Rethinking Standardisation of Network Management

Submission to the IAB's *Next Era of Network Management Operations* workshop (2024)

Rob Shakir <rjs@rob.sh>, 2024-09

To provide context for this submission: I began interacting with network equipment in 2001. I first wrote software that automated those interactions in approximately 2003. My career since has involved operating networks that range from being completely based on physical devices that are purchased from vendors who are well represented in the IETF, to those that are entirely realised in software running on Linux machines, to hybrids of the two.. The last 10+ years of my career has been focused on management of network systems. I have been the technical lead of the OpenConfig project – which has defined a number of technologies that have become widely adopted as network management interfaces, as well as aimed to take advantage of IETF technologies where possible.

The opinions that are expressed in this document are my personal opinions, and do not reflect those of my employers over the last two decades of being involved in building network systems.

Fundamentally, the problem with network management and the IETF today is that of recognising both what operators are aiming to achieve with more programmatic interaction with the network, as well as recognising the ecosystem and lifecycle expectations of software. For the IETF to be relevant to network operators it must embrace both ideas *and* do so in a way that allows operators that are interested in doing so to participate in an open process that makes material improvements to their infrastructure.

The need for iteration

Through the OpenConfig working group's experience of developing gNMI – which essentially was the cure to SNMP scalability problems for the networks in which I have worked – we found it was not sufficient to define a protocol in isolation. It needed focus on the implementation. Many lessons are learnt when translating a protocol from prose to code. Many more are learnt when deploying it within lab environments. A final set of lessons come when it is deployed at scale, especially within global contexts – where the realities of varied networks come into focus. It is impossible to know at design time what the initial standard needs to cover.

Going forward, it is my expectation that the technologies adopted by operators will be those that can be iterated on – such that learnings from each deployment stage can be addressed. There must be a closed loop between the developers of servers and clients – where implementation complexities and considerations in both layers be considered and design discussions can be held.

The standards ecosystem is not well suited to these iterations today. Many operators use open source components as part of their solutions, where the nature of writing drafts, and working through a long standards process to realise some solution is simply overhead, and simply too slow to develop usable implementations.

The IETF has two options as to how to engage in this environment:

- Move away from an approach where there is a goal of reaching a standard that can be carved into the stone tablet of an RFC. Rather, approach standards as a specification that can continually evolve, quickly and incrementally, in a way that has low-overhead and is accessible to developers whether they are working on open source software as a volunteer individual in their spare time, or whether they are employed by an equipment vendor who will fund their travel to meetings and cover their time to engage in the standards process. This represents a fundamental change to the way the network management areas of the IETF have been working.
- Continue to engage as a custodian of technologies that need clear interoperable standards, but do so in a way that allows for organisations to bring technologies that have already been developed for standardisation. Google brought two open source technologies – gNMI (draft-openconfig-rtgwg-gnmi-spec-01) and gRPC (draft-kumar-rtgwg-grpc-protocol-00) – to the IETF as technologies that existed and could be adopted by the IETF as interoperable standards. The high overhead process, resistance to externally-defined technologies, and complex standardisation process meant that these efforts detracted from the need to keep developing and adopting these technologies. For both IETF standardisation was abandoned. Even if they had been adopted, the ability to *change* these technologies – i.e., iterate on this initial standard – would have been lost. An approach that aims to have IETF act as the home of "stable" network management standards needs to continue to allow iteration outside of the core RFC standardisation process.

It is my firm belief that there is a need for open standards in network management – but the processes that exist today are not conducive to the participants that the IETF needs to hear from.

The need to ship

The standardisation process in the IETF is too slow – especially when compared to the timelines to meet business needs of today's networks (e.g., build new capacity, improve reliability etc.) . From proposing an initial draft to submitting it for publication is almost always measured in years. This velocity simply doesn't work for network management – especially when competing against approaches where once the code is committed, it can be deployed in production. Operators sending engineers to engage in the standards process very rarely (in my experience) see that standards engagement as an outcome in and of itself. The engagement in the standards process needs to bear fruit. Standards need to be shipped and implemented!

An example: OpenConfig – an informal working group between operators – was initiated partially through contacts made at the IETF. OpenConfig engaged significantly with the `netmod` working group on data models, and the use of the YANG language.

This engagement was heavy on time spent gardening, and low on the resulting yield of fruit. Resistance to change existing preconceptions, iterate on published outcomes, or consider partial solutions that could ship *now* rather than requiring large-scale rearchitecture meant that participants in OpenConfig considered that engagement with the IETF was not a good use of their time. Essentially – it was unclear that the engagement with the IETF would ever result in something that was shipped. Direct engagement with other operators, and following this, with system implementors results in shipped code much more reliably.

OpenConfig is not the only example of where slow standardisation and evolution has driven divisions in the industry. Papers such as 4D, and the move towards control and data-plane splits (e.g., OpenFlow) arguably were driven by the lack of evolution in the router software layer. Of course, one cannot lay the blame of this wider evolution solely at the feet of the IETF, since equipment vendors also needed to implement and ship these technologies – but incremental progress was not made in the IETF to address these concerns.

A split of operators and the IETF makes an already difficult situation worse. The IETF wants to develop relevant solutions to operators (as demonstrated both by the 2002 and the 2024 IAB workshops). Operators not engaging in the IETF due to mismatches in expectations means this input is lost.

To keep operators engaged, the IETF needs a velocity and approach to standardisation that allows for business goals to be incrementally realised.

The need for open source

Unlike other router-level engagements in the IETF – almost every interaction between client and server code that is used for network management is one that involves code that is not from a router vendor. Many client implementations are open source projects – some of which are funded by organisations that participate in the IETF, but many of which are not. Equally important is the fact that there are many different client implementations for these technologies – since they must integrate into different network management systems. In order for a technology to become usable for network operators – especially those with little or no software development expertise in their environments – we need to encourage open source programs that operators can build on top of.

Dave Ward observed this, and both presented at IETF as well as published blog posts around this requirement. Whilst the YANGCATALOG is his cited example of somewhere that brought together different SDOs and open source – from my perspective, there is little relevance of the IETF to open source developments of network management technologies at the router layer. In few places is there a bias towards enabling implementations. There seems to be little outreach from IETF to already adopted open source projects to determine what the problem that is being solved is, and this might be encouraged, if not adopted, by the IETF.

To be relevant to operators for network management – the IETF needs to ensure that it is engaging with open source in a way that unites developers towards common solutions, but leaves room for innovation, and values running solutions.

Equally, where the IETF is the instigator of new technologies, then it needs to invest in open, runnable code. Adoption of technologies by operators is simply not going to happen without there being some driving force behind making tooling available for those operators to adopt. A small subset of operators will invest in tooling to implement an ecosystem around an IETF standard, but time has proven that this will not be the majority.

The need for systems thinking

Business outcomes are what are realised by network management software. To achieve some business outcome, one needs to have a system design that can cater for the requirements – and fit technologies that exist into this design, or develop new ones where they do not. Many of the “Operator Requirements” in RFC3535 are those that require a system to be considered – for example, being able to focus on the configuration of the entire network requires a system that does not exist in a router-centric view of the network – i.e., some orchestration system. The individual components of this system cannot be designed in isolation – the requirements of the system place requirements on each layer.

The IETF has an allergy to architecture in the network management area. There is little or no publications or standards consideration for how different IETF technologies might fit together. This places constraints on the relevance of the technologies that are developed in the IETF.

For example, if one considers the “network-wide configuration” requirement of the 2002 workshop – thinking has converged towards needing a source-of-truth (SoT) for the network. An architecture that considers the existence of such a SoT has to consider what the authoritative source of truth for the network is – is it solely some software component running off-device, is the SoT backpopulated from the network itself, or is it some hybrid of the two? This question has significant implications when considering what the technologies needed for network management are. For example, is the interface between the NMS and network element solely a declarative interface, or does it need to have some handling for merging of different sources of truth based on their origin?

In order for the IETF to build solutions that are relevant to network management – reference architectures are likely to be a key component. A new operator must be able to understand how the technologies that the IETF defines fit together, and ultimately be able to map this architecture to how it achieves their business objectives.

The need for reuse

Historically, networking software running on routers has been considered to be a “special breed” of solutions. This has typically led to there being snowflake technologies defined that are specialised for network management. In numerous cases (YANG is a good example) the IETF has defined its own technology that specifically tries to cater for something in a “networking-specific way”. This problem hurts adoption of networking standards. Significant ecosystem development is required to make these technologies usable (we need networking-specific development, which doesn’t realise a business goal for an operator). It is also harder to hire software developers that have knowledge of these technologies vs. those that are commonly used in other parts of the industry.

Underpinning the arguments above – the IETF and future development of network management must consider how it reuses components that do not need to be specific to networking. The focus of development should be on ensuring that the cycles that are spent by participants (operators and vendors alike) focuses on defining parts of the problem that are truly networking specific, rather than re-defining technologies that could be reused from elsewhere.

In some cases, these technologies might be open source (as is argued in this document), but equally, could be from elsewhere in the IETF. Building vertically-integrated solutions that solve only a subset of the problem space of running a network results in complexity, as well as slow velocity.

What could the IAB do to influence change?

This submission aims to outline a set of problems with the current process that are beyond those that can be easily chartered into the IETF through existing or new working groups. Rather, the intent is to encourage discussion of the wider properties of how to ensure the continued evolution of network management in a way that avoids further fragmentation across the industry.

If it is the goal of the IETF to continue to be relevant in this area. IAB focus on the way of standardising software in a way that meets the needs of implementors and its consumers would be a good first step. Defining a process that is likely to meet the goals that are defined above seems critical.

Further to this, the IAB could encourage participation in the IETF to discuss the architecture of network management systems, to determine where there is commonality that can be exploited and used to build reusable standards and software across the industry.

Neither of these suggestions will create overnight change – but they may provide stepping stones to begin to unify different areas of industry activity, and reestablish the relevance of the IETF in this domain.