RFC3535, 20 Years Later from an Operator's Perspective (Deutsche Telekom)

Kristian Larsson k@centor.se
Kris Lambrechts kris@netedge.be
Ian Farrer ian.farrer@telekom.de

In 2002, the IAB organized a workshop on Network Management and the outcome was RFC3535. It's now been 20 years and it is time to evaluate what has been achieved as well as to look to the future.

1 We, the Authors

This document reflects the experience and opinions of the authors, who have been active in a development group for more than a decade at the forefront of using NETCONF / YANG and related technology in operating networks and building automation around these technologies. We pioneered the development of an automation system based on NETCONF and YANG and have early on been pushing the industry, and equipment vendors in particular, as well as performed interoperability testing and validation for early device software. We have built automation of configuration management and operational state retrieval for the largest router vendors on the market and used this to operate a nation-wide IP & optical network. We have actively worked with, and largely sponsored, the leading free and open source (FOSS) NETCONF / YANG server and datastore; Netopeer2 & Sysrepo (https://www.sysrepo.org/). We are using it to power NETCONF / YANG enabled management of well established areas as home gateways, custom AFTR software as well as more unusual scenarios like that of server workloads, e.g., DNS, DHCP, NTP, and Docker container management that are traditionally managed with an entirely different technology stack. It works great!

2 Retrospective on the 2002 Workshop & RFC3535

The workshop's recommendation to produce an XML-based configuration transport system, that become NETCONF, with a new single data definition language, i.e., YANG, was a resounding success. The natural continuation with NMDA, RESTCONF, and CORECONF has continued to deliver excellent, useful technology that unlocks new use cases for the industry.

More important than the development of any individual protocol (NETCONF / RESTCONF / CORECONF) and more important than any particular serialization (XML / JSON / CBOR) are the semantics and high level concepts that are specified through NETCONF, YANG, and NMDA. These can be described as model-driven declarative configuration that in turn have enabled a style of operations in management systems previously not possible. Going well beyond the original aspiration in RFC3535 of efficient bulk fetching, the industry, and the IETF has specified a whole new paradigm of retrieving data via streaming telemetry. The final architecture we now have is the apex of network device management and we dare say, on an conceptual level, has no real weaknesses. We can refine and optimize details but it is within the conceptual foundation we have laid down. More on the future of network automation systems later.

2.1 In Summary, Things that Went Well

- Promoting YANG and placing NMDA cross-protocols semantics front and centre
  - This has allowed YANG model-driven transport protocols from external parties, like gNMI. This speaks to the strong semantic foundation of YANG & NMDA.
- While YANG is certainly not without its warts, from a network engineering / operator perspective, YANG is a very very good fit. A few highlights:
  - Simple syntax with fairly straight forward semantics that people who are new to schemas and data modeling can easily learn
  - The clear separation of configuration and operational state!
    - And that of actions and notifications...
  - YANG's support for nested lists and compound keys is probably one of the defining features that make it very natural to express

data commonly found on devices. In contrast, the use of artificial indices in SNMP is the antithesis of how things appear in nature and introduces an "impedance mismatch" that makes it very difficult to model things in an elegant manner.

  - YANG has hit, by and large, the right level of accuracy in constraints and expressiveness of the data model, for example: default values are static which is simpler, but cannot express all cases.
  - The type of extensibility offered by augmentations fits very well to how IETF defines protocols and data models. It also allows vendors and others to augment in extra configuration. Most data modeling languages do not have such a provision, or implement extensibility in a much less useful manner.

- The right choice was made with regards to security / authentication / encryption. NETCONF requires SSH transport, with optional TLS support for security. RESTCONF relies on TLS. It was a good choice to layer on these established protocols, and the end result is much better than the situation with SNMP. SSH is a good default for NETCONF, sharing the same PKI setup used for interactive CLI sessions (typically already in place in networks).

- Beyond RFC3535, the industry, IETF and other SDOs (OpenConfig primarily) moved beyond RFC3535 and have specified semantics and protocols for streaming telemetry. A resounding success which gives an optimal event-driven reactive orchestration model.

- Eliminating SNMP was deemed "not an option". This proved to be wrong and the elimination of SNMP is now not just an option, but quite desirable due to the availability of multiple YANG-based transports doing a much better job.

- From RFC3535: `It would be nice to have a single data definition language for all programmatic interfaces (in case there happen to be multiple programmatic interfaces).`

  - Leading vendors now use the same YANG model to drive programmatic APIs as well as CLI!

2.2 Things that Went a Little Less Well

- The workshop observed that the implementation costs have to be low both on devices and managers. The outcome was something entirely different: The data constraints and transactional semantics of NETCONF & YANG meant they couldn't be implemented just by wrapping a thin veneer on top of an existing device's management system. Instead, existing solutions had to be reworked from scratch. Ultimately, this was a good thing and all large router vendors now have good transaction support enabling much more robust management than previously possible. However, this was not by any means low cost on the implementation side. Some vendors are still struggling.

- Service level models, like the IETF L3VPN, appear to not have gotten much traction. YANG and its transports are regarded as protocols for managing network devices and not used in higher levels, like the northbound API of a network manager / automation system. Technically, there is very little that prevents YANG from being used for service models and on the contrary, YANG is a better fit to express such APIs, with data constraints and transactional semantics, than many other schema languages.

- Standard device YANG models haven't seen much adoption. While the IETF has defined a number of device YANG models, ranging from system, interfaces to routing protocols. We are now at a point where one can configure a complete router for a basic SP network use case. However, there are basically no network equipment vendors that implement these models. OpenConfig is clearly ahead, with many of the models leading the IETF by years. This is also shown in equipment adoption (together with procurement requirements) where multiple of the large router vendors support some level of OpenConfig models, but no IETF device models.

  - Overall, the stability of the IETF models and the conformance to proper IETF YANG make them superior to OpenConfig whose models suffer from inconsistencies across versions, this in turn leads to operational and interoperability issues. The basic idea of OpenConfig with rapid iteration isn't bad, but it only works well when all elements are under the control of the same organization. When the device is produced by a vendor, the development iteration is lengthened to multiple months (at best) and all of a sudden,

the rapid iteration isn't very rapid. Instead, getting it right becomes a bit more important than just iterating fast. However, the IETF process would be improved by iterating model revisions faster.

- Nonetheless, thanks to YANG semantics, these models are much nicer to work with than any SNMP MIBs ever were.
- The use of off-box translation of IETF models could provide an alternative implementation approach that doesn't directly rely on network device vendors, making both implementation iterations shorter as well as providing a direct path to production.

3 Future of Network Management

At this point, new additions and changes to the YANG modeling language need to be very carefully considered. It is a stable and mature language that has been proven to work well in practice. We don't want to go down the path of scope and feature creep with an ever larger surface area of the language. In particular with backwards incompatible additions in the language that would require a new version of the language, we should be very careful.

3.1 On Network Managers

While device support for YANG and its model-driven transports has made significant progress, we haven't seen similar advancements in the capabilities of Network Managers.

Building automation for large and complex networks, as most SP networks are, is in essence about the creation and maintenance of layers of abstractions. The declarative model-driven nature of NETCONF & YANG, and its associated semantics through NMDA, allow for the creation of complete abstractions with an unprecedented simplicity. All of this is at a radically lower cost than any other previous solution. Utilizing this, we can build automation systems that allow us to focus on raising the level of abstraction, in turn unlocking new orchestration possibilities.

However, the lack of choice in commercial, off-the-shelf systems that implement such declarative methodology is seriously affecting the uptake of the protocol, due to the risk of vendor lock-in. There are even fewer credible open-source alternatives. Collaboration on open-source software, like Orchestron (`https://github.com/orchestron-orchestrator/`), could significantly help here.

Some networks still operate with open source or commercial tooling that fails to use a YANG model driven programmatic API. Instead they rely on CLI screen scraping with all its known failure modes. Systems that do speak NETCONF to devices, but where the generation of configuration is done by text rendering are equally fragile. Instead we want to fully exploit YANG, where we can validate code and templates at compile time instead of finding that a text based config template resulted in incorrect data at run time. This is exactly the kind of "shift left" tactics we need in order to realize robust network automation, a must if we want to advance beyond basic device management and orchestrate networks through higher level service abstractions.

Many managers are built on the principle of workflow execution, in which there is a basic "impedance mismatch" between ordering workflow execution at the top layer, and delivering declarative configuration changes to devices southbound. Workflows are a poor choice for abstraction, e.g., once a "create" workflow has been run, the abstracted view provided by the workflow disappears, the output is the low-level detail of the resulting device configuration. We are 'chained' to the lowest abstraction level (the device configuration) and real abstraction into service layers (e.g., RFS, CFS) is not possible.

Furthermore, we believe the convergence of configuration management and the collection and monitoring of operational state is essential in the Network Manager of the future. In the vast majority of SP networks, configuration management and the collection of statistics / telemetry data exists as two separate silos in both the organizational chart and technology stacks. This is a legacy which has no reason to exist in this day and age. To build complete abstractions we must consider both configuration and operational state. The IETF could do more to help bridge the gap by providing standardized operational monitoring models to match the configuration models; at both the device level and the network/service level. For example, the L3VPN SM has no state at all, so how do I know my VPN is working after I ordered it?

3.2 More Service Models

With the L3VPN SM and later additions, the IETF has raised the bar for what the interface between the Network Manager and the business should look like. A standardized model for a Layer 3 VPN product across business units (and potentially across operators) is a great feat.

- The implied use of YANG model-driven transport at this layer is just

as important. YANG is not just for devices, we should use YANG on the northbound of NMS / OSS!

- Promote the idea of IETF service models on the northbound of NMS / OSS / BSS.

The IETF should do more to promote its service models as a base line offering in the service catalogs of other SDOs (TMForum, MEF LSO, ...) by researching and standardizing interoperability models, specifying data conversion and/or embeddings, publishing implementation guidelines etc. Some networks are adopting TMF640/641 as models for service ordering in BSS, but how these interfaces can be interfaced with L3VPN SM for configuration is not specified; the TMF spec has a focus on service ordering metadata, like order creation time, order status, planned ready-for-service dates, etc., whereas the core of the service configuration itself is left for the implementer to fill in; the IETF L3VPN SM model on the other hand defines the L3VPN service config itself and has almost no service order related metadata. The combination of the relevant TMF specfications and IETF YANG service models would offer service providers a comprehensive and powerful solution.

3.3 The Future of Standardized Device YANG Models

The IETF now has a decent set of standardized YANG models for the configuration and management of routers. However, there are few, if any, implementations that support the lion's share of these YANG models and it remains to be seen if this will change in the future. As with SNMP MIBs, the incentives for implementing these on devices are not aligned between device vendors and SPs.

One interesting possibility would be using the Network Manager to perform the translation from standard IETF device YANG models to proprietary device YANG models. Such translation logic could benefit all users of a platform, such that the open sourcing and sharing such initiatives could become popular(even where incentives are aligned quite differently) Off-box translation can be iterated very quickly, reduces risk, has fewer requirements on the operational network, and delivers immediate value to any user.

3.4 Recommendations

A considerable amount of IETF time has been spent on the development and standardization of YANG modules. In order to guide future decisions,

it would be good to gauge current adoption and planned use of these technologies by SPs. E.g.,:

- Service level YANG models on managers, e.g., IETF L3VPN service model
- IETF YANG device/protocol models
  - Router related models, e.g., ietf-interfaces, isis, bgp
  - Network service models, e.g., DHCPv6 client/relay/server