I.J. Wireless and Microwave Technologies, **2011, 2, 1-7**
Published Online April 2011 in MECS (http://www.mecs-press.net)
DOI: 10.5815/ijwmt.2011.02.01

Available online at http://www.mecs-press.net/ijwmt

Research and Design of The Firewall Penetration Technology Serving to Informati on Sharing Systems

GAO Shou-ping[a], WANG Lu-da[b]

Computer Science Department, Xiangnan University, Chenzhou Chian

Abstract

In the current information sharing systems, there is a problem of firewall penetration. As for this problem and based on the existing TCP and UDP penetration schemes, we analyze all kinds of possible transmission environments in practical applications, and then present a technique of UDP coordinating with TCP to penetrate firewalls. Finally, we present the realization of this method: a network switch --XIPSwitch, which is independent from various applications. The realization method can further solve the firewall penetration problem of the internal terminals in the existing information sharing systems.

Index Terms: User packet protocol; transmission control protocol; penetration; network switch

1. Introduction

Currently, there is a common problem of firewall penetration existing in the information sharing systems (based on P2P). Firewall, one of basic network security tactics, is able to prevent the untrusty external network visitors from accessing the internal ones. If the communication between the external and internal network administrators is started by the later ones, it, as a result, is usually stopped by firewall and is particularly sensitive to TCP connection. Nevertheless, it’s necessary for the external administrators to start the communication in the information sharing systems. And, as all the administrators are protected in their own firewall, it’s the external administrator that starts the communication between different users every time. The existence of firewall has affected the normal operation of the information sharing systems. Therefore, there is a common problem about how to ensure free communication between the users protected by firewalls. In this paper, we present a kind of combination technique of penetrating firewalls, that is, making UDP coordinate with TCP, which helps solve the problem successfully. Network switch --XIPSwitch, a network adaptor independent from specific application mentioned in the final of the paper, means the realization of this technique.

* Corresponding author.
E-mail address: [a]gaoshoup@263.net, [b]wang_luda@163.com

2. Analyses of Existing Proposals

A. UDP Penetration

The current firewall is generally Cone NAT [1], in simple words, the port, which is distributed to the internal Internet by NAT every time is the same. It will remain unchanged even if it is connected to different hosts (except for being overdue) [2].There is a comparatively reliable solution towards the like NAT or firewall by using UDP penetration.

The circumstance for penetrating Cone NAT is presented in Fig. 1 (NAT A and NAT B are both Cone NAT). Assume that Client A is to send data like files to Client B, the process includes [2]:

1) Client A and B register on Server S respectively via their NAT A and B. Server S logs their NAT's IP addresses and port numbers. (Server S can only get the addresses and port numbers of NAT because it's NAT that communicate with Server S.)

2) Client A informs Server S of sending packet to Client B by NAT A.

3) Server S then sends NAT A's IP and port number of Client A to NAT B.

4) It won't work out like that Client B will send connection data to NAT A after receiving the information by Server S. (Because Client A never connects B, NAT A won't accept the data from Client B.)

5) Client B asks A to connect by informing Server S. After A succeeds in connecting B for receiving the order, terminal A builds up connection with B and sends data to B.

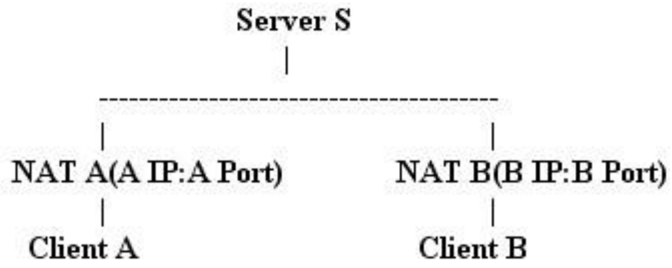

Figure 1. UDP Penetration Environment

The above steps can be successful because the connection's confirm doesn't have such a strict direction as TCP. Then in the UDP penetration circumstance, the task finished in Step (4) is started by Client B on internal network in order to make NAT B identify UDP stream which is about to build next. While it is Step (5) that precipitates Client A finally builds up UDP stream that can be identified by NAT B and started for internal network.

The problem that exists in UDP penetrating is about the performance. Due to the unreliability of UDP, users need to introduce the synchronous verification mechanisms by themselves so as to ensure the correctness and completeness of data transmission. But it has a great effect on the performance to realize these mechanisms on transport layer. In addition, we have some difficulty in realizing this way.

B. TCP Penetration [3]

TCP is more complete than UDP, but in the available schemes, a kind of new mechanism that builds up TCP connection directly between two hosts with NAT is adopted. This mechanism will work under the simple help of the third host. And then, direct TCP connection can be built between two hosts with NAT. Once it is built, the application program can realize mutual communication by the use of standard TCP.

The existing problem for TCP penetration is that it is in need of the third host located in public Internet to transfer and without considering the problem of identification and network flow control.

3. UDP Coordinating with TCP Firewall Penetration

In this paper, in view of the deficiency in the existing schemes, we present a scheme which is a combination of penetrating firewalls technique, namely, UDP connection is used to connect the clients with the server, while TCP connection is used between clients.

C. Analyses of Different Circumstances

In order to analyze all the possible transmission environments, we will discuss different transmission schemes under different Internet environments on the basis of topology structure that is shown in Fig.2, which aims at forming a complete solution scheme on penetrating firewalls. This topology includes a server, two external client terminals and two internal client terminals. And this topology is able to simulate all the possible transmission environments [4].

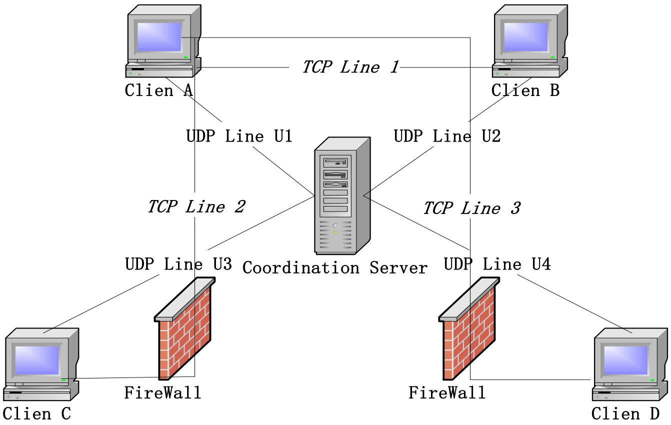

Figure 2. Transmission topology of Firewall Penetration

There were mainly two kinds of Internet transmission, namely, 1:1 and 1: m (one terminal to one and one terminal to more). In this paper, we will only discuss the circumstance of one to one (like MSN voicing service) because one to more transmission could be solved through asking to relay on one's own initiative or using BT models etc. , which is independent from the design of penetrating firewalls.

D. Connection between Client Terminals and the Server

As for the connection between clients and the server, there is bound to be a fixed IP in the server and the corresponding port should be opened no matter it is located in external or internal network to realize mutual connection successfully. Right now, there is no problem for client terminals to connect the server [2]. Principally, we have to solve the problem as to how the server sends out its notice about communication to the client terminals. The following are some usual solutions:

1) Fake information. This refers to the situation that when the server has got some message to be dispatched to the client terminal, the server terminal saves the message in the message pool instead of sending out to the client terminal. And the client terminal should make inquiries from the server about whether there is some message for him or not.

In this model, a reliable connection method—TCP connection is used for it takes the client terminals to connect the server terminal every time, thus it's regardless of the problem of penetrating firewalls [5]. However, to ascertain time interval is not easy. If it is too short, it will result in a heavy burden of network loading on the server terminal and most of the communication being wasted on invalid enquiries. While, if it is too long, its real time will not be sufficient. So the choosing of time intervals has to be determinate comprehensively by factors from all sides after full consideration, which is generally affected by online client terminals' number, communication amount, needs for application real time, etc. Meanwhile, it overheads some cost on realizing and maintaining the message pool. Obviously, this solution seems simple but not easy to realize. What's more, it is difficult to make clients satisfied.

2) Connection Being Kept. When the client terminal logins, it builds up connection with the server by way of TCP and keeps the connection all the time afterwards (or only keeps the session of this connection and then connects with the server by this session)[6]. Once the server gets some message to inform the client terminals, it sends out to them through the connection. At this time, the server terminal needs to keep a connection with every online client terminal and administrate them unifiedly by connection pool, which has a limit of online client terminals numbers, though. This solution is suitable for a comparatively smaller scale of client terminals numbers.

3) UDP Penetrating Firewall Connection. The client terminals build up connection with the server via UDP protocol. And the server terminal logs the IP addresses and port numbers of every client terminal. Hence, the server can communicate directly with clients via UDP protocol with simple principles.

However, as for UDP connection, it is necessary for the clients to set up a verification mechanism to verify their data's completeness. This of course makes it more difficult to realize the connection. To lower the difficulty in verification, we can limit the amount of communication data and assign one UDP packet for every time's communication. But as for communication that contains a greater amount of data, UDP is used only for coordinating while client terminals have to obtain the actual message from the server terminal by the use of TCP.

Considering the merits and demerits of the above ways, we finally decide to adopt UDP penetrating firewalls. Although it seems a little complicated on realization, it can lighten the burden of server's loading.

E. Connection between Client Terminals

The communication between client terminals is realized by TCP protocol (regardless of the circumstance of UDP for the moment). The discussion about solution of dealing with transmission under all circumstances of firewalls is as follows[7]:

1) Connection between clients on external network. In Fig. 2, suppose client terminal A wants to get in touch with terminal B, so A builds up a TCP connection directly with B. In this condition, since there is not a problem of penetrating firewalls, we don't have to discuss about it.

2) Connection between client terminals on internal and external network. Let's take the communication between client terminal A and C in Fig.2 as an example and discuss about it. When C is to communicate with A, a direct connection set up will work and is without the problem of coordination. And the circumstance is basically the same with external network's communication. Now we mainly need to have a discussion about communication between terminal A and C.

When terminal A is to communicate with C, A is unable to get in touch with C directly because C is on internal network. Now terminal A sends a coordinating request about asking C to contact it to the server. Then the server relays it to client C and C builds up TCP connection with A after receiving the request, so A communicates with C by this connection. In this way, it increases the server's cost of relaying a UDP packet for every communication.

3) Connection between Client Terminals on Internal Network. For instance, let's discuss about communication between client terminal C and D in Fig.2. This time, C first sends the server's coordinating application to D (The application kind is from internal network to internal network). Next, the server chooses a terminal A as its proxy from online external client terminals according to some algorithm, sends it a request of transferring and waits for A's confirm. The server, then, after receiving A's confirm, sends A's address to C and D. Right now C and D build up a TCP connection to A dividedly. Afterwards, all the data sent to A from C and D by this connection will be relayed until the communication is over. Right here, the choosing of A has a great effect on the speed of the transmission (we can also choose the way of transferring).

In specific application, the coordinating server doesn't work under no circumstances but usually under the circumstance of Internet program embodied with client terminals' programs. When Internet program logins or connects, it will send UDP to inform the coordinating server. Besides, in the actual application of public network, the choice of transferring station is usually a certain local server.

If C and D are in the same internal network, it will be absurd that C and D communicate by the transferring of A. In order to identify whether C and D are in the same network segment and support optimum algorithm of choosing A ,the server terminal needs to get more information like IP address , port number than the client terminals' information which includes internal network clients' addresses , port numbers, network performance statistics of different IP segment.

From the above analyses, we can draw a conclusion that this scheme works under all the different circumstances.

4. The Realization of XIPSwitch System.

In this chapter, we present a kind of network switch --- XIPSwitch, independent from various applications to realize the coordination of network transmission, internal and external network penetration, choosing of routine and the work of network performance statistics. The independent network switch is applicable to different information systems.

F. Systematic Structure and Functions

Systematic Structure refers to Fig.3.

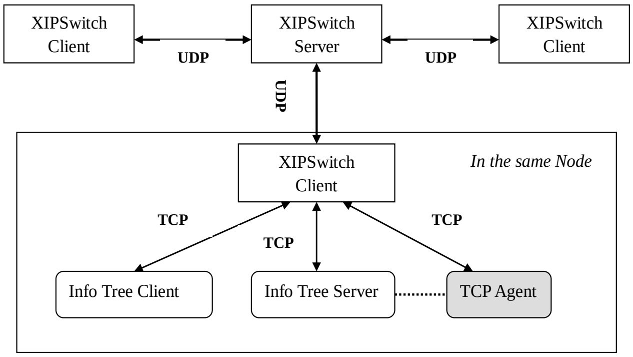

Figure 3. Network Switch and Its Alternating Way with Upper-layer Application.

XIPSwitch--a kind of network switch, a network coordinating module designed by the project, is in charge of providing service for local application to realize network routing and adjustment of penetrating firewalls. The functions that have been realized in general are as follows:

1) Get back to IP address according to fixed node number.

2) Ask for coordinating to penetrate firewalls. (communicate with internal network).

3) Communicate with server to protect the network information of every node.

4) Make network performance statistics so as to be used for optimizing routing.

G. Introduction of Systematic Modules.

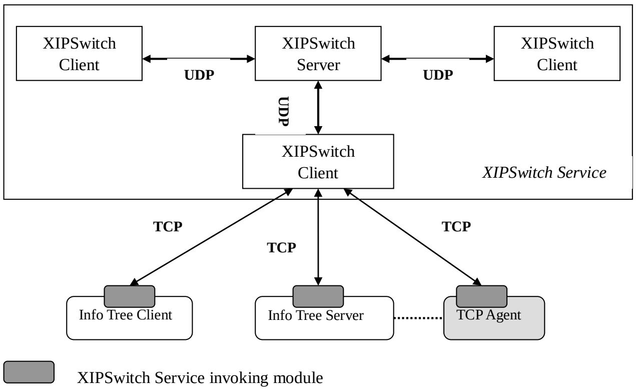

Figure 4. System Structure of XIPSwitch

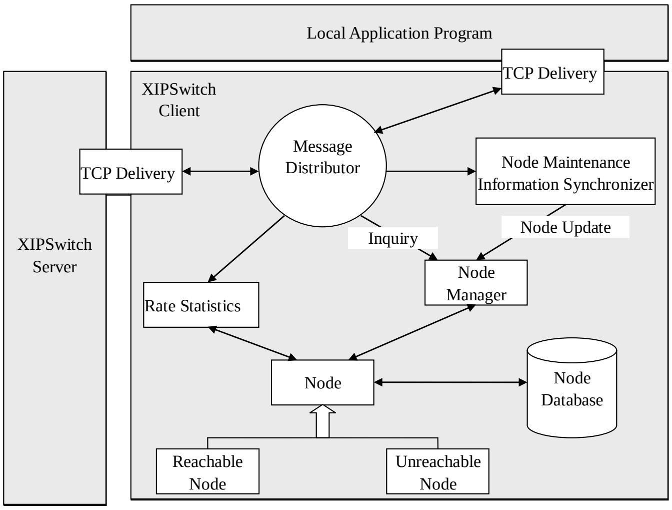

Figure 5. System Structure of Client Terminal

XIPSwitch is composed of two parts, that is, XIPSwitch Service and Service Invoking Module. Fig.4 shows the system structure of XIPSwitch, while Fig.5 displays the system structure of client terminals. Amongst, XIPSwitch offers the service of optimizing network and coordinating and responds to the upper application's request through some agreement to offer these services. While Service Invoking Module undertakes objects-oriented encapsulation towards the way of requesting to provide simple and clear interface for upper application's invoking. In this way, there are two advantages: firstly, it simplified the difficulties that lie in the development for upper application, whose developers can take advantage of XIPSwitch without knowing the details of interacting. Otherwise, it hides interacting details to the upper application. If the interacting protocol alters, they only need to change service invoking module and keep the interface unchanged so that the upper application doesn't need any change. We have discussed XIPSwitch's design and its interactive interface but we will leave the discussion of Service Invoking Module to TCP agent's design, which will be regarded as an independent module to be made important use of in upper application.

5. Conclusion

Information Sharing System has long been a hot spot, being researched especially in today's complicated network structure. To solve its problem of firewall penetration, we present a technique of combining firewall penetration---UDP coordinating with TCP, namely, the client terminals connect with the server through UDP, and the clients use TCP to connect mutually in this paper. In this paper we have confirmed the practicability of this scheme by analyzing all the transmission circumstances. What's more, on that basis, we have designed a network switch---XIPSwitch which is independent from various applications. It can be switched in the information sharing system to coordinate with network transmission, penetrate firewalls on internal and external network and finish routing and the work of network performance statistics for other applications.

However, XIPSwitch still have some deficiency when it comes to the efficiency and safety. Our efforts will be made towards the improvement of these aspects in the following steps.

Acknowledgment

The authors would like to thank Prof. Li Renfa and Dr. Zheng Fanzi for their suggestions.

This work is supported by Hunan Provincial Science & Technology Research Project (No. 2009FJ3194, No. 2010FJ6028), Chenzhou City Science & Technology Research Project (No.2009hj21) and Key Scientific Research Fund of Hunan Provincial Education Department (No.08A064).

References

[1] Egevang K B and Francis P, "The IP network address translation (NAT) ," S . RFC 1631 ,1994.

[2] Srisuresh P and Holdrege M, "IP network address translator (NAT) terminology and considerations," S . RFC 2663 ,1999.

[3] Xie Zhenyu and Xia Qingguo, "The research on establishing TCP connections between hosts behind NATs," J . Science Technology and Engineering.China, vol. 8, pp. 1090-1094, June 2006. (in Chiness)

[4] Lai Dian. "Research and implementation on UDP-based firewall penetration method," J . China Information Security, vol. 8, pp. 75-77, 2006. (in Chiness)

[5] Hu Hao, Dai Zhaojun and Sun Lechang, "Techniques of firewall defensive capability testing ,"J. Computer Applications, vol. 24 (7), pp. 99-101, 2004. (in Chiness)

[6] Wang Qingqing and Du Xu. TCP connectivity through network address translators,"J. Computer & Digital Engineering, vol. 34 (12), pp. 63-65, 2006. (in Chiness)

[7] Li Le and Hou Zhengfeng, "The design and implementation of high-speed firewall,"J . Modern Computer, vol. 07, pp. 20-21, 2006. (in Chiness)