Updated Specification of the IPv4 ID Field

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
   address/destination address/protocol tuple. If enforced, this
   uniqueness requirement would limit all connections to 6.4 Mbps for
   typical datagram sizes. 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 RFCs 791, 1122, and 2003 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.

Status of This Memo

   This is an Internet Standards Track document.

   This document is a product of the Internet Engineering Task Force
   (IETF). It represents the consensus of the IETF community. It has
   received public review and has been approved for publication by the
   Internet Engineering Steering Group (IESG). Further information on
   Internet Standards is available in Section 2 of RFC 5741.

   Information about the current status of this document, any errata,
   and how to provide feedback on it may be obtained at
   http://www.rfc-editor.org/info/rfc6864.

Copyright Notice

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 datagram lifetime (MDL) [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 MDL,
   which is typically interpreted as two minutes and is related to the
   recommended reassembly timeout [RFC1122]. 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
   (assuming a 2-minute MDL, using the analysis presented in [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. Further, some sources have been
   generating non-varying IPv4 IDs for many years (e.g., cellphones),
   which resulted in support for such in RObust Header Compression
   (ROHC) [RFC5225].

   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 ">>" preceding one or more indented
   lines indicate 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"
      (DF) flag, and a "More Fragments" (MF) flag [RFC791].

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

   IPv6 fragmentation differs from IPv4 fragmentation 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
   indicating that 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; thus, it 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 address/destination address/protocol tuple.

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

3.1. 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 be unique within the
   source address/destination address/protocol tuple when fragmentation
   is possible (e.g., DF=0) or when it has already occurred (e.g.,
   frag_offset>0 or MF=1).

Other uses have been envisioned for the IPv4 ID field. The field has
been proposed as a way to detect and remove duplicate datagrams,
e.g., at congested routers (noted in Section 3.2.1.5 of [RFC1122]) or
in network accelerators. It has similarly been proposed for use 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). This is discussed
further in Section 5.1.

The IPv4 ID field is used in some diagnostic 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. This is also discussed further in
Section 5.1.

The ID clearly needs to be unique (within the MDL, within the source
address/destination address/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 the MDL, within the source
address/destination address/protocol tuple) in those cases.

3.2. 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 64 KB, 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 or not a datagram is fragmented.

   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 MDL. It is
related to the reassembly timeout as a lower bound and the TCP
Maximum Segment Lifetime as an upper bound (both as noted in
[RFC1122]). Network delays are incurred in other ways, e.g.,
satellite links, which can add seconds of delay even though the Time
to Live (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 Commercial-Off-The-Shelf
(COTS) systems over wide-area paths have exhibited these speeds for
over a decade. This strongly suggests that IPv4 ID uniqueness has
been moot for a long time.

4. 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 Using the IPv4 ID field only for fragmentation

o Encouraging safe operation when the IPv4 ID field is used

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

There are two kinds of datagrams, which are defined below and used in
the following discussion:

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

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

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

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.

4.1. IPv4 ID Used Only for Fragmentation

Although RFC 1122 suggests that the IPv4 ID field has other uses,
including datagram de-duplication, such uses are already not
interoperable with known implementations of sources that do not vary
their ID. This document thus defines this field's value only for
fragmentation and reassembly:

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

Datagram de-duplication can still be accomplished using hash-based
duplicate detection for cases where the ID field is absent (IPv6
unfragmented datagrams), which can also be applied to IPv4 atomic
datagrams without utilizing the ID field [RFC6621].

In atomic datagrams, the IPv4 ID field has no meaning; thus, it can
be set to an arbitrary value, i.e., the requirement for non-repeating
IDs within the source address/destination address/protocol tuple 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 of atomic datagrams:

>> 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 --
either those datagrams that have 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 4, because such a test is completely backward compatible;
thus, this document 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 and
from some sources that do not vary the ID at all, and no adverse
impact has been observed. Duplicate suppression was suggested

[RFC1122] and has been implemented in some protocol accelerators, but
no impacts of IPv4 ID reuse have been noted to date. Routers are not
required to issue ICMPs on any particular timescale, and so IPv4 ID
repetition should not have been used for validation purposes; this
scenario has not been observed. Besides, repetition already occurs
and would have been noticed [RFC1812]. ICMP relaying at tunnel
ingresses is specified to use soft state rather than a datagram
cache; for similar reasons, if the latter is used, this should have
been noticed [RFC2003]. These and other legacy issues are discussed
further in Section 5.1.

4.2. Encouraging Safe IPv4 ID Use

This document also changes the specification of the IPv4 ID field to
encourage its safe use.

As discussed in RFC 1122, if TCP retransmits a segment, it may be
possible to reuse the IPv4 ID (see Section 6.2). 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. 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 for receivers to support
overlapping fragments.

4.3. IPv4 ID Requirements That Persist

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

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

Such sources include originating hosts, tunnel ingresses, and NATs
(including other address-sharing mechanisms) (see Section 5.3).

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

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

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

Specifically, 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.

5. Impact of Proposed Changes

   This section discusses the impact of the proposed changes on legacy devices, datagram generation in updated devices, middleboxes, and header compression.

5.1. Impact on Legacy Internet Devices

   Legacy uses of the IPv4 ID field consist of fragment generation, fragment reassembly, duplicate datagram detection, and "other" uses.

   Current devices already generate ID values that are reused within the source address/destination address/protocol tuple in less than the current estimated Internet MDL of two minutes. They assume that the MDL over their end-to-end path is much lower.

   Existing devices have been known to generate non-varying IDs for atomic datagrams for nearly a decade, notably some cellphones. Such constant ID values are the reason for their support as an optimization of ROHC [RFC5225]. This is discussed further in Section 5.4. Generation of IPv4 datagrams with constant (zero) IDs is also described as part of the IP/ICMP translation standard [RFC6145].

   Many current devices support fragmentation that ignores the IPv4 Don't Fragment (DF) bit. Such devices already transit traffic from sources that reuse the ID. If fragments of different datagrams reusing the same ID (within the source address/destination address/protocol tuple) arrive at the destination interleaved, fragmentation would fail and traffic would be dropped. Either such interleaving is uncommon or traffic from such devices is not widely traversing these DF-ignoring devices, because significant occurrence of reassembly errors has not been reported. DF-ignoring devices do not comply with existing standards, and it is not feasible to update the standards to allow them as compliant.

The ID field has been envisioned for use in duplicate detection, as
discussed in Section 4.1. Although this document now allows IPv4 ID
reuse for atomic datagrams, such reuse is already common (as noted
above). Protocol accelerators are known to implement IPv4 duplicate
detection, but such devices are also known to violate other Internet
standards to achieve higher end-to-end performance. These devices
would already exhibit erroneous drops for this current traffic, and
this has not been reported.

There are other potential uses of the ID field, such as for
diagnostic purposes. Such uses already need to accommodate atomic
datagrams with reused ID fields. There are no reports of such uses
having problems with current datagrams that reuse IDs.

Thus, as a result of previous requirements, this document recommends
that IPv4 duplicate detection and diagnostic mechanisms apply
IPv6-compatible methods, i.e., methods that do not rely on the ID
field (e.g., as suggested in [RFC6621]). This is a consequence of
using the ID field only for reassembly, as well as the known hazard
of existing devices already reusing the ID field.

5.2. Impact on Datagram Generation

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 MDL, 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 and TCP checksums are weak in this
   regard, but better than nothing).

Additional integrity checks can be employed using tunnels, as
supported by the Subnetwork Encapsulation and Adaptation Layer (SEAL)
[RFC5320], IPsec [RFC4301], or the Stream Control Transmission
Protocol (SCTP) [RFC4960]. 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 intervals that are smaller than typical
   MDL values.

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.

5.3. Impact on Middleboxes

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

The changes proposed in this document may not be implemented by
middleboxes; however, these changes are more likely to make current
middlebox behavior compliant than to affect the service provided by
those devices.

5.3.1. Rewriting Middleboxes

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

   Specifically:

   >> Address-sharing or rewriting devices MUST ensure that the IPv4 ID
      field of datagrams whose addresses or protocols 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, translated
   fragments already violate that requirement, as they repeat an IPv4 ID
   within the MDL for a given source address/destination
   address/protocol tuple.

   Such problems with transmitting fragments through NATs/ASMs/rewriters
   are already known; translation is typically 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.

5.3.2. Filtering Middleboxes

   Middleboxes also include devices that filter datagrams, such as
   network accelerators and firewalls. Some such devices reportedly
   feature datagram de-duplication that relies on IP ID uniqueness to
   identify duplicates, which has been discussed in Section 5.1.

5.4. 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

the assumption that the ID does not change (e.g., ROHC [RFC5225]),
where others include non-changing IDs via zero deltas (e.g., Enhanced
Compressed RTP (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.

5.5. Impact of Network Reordering and Loss

   Tolerance to network reordering and loss is a key feature of the
   Internet architecture. Although most current IP networks avoid
   gratuitous such events, both reordering and loss can and do occur.
   Datagrams are already intended to be reordered or lost, and recovery
   from those errors (where supported) already occurs at the transport
   or higher protocol layers.

   Reordering is typically associated with routing transients or where
   flows are split across multiple paths. Loss is typically associated
   with path congestion or link failure (partial or complete). The
   impact of such events is different for atomic and non-atomic
   datagrams and is discussed below. In summary, the recommendations of
   this document make the Internet more robust to reordering and loss by
   emphasizing the requirements of ID uniqueness for non-atomic
   datagrams and by more clearly indicating the impact of these
   requirements on both endpoints and datagram transit devices.

5.5.1. Atomic Datagrams Experiencing Reordering or Loss

   Reusing ID values does not affect atomic datagrams when the DF bit is
   correctly respected, because order restoration does not depend on the
   datagram header. TCP uses a transport header sequence number; in
   some other protocols, sequence is indicated and restored at the
   application layer.

   When DF=1 is ignored, reordering or loss can cause fragments of
   different datagrams to be interleaved and thus incorrectly
   reassembled and discarded. Reuse of ID values in atomic datagrams,
   as permitted by this document, can result in higher datagram loss in
   such cases. Situations such as this already can exist because there
   are known devices that use a constant ID for atomic datagrams (some
   cellphones), and there are known devices that ignore DF=1, but high
   levels of corresponding loss have not been reported. The lack of
   such reports indicates either a lack of reordering or a loss in such
   cases or a tolerance to the resulting losses. If such issues are

reported, it would be more productive to address non-compliant
devices (that ignore DF=1), because it is impractical to define
Internet specifications to tolerate devices that ignore those
specifications. This is why this document emphasizes the need to
honor DF=1, as well as that datagram transit devices need to retain
the DF bit as received (i.e., rather than clear it).

5.5.2. Non-atomic Datagrams Experiencing Reordering or Loss

   Non-atomic datagrams rely on the uniqueness of the ID value to
   tolerate reordering of fragments, notably where fragments of
   different datagrams are interleaved as a result of such reordering.
   Fragment loss can result in reassembly of fragments from different
   origin datagrams, which is why ID reuse in non-atomic datagrams is
   based on datagram (fragment) maximum lifetime, not just expected
   reordering interleaving.

   This document does not change the requirements for uniqueness of IDs
   in non-atomic datagrams and thus does not affect their tolerance to
   such reordering or loss. This document emphasizes the need for ID
   uniqueness for all datagram sources, including rewriting middleboxes;
   the need to rate-limit sources to ensure ID uniqueness; the need to
   not reuse the ID for retransmitted datagrams; and the need to use
   higher-layer integrity checks to prevent reassembly errors -- all of
   which result in a higher tolerance to reordering or loss events.

6. Updates to Existing Standards

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

6.1. 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.

   It later states 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.

6.2. Updates to RFC 1122

RFC 1122 states in Section 3.2.1.5 ("Identification: RFC 791
Section 3.2") that:

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.

6.3. 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. Specifically, RFC 2003 states the
following, where [10] refers to 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 RFC 6864.

7. Security Considerations

When the IPv4 ID is ignored on receipt (e.g., for atomic datagrams),
its value becomes unconstrained; therefore, that field can more
easily be used as a covert channel. For some atomic datagrams it is
now possible, and may be desirable, to rewrite the IPv4 ID field to
avoid its use as such a channel. Rewriting would be prohibited for
datagrams protected by the IPsec Authentication Header (AH), although
we do not recommend use of the AH to achieve this result [RFC4302].

The IPv4 ID also now adds much less to the entropy of the header of a
datagram. Such entropy might be used as input to cryptographic
algorithms or pseudorandom generators, although IDs have never been
assured sufficient entropy for such purposes. The IPv4 ID had
previously been unique (for a given source/address pair, and protocol
field) within one MDL, 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.

8. References

8.1. Normative References

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

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

   [RFC1812]   Baker, F., Ed., "Requirements for IP Version 4 Routers",
               RFC 1812, June 1995.

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

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

8.2. Informative References

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

   [Bo11]   Boucadair, M., Touch, J., Levis, P., and R. Penno,
               "Analysis of Solution Candidates to Reveal a Host
               Identifier in Shared Address Deployments", Work in
               Progress, September 2011.

   [De11]   Despres, R., Ed., Matsushima, S., Murakami, T., and O.
               Troan, "IPv4 Residual Deployment across IPv6-Service
               networks (4rd) ISP-NAT's made optional", Work in Progress,
               March 2011.

   [Pe11]   Perreault, S., Ed., Yamagata, I., Miyakawa, S., Nakagawa,
               A., and H. Ashida, "Common requirements for Carrier Grade
               NATs (CGNs)", Work in Progress, December 2012.

   [RFC1144]   Jacobson, V., "Compressing TCP/IP Headers for Low-Speed
               Serial Links", RFC 1144, February 1990.

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

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

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

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

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

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

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

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

   [RFC4443] Conta, A., Deering, S., and 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, September 2007.

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

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

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

   [RFC6145] Li, X., Bao, C., and F. Baker, "IP/ICMP Translation
              Algorithm", RFC 6145, April 2011.

   [RFC6219]   Li, X., Bao, C., Chen, M., Zhang, H., and 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.

   [RFC6621]   Macker, J., Ed., "Simplified Multicast Forwarding",
               RFC 6621, May 2012.

9. Acknowledgments

   This document was inspired by numerous discussions with the author by
   Jari Arkko, Lars Eggert, Dino Farinacci, and Fred Templin, as well as
   members participating in the Internet Area Working Group. Detailed
   feedback was provided by Gorry Fairhurst, Brian Haberman, Ted Hardie,
   Mike Heard, Erik Nordmark, Carlos Pignataro, and Dan Wing. This
   document originated as an Independent Submissions stream document
   co-authored by Matt Mathis, PSC, and his contributions are greatly
   appreciated.

   This document was initially prepared using 2-Word-v2.0.template.dot.

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