BEHAVE Working Group
Internet-Draft
Intended status: Informational
Expires: November 15, 2008

D. Wing
Cisco

May 14, 2008

Dynamic TCP Port Reuse for Large Network Address and Port Translators
draft-wing-behave-dynamic-tcp-port-reuse-00

Abstract

A strawman proposal is made to reduce public-facing TCP port use with large Network Address and Port Translators (NAPTs). This proposal attempts to preserve emergent NAT traversal techniques. It is anticipated that large NAPTs will be used for NAT64.

This document describes a strawman proposal for discussion on the BEHAVE mailing list, <https://www.ietf.org/mailman/listinfo/behave>.

Table of Contents

1. Introduction

With large NAPTs, it is desirable to re-use public TCP ports in order to conserve IPv4 address space [Iljitsch]. This paper describes three mechanisms to accomplish this goal for TCP.

UDP is not considered in this paper, as the primary goal of port re-use is for TCP applications.

2. Notational Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Algorithms

This document proposes three algorithms with increasing level of complexity. ALGO-1 works for TCP client/server applications (e.g., HTTP, SMTP, IMAP), ALGO-2 works for TCP peer-to-peer applications that use emergent NAT traversal techniques such as UNSAF [RFC3424] mechanisms to learn their public TCP port. The most advanced algorithm, ALGO-3, reuses public TCP ports for client/server applications while also allowing UNSAF applications to function. All three algorithms are presented in this paper to discuss the incremental improvements.

With all algorithms proposed below, a new TCP connection from an internal host cannot re-use the same public TCP source port and address to the same remote destination port and address (i.e., you cannot use the same 5-tuple for a new connection).

3.1. Algorithm for client/server applications (ALGO-1)

Upon seeing a new TCP SYN from the internal interface, the NAPT shall choose a public TCP ephemeral port that is not already used (this is today's REQ-1 in [I-D.ietf-behave-tcp]).

ALGO-1: If all of the public TCP ports are used, the NAPT shall choose a TCP port that is already used. It may select any port that has completed its TCP handshake. Once selected, that port cannot be chosen for additional TCP port re-use until it, too, has completed its TCP handshake.

Thus, the following situation could occur if the TCP handshake between H1 and H2 had completed, and then H1 needed another TCP connection with host H3:

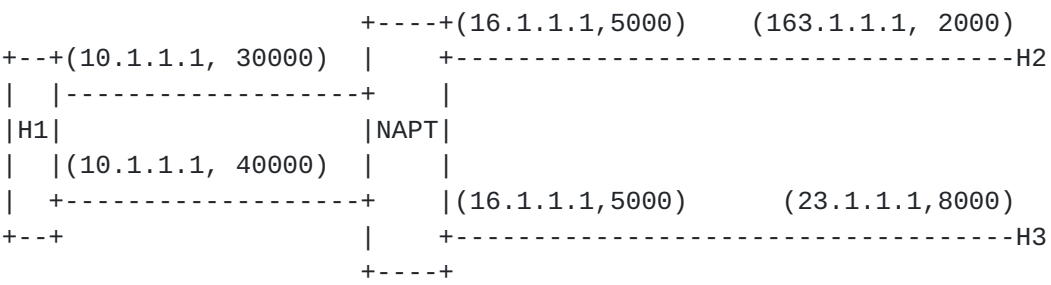

Figure 1: Same Source TCP Port on NAPT

ALGO-1 should work very well for client/server applications and provides the desired port overloading.

A drawback of ALGO-1 is that it violates REQ-1 of [I-D.ietf-behave-tcp]. This is because between the time the host performs UNSAF and learns its IP address and TCP port, the NAPT might re-use that public TCP port for another host (or application running on the same host) to re-use that same public TCP port. If the NAPT re-uses the TCP port with an UNSAF application (which expects an incoming TCP connection), an incoming TCP connection cannot be sent to the correct internal host -- thus breaking UNSAF. However, this can be improved upon with ALGO-2, below.

3.2. Algorithm for Peer-to-Peer Applications (ALGO-2)

Peer-to-peer TCP applications require a host, behind a NAPT, to listen for and respond to incoming TCP connections. In order to do so, many of these applications utilize UPnP or NAT-PMP to cause their IPv4 NAPT to forward incoming TCP connections to the host. After learning the public IPv4 and TCP port via UPnP or NAT-PMP, that information is communicated to other hosts participating in the peer-to-peer network. Because UPnP and NAT-PMP both utilize broadcast or multicast packets, and do not support nested NAPTs, they are not suitable for use by an ISP's NAPT.

For these reasons, it is anticipated that peer-to-peer TCP applications will migrate to using an UNSAF [RFC3424] technique (e.g., STUN [I-D.ietf-behave-rfc3489bis]). A host using an UNSAF technique learns its public IP address and TCP port and then tries to cause the NAPT to re-use that learned public IP address and TCP port for a subsequent connection to a different host (REQ-1 of [I-D.ietf-behave-tcp]). In order to cause the NAT to re-use the same public port for a new TCP connection, the host re-uses the same local

TCP port for the connection to the different host. The NAPT can take advantage of this characteristic of UNSAF applications to determine if the port can be re-used.

ALGO-2 requires TCP UNSAF applications signal they are using UNSAF (often they do this as a matter of their normal operation), and a change to the NAPT:

- TCP UNSAF applications need signal the NAPT that they are TCP UNSAF applications. Such TCP UNSAF implementations would need to make two connections from its same source TCP port to two different hosts, and make that connection within a finite length of time -- such as 30 seconds.

  (The first host is the host running the UNSAF protocol (e.g., the STUN server); the second host is any other host on the Internet (e.g., a remote host on the Internet, or NAPT's own public IP address, or even just a sink-hole IP address).

- 30 seconds after the TCP connection is established, one of two things would have occurred:

  * The host has re-used its internal TCP port for a connection to a different remote host. This indicates the host is using an UNSAF mechanism, and the NAT needs to conform to REQ-1 of [I-D.ietf-behave-tcp] for this connection, or;

  * The host has not re-used its internal TCP port for a connection to a different remote host. This indicates the NAPT can re-use the public TCP port for another connection.

This means short-lived connections (such as HTTP/1.0 connections) would receive less direct benefit from this p2p-friendly port-reuse scheme (but see , below). Longer lived client/server connections (e.g., IMAP, HTTP/1.1 with keepalive) would not trigger the p2p-friendliness described in this section and would reuse public TCP ports after the ~30 second wait to decide if the internal host was using UNSAF.

Thus, with ALGO-2 if the TCP handshake between H1 and H2 had completed and (within 30 seconds) H1 uses the same source port (30000) when making a connection within 30 seconds to H3, the NAPT will allocate the same external TCP port (5000):

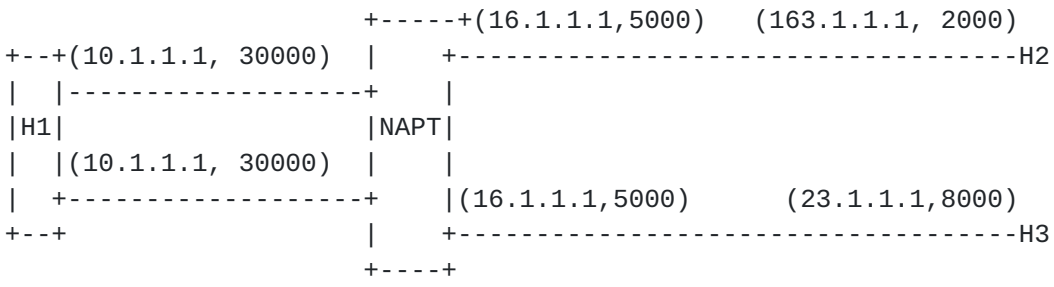

Figure 2: Same Source TCP Port on Host

When another host, H4, connects, the TCP SYN is forwarded to H1:

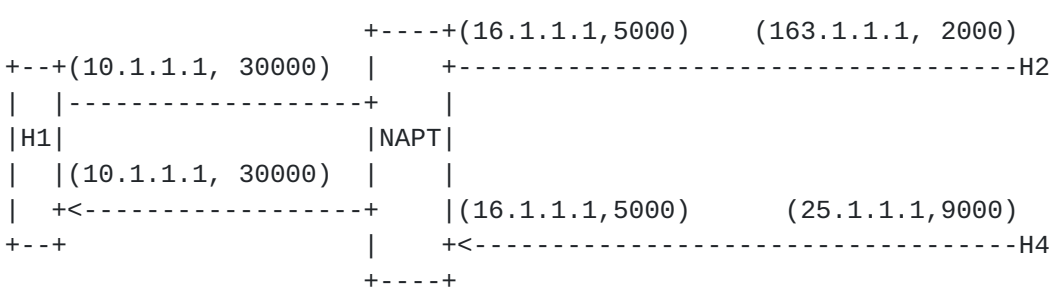

Figure 3: Incoming TCP connection

Cleanup: The NAPT can re-use the public TCP port once the TCP session has been torn down (TCP RST, FIN, or other similar shutdown indicator) and TIME_WAIT seconds have elapsed.

A drawback to ALGO-2 is that a specific port cannot be re-used until ~30 seconds have elapsed. By that time, some applications will have finished their TCP connection (e.g., short-lived HTTP connections). However, this can be improved upon with ALGO-3.

3.3. Algorithm for Client/Server and Peer-to-Peer (ALGO-3)

The final algorithm provides peer-to-peer friendliness while also providing better TCP port re-use -- when viewed at a system level. That is, with multiple TCP connections from multiple applications on multiple hosts, there is a good re-use of TCP ports. This is because multiple longer-lived TCP connections are allowed to persist on a specific port, and new connections are allowed to also re-use that same port. Once a connection has lived for its 30 seconds on a port, a new connection is allowed to re-use that same port.

The following algorithm further modifies ALGO-1's port selection to re-use TCP ports that are not needed by an UNSAF application.

ALGO-3: If a host has not re-used its internal TCP source port for a TCP connection within 30 seconds, the NAPT should prefer to re-use that public TCP port for the next TCP connection.

Discussion: After 30 seconds the NAPT knows the connections are client/server TCP connections because the internal host has not re-used the source TCP port. If the host does re-use its source TCP port, the NAPT MUST forward all incoming TCP connection requests to that host.

In this way, once a connection has been alive for 30 seconds and the host has demonstrated it doesn't need to an accept incoming connection (that is, the host has not exhibited characteristics of an UNSAF application), that public TCP port can be re-used by another (new) TCP connection.

The following figure shows H5 has established two TCP connections with two different servers (H6, H7), which the NAPT happened to place on the same public TCP port, 5000. This happened because the first connection had already persisted for 30 seconds and the NAPT decided it could re-use that port for another TCP connection. Then, after the second TCP connection had persisted for 30 seconds, H8 established a connection to yet another host and the NAPT decided to re-use port 5000 again. However, H8 is using UNSAF -- which the NAPT noticed because H8 connected to another host within 30 seconds using its same source TCP port (40000) [that connection is not shown in the diagram]. Thus, because of this, the NAPT will not re-use this public TCP port for another connection but instead forwards all incoming TCP connections (TCP SYNs) to port 40000 on H8.

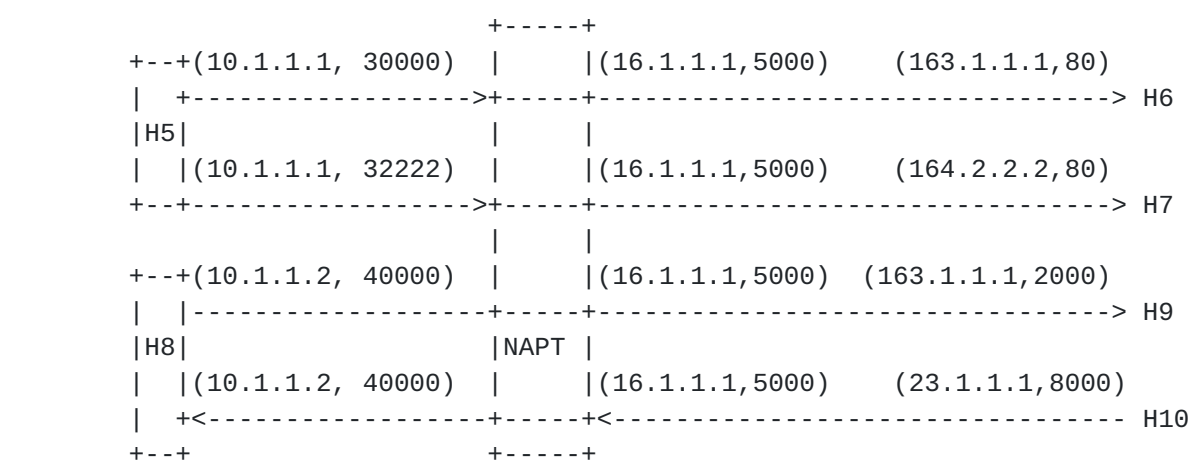

Figure 4: Client/Server and Peer-to-Peer

Cleanup: If a port was used by an UNSAF application, the NAPT can re-use that public TCP port once the UNSAF application has stopped (TCP RST, FIN, or other similar shutdown indication), and TIME_WAIT for the UNSAF application has elapsed.

4. Security Considerations

sTBD.

5. IANA Considerations

There are no IANA considerations for this document.

6. References

6.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

6.2. Informative References

[I-D.ietf-behave-rfc3489bis]
Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, "Session Traversal Utilities for (NAT) (STUN)", draft-ietf-behave-rfc3489bis-15 (work in progress), February 2008.

[I-D.ietf-behave-tcp]
Guha, S., "NAT Behavioral Requirements for TCP", draft-ietf-behave-tcp-07 (work in progress), April 2007.

[Iljitsch]
van Beijnum, I., "Scalability of endpoint independent mapping", May 2008, <http://www.ietf.org/mail-archive/web/behave/current/msg03821.html>.

[RFC3424] Daigle, L. and IAB, "IAB Considerations for UNilateral Self-Address Fixing (UNSAF) Across Network Address Translation", RFC 3424, November 2002.

Author's Address

Dan Wing
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134
USA

Email: dwing@cisco.com

Full Copyright Statement

Intellectual Property

Acknowledgment

This document was produced using xml2rfc v1.33 (of http://xml.resource.org/) from a source in RFC-2629 XML format.