IPv6 to Internet Standard

Bob Hinden

Background

- Goal is to move the core IPv6 RFCs to Internet Standard

- Internet Standard is defined in RFC 2026 as
  - An Internet Standard is characterized by a high degree of technical maturity and by a generally held belief that the specified protocol or service provides significant benefit to the Internet community.

RFC6410 Defines Advancement Process

- There are at least two independent interoperating implementations with widespread deployment and successful operational experience.
  1. (1) There are no errata against the specification that would cause a new implementation to fail to interoperate with deployed ones.
  2. (2) There are no unused features in the specification that greatly increase implementation complexity.
  3. (3) If the technology required to implement the specification requires patented or otherwise controlled technology, then the set of implementations must demonstrate at least two independent, separate and successful uses of the licensing process
  4. (4) If the technology required to implement the specification requires patented or otherwise controlled technology, then the set of implementations must demonstrate at least two independent, separate and successful uses of the licensing process.

Advancing Draft Standards

- Any protocol or service that is currently at the abandoned Draft Standard maturity level will retain that classification, absent explicit actions. Two possible actions are available:

  (1) A Draft Standard may be reclassified as an Internet Standard as soon as the criteria in Section 2.2 are satisfied.

  (2) At any time after two years from the approval of this document as a BCP, the IESG may choose to reclassify any Draft Standard document as Proposed Standard.

Updating RFCs

- RFC6410 doesn’t mention Updating RFCs
- Current advice from the ADs is that updating RFCs need to be incorporated
- Will have to show that updates have been implemented and meet RFC6410 criteria
- If no implementation experience, we can not include in bis version

Plan Presented at IETF93

- Re-classify to Internet Standard draft standard documents that require no changes. (IESG action)
- Start work on those that require updates. Restricted to errata and updates that meet the criteria for Internet standard.
- Phase 2 (Proposed standards documents)

Documents being Updated

- RFC2460 – Internet Protocol, Version 6 (IPv6) Specification
  - <draft-ietf-6man-rfc2460bis-00>
- RFC4291 – IP Version 6 Addressing Architecture
  - <draft-hinden-6man-rfc4291bis-06>

Documents Needing Update?

- RFC4443 – Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification
  - Updated by RFC4884 Extended ICMP Multipart Messages
  - Looking for a reviewer to evaluate if it can be reclassified as is, or does RFC4884 update need to be incorporated

Documents Ready to Advance

- RFC3596 – DNS Extensions to Support IP Version 6
- RFC1981 – Path MTU Discovery for IP version 6
  - Needs errata based on update to RFC2460 by <draft-ietf-6man-deprecate-atomfrag-generation-03>
- RFC4941 – Privacy Extensions for Stateless Address Autoconfiguration in IPv6

draft-ietf-6man-rfc2460bis-00

- All RFCs that update RFC2460 incorporated
  - RFC5095, RFC5722, RFC5871, RFC6437, RFC6564, RFC6935, RFC6946, RFC7045, RFC7112
- Errata ID: 2541, 4279, 2843 incorporated
- Also
  - Add instruction IANA Considerations to change references to RFC2460 to point to this document
  - Added paragraph acknowledging the authors of the updating RFCs
  - Remove old paragraph in Section 4 that should have been removed when incorporating the update from RFC7045.

Fragmentation Updates

- RFC5722
  - Don’t create and drop overlapping fragments
- RFC6946
  - Rule for processing Atomic fragments
- RFC7112
  - Require that all headers through the Upper-Layer Header are in the first fragment
- <draft-ietf-6man-deprecate-atomfrag generation-03>
  - Remove text to create atomic fragments on receipt of ICMP Packet Too Big with MTU < 1280

RFC5095 and RFC5871 Update

- RFC5095 Deprecated RH0
- RFC5871 Defined RH Allocation Guidelines

- Removed RH0 Routing Header text, replaced with:

```
The currently defined IPv6 Routing
Headers and their status can be found at
[IANA-RH].  Allocation guidelines for
IPv6 Routing Headers can be found in
[RFC5871].
```

RFC6437 Update

- Current specifications of the IPv6 Flow Label field and the Traffic Class as defined in [RFC2474] and [RFC3168]

6. Flow Labels

The 20-bit Flow Label field in the IPv6 header is used by a source to label sequences of packets to be treated in the network as a single flow.

The current definition of the IPv6 Flow Label can be found in[RFC6437].

7. Traffic Classes

The 8-bit Traffic Class field in the IPv6 header is used by the network for traffic management. The value of the Traffic Class bits in a received packet might be different from the value sent by the packet's source.

The current use of the Traffic Class field for Differentiated Services and Explicit Congestion Notification is specified in[RFC2474] and [RFC3168].

RFC6564 Update

- Defines Uniform Format for IPv6 Extension Headers
- Added new Section 4.8 that has recommendations for defining new Extension headers and options

4.8. Defining New Extention Headers and Options

No new extension headers that require hop-by-hop behavior should be defined.

New hop-by-hop options are not recommended because, due to performance restrictions, nodes may ignore the Hop-by-Hop Option header, drop packets containing a hop-by-hop header, or assign packets containing a hop-by-hop header to a slow processing path. Designers considering defining new hop-by-hop options need to be aware of this likely behaviour. There has to a very clear justification why any new hop-by-hop option is needed before it standardized.

Instead of defining new Extension Headers, it is recommended that the Destination Options header is used to carry optional information that need be examined only by a packet's destination node(s), because they provide better handling and backward compatibility. Defining new IPv6 extension headers is not recommended. There has to a very clear justification why any new extension header is needed before it is standardized.

If new Extension Headers are defined, they need to use the following format:

……

RFC6935 Update

- Change to support zero UDP checksums for tunneled packets

> As an exception to the default behaviour, protocols that use UDP as a tunnel encapsulation may enable zero-checksum mode for a specific port (or set of ports) for sending and/or receiving. Any node implementing zero-checksum mode must follow the requirements specified in "Applicability Statement for the use of IPv6 UDP Datagrams with Zero Checksums" [RFC6936].

RFC7045 Update

- Changed the requirement that hop-by-hop processing is a should, and note that some nodes won't process the Hop-by-Hop Option header

> The exception referred to in the preceding paragraph is the Hop-by-Hop Options header, which carries information that **should** be examined and processed by every node along a packet's delivery path, …
>
> It should be noted that due to performance restrictions nodes may ignore the Hop-by-Hop Option header, drop packets containing a hop-by-hop option header, or assign packets containing a hop-by-hop option header to a slow processing path. Designers planning to use a hop-by-hop option need to be aware of this likely behavior.

Updates from Mailing List Discussion (-01 version)

- **Added text that Extension headers must never be inserted by any node other than the source of the packet.**
- Change "must" to "should" in Section 4.3 on the Hop-by-Hop header, part of RFC7045. Should have been part of the RFC7045 update.
- **Added text that the Data Transmission Order is the same as IPv4 as defined in RFC791.**
- Updated the Fragmentation header text to correct the inclusion of AH and note no next header case.
- Change terminology in Fragment header section from "Unfragmentable Headers" to "Per-Fragment Headers”.
- **Removed paragraph in Section 5 that required including a fragment header to outgoing packets if a ICMP Packet Too Big message reporting a Next-Hop MTU less than 1280. This is based on the update in <draft-ietf-6man-deprecate-atomfrag-generation-03>.**
- **Changed to Fragmentation Header section to clarify MTU restriction and 8-byte restrictions, and noting the restriction on headers in first fragment.**

Inserting Extension Headers

- As a result of mailing list discussion added to Section 4. IPv6 Extension Headers
  - Extension headers must never be inserted by any node other than the source of the packet. IP Encapsulation must be used to meet any requirement for inserting headers, for example, as defined in [RFC2473]
- This was the intent of the specification and represents a clarification

Data Transmission Order

- Added text that the Data Transmission Order is the same as IPv4 as defined in RFC791

```
The data transmission order for IPv6 is
the same as for IPv4 as defined in
Appendix B of [RFC0791].
```

Deprecate Atomic Fragments

- RFC2460 update from <draft-ietf-6man-deprecate-atomfrag-generation-03>
- Removed from Section 5. Packet Size Issues

> In response to an IPv6 packet that is sent to an IPv4 destination (i.e., a packet that undergoes translation from IPv6 to IPv4), the originating IPv6 node may receive an ICMP Packet Too Big message reporting a Next-Hop MTU less than 1280. In that case, the IPv6 node is not required to reduce the size of subsequent packets to less than 1280, but must include a Fragment header in those packets so that the IPv6-to-IPv4 translating router can obtain a suitable Identification value to use in resulting IPv4 fragments. Note that this means the payload may have to be reduced to 1232 octets (1280 minus 40 for the IPv6 header and 8 for the Fragment header), and smaller still if additional extension headers are used.

- Plan is to publish deprecate-atomfrag as Informational and not update RFC2460

Clarification of Fragment Text

- Changed to Fragmentation Header section to clarify MTU restriction and 8-byte restrictions, and noting the restriction on headers in first fragment.

> The Fragmentable Part of the original packet is divided into fragments. The lengths of the fragments must be chosen such that the resulting fragment packets fit within the MTU of the path to the packets' destination(s). **Each complete fragment, except possibly the last ("rightmost") one, being an integer multiple of 8 octets long.**
>
> (3) Extension Headers, if any, and the Upper-Layer header. These headers must be in the first fragment. **Note: This restricts the size of the headers through the Upper-Layer header to the MTU of the path to the packets' destinations(s).**

Use of RFC2119 and Uppercase

- Discussion on the mailing list about use of RFC2119 and Uppercase keywords
- My thinking:
  - Formally RFC2119 is not required to be used, nor does it require upper case language.
  - IPv6 is widely implemented and current style has not been a hindrance to IPv6 deployment. Nor do any of the RFCs that update fix this kind of problem.
  - Rewriting all of the declarative text to use upper case “SHOULD/MUST” might break interoperability.
  - Moving to RFC2119 style has the potential to create problems that doesn’t exist today.
- Conclusion of discussion was to keep current style in RFC2460 (and RFC4291)

Open Issue

- Handling of exact duplicate fragments identified on the mailing list was left open
  - Some level 2 hardware may generate duplicate packets
    - Not sure how frequent a problem this is
- Current Fragmentation text treats this case as an overlapping fragment and discards all matching fragments
- Propose text that describes the issue, and suggests handling this case, but not requiring it
  - Low frequency event
  - Avoids requiring implementation changes

New Drafts Updating RFC2460

- IPv6 Hop-by-Hop Header Handling
  - draft-baker-6man-hbh-header-handling
- Transmission and Processing of IPv6 Options
  - draft-gont-6man-ipv6-opt-transmit,
- IPv6 Universal Extension Header
  - draft-gont-6man-rfc6564bis

RFC2460bis Next Steps

- Update draft with outcome from this meeting
- Make recommendation about obsoleting updating RFC
- Collect implementation reports on Updates
  - We need to be sure that all of the updates have been implemented and interoperate
  - Will need to back out unimplemented updates
- W.G. Last call

draft-hinden-6man-rfc4291bis-06

- All RFCs that update RFC4291 incorporated
  - RFC7371, RFC7346, RFC6052, RFC7136, RFC5952
- Errata incorporated
  - Errata IDs: 3480, 1627, 2702, 2735, 4406, 2406, 863, 864, 866
- Also
  - Updated references
  - Acknowledgement to authors of updating RFCs

RFC5952 Update

- New section 2.2.3 added
  - Recommendation for outputting IPv6 addresses
    - `This section provides a recommendation for systems generating and outputting IPv6 addresses as text.  Note, all implementations must accept and process all addresses in the formats defined in the previous two sections of this document.  The recommendations are as follows:`

      `.......`
- All addresses shown in the document were changed to lower case.

RFC7346 Update

- Added Realm-Local scope to the multicast scope table in Section 2.6, and add the updating text to the same section
- Scope Values:

0	reserved
1	Interface-Local scope
2	Link-Local scope
3	Realm-Local scope
4	Admin-Local scope
....	

RFC7346 Update (2)

Interface-Local, Link-Local, and Realm-Local scope boundaries are automatically derived from physical connectivity or other non-multicast-related configurations. Global scope has no boundary. The boundaries of all other non-reserved scopes of Admin-Local or larger are administratively configured. For reserved scopes, the way of configuring their boundaries will be defined when the semantics of the scope are defined.

According to [RFC4007], the zone of a Realm-Local scope must fall within zones of larger scope. Because the zone of a Realm-Local scope is configured automatically while the zones of larger scopes are configured manually, care must be taken in the definition of those larger scopes to ensure that the inclusion constraint is met.

Realm-Local scopes created by different network technologies are considered to be independent and will have different zone indices (see Section 6 of [RFC4007]). A router with interfaces on links using different network technologies does not forward traffic between the Realm-Local multicast scopes defined by those technologies.

RFC6052 Update

- Added text in Section 2.3 that points to the IANA registries that records the prefix defined in RFC6052 and a number of other special use prefixes.

```
The current assigned IPv6 prefixes and
references to their usage can be found
in the IANA Internet Protocol Version 6
Address Space registry [IANA-AD] and the
IANA IPv6 Special-Purpose Address
Registry [IANA-SP].
```

RFC7136 Update

- Deprecate the U and G bits in Modified EUI-64 format Internet IDs

Interface IDs must be viewed outside of the node that created Interface ID as an opaque bit string without any internal structure.

For all unicast addresses, except those that start with the binary value 000, Interface IDs are required to be 64 bits long. **If derived from an IEEE MAC-layer address, they must be constructed in Modified EUI-64 format.**

Modified EUI-64 format-based interface identifiers may have universal scope when derived from a universal token (e.g., IEEE 802 48-bit MAC or IEEE EUI-64 identifiers [EUI64]) or may have local scope where a global token is not **being used** (e.g., serial links, tunnel end-points) or where global tokens are undesirable (e.g., temporary tokens for privacy [RFC4941].

~~The use of the universal/local bit in the Modified EUI-64 format identifier is to allow development of future technology that can take advantage of interface identifiers with universal scope.~~

RFC7371 Update

- Change the flag bits and their definitions in Section 2.6

```
                             +-+-+-+-+
  ff1 is a set of 4 flags:   |X|R|P|T|
                             +-+-+-+-+
```

The high-order flag is reserved, and must be initialized to 0.

T = 0 indicates a permanently-assigned ("well-known") multicast address, assigned by the Internet Assigned Numbers Authority (IANA).

T = 1 indicates a non-permanently-assigned ("transient" or "dynamically" assigned) multicast address.

The P flag's definition and usage can be found in [RFC3306] **as updated by [RFC7371].**

The R flag's definition and usage can be found in [RFC3956] **as updated by [RFC7371].**

The X flag's definition and usage can be found in [RFC3956] **as updated by [RFC7371].**

- No information if this is implemented

RFC4291bis Next Steps

- Ready for working group adoption
- Collect implementation reports on Updates
- W.G. Last call

Summary

- Work on rfc2460bis and rfc4291bis close to being done
- Need to make decision on ICMPv6 (bis?)
- Collect and verify implementation of Updates
- Submit to IESG for Internet Standard
  - rfc2460bis, rfc4291bis
  - ICMPv6 RFC4443 (or rfc4443bis)
  - RFC3596, RFC1981, RFC2460, RFC4941

BACKUP SLIDES

Draft Standard documents

- RFC2460 – Internet Protocol, Version 6 (IPv6) Specification
- RFC4291 – IP Version 6 Addressing Architecture
- RFC4443 – Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification
- RFC3596 – DNS Extensions to Support IP Version 6
- RFC1981 – Path MTU Discovery for IP version 6
- RFC4861 – Neighbor Discovery for IP version 6 (IPv6)
- RFC4862 – IPv6 Stateless Address Autoconfiguration
- RFC4941 – Privacy Extensions for Stateless Address Autoconfiguration in IPv6
- RFC5072 – IP Version 6 over PPP

RFC2460: IPv6 Specification

- Status
  - 9 Updated by:
    - RH0 deprecation
    - Overlapping fragments (1 errata)
    - IANA considerations for routing types
    - flow label specification
    - uniform IPv6 extension header format
    - UDP checksum for tunneled packets (1 errata)
    - processing of atomic fragments
    - transmission and processing of IPv6 extension headers
    - implications of oversized IPv6 header chains
  - Two errata Held for Document update
- Proposal
  - RFC2460bis. Revise and re-classify as IS.
    Bob Hinden volunteered to be document editor.

RFC4291: IPv6 Addressing Architecture

- Status
  - 5 updated by:
    - IPv6 address text representation (1)
    - IPv6 addressing of IPv4/IPv6 translators
    - Significance of IPv6 Interface Identifiers
    - IPv6 multicast address scopes
    - Updates to the IPv6 multicast addressing architectures
  - 2 errata (no interoperability issue)
- Proposal
  - Re-classify RFC4291 to Internet Standard

RFC4443: ICMPv6

- Status:
  - 1 updated by:
    - Extended ICMP to Support Multi-Part Messages(1)
  - 4 errata (no interoperability issue)
- Proposal
  - Reclassify RFC4443 to Internet Standard.

RFC3596: DNS (AAAA)

- Status
  - No errata
  - No updated-by
- Proposal
  - Re-classify RFC3596 to Internet Standard

RFC1981:PMTUD

- Status:
  - No errata
  - No updated-by
- Proposal:
  - Re-classify as an Internet Standard

RFC4861: Neighbor Discovery

- Status:
  - 5 updated by:
    - IPv6 subnet model, links and subnet prefixes
    - Security issues with ipv6 fragmentation and ipv6 ND
    - NUD is too impatient
    - Enhanced duplicate address detection
    - Packet loss resiliency for router solicitations
  - 3 verified errata (interoperability arguable), 3 held for document update
- Proposal
  - Recycle at current level

RFC4862: SLAAC

- Status:
  - 1 updated by:
    - Enhanced duplicate address detection
  - 1 errata reported (no interoperability issue)
- Proposal
  - Recycle at current level

RFC4941: Privacy Addresses

- Status:
  - No updated by:
  - 3 verified errata (no interoperability issue), 4 held for document update
- Proposal:
  - Re-classify RFC4941 as Internet Standard

RFC5072: PPP

- Status:
  - No updated by
  - No errata
- Proposal
  - Phase 2 with rest of IPv6 over foo documents

Phase 2: IPv6 over foo?

- **RFC2464 – Transmission of IPv6 Packets over Ethernet Networks**
- RFC2467 – Transmission of IPv6 Packets over FDDI Networks
- RFC2470 – Transmission of IPv6 Packets over Token Ring Networks
- RFC2473 – Generic Packet Tunneling in IPv6 Specification
- RFC2491 – IPv6 over Non-Broadcast Multiple Access (NBMA) networks
- RFC2492 – IPv6 over ATM Networks
- RFC2497 – Transmission of IPv6 Packets over ARCnet Networks
- RFC2590 – Transmission of IPv6 Packets over Frame Relay Networks Specification
- RFC3146 – Transmission of IPv6 Packets over IEEE 1394 Networks
- RFC4338 – Transmission of IPv6, IPv4 and Address Resolution Protocol (ARP) Packets over Fibre Channel
- RFC4944 – Transmission of IPv6 Packets over IEEE 802.15.4 Networks
- RFC5121 – Transmission of IPv6 via the IPv6 Convergence Sublayer over IEEE 802.16 Networks
- RFC7428 – Transmission of IPv6 Packets over ITU-T G.9959 Networks