Computers and Electrical Engineering 67 (2018) 159–169

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

Handover: A mechanism to improve the reliability and availability of network services for clients behind a network address translator☆

Fu-Hau Hsu[a], Yan-Ling Hwang[b], Kai-Wei Chang[a,*], Chia-Hao Lee[a], Chuan-Sheng Wang[d], Chuan-Kai Kao[c], Zhi-Yao Zhong[a]

[a] Department of Computer Sciences and Information Engineering, National Central University, Taoyuan 320, Taiwan, ROC
[b] School of Applied Foreign Languages, Chung Shan Medical University, Taichung 402, Taiwan, ROC
[c] CyberTrust Technology Institute, Institute for Information Industry, Taipei 106, Taiwan, ROC
[d] Information and Communication Security Laboratory Chunghwa Telecom Co., Ltd. Yangmei, Taiwan, ROC

ARTICLE INFO

Article history:
Received 13 September 2017
Revised 13 March 2018
Accepted 14 March 2018

MSC:
00-01
99-00

Keywords:
Virtual machine
Live migration
Three-way handshake
NAT
DDoS

ABSTRACT

With the rapid growth of Infrastructure as a Service (IaaS), it becomes more important to increase reliability and availability of a service. In our patent, called *Connector* in the paper, we allow a server to be live migrated to another physical host with a different Internet Protocol (IP) address; meanwhile, all existing clients still can use their original connections to connect to the new server. However, a client sitting behind a network address translator cannot use Connector directly. In this research, we propose a new mechanism, called *Handover*, to allow a client sitting behind a network address translator to use Connector. We apply a fake three-way handshake to prevent the redirected traffic from being blocked by a network address translator router. Experimental results show that Handover is effective and the overhead of this changeover process is less than 0.2 s. Furthermore, it may be integrated into a Distributed Denial of Service (DDoS) defense system to guard a host against DDoS attacks.

1. Introduction

Nowadays, Infrastructure as a Service (IaaS) gradually becomes the new choice for many enterprises and individuals to build their own servers instead of using traditional equipment rental services. According to an analysis by Gartner, the market of IaaS is projected to grow 38.4 percent in 2016 to total $22.4 billion, up from $16.5 billion in 2015 [1]. This growth is expected to continue through 2017 and it shows a shift away from traditional Information Technology (IT) services to cloud-based services.

In 2005, a new approach to migrate an Operating System (OS) running services, which is called “Live Migration”, was proposed by Clark [2]. With this method, people can move a running virtual machine from a host to a different physical machines with very little down time. Later on Travostino also verified the feasibility of live migration in a Wide Area Network (WAN) environment [3]. At present, live migration is integrated into almost all mainstream virtualization platforms, such as

☆ Reviews processed and recommended for publication to the Editor-in-Chief by Guest Editor Dr. J-S Sheu.

* Corresponding author.
E-mail address: ccitf03001@ndu.edu.tw (K.-W. Chang).

https://doi.org/10.1016/j.compeleceng.2018.03.020
0045-7906/

Virtualbox [4], VMware [5], Kernel-based Virtual Machine (KVM) [6], and Xen [7]. However, there is no so much research discussing the problem that moving a Virtual Machine (VM) to a host with a different IP address may disconnect existing network connections [3,8–10]. Thus, a new question erased with live migration is if a server is live migrated to another host with a different IP, should all previous Transmission Control Protocol/Internet Protocol (TCP/IP) connections with the server be disconnected first and reestablished later after the live migration. Obviously, if communication between clients and the new server can utilize previous TCP/IP connections without the need to reestablish the TCP/IP connections, then the availability of services provided by the server can continue without any suspension after live migration.

In order to solve this problem, Bradford proposed a temporary network redirection scheme implemented by combining IP tunneling with Dynamic Domain Name System (DNS) [8]. When a VM migration which migrates a VM to a different host is about to be completed, an IP tunnel between the source and the destination will be set up. After that, all packets with the source IP address that is the VM's old IP address will be forwarded to the destination host. In addition, the Dynamic DNS entry for the services provided by the VM will also be updated. This ensures that future connections are directed to the VM's new IP address. Nevertheless, this technique needs the source host to take responsibility for the packet redirection until all connections to the old IP address are disconnected. Hence, the source host cannot be shut off as soon as the destination host takes over the VM. Rather, Travostino et al. [3] handles IP address changes by maintaining a correspondent host with a fixed IP address across migration. It requires the correspondent host to set up an IP tunnel to the host running the new VM, then every client access the service that is running on the new VM via this correspondent host. Ajila suggested to use gateway routers of each host's network to be responsible for packet redirection [9]. Similar to using a correspondent host, this mechanism also makes the source host not be directly involved with the packet redirection. Unfortunately, in certain situations, these methods may not be ideal solutions.

The above methods have the same weakness that the packet redirection is heavily depended on a central node. If the central node fails for some reasons, such as a DDoS attack, the old connections will be broken entirely. Therefore, in our patent, called *Connector* [11] in this paper, we proposed a new idea to deal with this issue. For a TCP/IP connection, called a *carrier TCP/IP connection* hereafter, between a client and a server S_{H1} in a physical host H_1, after the server is migrated to a new server S_{H2} on a host H_2 withe a different IP address, Connector can guarantee that the carrier TCP/IP connection will not be disconnected. In other words, the service that the client obtains will not be disable after the migration, even the server is migrated to a physical host with a different IP address. In this paper, we call the TCP/IP connection between the client and S_{H2} a *redirected connection*.

Connector provides a solution to migrate a virtual machine to another machine with a different IP address in another subnet while keeping its clients' carrier TCP/IP connections alive. Connector is an end-host solution; hence, it does not use a gateway router or an IP tunnel. Connector functions well with clients using public IP addresses. However, a client sitting behind an Network Address Translation (NAT) device cannot use Connector directly. We use NAT to represent a network address translation device or a network address translator hereafter. In order to let a TCP/IP connection between a server and a client passes through an NAT, the related three-way handshake traffic must pass through the NAT. But when using Connector, a client does not establish a new TCP/IP connection with the new server after live migration. A client and the new server just replace the previous server IP address in the related carrier TCP/IP connection with the new server's IP address to maintain the TCP/IP connection between the client and the server. Since the NAT does not see the three-way handshake packets associated with the TCP/IP connection between the client and the new server, an NAT does not allow IP packets related to this TCP/IP connection to pass through it; hence, the TCP/IP connection becomes invalid and a client cannot use Connector directly.

This paper proposes a solution, called *Handover*, to solve the problem that an NAT router blocks a redirected connection due to security concern. With Handover, clients sitting behind an NAT can still use Connector to maintain its connectivity with a sever after the server is live migrated to a different host with a different IP address. Handover utilizes a fake three-way handshake mechanism to ensure a redirected connection is recorded in the NAT router's "current connection" table. We think one of the major possible applications of Connector is to defense a system against DoS/DDoS attacks. When a network server is under DoS/DDoS attacks, administrators of the server can use Connector to move the server to a different host with a different IP address while still keeping existing TCP/IP connections connected. With the help of Handover, a client sitting behind an NAT can still obtain the protection provided by Handover. Besides, Connector and Handover can be used for load balance too. Furthermore, if an administrator needs to maintain or fix his server machine, he can also use Connector and Handover to move related service to a different host without stopping the service and then fix the machine. Our experiments show that Handover can work properly in almost all of the network environments even if a client is in a private network. Without modifying any module or kernel code of the operating system, Handover could be easily implemented by just installing some essential components which are available with apt-get.[1]

[1] A command line package management tool supplied with the Debian package.