**Updated Specification of the IPv4 ID Field**
**draft-ietf-intarea-ipv4-id-update-05.txt**


Status of this Memo

Copyright Notice

Abstract

   The IPv4 Identification (ID) field enables fragmentation and
   reassembly, and as currently specified is required to be unique
   within the maximum lifetime for all datagrams with a given
   source/destination/protocol tuple. If enforced, this uniqueness
   requirement would limit all connections to 6.4 Mbps. Because
   individual connections commonly exceed this speed, it is clear that
   existing systems violate the current specification. This document
   updates the specification of the IPv4 ID field in RFC791, RFC1122,
   and RFC2003 to more closely reflect current practice and to more
   closely match IPv6 so that the field's value is defined only when a
   datagram is actually fragmented. It also discusses the impact of
   these changes on how datagrams are used.

Table of Contents

## 1. Introduction

In IPv4, the Identification (ID) field is a 16-bit value that is unique for every datagram for a given source address, destination address, and protocol, such that it does not repeat within the Maximum Segment Lifetime (MSL) [RFC791][RFC1122]. As currently specified, all datagrams between a source and destination of a given protocol must have unique IPv4 ID values over a period of this MSL, which is typically interpreted as two minutes (120 seconds). This uniqueness is currently specified as for all datagrams, regardless of fragmentation settings.

Uniqueness of the IPv4 ID is commonly violated by high speed devices; if strictly enforced, it would limit the speed of a single protocol between two IP endpoints to 6.4 Mbps for typical MTUs of 1500 bytes [RFC4963]. It is common for a single connection to operate far in excess of these rates, which strongly indicates that the uniqueness of the IPv4 ID as specified is already moot.

This document updates the specification of the IPv4 ID field to more closely reflect current practice, and to include considerations taken into account during the specification of the similar field in IPv6.

## 2. Conventions used in this document

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].

In this document, the characters ">>" proceeding an indented line(s) indicates a requirement using the key words listed above. This convention aids reviewers in quickly identifying or finding this document's explicit requirements.

3. **The IPv4 ID Field**

   IP supports datagram fragmentation, where large datagrams are split
   into smaller components to traverse links with limited maximum
   transmission units (MTUs). Fragments are indicated in different ways
   in IPv4 and IPv6:

   o  In IPv4, fragments are indicated using four fields of the basic
      header: Identification (ID), Fragment Offset, a "Don't Fragment"
      flag (DF), and a "More Fragments" flag (MF) [RFC791]

   o  In IPv6, fragments are indicated in an extension header that
      includes an ID, Fragment Offset, and M (more fragments) flag
      similar to their counterparts in IPv4 [RFC2460]

   IPv4 and IPv6 fragmentation differs in a few important ways. IPv6
   fragmentation occurs only at the source, so a DF bit is not needed to
   prevent downstream devices from initiating fragmentation (i.e., IPv6
   always acts as if DF=1). The IPv6 fragment header is present only
   when a datagram has been fragmented, or when the source has received
   a "packet too big" ICMPv6 error message when the path cannot support
   the required minimum 1280-byte IPv6 MTU and is thus subject to
   translation [RFC2460][RFC4443]. The latter case is relevant only for
   IPv6 datagrams sent to IPv4 destinations to support subsequent
   fragmentation after translation to IPv4.

   With the exception of these two cases, the ID field is not present
   for non-fragmented datagrams, and thus is meaningful only for
   datagrams that are already fragmented or datagrams intended to be
   fragmented as part of IPv4 translation. Finally, the IPv6 ID field is
   32 bits, and required unique per source/destination address pair for
   IPv6, whereas for IPv4 it is only 16 bits and required unique per
   source/destination/protocol triple.

   This document focuses on the IPv4 ID field issues, because in IPv6
   the field is larger and present only in fragments.

4. **Uses of the IPv4 ID Field**

   The IPv4 ID field was originally intended for fragmentation and
   reassembly [RFC791]. Within a given source address, destination
   address, and protocol, fragments of an original datagram are matched
   based on their IPv4 ID. This requires that IDs are unique within the
   address/protocol triple when fragmentation is possible (e.g., DF=0)
   or when it has already occurred (e.g., frag_offset>0 or MF=1).

The IPv4 ID field can be useful for other purposes. The field has
been proposed as a way to detect and remove duplicate datagrams,
e.g., at congested routers (noted in Sec. 3.2.1.5 of [RFC1122]) or in
network accelerators. It can similarly be used at end hosts to reduce
the impact of duplication on higher-layer protocols (e.g., additional
processing in TCP, or the need for application-layer duplicate
suppression in UDP).

The IPv4 ID field is also used in some debugging tools to correlate
datagrams measured at various locations along a network path. This is
already insufficient in IPv6 because unfragmented datagrams lack an
ID, so these tools are already being updated to avoid such reliance
on the ID field.

The ID clearly needs to be unique (within MSL, within the
src/dst/protocol tuple) to support fragmentation and reassembly, but
not all datagrams are fragmented or allow fragmentation. This
document deprecates non-fragmentation uses, allowing the ID to be
repeated (within MSL, within the src/dst/protocol tuple) in those
cases.

## 5. Background on IPv4 ID Reassembly Issues

The following is a summary of issues with IPv4 fragment reassembly in
high speed environments raised previously [RFC4963]. Readers are
encouraged to consult RFC 4963 for a more detailed discussion of
these issues.

With the maximum IPv4 datagram size of 64KB, a 16-bit ID field that
does not repeat within 120 seconds means that the aggregate of all
TCP connections of a given protocol between two IP endpoints is
limited to roughly 286 Mbps; at a more typical MTU of 1500 bytes,
this speed drops to 6.4 Mbps [RFC791][RFC1122][RFC4963]. This limit
currently applies for all IPv4 datagrams within a single protocol
(i.e., the IPv4 protocol field) between two IP addresses, regardless
of whether fragmentation is enabled or inhibited, and whether a
datagram is fragmented or not.

IPv6, even at typical MTUs, is capable of 18.7 Tbps with
fragmentation between two IP endpoints as an aggregate across all
protocols, due to the larger 32-bit ID field (and the fact that the
IPv6 next-header field, the equivalent of the IPv4 protocol field, is
not considered in differentiating fragments). When fragmentation is
not used the field is absent, and in that case IPv6 speeds are not
limited by the ID field uniqueness.

Note also that 120 seconds is only an estimate on the maximum
datagram lifetime. It is loosely based on half maximum value of the
IP TTL field (255), measured in seconds, because the TTL was
originally specified as decremented not only for each hop, but also
for each second a datagram is held at a router (as implied in
[RFC791], although this has long since become a hopcount only).
Network delays are incurred in other ways, e.g., satellite links,
which can add seconds of delay even though the TTL is not decremented
by a corresponding amount. There is thus no enforcement mechanism to
ensure that datagrams older than 120 seconds are discarded.

Wireless Internet devices are frequently connected at speeds over 54
Mbps, and wired links of 1 Gbps have been the default for several
years. Although many end-to-end transport paths are congestion
limited, these devices easily achieve 100+ Mbps application-layer
throughput over LANs (e.g., disk-to-disk file transfer rates), and
numerous throughput demonstrations with COTS systems over wide-area
paths exhibit these speeds for over a decade. This strongly suggests
that IPv4 ID uniqueness has been moot for a long time.

## 6. Updates to the IPv4 ID Specification

This document updates the specification of the IPv4 ID field in three
distinct ways, as discussed in subsequent subsections:

o  Use the IPv4 ID field only for fragmentation

o  Avoiding a performance impact when the IPv4 ID field is used

o  Encourage safe operation when the IPv4 ID field is used

There are two kinds of datagrams used in the following discussion,
named as follows:

o  Atomic datagrams are datagrams not yet fragmented and for which
   further fragmentation has been inhibited.

o  Non-atomic datagrams are datagrams which either have already been
   fragmented or for which fragmentation remains possible.

This same definition can be expressed in pseudo code as using common
logical operators (equals is ==, logical 'and' is &&, logical 'or' is
||, greater than is >, and parenthesis function typically) as:

o  Atomic datagrams: (DF==1)&&(MF==0)&&(frag_offset==0)

o  Non-atomic datagrams: (DF==0)||(MF==1)||(frag_offset>0)

The test for non-atomic datagrams is the logical negative of the test for atomic datagrams, thus all possibilities are considered.

## 6.1. IPv4 ID Used Only for Fragmentation

Although RFC1122 suggests the IPv4 ID field has other uses, including datagram de-duplication, this document asserts that this field's value is defined only for fragmentation and reassembly:

>> IPv4 ID field MUST NOT be used for purposes other than fragmentation and reassembly.

Datagram de-duplication can be accomplished using hash-based duplicate detection for cases where the ID field is absent.

In atomic datagrams, the IPv4 ID field has no meaning, and thus can be set to an arbitrary value, i.e., the requirement for non-repeating IDs within the address/protocol triple is no longer required for atomic datagrams:

>> Originating sources MAY set the IPv4 ID field of atomic datagrams to any value.

Second, all network nodes, whether at intermediate routers, destination hosts, or other devices (e.g., NATs and other address sharing mechanisms, firewalls, tunnel egresses), cannot rely on the field:

>> All devices that examine IPv4 headers MUST ignore the IPv4 ID field of atomic datagrams.

The IPv4 ID field is thus meaningful only for non-atomic datagrams - datagrams that have either already been fragmented, or those for which fragmentation remains permitted. Atomic datagrams are detected by their DF, MF, and fragmentation offset fields as explained in Section 6, because such a test is completely backward compatible; this document thus does not reserve any IPv4 ID values, including 0, as distinguished.

Deprecating the use of the IPv4 ID field for non-reassembly uses should have little - if any - impact. IPv4 IDs are already frequently repeated, e.g., over even moderately fast connections. Duplicate suppression was only suggested [RFC1122], and no impacts of IPv4 ID reuse have been noted. Routers are not required to issue ICMPs on any particular timescale, and so IPv4 ID repetition should not have been used for validation, and again repetition occurs and probably could have been noticed [RFC1812]. ICMP relaying at tunnel ingresses is

   specified to use soft state rather than a datagram cache, and should
   have been noted if the latter for similar reasons [RFC2003].

## 6.2. Encourage Safe IPv4 ID Use

   This document makes further changes to the specification of the IPv4
   ID field and its use to encourage its safe use as corollary
   requirements changes as follows.

   RFC 1122 discusses that if TCP retransmits a segment it may be
   possible to reuse the IPv4 ID (see Section 8.2). This can make it
   difficult for a source to avoid IPv4 ID repetition for received
   fragments. RFC 1122 concludes that this behavior "is not useful";
   this document formalizes that conclusion as follows:

   >> The IPv4 ID of non-atomic datagrams MUST NOT be reused when
   sending a copy of an earlier non-atomic datagram.

   RFC 1122 also suggests that fragments can overlap [RFC1122]. Such
   overlap can occur if successive retransmissions are fragmented in
   different ways but with the same reassembly IPv4 ID. This overlap is
   noted as the result of reusing IPv4 IDs when retransmitting
   datagrams, which this document deprecates. However, it is also the
   result of in-network datagram duplication, which can still occur. As
   a result this document does not change the need to support
   overlapping fragments.

## 6.3. IPv4 ID Requirements That Persist

   This document does not relax the IPv4 ID field uniqueness
   requirements of [RFC791] for non-atomic datagrams, i.e.:

   >> Sources emitting non-atomic datagrams MUST NOT repeat IPv4 ID
   values within one MSL for a given source address/destination
   address/protocol triple.

   Such sources include originating hosts, tunnel ingresses, and NATs
   (including other address sharing mechanisms) (see Section 9).

   This document does not relax the requirement that all network devices
   honor the DF bit, i.e.:

   >> IPv4 datagrams whose DF=1 MUST NOT be fragmented.

   >> IPv4 datagram transit devices MUST NOT clear the DF bit.

In specific, DF=1 prevents fragmenting atomic datagrams. DF=1 also prevents further fragmenting received fragments. In-network fragmentation is permitted only when DF=0; this document does not change that requirement.

## 7. Impact on Datagram Use

The following is a summary of the recommendations that are the result of the previous changes to the IPv4 ID field specification.

Because atomic datagrams can use arbitrary IPv4 ID values, the ID field no longer imposes a performance impact in those cases. However, the performance impact remains for non-atomic datagrams. As a result:

>> Sources of non-atomic IPv4 datagrams MUST rate-limit their output to comply with the ID uniqueness requirements.

Such sources include, in particular, DNS over UDP [RFC2671].

Because there is no strict definition of the MSL, reassembly hazards exist regardless of the IPv4 ID reuse interval or the reassembly timeout. As a result:

>> Higher layer protocols SHOULD verify the integrity of IPv4 datagrams, e.g., using a checksum or hash that can detect reassembly errors (the UDP checksum is weak in this regard, but better than nothing), as in SEAL [RFC5320].

Additional integrity checks can be employed using tunnels, as in SEAL, IPsec, or SCTP [RFC4301][RFC4960][RFC5320]. Such checks can avoid the reassembly hazards that can occur when using UDP and TCP checksums [RFC4963], or when using partial checksums as in UDP-Lite [RFC3828]. Because such integrity checks can avoid the impact of reassembly errors:

>> Sources of non-atomic IPv4 datagrams using strong integrity checks MAY reuse the ID within MSL values smaller than is typical.

Note, however, that such frequent reuse can still result in corrupted reassembly and poor throughput, although it would not propagate reassembly errors to higher layer protocols.

## 8. Updates to Existing Standards

The following sections address the specific changes to existing protocols indicated by this document.

8.1. Updates to RFC 791

   RFC 791 states that:

      The originating protocol module of an internet datagram sets the
      identification field to a value that must be unique for that
      source-destination pair and protocol for the time the datagram
      will be active in the internet system.

   And later that:

      Thus, the sender must choose the Identifier to be unique for this
      source, destination pair and protocol for the time the datagram
      (or any fragment of it) could be alive in the internet.

      It seems then that a sending protocol module needs to keep a table
      of Identifiers, one entry for each destination it has communicated
      with in the last maximum datagram lifetime for the internet.

      However, since the Identifier field allows 65,536 different
      values, some host may be able to simply use unique identifiers
      independent of destination.

      It is appropriate for some higher level protocols to choose the
      identifier. For example, TCP protocol modules may retransmit an
      identical TCP segment, and the probability for correct reception
      would be enhanced if the retransmission carried the same
      identifier as the original transmission since fragments of either
      datagram could be used to construct a correct TCP segment.

   This document changes RFC 791 as follows:

   o  IPv4 ID uniqueness applies to only non-atomic datagrams.

   o  Retransmitted non-atomic IPv4 datagrams are no longer permitted to
      reuse the ID value.

8.2. Updates to RFC 1122

   RFC 1122 states that:

        3.2.1.5  Identification: RFC-791 Section 3.2

           When sending an identical copy of an earlier datagram, a
           host MAY optionally retain the same Identification field in
           the copy.

DISCUSSION:

Some Internet protocol experts have maintained that when a
host sends an identical copy of an earlier datagram, the new
copy should contain the same Identification value as the
original.  There are two suggested advantages:  (1) if the
datagrams are fragmented and some of the fragments are lost,
the receiver may be able to reconstruct a complete datagram
from fragments of the original and the copies; (2) a
congested gateway might use the IP Identification field (and
Fragment Offset) to discard duplicate datagrams from the
queue.

This document changes RFC 1122 as follows:

o  The IPv4 ID field is no longer permitted to be used for duplicate
   detection. This applies to both atomic and non-atomic datagrams.

o  Retransmitted non-atomic IPv4 datagrams are no longer permitted to
   reuse the ID value.

## 8.3. Updates to RFC 2003

This document updates how IPv4-in-IPv4 tunnels create IPv4 ID values
for the IPv4 outer header [RFC2003], but only in the same way as for
any other IPv4 datagram source. In specific, RFC 2003 states the
following, where ref. [10] is RFC 791:

     Identification, Flags, Fragment Offset

        These three fields are set as specified in [10]...

This document changes RFC 2003 as follows:

o  The IPv4 ID field is set as permitted by this document.

## 9. Impact on Middleboxes

Middleboxes include rewriting devices that include network address
translators (NATs), address/port translators (NAPTs), and other
address sharing mechanisms (ASMs). They also include devices that
inspect and filter datagrams that are not routers, such as
accelerators and firewalls.

**9.1. Rewriting Middleboxes**

   NATs and NAPTs rewrite IP fields, and tunnel ingresses (using IPv4
   encapsulation) copy and modify some IPv4 fields, so all are
   considered sources, as do any devices that rewrite any portion of the
   source address, destination address, protocol, and ID tuple for any
   datagrams [RFC3022]. This is also true for other ASMs, including 4rd,
   IVI, and others in the "A+P" (address plus port) family [Bo11] [De11]
   [RFC6219]. It is equally true for any other datagram rewriting
   mechanism. As a result, they are subject to all the requirements of
   any source, as has been noted.

   NATs/ASMs/rewriters present a particularly challenging situation for
   fragmentation. Because they overwrite portions of the reassembly
   tuple in both directions, they can destroy tuple uniqueness and
   result in a reassembly hazard. Whenever IPv4 source address,
   destination address, or protocol fields are modified, a
   NAT/ASM/rewriter needs to ensure that the ID field is generated
   appropriately, rather than simply copied from the incoming datagram.
   In specific:

   >> Address sharing or rewriting devices MUST ensure that the IPv4 ID
   field of datagrams whose address or protocol are translated comply
   with these requirements as if the datagram were sourced by that
   device.

   This compliance means that the IPv4 ID field of non-atomic datagrams
   translated at a NAT/ASM/rewriter needs to obey the uniqueness
   requirements of any IPv4 datagram source. Unfortunately, fragments
   already violate that requirement, as they repeat an IPv4 ID within
   the MSL for a given source address, destination address, and protocol
   triple.

   Such problems with transmitting fragments through NATs/ASMs/rewriters
   are already known; translation is based on the transport port number,
   which is present in only the first fragment anyway [RFC3022]. This
   document underscores the point that not only is reassembly (and
   possibly subsequent fragmentation) required for translation, it can
   be used to avoid issues with IPv4 ID uniqueness.

   Note that NATs/ASMs already need to exercise special care when
   emitting datagrams on their public side, because merging datagrams
   from many sources onto a single outgoing source address can result in
   IPv4 ID collisions. This situation precedes this document, and is not
   affected by it. It is exacerbated in large-scale, so-called "carrier
   grade" NATs [Pe11].

Tunnel ingresses act as sources for the outermost header, but tunnels act as routers for the inner headers (i.e., the datagram as arriving at the tunnel ingress). Ingresses can always fragment as originating sources of the outer header, because they control the uniqueness of that IPv4 ID field and the value of DF on the outer header independent of those values on the inner (arriving datagram) header.

## 9.2. Filtering Middleboxes

Middleboxes also include devices that filter datagrams, including network accelerators and firewalls. Some such devices reportedly feature datagram de-duplication, which relies on IP ID uniqueness to identify duplicates. Such accelerators already risk dropping non-duplicate datagrams because of early ID reuse, and, as a result of earlier requirements:

>> Datagram de-duplication mechanisms MUST ignore the ID values on atomic datagrams.

## 10. Impact on Header Compression

Header compression algorithms already accommodate various ways in which the IPv4 ID changes between sequential datagrams [RFC1144] [RFC2508] [RFC3545] [RFC5225]. Such algorithms currently assume that the IPv4 ID is preserved end-to-end. Some algorithms already allow assuming the ID does not change (e.g., ROHC [RFC5225]), where others include non-changing IDs via zero deltas (e.g., ECRTP [RFC3545]).

When compression assumes a changing ID as a default, having a non-changing ID can make compression less efficient. Such non-changing IDs have been described in various RFCs (e.g., footnote 21 of [RFC1144 and cRTP [RFC2508]). When compression can assume a non-changing IPv4 ID - as with ROHC and ECRTP - efficiency can be increased.

## 11. Security Considerations

When the IPv4 ID is ignored on receipt (e.g., for atomic datagrams), its value becomes unconstrained; that field then can more easily be used as a covert channel. For some atomic datagrams - notably those not protected by IPsec Authentication Header (AH) [RFC4302] - it is now possible, and may be desirable, to rewrite the IPv4 ID field to avoid its use as such a channel.

The IPv4 ID also now adds much less entropy of the header of a datagram. The IPv4 ID had previously been unique (for a given source/address pair, and protocol field) within one MSL, although

this requirement was not enforced and clearly is typically ignored.
The IPv4 ID of atomic datagrams is not required unique, and so
contributes no entropy to the header.

The deprecation of the IPv4 ID field's uniqueness for atomic
datagrams can defeat the ability to count devices behind a
NAT/ASM/rewriter [Be02]. This is not intended as a security feature,
however.

## 12. IANA Considerations

There are no IANA considerations in this document.

The RFC Editor should remove this section prior to publication

## 13. References

### 13.1. Normative References

[RFC791]   Postel, J., "Internet Protocol", RFC 791 / STD 5, September
           1981.

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

[RFC1812]  Baker, F. (Ed.), "Requirements for IP Version 4 Routers",
           RFC 1812 / STD 4, Jun. 1995.

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

[RFC2003]  Perkins, C., "IP Encapsulation within IP", RFC 2003,
           October 1996.

### 13.2. Informative References

[Be02]     Bellovin, S., "A Technique for Counting NATted Hosts",
           Internet Measurement Conference, Proceedings of the 2nd ACM
           SIGCOMM Workshop on Internet Measurement, Nov. 2002.

[Bo11]     Boucadair, M., J. Touch, P. Levis, R. Penno, "Analysis of
           Solution Candidates to Reveal a Host Identifier in Shared
           Address Deployments", (work in progress), draft-boucadair-
           intarea-nat-reveal-analysis, Sept. 2011.

[De11]    Despres, R. (Ed.), S. Matsushima, T. Murakami, O. Troan,
          "IPv4 Residual Deployment across IPv6-Service networks
          (4rd)", (work in progress), draft-despres-intarea-4rd, Mar.
          2011.

[Pe11]    Perreault, S., (Ed.), I. Yamagata, S. Miyakawa, A.
          Nakagawa, H. Ashida, "Common requirements of IP address
          sharing schemes", (work in progress), draft-ietf-behave-
          lsn-requirements, Mar. 2011.

[RFC1144] Jacobson, V., "Compressing TCP/IP Headers", RFC 1144, Feb.
          1990.

[RFC2460] Deering, S., R. Hinden, "Internet Protocol, Version 6
          (IPv6) Specification", RFC 2460, Dec. 1998.

[RFC2508] Casner, S., V. Jacobson. "Compressing IP/UDP/RTP Headers
          for Low-Speed Serial Links", RFC 2508, Feb. 1999.

[RFC2671] Vixie,P., "Extension Mechanisms for DNS (EDNS0)", RFC 2671,
          Aug. 1999.

[RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network
          Address Translator (Traditional NAT)", RFC 3022, Jan. 2001.

[RFC3545] Koren, T., S. Casner, J. Geevarghese, B. Thompson, P.
          Ruddy, "Enhanced Compressed RTP (CRTP) for Links with High
          Delay, Packet Loss and Reordering", RFC 3545, Jul. 2003.

[RFC3828] Larzon, L-A., M. Degermark, S. Pink, L-E. Jonsson, Ed., G.
          Fairhurst, Ed., "The Lightweight User Datagram Protocol
          (UDP-Lite)", RFC 3828, Jul. 2004.

[RFC4301] Kent, S., K. Seo, "Security Architecture for the Internet
          Protocol", RFC 4301, Dec. 2005.

[RFC4302] Kent, S., "IP Authentication Header", RFC 4302, Dec. 2005.

[RFC4443] Conta, A., S. Deering, M. Gupta (Ed.), "Internet Control
          Message Protocol (ICMPv6) for the Internet Protocol Version
          6 (IPv6) Specification", RFC 4443, March. 2006.

[RFC4960] Stewart, R. (Ed.), "Stream Control Transmission Protocol",
          RFC 4960, Sep. 2007.

[RFC4963] Heffner, J., M. Mathis, B. Chandler, "IPv4 Reassembly
          Errors at High Data Rates," RFC 4963, Jul. 2007.

   [RFC5225] Pelletier, G., K. Sandlund, "RObust Header Compression
             Version 2 (ROHCv2): Profiles for RTP, UDP, IP, ESP and UDP-
             Lite", RFC 5225, Apr. 2008.

   [RFC5320] Templin, F., Ed., "The Subnetwork Encapsulation and
             Adaptation Layer (SEAL)", RFC 5320, Feb. 2010.

   [RFC6219] Li, X., C. Bao, M. Chen, H. Zhang, J. Wu, "The China
             Education and Research Network (CERNET) IVI Translation
             Design and Deployment for the IPv4/IPv6 Coexistence and
             Transition", RFC 6219, May 2011.

## 14. Acknowledgments

Author's Address

   Joe Touch
   USC/ISI
   4676 Admiralty Way
   Marina del Rey, CA 90292-6695
   U.S.A.

   Phone: +1 (310) 448-9151
   Email: touch@isi.edu