Methods for rapidly testing node reachability on multiple virtual private networks and evaluation

Naoki Tateishi, Naoyuki Tanji, Mitsuho Tahara and Hikaru Seshake
NTT Network Service Systems Laboratories
NTT Corporation
Musashino-city, Tokyo, 180-8585 Japan
tateishi.naoki@lab.ntt.co.jp

Abstract—**Reachability checks with pings are provided as supplementary VPN services of telecom carriers. With the expansion of cloud services, VPN monitoring services are also expanding. However, current routers do not have the short-interval reachability checks necessary for large-scale VPNs. Conventional ping tools can only check the reachability of one VPN (IP-network) at a time. In this paper, we devise ways to improve these ping tools. In order to test the reachability of a number of VPNs simultaneously, it is necessary to devise a routing table with a service ID and an interface mapping table. To ensure performance, it is necessary to invent new methods to send many packets while controlling the number of ping sending units and methods to order reachability checks of the VPNs. The improved ping tools can test the reachability of many VPNs simultaneously within several minutes.**

Keywords-component, large-scale network, VPN, health check, ICMP

I. INTRODUCTION

As cloud services expand, cloud service providers and customers are placing increasing demands on VPN services. Health checks for VPNs are currently provided as a supplemental service [1][2]. In general, health check services use a ping (ICMP echo) function to check VPNs implemented by routers [3]. The number of VPNs and related equipment is increasing, and there is an increasing demand for shortening the health check interval. However, the ping function implemented by routers is insufficient for testing large numbers of VPNs within a short interval. Conventional ping tools can check the reachability only one IP network (1 VPN) at a time and hence are too slow to be applied to many IP networks (Many VPNs).

We have already proposed speedup methods for ping and SNMP [4][5][6]. This paper proposes methods to improve ping tools for reachability checks for large-scale VPNs. There are three problems with using conventional ping tools to check the reachability of VPNs. The 1st problem is lack of scalability to check reachability of large scale network. The 2nd problem is that only 1 VPN can be checked at a time. The 3rd problem is there is no way to ensure the scalability of the current methods with respect to test equipment and VPNs with many timeouts. To solve 1st problem, we propose the method to send multiple ICMP echo at once. To solve the 2nd problem, we propose a method to use the service-ID (SVID) and interface (IF) pairing information with commonly used routing information to decide the interface for sending and receiving packets. To solve the 3rd problem, we propose the method to control the order of the reachability tests for VPNs in order to finish the test in certain terms.

In the 2nd section, we review the various reachability tests, and in the 3rd section, we explain the problem of using ping tools to check the reachability of VPNs. In the 4th section, we propose methods to solve these problems. The 5th section discusses the results of an evaluation of these methods.

II. REACHABILITY TEST METHODS

In order to detect fault promptly in large-scale networks consisting of different equipment, speed and versatility were the important considerations for collecting network state information. The methods developed during this period are broadly classified into log checks such as syslog [7], packet checks on the route such as netflow [8], and pollings such as ping.

Detailed information can be collected by checking logs. However, in some cases, it is necessary to collect a lot of logs from a lot of equipment and it requires long time. The existence of packets can be easily analyzed by checking packets on the route. However, to check end-to-end packet flows, packets have to be checked at a lot of equipment and the load of this sort of probe tends to be high. Using polling, it is difficult to analyze the cause, but determining whether packets have been forwarded from one piece of equipment to other equipment can be easily and directly checked.

The common polling tests include ethernet OAM [9], ICMP ping, and ping for each service (ex. HTTP ping, SNMP [10]). Ethernet-OAM can be used for testing reachability inside individual ethernet subnets. Pings for each service require each service to be started by the equipment. However, the polling test can be done without a service having to be started. The ICMP protocol stack must be implemented with the IP protocol stack, so a lot of equipment supports it. Thus, ICMP ping is ordinarily used for polling reachability tests.

There are numerous conventional ping tools, such as hping [11] and fping [12]. However, these tools are not intended for testing tens of thousands of nodes simultaneously or multiple VPNs, so these ping tools are not suitable for situations in which there are a great many VPNs (large-scale VPNs).

978-3-901882-50-0 ©2013 IFIP

III. Problem of using conventional ping tools to check the reachability of VPNs

The requirements for the ping tool can be defined in terms of the number of VPNs, the number of devices in one VPN, and the frequency of reachability checks. For instance, a large Japanese telecom carrier has a few thousand VPNs. The problem faced by such networks is that the reachability check by ping must finish in a few seconds in order to provide high SLA (service level agreement) services.

The ping function implemented in routers is generally used for the reachability check of VPNs (Fig. 1, left). The scalability of this ping function is insufficient for testing many VPNs simultaneously, but it is difficult to speed-up this function in routers. Thus, we looked into developing a method of connecting ping servers to VPN routers (Fig. 1, right).

In this environment, to enable frequent reachability tests, the tests must be able to be conducted simultaneously and scalability must be ensured. In addition, there is a need for a method of testing multiple VPNs simultaneously. Conventional ping tools cannot test the reachability of multiple VPNs simultaneously. The problem of using ping tools with multiple VPNs is in Fig. 2. In this network, the equipment in VPN-1 has the same IP address as other equipment in VPN-2. Two prefix and next-hop records for each VPNs are registered in the routing table. If conventional ping tools are used for the reachability check of equipment A01, only the IP address of A01 is inputted. At this time, the ping tools refer to the routing table, but since the record selection rules are not defined, it is not certain whether the conventional ping tool can select an NW-A record. If the ping tools select a lower record of the routing table, they cannot test the reachability of the intended equipment.

The time taken for the reachability test can be reduced by executing a number of packet sending and receiving units at the same time. However, if these units max out the resources of the server, increasing their number will not speed up the check because of the overhead for switching tasks. To reduce the time for the test by utilizing server resources, the number of executing units must be limited.

If the targets VPNs are assigned to units in a fixed manner, as the result will be a biased assignment causing idle running packet sending and receiving units. Consequently, the reachability test takes a longer time. To reduce the testing time by making maximum use of the packet sending and receiving units, we propose to assign target VPNs to these units dynamically and efficiently use time-out periods.

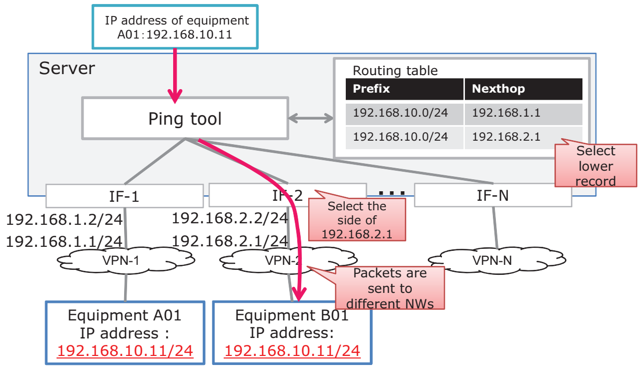

Figure 2. Problem of using ping tools in reachability test of VPNs

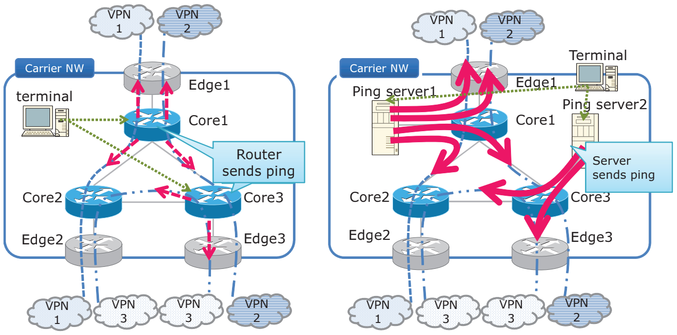

Figure 1. (Left) Ping from routers
(Right) Ping from servers attached to routers

IV. Proposal

We present a method to check the reachability of multiple VPNs simultaneously and a method to improve the scalability of the check.

A. *Sending multiple ICMP ECHO at once*

In the current ping method, one process sends packets to one target (Figure 3 (a)). However, this method is problematic in large-scale networks. We can only execute hundreds of processes simultaneously because many operating systems only permit executions of up to this amount. In addition, we cannot shorten the intervals for sending pings because this method must wait for a reply after sending packets until a timeout occurs.

We solved these problems by dividing the task into threads and by parallelization. Figure 3 (b) shows an example. We assign one thread to send packets and another thread to receive packets. We can send many packets at a faster pace by parallelizing the sending and receiving threads so that they are simultaneous. In addition, the testing period can be shortened by using multi-packet sending and receiving units. This method is also effective at minimizing the testing period because it can finish the reachability test for the nodes that cannot reply quickly.

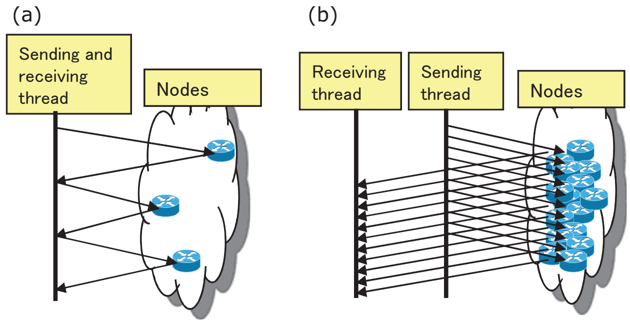

Figure 3. (a) Current ping method (b) Proposed method

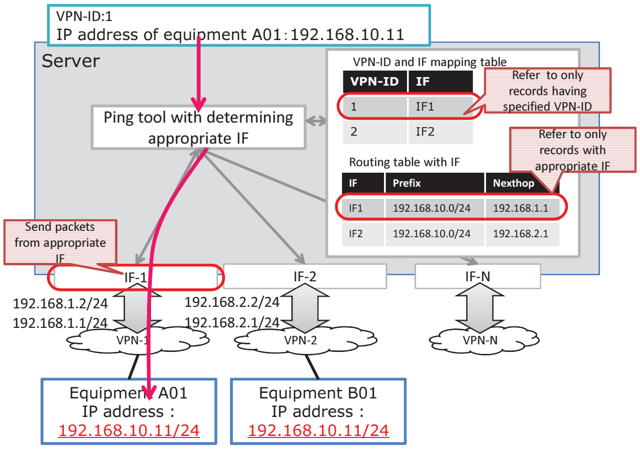

Figure 4. Method of simultaneously testing multiple VPNs

B. *Mapping mechanism so that a single node can assume multiple "virtual interfaces"*

We devised a method that uses a network service ID (SVID) that is uniquely assigned to each VPN, in addition to the IP address of the target nodes.

Before the reachability test, the SVID-IF mapping table and the routing table with the IF information are configured. Before the test, the SVID and IP address are specified by users. The ping tool refers to the SVID-IF mapping table to determine appropriate interfaces to send pings to, and then it looks up only the records which have the interface for sending pings from the routing table with the IF information (Fig. 4).

Using this method, even if the ping tool is connected to a number of VPNs, it can send pings using the appropriate interface and get test results from the intended equipment.

C. *Optimizing the operation of each "unit", using "time out" periods to probe other VPNs*

To reduce the testing time by making maximum use of the packet sending and receiving units, we use dynamically assign the target VPNs to these units and efficiently use the time-out periods.

1) *Dynamic assignments of target VPNs*

If the target VPNs are assigned to packet sending and receiving units in a fixed manner, the result will be biased; some packet sending and receiving units will spend more time than other units. To prevent biased assignment, we establish a queue of the reachability test tasks. If a packet sending and receiving unit finishes a reachability test of a VPN, another VPN are assigned to this unit.

2) *Efficient use of time-out waiting period*

The receiving thread of packet sending and receiving unit waits for the time-out period. And if the receiving thread does not receive reply packet, the unit resends packets. The probability of getting a reply decreases with each iteration of packet resending. Therefore, the probability of getting a reply in the last iteration is very small and tresource consumption is low at the time-out of the last iteration.

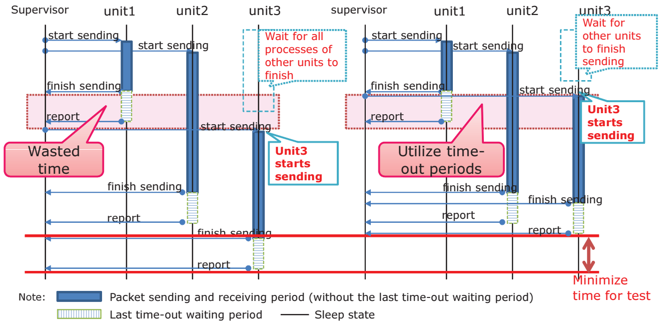

Figure 5. (Left) Unit 3 starts pinging after report of unit 1 (Right) Unit 3 starts pinging after unit 1 finishes

To utilize the resource of the ping server efficiently, the packet sending and receiving units that are booted in excess of the number of parallel executions are put in pause mode. If a unit starts waiting for a reply during the time-out period of the last iteration, the other waiting units start the reachability test of the other VPNs, and consequently the time taken to do the reachability tests for all of the VPNs becomes shorter.

We can illustrate this idea as follows. Suppose that the number of booted units is 3, the number of activated units for sending packets is 2, and the number of target VPNs is 3 (Fig. 5). The supervisor divides units into ones sending and receiving packets (state 1), ones waiting the reply of the last iteration (state 2), and paused units (state 3)

Firstly, the supervisor assigns VPN1 and 2 to unit 1 and unit 2. When unit 1 starts to wait for the reply of the last iteration of packet resending, the supervisor assigns VPN3 to unit 3. Unit 3 then starts the reachability test for the VPN.

Compared with the conventional method in which unit 3 starts the test after unit 1 finishes its time for waiting for the reply of the last iteration, our proposal shortens the testing time.

V. Evaluation

In this section, we evaluate the effect of parallelizing the ping packet sending and receiving units and scheduling of the tests for the VPNs.

A. *Environments*

Target server emulates large scale network and this server was directly connected to ping server. Our proposal is implemented by C language and it works on the ping server. We prepared two environments to the target server for the evaluation of parallelizing the ping packet sending and receiving units and scheduling of the tests. The number of trial is 100. The average testing time and standard deviation (SD) are measured at both environments.

Environment-A was used to evaluate the effect of parallelizing ping packet sending. Environment-A was composed of 1 VPN containing 100,000 nodes.

In addition, Environment-B was used to evaluate the scheduling of the tests. Environment-B was composed of 2000 VPNs on the assumption that VPN services are provided by telecom carriers. The each VPN had 10 nodes. The environment is determined by reference to VPN services in real telecom network. Not so many nodes are connected to VPNs for small corporations, and the number of these corporations is large.

We set two conditions and measured the time taken by the reachability test for all nodes in environments A and B. The conditions are:

1. All nodes respond to pings. (best case)
2. At each VPN, 1 node does not respond to ping. (worst case)

The test parameters in environment-B are shown as below. Dynamic assignment of VPNs was deactivated in pattern 1 and activated in patterns 2 and 3. Use of time-out period was activated only in pattern 3. In pattern 1, the number of nodes assigned to each unit was kept as equal as possible.

The server to execute our software had 2 CPUs (Xeon X5570 2.96 GHz which has 4 cores) and 16 GB memory. The maximum number of retries was 4. In a domestic VPN service provided by Japanese telecom carrier, the maximum one-way delay is guaranteed to be less than 35 ms. In consideration of this, the timeout period was set to 500 ms. The maximum number of retries was 4.

B. Evaluation results and consideration

In environments A1, in best case, Fping require 2.43 minutes with a SD of 0.15sec to finish the reachability test in best case and 11.8 minutes with a SD of 0.35sec in worst case. However, our tools require about 0.90 seconds with a SD of 0.026sec in best case and about 3.70 seconds with a SD of 0.025sec in worst case. Parallelization of packet sending and receiving has a positive impact to shorten testing time.

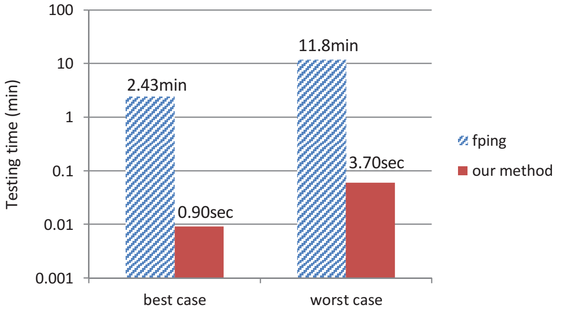

Figure 6. Results in Environment-A

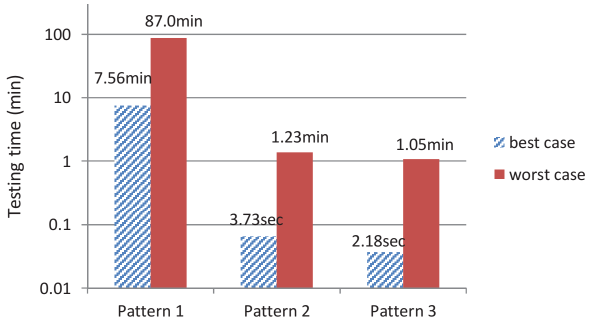

Figure.7 Results in Environment-B

Comparing patterns 1, 2, and 3 in environment B, we found that the difference in testing time. In best case, at pattern 1, 7.56 minutes with a SD of 0.12sec was required to finish test, but at pattern 2, 3.91 seconds with a SD of 0.016sec was required (64 threads are enabled) and the time is less than 0.9% of first case. In addition, at pattern 3, only 2.18 seconds with a SD of 0.014sec was required and the time is less than 55% of second case. In worst case, at pattern1, 87.0 minutes with a SD of 0.017sec was required to finish test, but at pattern 2, 1.23 minutes with a SD of 0.01sec was required and the time is less than 1.6% of first case. In addition, at pattern 3, 1.05 minutes with a SD of 0.001sec was required and the time is less than 78% of second case (Fig. 7). The two proposed methods were shown to have a positive impact by reducing the time for the reachability test of multiple VPNs.

VI. Conclusions

We presented a method for a simultaneous reachability test of multiple VPNs, including a means of speeding up packet sending and receiving and for a means of scheduling reachability tests of multi-VPNs. In the future, we will evaluate our methods in a real environment.

References

[1] KDDI Powered Ethernet, http://www.kddi.com/english/business/powered_ethernet/index.html

[2] Global e-vlan services http://www.ntt.com/gevlan_e/

[3] Postel. J., "Internet Control Message Protocol", STD 5, RFC 792, September 1981.

[4] Naoki Tateishi, Mitsuho Tahara, Yu Miyoshi and Souhei Majima, “Rapid test of node reachability with congestion control and its evaluation,” IEICE Tech. Rep., vol. 107, TM2007-67, pp. 85-90. , March 2008.

[5] Naoki Tateishi et al, “Methods for rapidly testing node reachablity with congestion control and evaluation”, APNOMS2008, 2008/10.

[6] Naoki Tateishi, Sabur Seto and Hikaru Seshake, “A Method to SNMP manager to a large scale network (Encouragement Talk),” IEICE Tech. Rep., vol. 109, no. 378, ICM2009-38, pp. 19-24, Jan. 2010.

[7] R. Gerhards, The Syslog Protocol, RFC 5424, March 2009

[8] B. Claise, "Cisco Systems NetFlow Services Export Version 9," RFC 3954, Jul. 2004.

[9] "IEEE Standard for Local and metropolitan area networks - Virtual Bridged Local Area Networks, Amendment 5: Connectivity Fault Management", IEEE 802.1ag, 2007.

[10] D. Harrington, R. Presuhn, B. Wijnen, An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks, RFC3411, Dec. 2002.

[11] Hping - Active Network Security Tool, http://www.hping.org/

[12] fping, http://fping.org/