

Geoff Huston
April 2016

Declaring IPv6 an “Internet Standard”

I’ve already shared my thoughts following a session of the IPv4 Sunset Working Group at IETF 95 that considered whether to declare IPv4 an “Historic” specification. Of course, as one would expect for a meeting of a Standards Development Organization (SDO), that wasn’t the only standards process discussion through the week. Another session, this time in the IPv6 Maintenance Working Group, considered the related topic of whether to make the IPv6 specification a full Internet Standard. Let’s look at that proposal.

The role of the Internet Engineering Task Force is not, as it’s name might suggest, to design and construct computer networks. Its role is not in fact to perform or even oversee engineering-related activities. Its role is one level removed, namely to produce workable specifications that not only allow others to design and construct functional computer networks, but also to produce specifications that allow independently produced artefacts that comply with such specifications to interoperate in intended ways. The IETF is a Standards Development Organization, and its role is to develop standards.

In a broader context this is not a novel task by any means. Open interoperable standards became the foundation of the industrial world and are certainly the bedrock of our post-industrial world. No matter where you look, from vehicles, houses, and household goods, through to many other aspects of the urban environment, what you see and interact with is manufactured according to a set of standard specifications. These standards that define the strength of timber to be used to build houses, or the level of fire resistance used in the material to clad high rise towers, and so many other things to allow for interoperability, safety and security in our built environment.

Obviously, standards do not enter this world in a pristine ready-to-use manner. While the result may be like Lego products, a set of interoperable components that can be assembled to make a functioning whole in a myriad of ways, the standard itself is often the result of protracted effort. Often this effort pulls in a diversity of interests into a committee or a study group, and the group effort is to arrive at an outcome that is acceptable to most, if not all, of the assembled interests. Conventionally this standards process tends to rely on representation rather than direct participation, and often reaches its position by simple processes such as voting. Back in the late eighties the then new IETF prided itself on being different. Dave Clark coined a phrase at the time that became the mantra of the IETF: “We reject kings, presidents and voting. We believe in rough consensus and running code.”

While this statement may be attributed to the hubris of a youthful organization wanting to assert its own unique identity, there was a certain implied deep criticism of other communications standards groups in that phrase. Around the late 1980’s the communications world was looking to shift towards flexible high capacity digital networks and a number of technical standards committees had set their best and brightest to the task. Time Division switches were just too inefficient and had significant scaling issues, so packet switching was an obvious place to explore. But the results, if the Asynchronous Transfer

Mode (ATM) specification was an indicator, were a splendid illustration of the pitfalls of committee work. In this case representatives from European telephone bodies wanted a 32-byte payload to facilitate the simple adaptation of voice systems into the network, particularly with respect to reduced jitter and eliminating the need for echo cancellation. Their counterparts from the other side of the Atlantic advocated a 64-byte payload that appeared to offer a workable compromise for data traffic. The committee result, namely a vote for a compromise 48-byte payload and a 5-byte cell header was a result that was more of a camel, and rather than being the mainstay of the communications enterprise over the ensuing decades ATM remains as a testament to the inanity of committee compromises in technical standards. It lives on as the mainstay of an unfulfilled alternate fantasy world for the more deluded telephants, where the world decided to use ATM everywhere instead of IP! Perhaps in the ultimate cruel blow to its legacy, a Google search for "ATM" these days is interpreted as a request for cash machine locations!

The IETF's avowed eschewing of voting is illustrated best by a comment that it makes absolutely no sense for a committee to vote on the value of pi! The comment about kings and presidents appeared to me to be a stated preference for direct participation rather than representation. The IETF encouraged the developer writing the code, or the engineer designing the system, to use the IETF to talk directly to peers undertaking similar work. Empowering the direct participating of such folk necessarily meant discouraging representation and indirection, and also meant redefining "leadership" as "facilitation" and not "direction setting".

The, perhaps in retrospect naive, expectation was that the quest for efficiency, interoperability, simplicity and scalability would naturally yield the "right" answers, and that voting would be at best confusing and at times just a misleading distraction. The IETF's intended approach was to test ideas through implementation and experimentation, and then fold this experience back into the standards process. The standard specifications would be produced with the assurance that these specifications had been used to produce implementations, and the implementations were mutually compatible.

The ensuing years have not been all that kind to these youthful ideals. These days the IETF is perhaps a little more open to participants in its working groups when compared to other standards bodies, but voting is now commonplace as a means of exposing points of agreement and disagreement within a working group. And as for kings and presidents, the Internet Architecture Board appears to think that from time to time it has to provide the IETF with "leadership" of some form or another. Running code is a dimly remembered aspiration in many cases, and interpreting the IETF's copious written outputs is in itself an art form.

Requests for Comments and Internet Standards

What are these written outputs?

The theory, as documented in RFC 2026, is that all work starts with an *internet draft*. These documents were intended to be rough sketches of ideas, contributed into the IETF process as raw inputs. They are published for no longer than 6 months, and are often updated in their lifetime.

Of course the Internet forgets nothing and these temporary documents are no exception. These days' internet drafts not only form an entire document collection in their own right, but have been cited by buyers and vendors alike as a surrogate standard specification when the need arises.

A draft may be adopted by a Working Group and further refined, progressing through a number of iterations. When its thought to be fully cooked the specification heads into a review phase and subsequent publication as a *Request for Comment* (RFC). (Although these days its a bit of a mystery as to exactly who in the IETF is requesting a comment, from whom, and why!). There are five types of RFCs: *Standards Track*, *Best Current Practice*, *Experimental*, *Informational* and *Historic*. The intention of these classifications is pretty straightforward, as their names suggest.

Within the *Standards Track* are three sub classifications: *Proposed Standard*, *Draft Standard* and *Internet Standard*.

Proposed Standard specifications are entry level specifications. While the specification is intended to be relatively stable and its been widely reviewed, it has not necessarily been used to generate implementations (in other words, running code is optional for Proposed Standards) nor is it necessary to have operational experience, even if there are working implementations of the specification. A Proposed Standard may change, or it may be superseded or even abandoned. Practically, Proposed Standards encompass a panoply of documents which a complete spectrum of maturity levels. Some working groups have been diligent in their development and review processes such that the resultant Proposed Standard functionally meets the intentions of a full Internet Standard. Their documents are clear, and complete. The technical specification is stable and mature. And there is extensive operational experience. Other Proposed Standards are highly speculative, and really fall short of the practical requirements of a functional technical standard.

Draft Standards are distinguished by have at least two independent and interoperable implementations of the specification, including all options and features of the specification. The subliminal message of a Draft Standard is stability, maturity and utility. What may be missing is operational experience, particularly at scale, but that are considered to be working prototypes of the final Internet Standard and there is no practical impediment to their use as technical specifications.

Internet Standards are specifications that have enjoyed "significant implementation and successful operational experience" according to RFC 2026. Not only is the specification complete and stable in every respect, described as "technical maturity", it is also believed to be useful, or again to quote RFC2026, "a generally held belief that the specified protocol or service provides significant benefit to the Internet community."

The State of RFCs

Given this quick description of the IETF's Standards Process, let's now look at RFCs by the numbers.

At the time of writing this note we appear to be up to RFC 7842. However, some 79 RFC numbers are listed as "Not Issued". There are a further 45 number gaps in the public RFC document series, leaving 7,118 documents in the RFC series.

Of these 7,118 RFCs, 288 are classified as *Historic* and 888 of the earlier RFCs are marked with an *Unknown* status. 2,264 are *Informational*, 274 are classified as *Best Current Practice*, and 460 are *Experimental*. The remaining 3,292 RFCs, or 46% of the entire corpus of RFC documents, are *Standards Track* documents.

Of these 3,292 Standards Track documents some 3,045, or 92% of the Standards Track collection, are at the first stage, namely *Proposed Standard*. This strongly suggests that many proponents apply energy to get a specification into the Standards Track, but are not overly fussed about expending further effort in progressing the document any further once it reaches the initial Standards Track designation. This strongly suggests that for many there is little practical difference between Proposed, Draft and Internet Standards.

A further 141 documents are *Draft Standards*, and just 106 are *Internet Standards*. This latter number is slightly misleading, as 7 of these documents are obsoleted by other RFCs, leaving just 99 current Internet Standards documents.

So if the objective of the IETF is to foster the development of *Internet Standards* specifications, then strictly speaking it has not enjoyed a very stellar record over its 30-year history. Almost one half of these Internet Standard specifications were generated in the 1980s (42 RFCs have original publication dates in the 1980s), just 19 in the 1900's and 26 in the 2000's. There were none in 2011. 4 in 2011, 1 in 2012, 2 in 2013, 1 in 2014, none in 2015. There have been 3 so far in 2016.

The numbers would suggest that if the broader industry even looks behind the subtleties of the RFC classification process, and it probably does not, then *Proposed Standard* certainly appears to be more than enough, and much of the Internet today is constructed on stable mature specifications that the IETF calls *Proposed Standards*.

The IPv6 Standard

Now let's turn to IPv6.

Some 371 RFC's have "IPv6 in their title.

146 of these RFCs are *Informational*, 4 of these are *Historic*, 23 are *Experimental*, 5 are *Best Current Practice* and the remaining 193 are *Standards Track* documents. Of these 193 documents, 24 are already obsoleted, 164 are *Proposed Standards*, just 5 are *Draft Standards* and there are no *Internet Standards*.

The question considered at the 6MAN WG meeting was whether it is time to promote IPv6, as specified in RFC2460, as an *Internet Standard*.

The review of RFCs above would tend to the view that this is not an action that has any particular significance outside of the IETF. The larger world appears to be perfectly happy to use Proposed and Draft Standard specifications on an equal footing to Internet Standards. So it's not about the external perception of IPv6, or at least that's what the existing practice seems to indicate. But there is an internal message from the IETF back to itself, saying that as an Internet Standard then we are done. IPv6 is cooked and ready. No more fiddling with the bits! And the question I have is: Are we really ready? Is this specification now ready to be considered a stable and mature standard?

Just to remind ourselves, RFC2026 says that the criteria for an Internet Standard is:

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.

I don't think that there is any doubt about the "significant benefits" part of this criteria when considering IPv6.

But the issue of "technical maturity" is one I'd like to explore a little further. In so many ways IPv6 is just IPv4 with larger address fields. There are just a few technical differences, but its these differences

that continue to be a source of technical exploration and potential uncertainty. IPv6 added a 20-bit Flow Identifier field to the header. IPv6 cemented the “Don’t Fragment” flag to ON. IPv6 altered the IPv4 options to IPv6 Header Extensions. And finally, IPv6 explicitly allowed multi-addressing. I believe that its these four alterations that continue to generate subtle effects that have a bearing on the perception of the IPv6 specification as stable and mature.

Extension Headers

IPv4 used a variable header size, and allowed for up to 40 octets of “IP options”. They never enjoyed much use, and while “landmark” routing (Loose Source Routing Option) and “source” routing (Strict Source Routing Option) were fun to play with, they were a security nightmare.

The packet could not be efficiently processed through all the various forms of deployed routers, and a common implementation path was for the router to generate an exception and use software processing for a packet with IP options. These days’ packets with IPv4 options are variously ignored or discarded.

IPv6 took this construct and replaced it with a fixed length IP header and a chained list of “Next Headers”. At the end of the chain was the Upper Layer header, conventionally, but not exclusively, a UDP or TCP header. The chain could be populated by options relating to the intermediate forwarding decisions, handling by the destination, security options and support for mobility. As with IPv4 it is still challenging to process IPv6 packets with extension headers at wire speed, and there are reports that rather than process IPv6 packet headers with extensions a number of routers simply elect to drop IPv6 packets if they have extension headers. A report on the rather disturbing level of packet drop relating to IPv6 Extension headers is in the process of being published as an Informational RFC. (The URL of this internet draft is: <https://datatracker.ietf.org/doc/draft-ietf-v6ops-ipv6-ehs-in-real-world/>)

While this document postulates that this situation will improve over time, it is difficult to see how, which means that the practical lesson from this document is very simple: don’t use IPv6 extension headers if you want your IPv6 packet to be delivered!

And if IPv6 Extension headers are just not widely supported then are they really part of the IPv6 Standard?

To quote from RFC2026:

The requirement for at least two independent and interoperable implementations applies to all of the options and features of the specification. In cases in which one or more options or features have not been demonstrated in at least two interoperable implementations, the specification may advance to the Draft Standard level only if those options or features are removed.

Now there is a distinction here between demonstrated interoperation in just two selected implementations and demonstrated interoperation between all implementations, and in the case of IPv6 Extended headers it likely meets the first criteria to two selected implementations, but appears to fail the second, when looking for interoperability across all implementations.

Fragmentation

Which leads to the topic of fragmentation control in IPv6. Fragmentation is controlled by a Fragmentation Control IPv6 Extended Header. The same report notes a 30% to 40% drop rate for IPv6 packets with the IPv6 Fragmentation header present.

There is no concept of fragmentation in IPv6 within the network itself. When a packet is too large for the next hop the packet is discarded and an ICMP6 Packet Too Big is sent back to the source address contained in the IPv6 packet header. There are many reports of ICMP messages being filtered in the network, making the receipt of the ICMP6 message at best a haphazard affair.

Once forward fragmentation is eliminated from the IP protocol then is what's left in an IP level problem? A so far untried approach in IP is to truncate the packet and pass what fits forward to the destination with some explicit IP truncation flag set. And then there is the approach of performing Path MTU discovery that does not rely on RFC4821. Or we could revert to forward fragments and mimic the IPv4 options and behaviour. And then there is the other option, namely to remove IPv6 fragmentation from the protocol spec altogether, as described in a now-expired Internet draft "IPv6 Fragment Header Deprecated" (<https://datatracker.ietf.org/doc/draft-bonica-6man-frag-deprecate/>)

Whatever the way it is resolved, there is a common starting point here. IPv6 Packet Fragmentation, as described right now, does not appear to be all that robust out there in deployed networks, and that seems to undermine an assertion that this particular aspect of the specification is "mature".

Multi-Addressing

The other change introduced by IPv6 is the concept of multiple addresses. In IPv4 it is assumed that all IP hosts connected to a common network domain (normally a MAC broadcast domain) share a common prefix, and each connected host's interface to this network has a unique IP address. An IPv4 host would conventionally acquire multiple IPv4 addresses if it was to be connected to multiple discrete networks via multiple network interfaces.

IPv6 addressing moved away from an implicit binding to broadcast domains. It replaced the concept of a Layer 2 broadcast domain with a Layer 2 Multicast domain. Now in this new model each IP host binds itself into each of the local networks using its Layer 2 MAC address, expanding it to 64 bits, and using a constant IPv6 prefix of fe80::/64 to construct a local IPv6 address for each connected interface. This allows the host to then access the local network's multicast subnet, and thereby listen to Router Advertisements. The host can pick up these router advertisements and use the router's 64-bit network prefix as the prefix to use for the host's globally scoped address. There are other mechanisms to acquire an address in IPv6, but the essential characteristic of IPv6 is that there is no longer the required concept of a fixed address prefix that is associated with each Layer 2 broadcast domain.

So when a host with multiple IPv6 addresses wishes to communicate with a remote host with multiple IPv6 addresses, which address pair should be selected? RFC6724 offers some advice here, but if you want to do the right thing by BCP38, then you need to select source addresses that are aligned to the IP address of the site's egress router used to reach the destination address. All this is fine in theory, but we just can't quite figure out how to do this in IPv6. The problem is that source address spoofing is the scourge of today's Internet, and right now the Multi-Homed IPv6 end site environment appears to be adding to this problem, rather than assisting to mitigate it.

The Flow Identifier Field

The introduction of the 20-bit flow label was, to quote RFC2460, the Draft Standard for IPv6

This aspect of IPv6 is, at the time of writing, still experimental and subject to change as the requirements for flow support in the Internet become clearer. Hosts or routers that do not support the functions of the Flow Label field are required to set the field to zero when originating a packet, pass the field on unchanged when forwarding a packet, and ignore the field when receiving a packet.

This hardly fills the reader with confidence if they are looking for reassurance about the maturity of the specification.

A subsequent specification for the IPv6 Flow Label, the Proposed Standard RFC6437, provides some useful clarification of the goals of the Flow Label, and its applicability in multi-path contexts to assist

routers to preserve a single path for each discrete end-to-end packet flow. However, the accompanying rationale, RFC 6436, provides the observation that the IPv6 flow label is “hardly used in practice in widespread IPv6 implementations, although some operating systems do set it.” This document also notes various proposed mutually inconsistent uses for this Flow Label field.

Whether the new proposal gains traction and acceptance, or whether RFC6437 will be just one more mutually inconsistent proposal for interpretation and use of the Flow Label remains to be seen. But again what is evident that this is an option in the IPv6 header does is not uniformly interpreted or even used.

Are we ready for an Internet Standard for IPv6?

In some ways the role of a Standards Body in a changing world is a challenging one. It can take a snapshot of a technology and label it a “standard” but if the technology is alive and evolving then its also true that the next snapshot will be different. We continue to learn and evolve. In that respect, one view of an Internet Standard is a tombstone. Its a marker of a technology that is no longer changing or evolving.

It’s heartening to observe that we are still learning about new and different ways to operate packet networks. It’s equally heartening that we do not have the hubris to assert that we have attained perfection in these specifications. They are not finished, and there is still more to understand, and potentially more to change.

From time to time the IETF performs some introspection on its standards development process, and inevitably the observation is made that it appears that is no one, except the IETF itself, who cares about the distinction between *Proposed*, *Draft* and *Internet Standards*. Yes, there are many standards, and no, they they are not all mutually consistent. But perhaps all of this is simply the hallmark of a vibrant and flexible technology. As the range of application of the technology increases, and as our breadth and want to keep refining the technology. And there is nothing wrong with that.

For me its pretty clear that IPv6 is a usable and useful specification. It allows for interoperability, and it is maturing. It’s a good IETF Standard.

But like many others, that’s where I find myself stopping. I really can’t appreciate the finer nuances of the various sub categories of IETF Standards Track documents. I strongly prefer the early pragmatic view of the IETF. Where IPv6 is concerned we don’t have perfection. We do appear to have achieved a rough consensus for most of the specification and there is certainly running code. And that’s good enough for me!

Author

Geoff Huston B.Sc., M.Sc., is the Chief Scientist at APNIC, the Regional Internet Registry serving the Asia Pacific region. He has been closely involved with the development of the Internet for many years, particularly within Australia, where he was responsible for building the Internet within the Australian academic and research sector in the early 1990's. He is author of a number of Internet-related books, and was a member of the Internet Architecture Board from 1999 until 2005, and served on the Board of Trustees of the Internet Society from 1992 until 2001 and chaired a number of IETF Working Groups. He has worked as an Internet researcher, as an ISP systems architect and a network operator at various times.

www.potaroo.net

Disclaimer

The above views do not necessarily represent the views or positions of the Asia Pacific Network Information Centre.