Network Working Group K. Naito
Internet-Draft A. Matsumoto
Intended status: Informational NTT
Expires: March 7, 2013 September 3, 2012

NAT TIME_WAIT reduction
draft-naito-nat-time-wait-reduction-02

Abstract

When network address translation (NAT) is used in an address resource restricted environment, or when a lot of users are located under a NAT device, IP addresses and port resources may be eaten up, and this affects user experiences very negatively. This situation can be greatly mitigated by tweaking mapping behavior and session timer handling in NAT functions. This document proposes extension for optimizing NAT IP address and port resources in address resource restricted environments. The extension makes use of TCP timestamps and sequence numbers for TIME_WAIT assassination.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on March 7, 2013.

1. Introduction

After IPv4 addresses run out, IPv4 address resources will be further restricted site-by-site. If global IPv4 address are shared between several clients, assignable port resources at each client will be limited.

NAT is a tool that is widely used to deal with this IPv4 address shortage problem. However, the demand for resources to provide Internet access to users and devices will continue to increase. IPv6 is a fundamental solution to this problem, but the deployment of IPv6 will take time.

In some cases, e.g. browsing a dynamic web page for a map service, a lot of sessions are used by the browser, and a number of ports are eaten up in a short time. What is worse is that when a NAT is between a PC and a server, TIME_WAIT state of each TCP connection is kept for certain period, typically for four minutes, which consumes port resources. Therefore, new connections cannot be established.

This problem is caused or worsened by the following behavior.

> TIME_WAIT state assigned for a TCP connection remains active for 2MSL after the last ACK to the last FIN is transferred.

We propose mechanisms to change the above behavior that make it possible to save addresses and ports resources.

1.1. TCP TIME_WAIT

The TCP TIME_WAIT state is described in RFC793 [RFC0793]. The TCP TIME_WAIT state needs to be kept for 2MSL before a connection is CLOSED, for the reasons below.

1: In the event that packets from a session are delayed in the in-between network, and delivered to the end relatively later, we should prevent the packets from being transferred and interpreted as a packet that belongs to a new session.
2: If the remote TCP has not received the acknowledgment of its connection termination request, it will re-send the FIN packet several times.

These points are important for the TCP to work without problems.

1.2. TIME_WAIT Assassination

A TCP server MAY accept a TCP SYN for the 5-tuple session that is just finished and marked as TIME_WAIT state,as far as the TCP sequence number is increased. This is known as TIME-WAIT assassination. It should also be noted that some assassination hazards are described in RFC1337 [RFC1337].

1.3. Protect Against Wrapped Sequence numbers (PAWS)

The TCP sequence number wraps frequently especially in a high bandwidth session. PAWS is used to prevent old duplicate packets that occurred in a previous session from being transferred to the new session whose valid TCP sequence numbers happen to overlap with the old duplicate packets. This is implemented by introducing TCP timestamp option, and checking the timestamp option value of each packet. PAWS is described in RFC1323 [RFC1323].

2. NAT resource optimizing extension proposal

2.1. Apply RFC6191 to NAT

RFC 6191 [RFC6191] defines a mechanism for reducing the TIME_WAIT state using TCP timestamps and sequence numbers. This document proposes to apply this RFC6191 [RFC6191] mechanism at NAT. By tracing timestamp and sequence number values in NAT that manages states of traversing TCP sessions, a TIME_WAIT remaining wait-time can be reduced to zero, when a TCP-SYN packet carrying a larger timestamp or sequence number value arrives. In this case, PAWS works to discard old duplicate packets at NAT. A packet can be discarded as an old duplicate if it is received with a timestamp or sequence

number value less than a value recently received on the connection. When there are several clients with nonsuccessive timestamp or sequence number values are connected to a NAT device (i.e. not monotonically increasing among clients), it prevents some clients from getting a port to start a connection for a long time because other clients with larger timestamp or sequence number values are preferred. Two workarounds for this issue are described below.

2.1.1. Rewrite timestamp and sequence number values at NAT

Rewrite timestamp and sequence number values of outgoings packets at NAT to be monotonically increasing.

2.1.2. Split an assignable number of port space to each client

Set some rules among clients connecting to NAT, e.g., split assignable ports between clients. This MAY be done by distributing rules to clients via NAT.

2.2. Resend the last ACK to the resended FIN

In case the remote TCP could not receive the acknowledgment of its connection termination request, NAT, on behalf of clients. resends the last ACK packet when it recieves an FIN packet of the previous connection, and when the state of the previous connection is deleted from the NAT. This mechanism should be used when clients starts closing process, and the remote host could not receive the last ACK.

2.3. Remote host behavior of several implementations

To solve the port shortage problem on the client side, the behavior of remote host should be compliant to RFC 6191 [RFC6191] or the mechanism written in 4.2.2.13 of RFC1122 [RFC1122], since NAT may reuse the same 5 tuple for a new connection.We have investigated behaviors of OSes (e.g., Linux, FreeBSD, Windows, MacOS), and found that they implemented the server side behavior of the above two.

3. Security Considerations

Security issues are not discussed in this memo.

4. Normative References

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.

[RFC1122] Braden, R., "Requirements for Internet Hosts - Communication Layers", STD 3, RFC 1122, October 1989.

[RFC1323] Jacobson, V., Braden, B., and D. Borman, "TCP Extensions for High Performance", RFC 1323, May 1992.

[RFC1337] Braden, B., "TIME-WAIT Assassination Hazards in TCP", RFC 1337, May 1992.

[RFC6191] Gont, F., "Reducing the TIME-WAIT State Using TCP Timestamps", BCP 159, RFC 6191, April 2011.

Appendix A. Revision History

02: Changed intended status to "informational".

01: 'draft-naito-nat-resource-optimizing-extension-01' was divided into two drafts after IETF83 meeting.
'draft-naito-nat-resource-optimizing-extension-01' containes two mechanisms. One mechanism, TIME_WAIT reduction is written in this draft, and the other is written in
'draft-naito-nat-port-overlapping'.

Authors' Addresses

Kengo Naito
NTT NT Lab
3-9-11 Midori-Cho
Musashino-shi, Tokyo 180-8585
Japan

Phone: +81 422 59 4949
Email: naito.kengo@lab.ntt.co.jp

Arifumi Matsumoto
NTT NT Lab
3-9-11 Midori-Cho
Musashino-shi, Tokyo 180-8585
Japan

Phone: +81 422 59 3334
Email: arifumi@nttv6.net