August 2021
Geoff Huston

Running Code

There was an interesting discussion in a working group session at the recent IETF 111 meeting over a proposal that this working group should require at least two implementations (presumably independently developed implementations) of a working group draft before the working group would consider the document ready for submission to the IESG for progression to publication as an RFC.

What's going on here?

Some Background to "Running Code"

To provide some background to this discussion we should cast our mind back to the 1970's and 1980's and the industry's efforts at that time to define an *open* network standard. At that time the computer vendors had separately developed their own proprietary networking protocols, built to widely differing architectural principles and intended to meet very diverse objectives. IBM used *SNA*, Digital had *DECnet*, Apple had *Appletalk* and Wang had the delightfully named *WangNet*, to name just a few. While this worked well for the larger vendors of computers, customers were increasingly frustrated with this implicit form of vendor lock-in. Once they had purchased a mainframe from one vendor then they were locked in to purchase the entirety of their IT infrastructure from the same vendor, as other vendor's equipment simply would not work with the installed equipment. Customers were trapped and vendors knew that and at times ruthlessly exploited that condition by charging a premium for their peripherals, such as terminals, printers, and data entry systems. Customers wanted to break apart their IT environment, and source peripherals, mainframe systems, and indeed their entire network as separate transactions. What they needed was a common standard for these components so that a number of vendors could provide individual products that would interoperate within the customer's network.

This effort to define a vendor-neutral common network environment was taken up through a program called *Open Systems Interconnection* (OSI), a reference model for computer networking promulgated by the International Organization for Standardization and International Electrotechnical Committee (ISO/IEC).

The program's intentions were doubtless highly worthy and laudable, despite the sluggish support from some of the larger computer vendors. It produced an impressive stack of paperwork, held many meetings in many Fine Places and doubtless the participants enjoyed many Fine Dinners, but in terms of the technology it managed to define its outcomes were a little more mundane. At the transport level two competing mutually incompatible technologies that were incorporated into a single OSI transport standard, and at the application level there was an incomprehensible jumble of text that did not lend itself to working code, or even consistent human interpretation! There were many problems inherent in the OSI effort, including notably the unwillingness of existing vendors to desert their proprietary platforms and embrace an open and vendor-neutral technology. There was also the standards-making process itself that attempted to resolve differences through compromise, often termed in this particular context, the art of making everyone equally unhappy with the outcome. One need only look the process where the relevant ATM standards body was unable to decide between 32 and 64-octets for the payload size and compromised to a rather odd size of 48 octets for a call payload, which when coupled to a 5-octet header, resulted in an ATM cell size of a rather bizarre value of 53-octets.

The community of folk who were working on the development of the Internet protocols at the time were increasingly dismissive of the OSI efforts. There were glaring disconnects between the various optimistic statements of public policy in promoting open systems and OSI in particular (such as the GOSIP profiles adopted by public sectors in many national environments) and the practical reality that the OSI protocol suite was simply undeployable and the various implementations that existed at the time were fragmentary in nature. Perhaps more concerning was that it was altogether dubious whether these various OSI implementations interoperated in any useful way!

At the same time the IP effort was gaining momentum. Thanks to a DARPA-funded project, an implementation for the TCP/IP protocol stack on Unix was available as an open-source package from the good folk at Berkley, and the result was a startlingly good operating system in the form of Unix coupled with a fully functional and amazingly versatile networking protocol in the form of IP, and all for essentially no cost at all for the software.

There was a period of an evident rift between policy and practice in the industry, where many public sector procurement processes were signed up to a Government OSI Profile (or GOSIP) as a means of incenting vendors to commit to providing OSI-based services while at the same time many vendors and customers were embracing TCP/IP as a practical and fully functional technology. These vendors were also assisting these same public agencies in writing boilerplate excuses as to why GOSIP might be fine for others but was inappropriate for them when considering the agency's particular circumstances and requirements. On the one side the IP folk could not understand why anyone could sign up to non-functional technology, while on the other side the OSI folk, particularly those in Europe, could not understand why anyone could be led into committing into a common networking technology that was wholly and completely controlled by the United States. The links between the initial program instigators of IP, the US Defence Advanced Research Project Agency, and the implicit involvement of US government itself were an anathema to some folk, who took to pointedly calling the protocol "DoD IP" as a direct reference to its US military origins. The IP folk were keen to avoid an overt confrontation at a political level and through the late 1980's, as IP gained traction in the global research community, they were consistent in calling the Internet "an experiment" in broad scale networking technology, with the avowed intention of further informing the development of OSI into a deployable and functional technology platform.

Things can to a head in mid 1992 when the Internet Architecture Board grappled with the then topical question of scaling the Internet's routing and addressing issues and published a now infamous statement that advocated further development of the IP protocol suite by using the CLNS part of OSI. This excited a strong reaction in the IP community and resulted not only in a reversal of this stated direction to use CLNS in IP, but also resulted in a complete restructuring of the IETF itself, including the IAB! This included a period of introspection as to why IP was becoming so popular and what were the essential differences in the standardisation processes used by the technical committees in ISO/IEC and the IETF.

Dave Clark of MIT came up with a pithy summarisation of the IETF's mode of operation at the time in his A Cloudy Crystal Ball/Apocalypse Now presentation at the July 1992 IETF meeting: "We reject kings, presidents, and voting. We believe in rough consensus and running code." The first part of this statement is a commentary on the conventional standard process, where delegates to the standard meeting vote on proposals and adoption is by a majority vote. Dave was trying to express the notion that it's not which company or interest that you might want to represent at a meeting, or just a simple head count of those in favour and those opposed that should matter in the IETF. He wanted to express a more pragmatic observation about whether what you are proposing makes sense to your peers! Implicitly, it was also a message to the IAB at the time that making ex-cathedra pronouncements as to the future direction of IP was simply not a part of the emerging culture of the IETF. The second part of this mantra, namely "running code" was a commentary about the IETF's standards process itself.

The issue of whether the IETF has reverted to voting over the intervening years, or not, is a topic for another time. Here let's look at the "running code" part of this mantra for the IETF in a bit more detail.

The IETF Standards Process

The entire concept of a "standard" in the communications sector is to produce a specification of a technology that allows different folk to produce implementations of the technology that interoperate with each other in a completely transparent manner. An implementation of a protocol should not have to care in the slightest if the counterpart it is communicating with over the network is a clone of its own implementation of the standard or an implementation generated by someone else. It should not be detectable, and it should certainly not change the behaviour of the protocol. That's what "interoperable" was intended to mean in this context.

There were a few other considerations about this form of industry standard, including the consideration that the standard did not implicitly favour one implementation or another. A standard was not intended to be a competitive bludgeon where one vendor could extract an advantage by making their technology the "standard" in an area, nor was it intended to be a tombstone of a technology where no vendor was willing to implement a standard because they were unable to make money from implementing a standard as it was no longer current or useful anymore.

However, "running code" expresses something which at that time of OSI expressed a more fundamental aspect of a technology specification. The standard was sufficiently self-contained that a consumer of the standard could take the specification and implement the technology it described and produce a working artefact that interoperated cleanly with any other implementation based on the same specification. The specification did not require any additional information to be able to produce an implementation. This is the longer version of the intent of "running code."

What this means in practice is described at length in RFC2026: "an Internet Standard is a specification that is stable and well-understood, is technically competent, has multiple, independent, and interoperable implementations with substantial operational experience, enjoys significant public support, and is recognizably useful in some or all parts of the Internet." However, the reader should be aware of a subtle shift in terminology here. The statement in RFC2026 is not referring to a published RFC document, or a working draft that has been adopted by a IETF Working Group. It's referring to an "Internet Standard". As this RFC describes, there is a track that a specification is expected to progress through, from a Proposed Standard to a Draft Standard to an Internet Standard. This process was updated in 2011 with the publication RFC6410 which recognised that there was considerable confusion of the exact role of the Draft Standard phase within the process, illustrated by the observation that remarkably few specifications were moving from Proposed Standard to Draft Standard. So RFC6410 revised this to describe a two-step process of the "maturation" of an Internet Standard. The stages in the IETF Standards process are:

> Proposed Standard: A Proposed Standard specification is generally stable, has resolved known design choices, is believed to be well-understood, has received significant community review, and appears to enjoy enough community interest to be considered valuable. However, further experience might result in a change or even retraction of the specification before it advances. Usually, neither implementation nor operational experience is required for the designation of a specification as a Proposed Standard.
>
> Internet Standard: A specification for which significant implementation and successful operational experience has been obtained [...]. An Internet Standard (which may simply be referred to as a 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.

What's missing from RFC6410 is the stage of a Draft Standard, which in the earlier RFC included a requirement for "running code", namely that "at least two independent and interoperable implementations from different code bases have been developed, and for which sufficient successful operational experience has been obtained" (RFC2160)

It appears that the IETF had learned to adopt a flexible attitude to "running code". As RFC6410 notes: "Testing for interoperability is a long tradition in the development of Internet protocols and remains important for reliable deployment of services. The IETF Standards Process no longer requires a formal interoperability report, recognising that deployment and use is sufficient to show interoperability.".

> This strikes me as a good example of a "Well, Yes, but No!" form of evasive expression that ultimately eschews any formal requirement for "running code" in the IETF standards process.

RFC6410 noted that: "The result of this change is expected to be maturity-level advancement based on achieving widespread deployment of quality specifications. Additionally, the change will result in the incorporation of lessons from implementation and deployment experience, and recognition that protocols are improved by removing complexity associated with unused features."

How did all this work out? Did anyone listen? Let's look at the numbers to find out.

RFCs by the Numbers

At the time of writing in August 2021 we appear to be up to RFC 9105 in the series of published RFCs. However, some 184 RFC numbers are currently listed as "Not Issued". There are a further 25 number gaps in the public RFC document series, leaving 8,896 documents in the RFC series.

Of these 8,896 RFCs, 331 are classified as Historic and 887 of the earlier RFCs (prior to November 1989) are marked with an Unknown status. 2,789 are Informational, 300 are classified as Best Current Practice, and 522 are Experimental. The remaining 4,832 RFCs, or 54% of the entire corpus of RFC documents, are Standards Track documents.

Of these 4,832 Standards Track documents some 3,806, or 79% of the Standards Track collection, are at the first stage, namely Proposed Standard. A further 139 documents are Draft Standards and have been stranded in this state since 2011 since the publication of RFC6410.

Just 122 RFCs are Internet Standards. To be accurate, there are currently 85 Internet Standard specifications, each of which incorporate one or more component RFCs from this total set of 122 RFCs. That's just 2.5% of the total number of standards track RFCs. Almost one half of these Internet Standard specifications were generated in the 1980s (there 47 RFCs that are an Internet Standard or are part of an Internet Standard that have original publication dates in the 1980s or earlier), just 21 in the 1900's and 28 in the 2000's. A further 26 RFCs were published as Internet Standards in the 10 years since RFC6410 was published in 2011. Given the accelerating rate of RFC publication of this same period, it could be inferred that the quality of these RFC-published specifications is falling dramatically, given that the proportion of standards track RFCs that reach a level of full maturity as an Internet Standard is falling dramatically. There is, however, an alternative and more likely conclusion from these numbers.

That conclusion is that many IETF participants apply their energy to get a specification into the standards track at the first, or Proposed Standard, level, but are not overly fussed about expending further effort in progressing the document any further once it reaches this initial Standards Track designation. This strongly suggests that for many there is no practical difference between Proposed Standard and a full Internet Standard.

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 and these 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 sufficient and distinctions related to formal validation of "running code" or other aspects of the maturation of a technical specification is a piece of largely ignored IETF mythology.

Working Groups and Running Code

The formal requirement for running and interoperable code may have been dropped from the IETF standards process but some form of a requirement for implementations of a proposed specification is still part of the process of some IETF Working Groups. In IDR (Inter-Domain Routing), where there are 24 active drafts in the working group, it has been a common practice to request reports of implementations of draft specifications as a part of the criteria advancement of a draft through the Working Group, although this appears to be applied in various ways for various drafts!

In other cases, such as DNSOP (DNS Operations) there has been pushback from DNS software vendors against feature creep in drafts in this space (known as the "DNS Camel" after a now infamous presentation at IWTF 101 that decried the feature bloat that was being imposed on the DNS (https://www.ietf.org/blog/herding-dns-camel/). The response from some vendors is not to implement any DNSOP working group drafts (of where there are 17 such active documents in the working group at present) and await their publication as a Proposed Standard RFC as a precondition of any code development.

At IETF 111 there was a discussion in the SIDROPS (Secure Inter-Domain Routing Operations) to introduce an IDR-like requirement for implementations as some form of requirement precondition for a draft to progress to RFC publication, although in the content of an operations working group (as distinct from a protocol development working group) the intent of such a move by SIDROPS is probably only going to add to the levels of confusion rather than to add any clarity!

It appears that various working groups in the IETF have various positions on what "running code" actually means and whether any requirement should be folded into the Working Group's processes. Perhaps that spectrum of variance within the IETF reflects a deeper level of confusion about what we mean by "running code" in the first place. Some Proposed Standards have already had implementations and some level of interoperability tested by the Working Group before publication as a Proposed Standard RFC. Some do not. And the RFC itself does not record the various processes used by the Working Group to get the specification to the state of a Proposed Standard. No wonder folk get confused!

What do we mean by "running code" anyway?

I guess that this question lies at the heart of the conversation.

On the one hand, the phrase was intended to be a summation of the original set of criticisms of the ISO/IEC effort with OSI. If an organisation generates its revenue by selling paper copies of standards documents, as was the common case at the time, then producing more paper-based standard specifications was the way the organisation continued to exist. As was characterised at the time, this situation had degenerated into simply writing technical specifications for technologies that simply did not exist, or "paperware about vapourware". The IETF wanted to distinguish its efforts in a number of ways: It wanted to produce standard specifications that were freely available and available for free. It wanted to produce specifications that were adequately clear, so that they were able to guide implementers to produce working code, and adequately complete, so that multiple independent implementations based on these specifications would interoperate.

At the same time the IETF implicitly wanted a lot more than just this "elemental" view of standard specifications. It was not particularly interested in merely a disparate collection of specifications that met the objective of each specification supporting interoperable implementations. It wanted more. It wanted an outcome that the collection of such specifications described a functional networked environment. I guess that this larger objective could be summarised as a desire to produce a collection of specifications that each supported running code that, taken together, was capable of supporting a functional networked environment that passed supported running packets! This “running network” objective was an intrinsic property of the earlier vendor-based proprietary network systems of the 1980s and was the avowed intention of the OSI effort. The intention was that consumers could purchase components of their networked environment from different vendors, and at different times, and put them together to construct a functional network. The parts have to work together to create a unified and coherent whole, and the IETF certainly embraced this objective.

However, the IETF thinking evolved to take on a grander ambition. With the collapse of the OSI effort in the early 1990's it was clear that there was only one open network architecture left standing, and that was IP. So, the IETF added a further, and perhaps even more challenging ambition to the mix. The technology specified through the IETF process had to scale. It had to be fit for use within the Internet of today and tomorrow. The specifications that can be used for tiny deployments involving just a couple of host systems also should be applicable to vast deployments that span millions and even billions of connected devices.

When we think about the intended scope of this latter objective, the entire exercise of producing specifications that support "running code" became a whole lot more challenging as a result. The various implementations of a specification had to interoperate and play their intended role in supporting a coherent system that is capable of scaling to support truly massive environments. But are we capable of predicting such properties within the scope of the specification of the technology? No one expected the BGP protocol to scale to the extent that it has. On the other hand, our efforts to scale up robustly secure network associations has been consistently found wanting. Scalability is a hard property to predict in advance.

At best we can produce candidate technologies that look like they might be viable in such a context, but ultimately, we will only know if the specifications can meet such expectations when we get to evaluate it in the light of experience. If the intended definition of an Internet Standard is a specification that has all of these attributes, including scalability, then at best it’s a specification that is an historical document that merely blesses what has worked so far. However, it has little practical value to consumers and vendors who are looking to further refine and develop their digital environment along the path to such levels of scaling in deployment.

Maybe it’s the case that Proposed Standards specifications are good enough. They might scale up and work effectively within the Internet. They might not. We just don’t know yet. The peer review process in the working group and in IESG review has performed a basic sanity test, hopefully, that the proposed specification is not harmful, frivolous or contradictory, and appears relatively safe to use.

Maybe that’s enough. Perhaps that as much as the IETF could or should do. A standard specification, not matter how carefully it may have been developed, is not the same as a cast-iron assurance of quality of performance of the resultant overall system in which it is used. Such a specification cannot guide a consumer through a narrowly constrained single path through a diverse environment of technology choices. A standard in this case is at best a part of an agreement between a provider and a consumer that the goods or service being transacted has certain properties. If many consumers express a preference to use a particular standard in such agreements, then producers will provide goods and services that conform to such standards. If they do not, then the standard is of little use. What this means is that in many ways the role of a standard specification in this space and the determination as to whether or not a standard is useful is ultimately a decision of the market, not the IETF.

Perhaps the most appropriate aspiration of the IETF is to produce specifications that are useful to consumers. As long as consumers are willing to cite these specifications in the requirements for the goods and services that they purchase, then the greater the motivation on the part of producers to provide goods or services that conform to these standard specifications.

Running Code?

And what about "running code"?

Maybe RFC6410 was correct in dropping a formal requirement for multiple interoperable implementations of a specification. The judgement of whether a standard is of sufficient completeness and clarity to support the implementation of running code is a function that the market itself is entirely capable of performing, and the additional effort in front-loading the standards development process with such implementations of various proposals can be seen as an imposition of additional cost and effort within the standardisation process.

On the other hand, "running code" is a useful quality filter for proposed specifications. If a specification is sufficiently unclear that it cannot support the development of interoperable implementations then it's not going to be a useful contribution, and it should not be published. No matter how many hums of approval or how many votes on the IESG ballots to endorse a specification as a Proposed Standard, if the specification is unclear, ambiguous, woefully insecure or just dangerous to use then it should not be a Proposed Standard RFC. And if we can't use the specification to produce running code, then the specification is pretty useless!

I think personally that there is a place for "running code" in today's IETF, and it should be part of the process of peer review of candidate proposals that guide a Working Group's deliberations on advancing a proposal to a Standards Track RFC. It would also be helpful if notes on implementation experience were able to be stapled to these documents as a set of helpful guides the next set of folk who want to use these standard specifications. In so many other digital realms we've managed to embrace living documents that incorporate further experience. Wikipedia, for all its faults, is undoubtedly still a major resource for our time. We probably need to get over our nostalgic obsession with lineprinter paper in RFCs and think about the intended role of these specifications within the marketplace of standards and technology. And while I'm on my wish list, maybe we should just drop all pretence as to some subtle distinction between Proposed and Internet Standards, and just call the lot "Standards" and be done with it!

Disclaimer

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

Author

Geoff Huston AM, B.Sc., M.Sc., is the Chief Scientist at APNIC, the Regional Internet Registry serving the Asia Pacific region.

www.potaroo.net