operators cannot always accomplish such tasks correctly and rapidly. For these reasons, the configuration tasks are an obstruction to the rapid preparation of a new VPN; hence, a means of auto-configuring the DHCP server is important.

The existing provisioning methods of Internet VPNs, and a comparison with our requirements



Table 1. The existing provisioning methods of Internet VPNs, and a comparison with our requirements.

| | | Requirement (1): Rapid preparation of a new VPN | | Requirement (2): Rapid joining to the user's desired VPN | Requirements (3): Provisioning of the hosts' configurations in the user's network |
|------------------------------------|--|---|---|--|--|
| | | (1a) Preparation for the provisioning of the VPN routers' configurations | (1b) Preparation for the provisioning of the hosts' configurations | | |
| (a) Router-oriented methods | UMU-PBNM[4] | No Manual configurations of the management servers are required. | No It cannot provide the DHCP based hosts' configurations. | No Manual configurations of the management servers are required. | No It cannot provide the DHCP based hosts' configurations. |
| | X-Bone[5] | No It relies on the IP multicast, and is hence difficult to deploy through the Internet environment. | No It cannot provide the DHCP based hosts' configurations. | No The VPN routers are automatically joined to the invited VPNs disregarding the users' desire. | No It cannot provide the DHCP based hosts' configurations. |
| | DVC[6] | Yes The users can prepare a new VPN through the operations on the Java-based GUI of the VPN router. | No It cannot provide the DHCP based hosts' configurations. | No The users need to specify all the hosts in their network before any joining. When a user wants to join a VPN, they must also specify all the hosts that they wish to use in the VPN. | No It cannot provide the DHCP based hosts' configurations. |
| (b) Host-oriented methods | RFC2764[3] Note: It provides a framework for the VPRN, rather than the Internet VPN. It is referred to only to discuss the provisioning method of the hosts' configurations. | - | No Manual configurations of the management servers are required. | - | Yes The users can easily retrieve the host configurations via the DHCP. |

Existing provisioning methods of Internet VPNs and comparison with our requirements

Table 1 shows the existing provisioning methods for the configuration of the VPN routers and hosts. They are divided into two categories: (a) router-oriented methods and (b) host-oriented methods.

- (a) Router-oriented methods ([4], [5], and [6]): these methods intend to terminate VPN tunnels at VPN routers and/or hosts.

- (b) Host-oriented methods ([3]): these methods intend to terminate VPN tunnels only at VPN routers, and do not require the hosts to terminate the VPN tunnels.

Table 1 shows these methods alongside our requirements. To the best of our knowledge, they cannot satisfy our requirements simultaneously for the following reasons.

(a) Router-oriented methods

- The UMU-PBNM (University of Murcia Policy-Based Network Management) [4] requires time-consuming manual re-configuration of the management server when preparing a new VPN and joining a VPN. For this reason, they can satisfy neither Requirements (1a) nor (2).

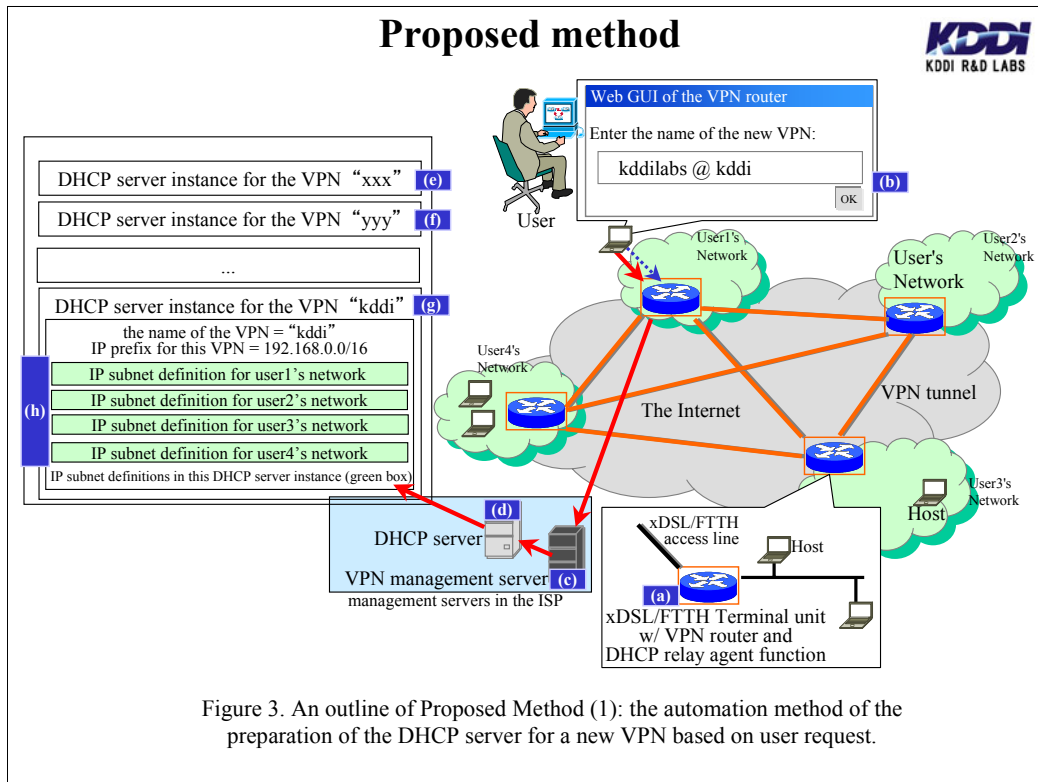
- In the X-Bone [5], firstly a VPN router must send an "invitation" message to create a new VPN, toward the management server of the VPN router. Then, this server forwards the invitation message to other VPN routers via the IP multicast. Next, the other VPN routers send the responses with such multicast messages toward the server, and finally a new VPN is prepared. This dependence on IP multicast makes it almost impossible to deploy [5] on the Internet, where the routing of the IP multicast packet is not widely available. In addition, VPN routers are automatically joined to the invited VPNs regardless of the user's desire. So it does not meet Requirement (2). However, [5] includes a salient feature that the management server can automatically determine and provision VPN router configuration.

- The DVC (Dynamic VPN Controller) [6] can automatically determine and provision configuration of VPN routers. Users are also allowed to create a new VPN. Furthermore, users must specify all the hosts they wish to use in a VPN upon joining. While a VPN router provides Java-based GUI to facilitate such operations, it is still time-consuming. More precisely, in order to configure a particular host, a user must investigate its IP address or DNS name of the host in advance, implying that [6] cannot satisfy Requirement (2).

- [4], [5], and [6] cannot seemingly provide the DHCP-based host configuration, because they configure the VPN routers and hosts by the same protocol as one to configure VPN tunnels. Note that most of the existing DHCP implementations cannot configure VPN tunnels. Therefore, they do not satisfy Requirement (1b) or (3).

(b) Host-oriented methods

- RFC2764 [3] defines a framework of the provisioning method of host's configurations as part of the VPRN framework. In this framework, an ISP is assumed to install DHCP or RADIUS servers. Then, an edge router (the VPN router for the VPRN) acts as a DHCP relay agent or a "mini-DHCP" server, which is delegated an IP prefix from the DHCP or RADIUS server. Users can easily retrieve host configuration from the DHCP server through the DHCP relay agent, or directly from the mini-DHCP server itself. A similar method for this framework may also be applicable to an Internet VPN. However, the DHCP server or RADIUS server still needs to be configured to (i) be able to provision host configuration for all VPNs provisioned by the ISP, and (ii) be consistent with the network topology of each VPN. [3] does not include a means to auto-configure such server, meaning that the framework does not satisfy Requirement (1b).



Proposed provisioning method

As shown in Table 1, most of the existing methods do not satisfy our requirements simultaneously. Therefore, we propose a provisioning method for Internet VPNs that does satisfy the requirements.

In our proposal, the provisioning of Internet VPNs is provided as an additional service of the xDSL or FTTH, and their terminal units also serve as a VPN router and a DHCP relay agent (Fig.3 (a)), to avoid any problem for the users in configuring the terminal units corresponding to the VPN being joined. The unit also provides a simple web GUI (Fig.3 (b)), to reduce the complexity and amount of configuration task for creation and/or joining of the users' own Internet VPNs. We also introduce two management servers placed at the ISP: the VPN management server (Fig.3 (c)) that automatically generates and provisions the VPN routers' configurations, and the DHCP server (Fig.3 (d)) that provides the basic hosts' IP configurations in the users' networks.

Our proposed method is divided into two methods, which satisfy the requirements simultaneously in an integrated manner.

Proposed Method (1): the auto-configuration method of the DHCP server for a new VPN based on user request

This method aims to satisfy Requirements (1b) and (3). Our proposed auto-configuration method is outlined below, along with Fig.3.

- The DHCP server instances for each VPN (Fig.3 (e), (f) and (g)) are auto-configured to have IP subnet definitions (Fig.3 (h)) for the IP subnets exists in each VPN, and started upon the creation of a new VPN. Thus, a single DHCP server serves all existing VPNs simultaneously. While this has not been experimentally examined, this multi-instance architecture is expected to reduce the interference between the DHCP server's provisioning operations for hosts' configurations in each VPN, compared to the architecture where a single DHCP server instance serves all VPNs simultaneously, in terms of the time taken for hosts' configurations.

- The proposed method supports both the auto-configuration of the DHCP server for L2 VPNs and L3 VPNs. However, L2 VPNs are provided as default in order to avoid confusing the users. Relatively advanced users may optionally choose L3 VPN. L2 VPNs may be a better choice when users are not concerned with routing efficiency, because the unlimited broadcast domain makes it easy for service discovery applications, such as the "Network Computers" in the Windows PCs, to find other hosts (or services) within the VPN.

Note that a user who makes a new VPN, can restrict participant users of a new VPN, by enumerating the user-ids that are allowed to join, when entering the name of the new VPN into the web GUI. Because participant users of personal group work typically comprise a small number of friends or family of the user who create the VPN, such simple limitations may be sufficient.

Proposed method (contd.)

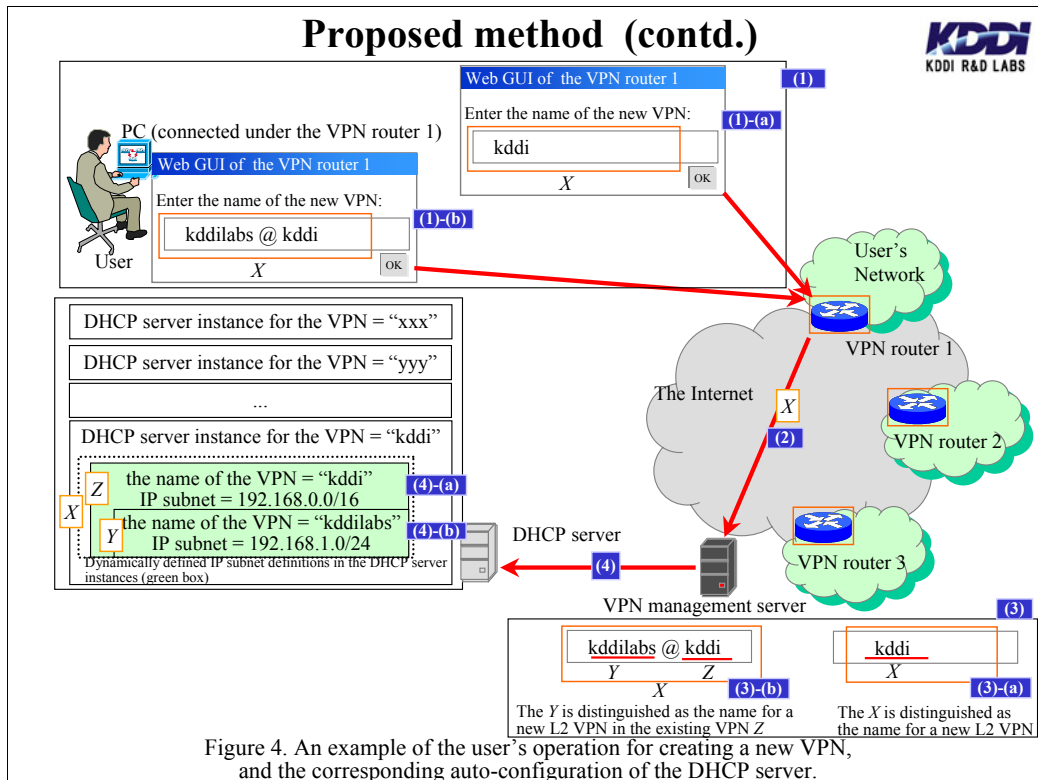


Figure 4. An example of the user's operation for creating a new VPN, and the corresponding auto-configuration of the DHCP server.

An example of the creation of a new VPN and the corresponding auto-configuration of the DHCP server

An example of user operations for creating a new VPN and the corresponding auto-configuration of the DHCP server is shown in below, along with Fig.4. We assume all users are registered with an ISP providing xDSL or FTTH service for them, meaning the VPN management server on the ISP can correctly authenticate the users (using the external RADIUS server, for example). Of course, each user has his own unique user-id assigned by the ISP. However, the authentication process is omitted from the following example for the sake of simplicity. The user can use any one of the VPN routers when creating a new VPN.

(1) A user opens the web GUI of VPN routers (resembling Fig.4 (1)), using a web browser on the PC connected to the VPN router. Then the user enters a brief and arbitrary word X as "the name of the new VPN". For example, the user enters "kddi" as X in Fig.4 (1)-(a), or "kddilabs@kddi" as X in Fig.4 (1)-(b).

(2) The VPN router sends the word X to the VPN management server using the SSL (Fig.4 (2)).

(3) The VPN management server parses the word X according to rules shown below:

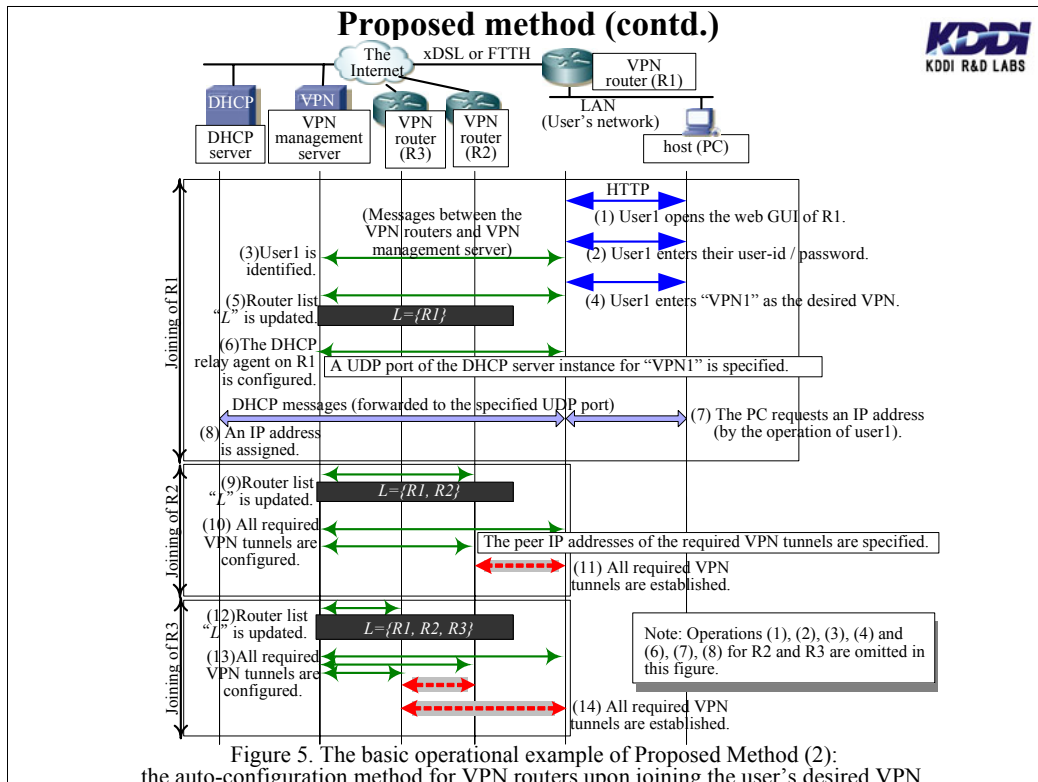
(3)-(a) If the word X does not contain "@" (an at mark) like "kddi" in Fig.4 (3)-(a), then the VPN management server identifies X as the name for the new L2 VPN. In this case, the DHCP server is prepared to assign IP addresses to the hosts in this new L2 VPN.

(3)-(b) If the word X contains "@" as in Fig.(3)-(b), then the VPN management server first parses it into two parts. Y is the part of X in front of "@" ("kddilabs" in Fig.4 (3)-(b)), while Z is the part of the X coming after "@" ("kddi" in Fig.4 (3)-(b)). Next, if Z stands for the L2 VPN previously created by this user, then the VPN management server identifies Y as the name for the new L2 VPN, which the user desired to create as part of the previously created VPN Z . In this case, the DHCP server instance for Z must be prepared to assign IP addresses to the hosts in an IP subnet for Y . In addition, the IP subnet for Y must be defined within the IP subnet of Z . As a result, Y and Z must be prepared as a new L3 VPN. If Z is indeterminate, then certain error messages are presented to the user through the web GUI.

(4) The VPN management server sends certain commands to the DHCP server, in order to prepare the new IP subnet(s) for a new VPN, corresponding to the predetermined configurations of the DHCP server in the process (3) (Fig.4 (4)).

When the case is (3)-(a), to prepare a new L2 VPN X , the VPN management server sends certain commands toward the DHCP server; in order to start a new DHCP server instance, there is a definition of the IP subnet for X . For example, a DHCP server instance for the VPN "kddi" is started, with the IP subnet 192.168.0.0/16 (Fig.4 (4)-(a)). This IP subnet prefix (192.168.0.0) and the prefix length (/16) are pre-defined by the ISP operator as the default settings of the DHCP server instance. The VPN management server specifies a UDP port for the DHCP server instance upon sending the commands, so that the DHCP server instance for X receives the DHCP messages only from the hosts in X . We further describe the relaying of the DHCP message in the next section.

When the case is (3)-(b), to prepare a new L2 VPN Y within the existing L2 VPN Z , the VPN management server sends certain commands toward the DHCP server, in order to add the definition of an IP subnet for the Y to the existing DHCP server instance for the Z . For example, the IP subnet 192.168.1.0/24 for the VPN "kddilabs@kddi" is dynamically and additionally defined in the DHCP server instance for the VPN "kddi" (Fig.4 (4)-(b)). This prefix length (/24) is also pre-defined by the operator of the ISP as the default setting for the DHCP server instance.



Proposed Method (2): the auto-configuration method of the VPN routers upon joining the user's desired VPN

In this section, we describe the auto-configuration method of the VPN routers upon joining the user's desired VPN. The basic operation of this method is shown in Fig.5, whereby the VPN management server rapidly and automatically configures all required VPN tunnels, whenever the user enters "the desired VPN", which is the name of the VPN the user wishes to join, to the web GUI of the VPN router. Note that the user must enter the desired VPN through which the web GUI of the VPN router that the user wants to join to the desired VPN with it. As the prerequisite for joining, each of the VPN routers is assigned a unique identifier from the VPN management server, upon the bootstrap procedure of the VPN routers.

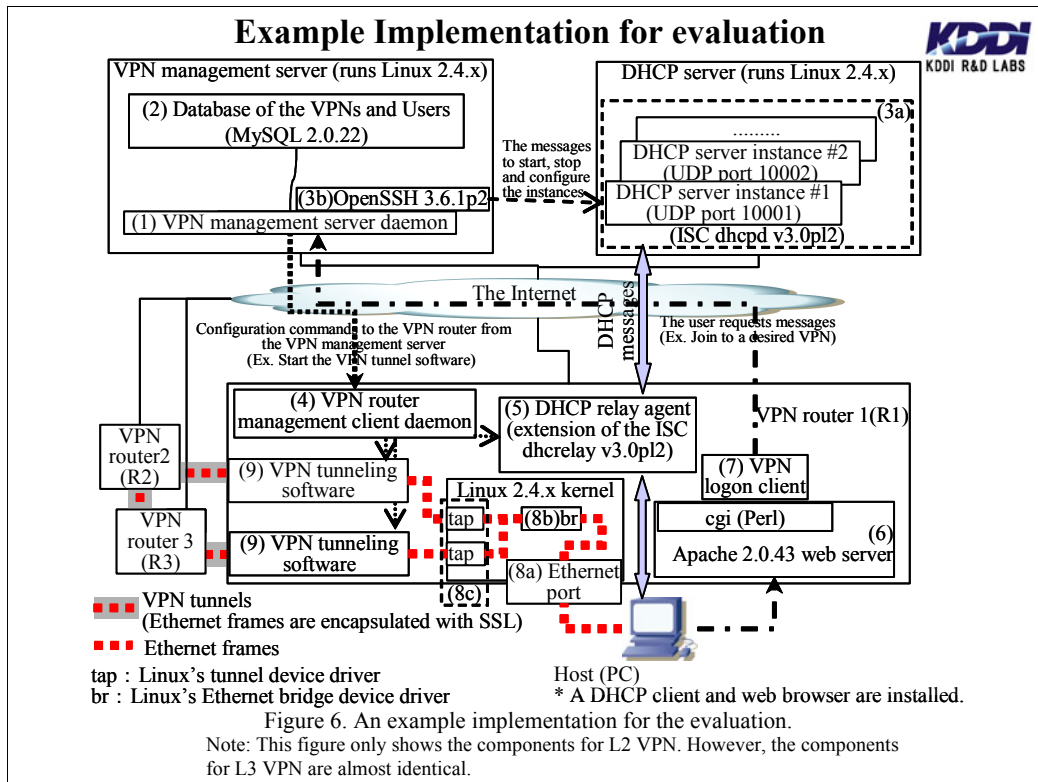
A basic operation example of Proposed Method (2)

An example of the auto-configuration operation involving the three VPN routers (R1, R2, and R3) joined to a L2 VPN sequentially is shown in Fig.5. We note that the operations for L3 VPNs are almost the same.

- (1) A user "user1" who is working with a PC connected under R1 enters the word "VPN1" as the desired VPN into the web GUI of R1 (Fig.5 (4)).
- (2) The R1 sends the desired VPN to the VPN management server along with the R1 identifier. The VPN management server then adds R1 into the "participant router list" L for the VPN1. Thus the $L = \{R1\}$ at this time (Fig.5 (5)). The VPN management server configures all required VPN tunnels between routers in the L . However, the number of all the required VPN tunnels is 0 at this time.
- (3) In order to begin assigning IP addresses to the hosts connected under R1, the VPN management server configures the DHCP relay agent on R1, as the agent to relay the DHCP messages from the hosts toward the UDP port of the corresponding DHCP server instance (Fig.5 (6)). This UDP port is prepared in Process (4) of Feature (1) as described in the previous section.
- (4) Another user "user2" who is working with a PC connected under R2, enters "VPN1" as the desired VPN, in the same manner as did user1. The L is then updated to $\{R1, R2\}$ (Fig.5 (9)). Consequently, the VPN management server configures a VPN tunnel between R1 and R2 (Fig.5 (10), (11)), as part of all the required VPN tunnels. Note that the detailed configurations of the VPN tunnels vary according to the type of the VPN tunnels and the implementations, such as IPSec with Openswan or SSL with OpenVPN. Our VPN management server can determine appropriate configurations for all supported types of VPN tunnels and the implementations.
- (5) Another user "user3" who is working with a PC connected under R3, enters "VPN1" as the desired VPN in the same manner as did user1. Then the L is updated to $\{R1, R2, R3\}$ (Fig.5 (12)). Consequently, the VPN management server configures all the required VPN tunnels between R1, R2 and R3 (Fig.5 (13)).

As aforementioned, the idea of generating configurations of all required VPN tunnels was already proposed ([5], [6]), and is not, in itself, novel. However, we note that the important difference between our method and [5], [6] is that the users are allowed to join their desired VPN while only a few operations, such as entering the desired VPN name, are required.

We also note that the configurations of the DHCP relay agents (described in Process (3) of the following example), which is conjunct with Proposed Method (1), can be distinguished from the existing methods.



An example implementation for the evaluation

We implemented a provisioning system for the configuration of VPN routers and hosts, which includes our aforementioned proposal. The aim of this system is to evaluate our proposal in terms of achieving our requirements. The system components are shown in Fig.6.

The VPN management server

- (1) The VPN management server daemon (Fig.6 (1)) is implemented from scratch using C language. This daemon handles most of the message transaction between itself and the VPN router's management client daemon (mentioned below), for creating a new VPN and joining to the desired VPN, etc.
- (2) A Relational Database of the VPNs (Fig.6 (2)) tracks the name of the VPNs and the *L* (the list of participant VPN routers of a VPN), for example. The user authentication information, such as the user-id and password are also tracked in this database, instead of the external RADIUS or LDAP server being used.

The DHCP server

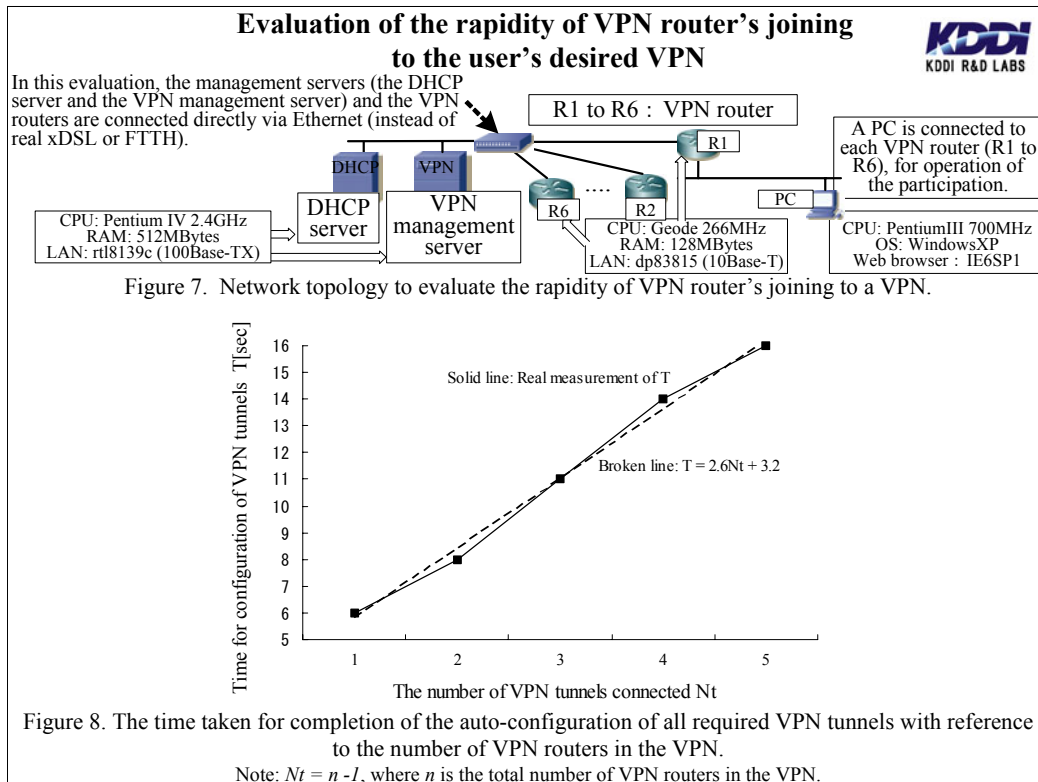
- (3) Multiple DHCP server instances (Fig.6 (3a)) are used for IP address assignment of the hosts in a number of VPNs. These instances are started by the VPN management server daemon using the OpenSSH (Fig.6 (3b)), with the specified listening UDP port, in order to receive the IP address assignment request only from the hosts in a particular VPN.

The VPN routers

- (4) The VPN router management client daemon (Fig.6 (4)) is implemented from scratch with C language. The daemon handles the message transactions between the VPN management server daemon and this client daemon, and starts/stops the VPN tunneling software (mentioned below) with a specified configuration (such as the destination IP address of the VPN tunnel establishment session, etc.) from the VPN management server daemon. The daemon also starts/stops the DHCP relay agent (mentioned below).
- (5) The DHCP relay agent (Fig.6 (5)) is implemented as an extension of the ISC dhcrelay v3.0pl2, in order that the VPN router client daemon can start the agent by specifying a forwarding UDP port, where the DHCP server instance for the VPN is listening.
- (6) In order to provide an interactive user interface, the web server (Fig.6 (6)) is used. The VPN logon client (mentioned below) is executed via certain cgi (perl) scripts from the apache, in order to process the user's login, joining to a desired VPN, etc.
- (7) The VPN logon client (Fig.6 (7)) is implemented from scratch with C language. This is not a daemon program. This program is executed from the cgi, only when the transmission of the request messages to the VPN management server daemon is required. User authentication or their joining to a desired VPN are examples of such requests.
- (8) Ethernet frames received at an Ethernet port (Fig.6 (8a)) are forwarded to the destination Ethernet address (that may exist under other VPN routers) via the "br" (Fig.6 (8b)), and the "tap" (Fig.6 (8c)). One tap corresponds to each one of another VPN router. The br also runs the STP (Spanning Tree Protocol) in order for the routing of Ethernet frames.
- (9) Some VPN tunneling software packages (such as OpenVPN for SSL tunneling or Openswan for IPsec tunneling, Fig.6 (9)) are used to capsule the Ethernet frames into IP packets.

The message formats and the protocols

The message formats and protocols used (i) between the VPN management server daemon and the VPN router client daemon and (ii) between the VPN management server daemon and the VPN logon client are originally defined by us, on top of the TCP/IP and SSL. Detailed message formats and protocols are not shown here. The message format and protocol used (iii) between the DHCP server instance and the DHCP relay agent and (iv) between the DHCP relay agent and the DHCP client on the hosts conforms to the standard of the DHCP.



Evaluation of the proposed provisioning method

In this section, we evaluate the proposed provisioning method in terms of the satisfaction of our requirements, as follows. We used PC hardware with a fairly modest specification of 2.4GHz Pentium IV, 512MBytes RAM and a FastEthernet port as the DHCP server. Another PC, with the same specifications as the DHCP server, was used as the VPN management server. The PC had a 266MHz Geode SC1200, 64MBytes RAM and FastEthernet ports used as the VPN router.

Evaluation of the rapidity of the joining to a user's desired VPN

Our proposal tries to satisfy Requirement (2), through the VPN management server automatically and rapidly generating the VPN router configuration, and throws them into the VPN routers, based on user request.

Therefore, we evaluate the time taken for this configuration operation, as the value of n , which represents the total number of VPN routers participating in a VPN. In this evaluation, the testbed network as shown in Fig.7 was used. This testbed consisted of up to the 6 VPN routers; a number considered sufficient to simulate a VPN for personal group work. In this evaluation, the VPN routers were connected in full-meshed and peer-to-peer topology. In addition, the OpenVPN was used as the VPN tunnel software, to establish SSL/L2 VPN tunnels.

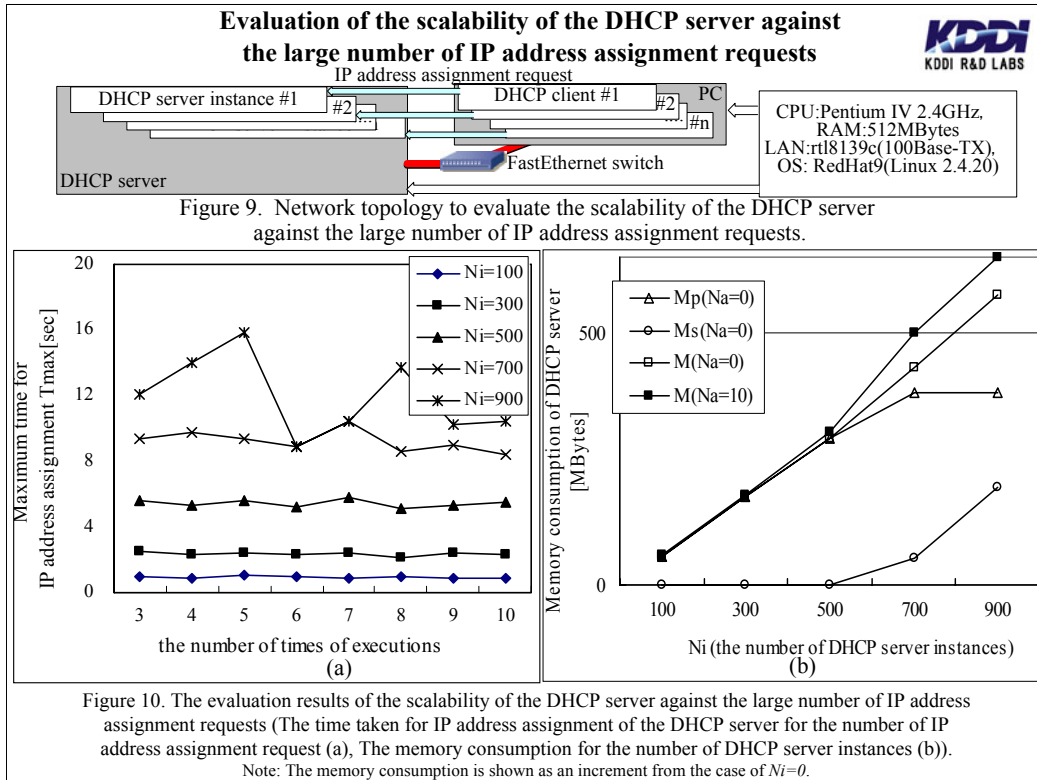
The measurement is executed as follows. Firstly, $(n-1)$ of VPN routers are joined into the VPN in advance. Next, a VPN router joins to the same VPN, by means of entering "the desired VPN" into the web GUI on the PC connected under each of the VPN routers. We measured time T defined as the period from time $T1$ until time $T2$, where $T1$ is the time that the VPN logon client sent the joining request message (shown as an arrow between (4) and (5) in Fig.5), and $T2$ is the time that the establishment of all required VPN tunnels are completed. Both $T1$ and $T2$ were measured by the log file of the VPN routers.

Evaluation result

The evaluation result of the rapidity of the joining to a user's desired VPN is shown in Fig.8. Nt represents the number of the VPN tunnels established in each joining of the VPN router. Because the total number of VPN tunnels between n of VPN routers is $n(n-1)/2$, and the total number of VPN tunnels between $(n-1)$ of VPN routers is $(n-1)(n-2)/2$, so Nt is equal to $n(n-1)/2 - (n-1)(n-2)/2 = n-1$.

The solid line in the result shows the actual measurements, while the broken line shows the linear approximation by $T = 2.6Nt + 3.2$.

First, the actual measurements show that the values of T are always below about 16[sec] for this evaluation. That seems apparently small in comparison with the case where the configuration tasks are manually performed. Moreover, the linear approximation seems to give a fairly good estimation in this range of Nt . From the inclination of the broken line (2.6), we speculate that our evaluation system requires about 2.6 seconds to establish a VPN tunnel in this range Nt . As aforementioned, the actual configuration of a VPN tunnel varies according to the tunnel type and the implementation. Hence, the time taken to establish one may vary according to the type of VPN tunnel involved and its implementation.



Evaluation of the scalability of the DHCP server against the large number of IP address assignment requests

We aim to satisfy Requirement (3) by provisioning the host configurations from the ISP's auto-configured, multiple-instanced DHCP server. In order to evaluate the scalability of this DHCP server, we used the following two metrics:

- (a) The time taken for the IP address assignment of the DHCP server for the number of simultaneous IP address assignment requests.
- (b) The memory consumption of the DHCP server for the number of the DHCP server instances. This number is equal to the number of the VPNs served by the DHCP server.

In order to measure the above metrics, the testbed network shown in Fig.9 is used. The measurement is executed as follows. A given number of N_i of the DHCP server instances is started on the DHCP server. We use 100, 300, 500, 700 and 900 as the values of N_i . At this time, the value of N_a , which is the number of IP addresses assigned from each instance, is equal to 0. Subsequently, the physical memory consumption M_p , the swap memory consumption M_s , and M , which is the sum of M_p and M_s , are measured using the *free* command. Next, for each case of N_i , the PC starts N_i of the DHCP clients sequentially, using a tiny *bash* script. Each individual DHCP client sends a DHCPDISCOVER message, which is an IP address assignment request message, toward the specified forwarding UDP port of the DHCP server. Note that an UDP port is exclusively used by the pair of the DHCP client and the DHCP server instance respectively. Then time T , defined as the period from the start of the DHCP clients (almost the same time as the start of the *bash* script) until the exit of all DHCP clients (each DHCP clients exits with a certain return value to the *bash* script), is measured using the *time* command. The *bash* script is executed 10 times in total for each case of N_i , with intervals of a few seconds. Subsequently, the T_{max} , which is the maximum value of T , is measured for each execution. Finally, after 10 times executions, M is measured again. The DHCP client used in this evaluation is ISC *dhclient* v3.0pl2.

Evaluation result

In Fig.10 (a), the value of T is stable at below 6 sec for a relatively small value of N_i (below or equal to 500). However, the value of T tends to be unstable and long, for the relatively large value of N_i (above or equal to 700), in comparison with the case of the smaller value of N_i . Based on Fig.10 (b), we can see that the value of the memory consumption M_p and M have increased corresponding to the value of N_i , at a rate of about 0.5Mbytes per single instance, where the value of N_a is equal to 0 (none of the IP addresses are assigned) and the value of N_i is relatively small (below or equal to 500). Note that the value of the swap memory consumption M_s is almost at 0Mbytes for this relatively small value of N_i (below or equal to 500), and begins to increase for the relatively large value of N_i (above or equal to 700). This behavior seems to correlate with the unstable and relatively long value of T in Fig.10 (a) for this range of value of N_i . The frequent memory swap-in/out operations of the server's OS are thought to be caused by a simultaneous and large number of IP address assignment requests, in the value range of N_i above or equal to 700. Then, the IP address assignment operations are thought to be disturbed by the swap-in/out operations. Based on the comparison of M between the case where N_a is 0 and N_a is 10 as shown in Fig.10 (b), N_a , which is expected to correlate with the number of hosts in the VPN, has a relatively small impact on M .

Therefore, a generic PC server system, such as used in this evaluation, seems to be able to serve hosts in up to 500 VPNs, while maintaining a relatively small (below 6 seconds) IP address assignment time.

Conclusions



In this paper:

- An automated provisioning method of Internet VPNs is proposed, to enable the ad-hoc usage.
 - The auto-configuration of the DHCP server for the creation of a new VPN is introduced.
 - VPN routers are totally auto-configured by the VPN management server when joining the user's desired VPN.
 - The users can accomplish such tasks with less complexity using the simple web GUI.
- Certain evaluation results obtained in the small network testbed are shown.
 - The auto-configuration of VPN routers was completed in at most 16 seconds, which is less than the time for the manual configuration.
 - The hosts in up to 500 VPNs were configured within 6 seconds, using a single and fairly modest DHCP server with 512Mbytes of memory and a 2.4GHz CPU. Therefore the proposed method is considered adequately scalable.

Future considerations:

- Further evaluations is required to examine under more realistic environment.
 - The time taken for the auto-configuration of the VPN routers, for the case where another type of the VPN tunnels is used.
 - The time taken for the hosts' configuration, for the case where xDSL or FTTH are used.

Conclusions

In this paper, we proposed a new provisioning method of Internet VPN to enable ad-hoc personal group work across the Internet. In our method, the auto-configuration of the DHCP server based on users' request for the creation of a new VPN is introduced, for the configuration of the hosts' in the users' networks. In addition, VPN routers are totally auto-configured when joining the user's desired VPN.

The users can accomplish such tasks with less complexity, using the simple web GUI. Moreover, the hosts in the users' networks can obtain the configuration from the DHCP server via the auto-configured DHCP relay agent, so that the ad-hoc usage of Internet VPNs can be achieved.

Based on the evaluations in our small network testbed, the auto-configuration of VPN routers was completed in at most 16 seconds, which is less than the time for the manual configuration. We also showed that hosts in up to 500 VPNs were configured within 6 seconds, using a single and fairly modest DHCP server with 512Mbytes of memory and a 2.4GHz CPU. Within this number of VPNs, therefore, the proposed method is adequately scalable.

Additional evaluations are still needed, in order to examine the proposed method under a more realistic environment. Firstly, the time for the auto-configuration of VPN routers may vary depending on the type of VPN tunnels and their implementation. Therefore, further evaluations are required when various types of VPN tunnels are deployed. Secondly, our testbed does not use the xDSL or FTTH connection between the DHCP server and hosts. Hence, evaluations for the case where such connections are used are required. Drop of the DHCP/UDP packets might occur in such cases, and this may lead to performance degradation in terms of the time for the hosts' configurations.

Acknowledgments

We are indebted to Mr. Tohru Asami, President, CEO of KDDI R&D Laboratories Inc., for his continuous encouragement with respect to this work. This work is partly supported by the Ministry of Public Management, Home Affairs, Posts and Telecommunications.

References

- [1] D. Hoffman, T. Novak and A. Venkatesh, "Has The Internet Become Indispensable?," Communications of the ACM, Vol.47, No.7, pp.37-42, July 2004.
- [2] M. Lad, "Dynamic VPN Deployment Issues," Department of Computer Science, University College London, Ver.2, June 2003.
- [3] IETF RFC2764. <http://www.ietf.org/rfc/rfc2764.txt> (accessed at April 2005).
- [4] A. Gomez, G. Martinez and O. Canovas, "New Security Service based on PKI," Further Generation Computer Systems, Vol.19, pp.251-262, Elsevier Science, January 2003.
- [5] J. Touch, "Dynamic Internet Overlay Deployment and Management Using X-Bone," Computer Networks, Vol.36, No.2-3, pp.117-135, July 2001.
- [6] "Dynamic VPN Controller (DVC) Demonstrator Project Report," Version 1.2, NRNS Incorporated, Canada, October 2002.