TSVWG   R. Penno
Internet-Draft   Cisco
Intended status: Best Current Practice   S. Perreault
Expires: May 23, 2014   Viagenie
                                                           S. Kamiset
                                                    Insieme Networks
                                                        M. Boucadair
                                                      France Telecom
                                                            K. Naito
                                                                 NTT
                                                   November 19, 2013

Network Address Translation (NAT) Behavioral Requirements Updates
draft-ietf-tsvwg-behave-requirements-update-00

Abstract

   This document clarifies and updates several requirements of RFC4787,
   RFC5382 and RFC5508 based on operational and development experience.
   The focus of this document is NAPT44.

Requirements Language

   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 RFC 2119 [RFC2119].

Table of Contents

1. Terminology

The reader should be familiar with all terms defined in RFC2663
[RFC2663],RFC4787 [RFC4787],RFC5382 [RFC5382],RFC5508 [RFC5508]

2. Introduction

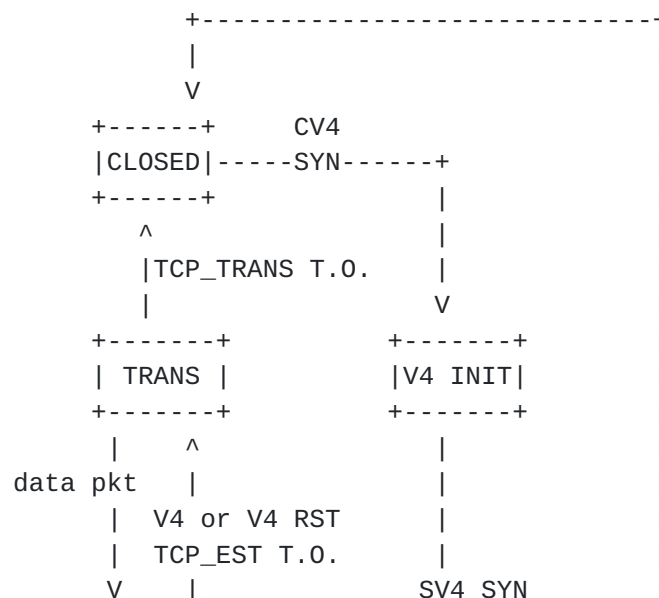[RFC4787], [RFC5382] and [RFC5508] greatly advanced NAT
interoperability and conformance. But with widespread deployment and
evolution of NAT more development and operational experience was
acquired some areas of the original documents need further
clarification or updates. This documents provides such
clarifications and updates.

2.1. Scope

This document focuses solely on NAPT44 and its goal is to clarify,
fill gaps or update requirements of [RFC4787], [RFC5382] and
[RFC5508]. It is out of the scope of this document the creation of
completely new requirements not associated with the documents cited
above. New requirements would be better served elsewhere and if they
are CGN specific in an update to [RFC6888].

3. TCP Session Tracking

[RFC5382] specifies TCP timers associated with various connection
states but does not specify the TCP state machine a NAPT44 should use
as a basis to apply such timers. The TCP state machine below,
adapted from [RFC6146], provides guidance on how TCP session tracking
could be implemented - it is non-normative.

```
                           +----------------------------+
                           |                            |
                           V                            |
                       +------+     CV4                 |
                       |CLOSED|-----SYN------+          |
                       +------+              |          |
                         ^                   |          |
                         |TCP_TRANS T.O.     |          |
                         |                   V          |
                       +-------+         +-------+      |
                       | TRANS |         |V4 INIT|      |
                       +-------+         +-------+      |
                         |    ^              |          |
                   data pkt   |              |          |
                         |  V4 or V4 RST     |          |
                         |  TCP_EST T.O.     |          |
                         V    |              SV4 SYN    |
```

```
                    +--------------+          |              |
                    | ESTABLISHED  |<---------+              |
                    +--------------+                         |
                       |          |                          |
                     CV4 FIN    SV4 FIN                       |
                       |          |                          |
                       V          V                          |
                 +---------+   +----------+                   |
                 |CV4 FIN  |   | SV4 FIN  |                   |
                 |   RCV   |   |   RCV    |                   |
                 +---------+   +----------+                   |
                       |          |                          |
                    SV4 FIN    CV4 FIN                  TCP_TRANS
                       |          |                        T.O.
                       V          V                          |
                 +----------------------+                    |
                 | CV4 FIN + SV4 FIN RCV|--------------------+
                 +----------------------+
```
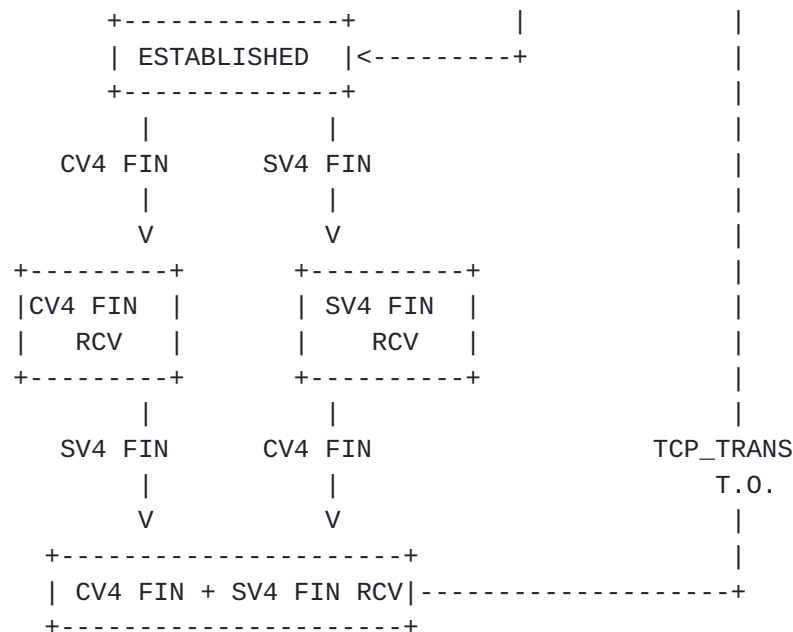
3.1. TCP Transitory Connection Idle-Timeout

[RFC5382]:REQ-5 The transitory connection idle-timeout is defined as
the minimum time a TCP connection in the partially open or closing
phases must remain idle before the NAT considers the associated
session a candidate for removal. But the document does not clearly
states if these can be configured separately. This document
clarifies that a NAT device SHOULD provide different knobs for
configuring the open and closing idle timeouts. This document
further acknowledges that most TCP flows are very short (less than 10
seconds) [FLOWRATE][TCPWILD] and therefore a partially open timeout
of 4 minutes might be excessive if security is a concern. Therefore
it MAY be configured to be less than 4 minutes in such cases. There
also may be cases that a timeout of 4 minutes might be excessive.
The case and the solution are written below.

3.2. TIME_WAIT State

The TCP TIME_WAIT state is described in [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.

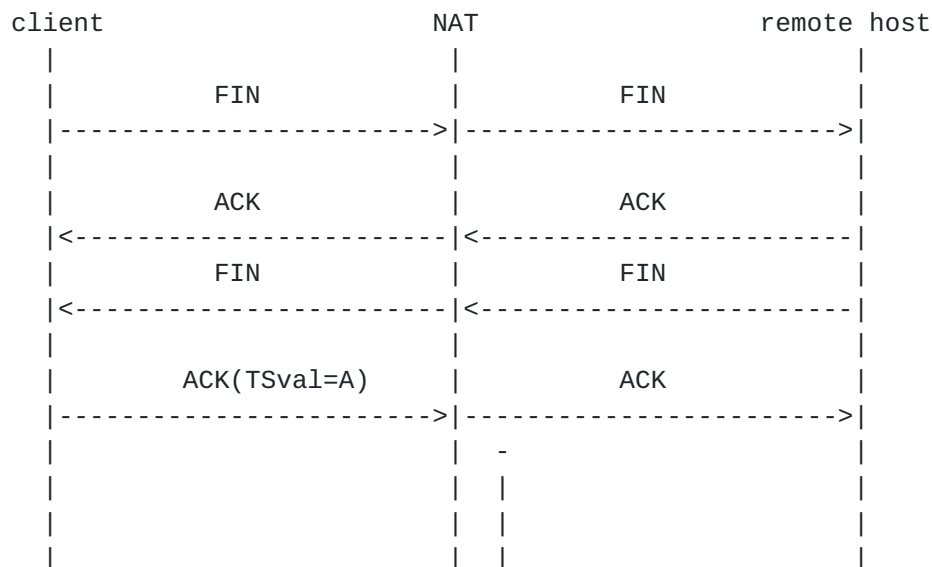These points are important for the TCP to work without problems.

[RFC5283] leaves the handling of TCP connections in TIME_WAIT state unspecified and mentions that TIME_WAIT state is not part of the transitory connection idle-timeout. If the NAT device honors the TIME_WAIT state, each TCP connection and its associated resources is kept for a certain period, typically for four minutes, which consumes port resources.
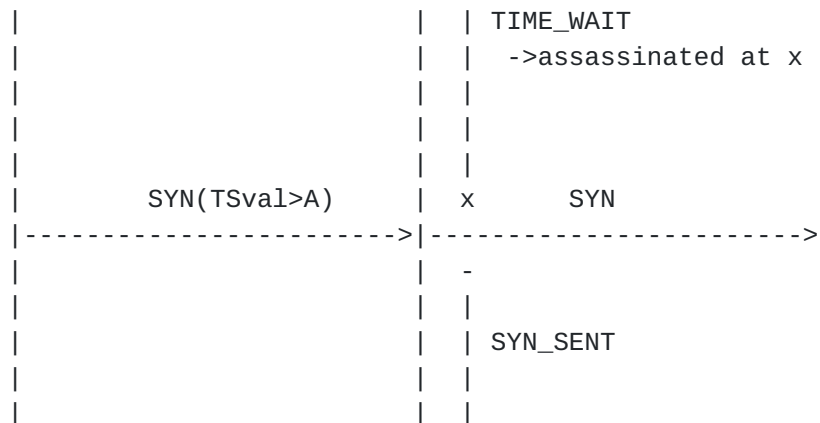
[RFC6191] explains that in certain situation it is necessary to reduce the TIME_WAIT state and defines such a mechanism using TCP timestamps and sequence numbers. When a connection request is received with a four-tuple that is in the TIME-WAIT state, the connection request may be accepted if the sequence number or the timestamp of the incoming SYN segment is greater than the last sequence number seen on the previous incarnation of the connection.

[N.E] This document specifies that a NAT device should keep TCP connections in TIME_WAIT state unless it implements the proposal below?

3.2.1. Proposal: Apply RFC6191 and PAWS to NAT

This section proposes to apply [RFC6191] mechanism at NAT. This mechanism MAY be adopted for both clients' and remote hosts' TCP active close.

```
          client                    NAT                remote host
            |                        |                       |
            |          FIN           |          FIN          |
            |----------------------->|---------------------->|
            |                        |                       |
            |          ACK           |          ACK          |
            |<-----------------------|<----------------------|
            |          FIN           |          FIN          |
            |<-----------------------|<----------------------|
            |                        |                       |
            |       ACK(TSval=A)      |          ACK          |
            |----------------------->|---------------------->|
            |                        | -                     |
            |                        | |                     |
            |                        | |                     |
            |                        | |                     |
```

```
            |                        |  | TIME_WAIT            |
            |                        |  |   ->assassinated at x |
            |                        |  |                       |
            |                        |  |                       |
            |                        |  |                       |
            |      SYN(TSval>A)      |  x      SYN              |
            |----------------------->|----------------------->|
            |                        |  -                      |
            |                        |  |                       |
            |                        |  |  SYN_SENT             |
            |                        |  |                       |
            |                        |  |                       |
```

(postamble)

Also, 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.

To make these mechanisms work, we should concern the case that there
are several clients with nonsuccessive timestamp or sequence number
values are connected to a NAT device (i.e. not monotonically
increasing among clients). Two mechanisms to solve this mechanism
and applying [RFC6191] and PAWS to NAT are described below. These
mechanisms are optional.

3.2.1.1. Rewrite timestamp and sequence number values at NAT

Rewrite timestamp and sequence number values of outgoings packets at
NAT to be monotonically increasing. This can be done by adopting
following mechanisms at NAT.

A: Store the newest rewritten value of timestamp and sequence number
   as the "max value at the time".

B: NAT rewrite timestamp and sequence number values of incoming
   packets to be monotonically increasing.

When packets come back as replies from remote hosts, NAT rewrite
again the timestamp and sequence number values to be the original
values. This can be done by adopting following mechanisms at NAT.

C: Store the values of original timestamp and sequence number of
   packets, and rewritten values of those.

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

Adopt following mechanisms at NAT.

A: Choose clients that can be assigned ports.

B: Split assignable port numbers between clients.

Packets from other clients which are not chosen by these mechanisms are rejected at NAT, unless there is unassigned port left.

3.2.1.3. Resend the last ACK to the retransmisstted FIN

We need to solve another scenario to make RFC6191 work with NAT. In the case the remote TCP could not receive the acknowledgment of its connection termination request, the NAT device, on behalf of clients, resends the last ACK packet when it receives a FIN packet of the previous connection, and when the state of the previous connection has been deleted from the NAT. This mechanism MAY be used when clients starts closing process, and the remote host could not receive the last ACK.

3.2.1.4. 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 [RFC6191] or the mechanism written in 4.2.2.13 of [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.

3.3. TCP RST

[RFC5382] leaves the handling of TCP RST packets unspecified. This document does not try standardize such behavior but clarifies based on operational experience that a NAT that receives a TCP RST for an active mapping and performs session tracking MAY immediately delete the sessions and remove any state associated with it. If the NAT device that performs TCP session tracking receives a TCP RST for the first session that created a mapping, it MAY remove the session and the mapping immediately.

4. Port Overlapping behavior

   [RFC4787] [RFC5382]: REQ-1 Current RFCs specifiy a specific port
   overlapping behavior, i.e., that the external IP:port can be reused
   for connections originating from the same internal source IP:port
   irrespective of the destination. This is known as endpoint-
   independent mapping. This document clarifies that this port
   overlapping behavior can be extended to connections originating from
   different internal source IP:ports as long as their destinations are
   different. This known as EDM (Endpoint Dependent Mapping). The
   mechanism below MAY be one optional implement to NAT.

   If destination addresses and ports are different for outgoing
   connections started by local clients, NAT MAY assign the same
   external port as the source ports for the connections. The port
   overlapping mechanism manages mappings between external packets and
   internal packets by looking at and storing their 5-tuple (protocol,
   source address, source port, destination address, destination port) .
   This enables concurrent use of a single NAT external port for
   multiple transport sessions, which enables NAT to work correctly in
   IP address resource limited network.

   Discussions:

   [RFC4787] and [RFC5382] requires "endpoint-independent mapping" at
   NAT, and port overlapping NAT cannot meet the requirement. This
   mechanism can degrade the transparency of NAT in that its mapping
   mechanism is endpoint-dependent and makes NAT traversal harder.
   However, if a NAT adopts endpoint-independent mapping together with
   endpoint-dependent filtering, then the actual behavior of the NAT
   will be the same as port overlapping NAT.

5. Address Pooling Paired (APP)

   [RFC4787]: REQ-2 [RFC5382]:ND Address Pooling Paired behavior for NAT
   is recommended in previous documents but behavior when a public IPv4
   run out of ports is left undefined. This document clarifies that if
   APP is enabled new sessions from a subscriber that already has a
   mapping associated with a public IP that ran out of ports SHOULD be
   dropped. The administrator MAY provide a knob that allows a NAT
   device to starting using ports from another public IP when the one
   that anchored the APP mapping ran out of ports. This is trade-off
   between subscriber service continuity and APP strict enforcement.
   (NE: It is sometimes referred as 'soft-APP')

6. EIF Security

[RFC4787]:REQ-8 and [RFC5382]:REQ-3 End-point independent filtering could potentially result in security attacks from the public realm. In order to handle this, when possible there MUST be strict filtering checks in the inbound direction. A knob SHOULD be provided to limit the number of inbound sessions and a knob SHOULD be provided to enable or disable EIF on a per application basis. This is specially important in the case of Mobile networks where such attacks can consume radio resources and count against the user quota.

7. EIF Protocol Independence

[RFC4787]:REQ-8 and[RFC5382]: REQ-3 Current RFCs do not specify whether EIF mappings are protocol independent. In other words, if an outbound TCP SYN creates a mapping, it is left undefined whether inbound UDP packets destined to that mapping should be forwarded. This document specifies that EIF mappings SHOULD be protocol independent in order allow inbound packets for protocols that multiplex TCP and UDP over the same IP: port through the NAT and also maintain compatibility with stateful NAT64 RFC6146 [RFC6146]. But the administrator MAY provide a configuration knob to make it protocol dependent.

8. EIF Mapping Refresh

[RFC4787]: REQ-6 [RFC5382]: ND The NAT mapping Refresh direction MAY have a "NAT Inbound refresh behavior" of "True" but it does not clarifies how this applies to EIF mappings. The issue in question is whether inbound packets that match an EIF mapping but do not create a new session due to a security policy should refresh the mapping timer. This document clarifies that even when a NAT device has a inbound refresh behavior of TRUE, such packets SHOULD NOT refresh the mapping. Otherwise a simple attack of a packet every 2 minutes can keep the mapping indefinitely.

8.1. Outbound Mapping Refresh and Error Packets

In the case of NAT outbound refresh behavior there are certain types of packets that should not refresh the mapping even if their direction is outbound. For example, if the mapping is kept alive by ICMP Errors or TCP RST outbound packets sent as response to inbound packets, these SHOULD NOT refresh the mapping.

9. EIM Protocol Independence

[RFC4787] [RFC5382]: REQ-1 Current RFCs do not specify whether EIM are protocol independent. In other words, if a outbound TCP SYN creates a mapping it is left undefined whether outbound UDP can reuse such mapping and create session. On the other hand, Stateful NAT64

[RFC6146] clearly specifies three binding information bases (TCP, UDP, ICMP). This document clarifies that EIM mappings SHOULD be protocol dependent . A knob MAY be provided in order allow protocols that multiplex TCP and UDP over the same source IP and port to use a single mapping.

10. Port Parity

   A NAT devices MAY disable port parity preservation for dynamic mappings. Nevertheless, A NAT SHOULD support means to explicitly request to preserve port parity (e.g., [I-D.pcp-port-set]).

11. Port Randomization

   A NAT SHOULD follow the recommendations specified in Section 4 of [RFC6056] especially: "A NAPT that does not implement port preservation [RFC4787] [RFC5382] SHOULD obfuscate selection of the ephemeral port of a packet when it is changed during translation of that packet. A NAPT that does implement port preservation SHOULD obfuscate the ephemeral port of a packet only if the port must be changed as a result of the port being already in use for some other session. A NAPT that performs parity preservation and that must change the ephemeral port during translation of a packet SHOULD obfuscate the ephemeral ports. The algorithms described in this document could be easily adapted such that the parity is preserved (i.e., force the lowest order bit of the resulting port number to 0 or 1 according to whether even or odd parity is desired)."

12. IP Identification (IP ID)

   A NAT SHOULD handle the Identification field of translated IPv4 packets as specified in Section 9 of [I-D.ietf-intarea-ipv4-id-update].

13. ICMP Query Mappings Timeout

   Section 3.1 of [RFC5508] says that ICMP Query Mappings are to be maintained by NAT device. However, RFC doesn't discuss about the Query Mapping timeout values. Section 3.2 of that RFC only discusses about ICMP Query Session Timeouts. ICMP Query Mappings MAY be deleted once the last the session using the mapping is deleted.

14. Hairpinning Support for ICMP Packets

   [RFC5508]:REQ-7 This requirement specifies that NAT devices enforcing Basic NAT MUST support traversal of hairpinned ICMP Query sessions. This implicitly means that address mappings from external address to internal address (similar to Endpoint Independent Filters) MUST be

   maintained to allow inbound ICMP Query sessions. If an ICMP Query is
   received on an external address, NAT device can then translate to an
   internal IP. [RFC5508]:REQ-7 This requirement specifies that all NAT
   devices (i.e., Basic NAT as well as NAPT devices) MUST support the
   traversal of hairpinned ICMP Error messages. This too requires NAT
   devices to maintain address mappings from external IP address to
   internal IP address in addition to the ICMP Query Mappings described
   in section 3.1 of that RFC.

15. IANA Considerations

   TBD

16. Security Considerations

   In the case of EIF mappings due to high risk of resource crunch, a
   NAT device MAY provide a knob to limit the number of inbound sessions
   spawned from a EIF mapping.

   [TCP-Security] contains a detailed discussion of the security
   implications of TCP Timestamps and of different timestamp generation
   algorithms.

17. Acknowledgements

   Thanks to Dan Wing, Suresh Kumar, Mayuresh Bakshi, Rajesh Mohan and
   Senthil Sivamular for review and discussions

18. References

18.1. 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.

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

   [RFC2663] Srisuresh, P. and M. Holdrege, "IP Network Address
              Translator (NAT) Terminology and Considerations", RFC
              2663, August 1999.

[RFC3605] Huitema, C., "Real Time Control Protocol (RTCP) attribute
           in Session Description Protocol (SDP)", RFC 3605, October
           2003.

[RFC4787] Audet, F. and C. Jennings, "Network Address Translation
           (NAT) Behavioral Requirements for Unicast UDP", BCP 127,
           RFC 4787, January 2007.

[RFC5382] Guha, S., Biswas, K., Ford, B., Sivakumar, S., and P.
           Srisuresh, "NAT Behavioral Requirements for TCP", BCP 142,
           RFC 5382, October 2008.

[RFC5508] Srisuresh, P., Ford, B., Sivakumar, S., and S. Guha, "NAT
           Behavioral Requirements for ICMP", BCP 148, RFC 5508,
           April 2009.

[RFC6056] Larsen, M. and F. Gont, "Recommendations for Transport-
           Protocol Port Randomization", BCP 156, RFC 6056, January
           2011.

[RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful
           NAT64: Network Address and Protocol Translation from IPv6
           Clients to IPv4 Servers", RFC 6146, April 2011.

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

[RFC6888] Perreault, S., Yamagata, I., Miyakawa, S., Nakagawa, A.,
           and H. Ashida, "Common Requirements for Carrier-Grade NATs
           (CGNs)", BCP 127, RFC 6888, April 2013.

18.2. Informative References

[FLOWRATE]
           Zhang, Y., Breslau, L., Paxson, V., and S. Shenker, "On
           the Characteristics and Origins of Internet Flow Rates", .

[I-D.ietf-pcp-port-set]
           Qiong, Q., Boucadair, M., Sivakumar, S., Zhou, C., Tsou,
           T., and S. Perreault, "Port Control Protocol (PCP)
           Extension for Port Set Allocation", draft-ietf-pcp-port-
           set-03 (work in progress), October 2013.

[I-D.naito-nat-resource-optimizing-extension]
           Kengo, K. and A. Matsumoto, "NAT TIME_WAIT reduction",
           draft-naito-nat-resource-optimizing-extension-02 (work in
           progress), July 2012.

[TCPWILD] Qian, F., Subhabrata, S., Spatscheck, O., Morley Mao, Z.,
           and W. Willinger, "TCP Revisited: A Fresh Look at TCP in
           the Wild", .

Authors' Addresses

   Reinaldo Penno
   Cisco Systems, Inc.
   170 West Tasman Drive
   San Jose, California 95134
   USA

   Email: repenno@cisco.com

   Simon Perreault
   Viagenie
   2875 boul. Laurier, suite D2-630
   Quebec, QC G1V 2M2
   Canada

   Email: simon.perreault@viagenie.ca

   Sarat Kamiset
   Insieme Networks
   California

   Mohamed Boucadair
   France Telecom
   Rennes 35000
   France

   Email: mohamed.boucadair@orange.com

   Kengo Naito
   NTT
   Tokyo
   Japan

   Email: kengo@lab.ntt.co.jp