NETCONF : Ready for Primetime or Work in Progress ?

Author: David French

January 2009

Abstract

This paper highlights some of the factors to be considered when adding Netconf-style management functionality to a network device roadmap. The paper discusses the background and current status of Netconf technology as well as serious issues concerning its limitations and ability to deliver on the claims being made. The conclusion lists a series of recommendations for network equipment providers.

Silicon & Software Systems

Email: info@s3group.com

Web: http://www.s3group.com/

Version: 1.0

Silicon & Software Systems

Table of Contents

BSS	Business Support System
COTS	Commercial Off The Shelf Software
DML	Device Modelling Language
EMS	Element Management System
HTML	Hypertet Markup Language
IETF	Internet Engineering Task Force
IAB	Internet Architecture Board
IRTF	Internet Research Task Force
MIB	Management Information Base
NMS	Network Management System
OSPF	Open Shortest Path First
OSS	Operations Support System
S3	Silicon & Software Systems
SMI	Structured Management Information
SSH	Secure SHell
UML	Unified Modelling Language
XML	eXtensible Markup Language
XSD	XML Schema Description

1 Introduction : Device Management Background and Challenges

Networked devices are becoming more numerous, more varied and more complex. Existing device classes such as routers are increasingly gaining sophisticated functionality such as flow-state routing. New classes of device functionality such as session border controllers and media gateways are also emerging. Previous distinctions that existed between telecom and datacom are dissolving in a tide of internet protocol packets. Moreover the pace of this change is accelerating with new service classes, applications and availability requirements impacting the plans of network equipment providers and the expectations of their customers, the network operators.

Device management needs to evolve to deal with these developments. Previous approaches to device management now face severe scaling, performance, interoperability and functional limitations.

Failure of networked devices can have extreme consequences. For example, on August 28th, 2002 AT&T suffered an almost three hour outage in their Chicago network which had nationwide implications. The problem was traced to manual misconfiguration of the OSPF routing stack in a backbone router. [1]

Optimizing the use of existing network assets is vital to extracting maximum value from the high levels of capital expenditure present in existing networks. Potentially even more serious than individual outages due to mis-configuration is the risk of wasting expensive network capacity due to deficiencies in existing network provisioning and management approaches. Reconfiguring networks to meet rapidly changing demands is currently far too difficult, too expensive and too error-prone.

2 Industry Response

The Internet Engineering Task Force (IETF) has recognized the challenges facing device management and is active in the area of operations and maintenance standardization to help network operators manage their networks.

As part of this activity and in recognition of problems in current approaches the Internet Architecture Board (IAB) held a workshop in June 2002 attended by representatives of industry leaders and world experts on existing management technologies. Discussions focused on the challenges faced by network operators and specifically on the limitations of the Simple Network Management Protocol (SNMP), whose earliest RFC dates from 1988. A report was given to the IETF and is contained in the informational memo entitled RFC3535 [2].

A large number of the observations recorded concern the lack of standard SNMP MIBs, their unsuitability for configuration purposes, the high costs of standardization and the high costs of implementation on devices and managers. In summary, while SNMP is widely deployed and very useful it was judged not to have met the needs of device management and also that some of the failure can be traced to deficiencies in the MIBs. The use of proprietary CLI formats for configuration and the lack of a common data model were also identified as serious barriers to interoperability.

Reflecting work already done by network equipment vendors in the late 90's the report also contained observations on the use of XML for network management, pointed out that XML has a robust schema language called XSD and also that there is a lack of commonly accepted management-specific XML schemas.

Among the eight recommendations contained in Section 6 of RFC 3535 :

5. The workshop recommends, with strong consensus from the operators and rough consensus from the protocol developers, that the IETF/IRTF should spend resources on the development and standardization of XML-based device configuration and management technologies (such as common XML configuration schemas, exchange protocols and so on).

XML-based management interfaces already existed at this time. The prime example was the XML API provided by JUNOS® from Juniper Networks. A 2001 white paper from Juniper Networks entitled “XML based Network Management” [7] describes how this works and what it can offer.

As an existing system, Juniper’s XML-based management interface was shared with the internet community and forms the basis for the standardization efforts being carried out in response to recommendation 5 of RFC 3535. The standardization efforts take place under the auspices of the IETF in the Netconf working group. Work began in 2003 and follows the Netconf charter [3].

3 Current Netconf Standards

Since 2003 the Netconf Standards Working group has published two main standards. These are RFC 4741 in December 2006 (95 pages) and RFC 5277 (35 pages including detailed but non-normative XML examples) from July 2008. RFC 5277 deals with the optional notifications feature in Netconf.

There are also three lesser standards which provide details on the use of underlying transport protocols in a Netconf system. These are RFC 4742, 4743, 4744 and concern SSH (10 pages), SOAP (20 pages) and BEEP (10 pages). A technical errata document on RFC 4741 has also been published. Of these three only SSH is required in a Netconf system. Using SSH as a transport layer is a standard approach to securing otherwise insecure protocols.

RFC 4741 [6] is the main document and defines an essential building block of Netconf. Starting from an initial Juniper-edited draft in August 2003, RFC 4741 went through 12 drafts before release in December 2006. Following further discussions and closer examination a revision (entitled RFC4741-bis) is now being planned to add clarifications and amendments.

RFC 4741 builds on the potential of the approach outlined in the earlier Juniper White Paper on XML Network Management. The standard lists required and optional capabilities, defines XML wrappers, handshaking and capability discovery and transport protocol needs.

RFC 4741 is also explicit that the content layer is outside of its scope:

Section 1.1

4. The content layer is outside the scope of this document. Given the current proprietary nature of the configuration data being manipulated, the specification of this content depends on the NETCONF implementation. It is expected that a separate effort to specify a standard data definition language and standard content will be undertaken.

The same scope limitation applies to RFC 5277 through its reliance on a non-normative “fictional” schema.

4 Ongoing standardization work

In addition to the published standards, work continues on two major drafts. These are Netconf monitoring schema (currently at draft 03, 44 pages including a large amount of non-normative content) and a content definition or model format (currently at draft 02, 155 pages). The content definition format proposal is provisionally scheduled to be available in late 2009. [8]

Both of these are concerned with the content layer definition which is outside the scope of RFC 4741. Several competing technologies could be used for content layer definition. The XML standards RelaxNG and XSD are possible but have proven unwieldy in different degrees. Two proprietary solutions are also being considered for adaptation into the standards.

As already mentioned the central document RFC 4741 is being revised. This will result in RFC4741-bis provisionally scheduled for Q2, 2009.

At a higher level, the Netconf charter itself is being revised following IETF 73 to accommodate an incremental development phase. Several additional items such as notification content, access control and Netconf access to SMI-based MIB data have been omitted from the new scope and postponed until there is a community consensus to take them on.

As usual within the standards process various proprietary solutions are being promoted and the winners are not necessarily the technically superior options. There are commercial reasons to use standards to protect existing investments in XML-based management technology by a variety of vendors. The process of creating standards sometimes should not be too closely scrutinized by idealists.

In any case, the current status of work means that Netconf is incomplete as a useful standard and will remain so at least until the end of the decade when a standard content definition format should be available.

5 Netconf Concerns

5.1 XML

It is notable in RFC 3535 that the consensus among network operators for an XML-based management protocol was stronger than among the protocol developers. It is reasonable to assume that protocol developers had greater experience of device capabilities and format limitations. It could also be inferred that XML processing is more amenable to off-device systems.

When XML appeared as a derivative of SGML used for document storage in the late 90s, it was touted as a panacea for every application where data transfer and representation is involved. The initial enthusiasm has since waned to a more realistic appraisal of its uses. Several disadvantages of XML have become apparent. These include a high level of redundancy which affects transmission and processing costs and a more problematic need to force XML encoded data into a hierarchical model.

XML is human readable text as opposed to binary. This is often quoted as a major advantage based on small examples. RFC 3535 notes that XML is verbose. When encoding data this verbosity leads to documents for which humans need dedicated editors which can cope with the large filesizes involved and provide a manageable view on the underlying document. Standard text processing utilities such as UNIX diff are not useful on XML unless a special profile called Canonical XML is used which imposes formatting conventions.

Although often referred to as such, XML itself is not a single standard and actually exists in different flavours (version 1.0 dating from 1998 with a fifth edition dating from November 26th, 2008) and version 1.1 from 2004, updated in 2006.

For these and other reasons XML technology has limited usage within networked devices beyond providing HTML web interfaces and some proprietary protocols. The consequence is that the XML building blocks such as XML parsers, XML databases and even expensive XPath implementations have to be recreated at a carrier-grade quality level and then allocated scarce resources on the device. Unlike language libraries and operating system utilities these costs are not being shared with other parts of the system.

Probably the most serious concern is the one pointed out by RFC3535:

- XML alone is just syntax. XML schemas must be carefully designed to make XML truly useful as a data exchange format.

5.2 Schemas

Standard and proprietary MIBs (Management Information Base) are an essential part of SNMP management protocol usage. Netconf currently lacks MIBs.

SNMP MIBs depend on the SMI (Structure of Managed Information) which defines their format. As Netconf lacks a content layer definition the equivalent of standard SNMP MIBs for Netconf cannot be written.

As mentioned in RFC 3535, XML has a robust schema language called XSD (XML Schema W3C). This is one of several, others are RelaxNG, (REgular LAnguage for XML Next Generation) and DTD (Document Type Definition).

XML-related technology is intent on tackling big issues and consequently takes a necessarily abstract high level approach to the majority of requirements. At least four individual attempts to implement Netconf followed the initial mention of XSD and tried to define the content layer of Netconf messages (the actual data) using this technology. Usability issues were cited as the major barrier to progress and the efforts were abandoned. XSD was not designed for this specific task and imported too much unnecessary functionality while lacking application specific features.

The content layer definition efforts then turned to use of RelaxNG rather than XSD. Despite the perceived advantages of RelaxNG some implementors abandoned their standards-based efforts at this point and instead decided to invent their own in-house modelling languages. The consequence of that decision is that all of their Netconf configuration data had to be expressed in various divergent proprietary formats.

Some standardization work around application specific configuration, such as the proposed RFC 5388 “Information Model and XML Data Model for Traceroute Measurements”, is proceeding using existing XML technology and without creating new formats.

The original aim of commonly accepted standardized management schemas or MIBs cannot be realised until at least the underlying format is standardised or an existing and proven technology such as RelaxNG or XSD is accepted by the network management community. The possibility of multiple schema formats being in use simultaneously defeats the original intentions.

One effort to standardise a format was made in a BOF (Birds Of a Feather) proposal to IETF in late 2007. The IETF perceived this as an effort to create yet another modelling language to add to the very large number of application specific formats and modelling languages already existing. The proposal was rejected and moved to the operations and management area within IETF instead.

Regarding Netconf, the official description of the new working group states that :

“The specifications do not include a modeling language or accompanying rules that can be used to model the management information that is to be configured using NETCONF. This has resulted in inconsistent syntax and interoperability problems. “

Within this working group several proprietary modelling languages have since been proposed, including one called Kalua from Nokia Siemens Networks. This was rejected in the second half of 2008. Deliberations are ongoing and there is hope to reach a consensus and a recommended proposal in late 2009.

RFC 3535 states that

“18. The specification costs for data models have to be low.”

The status of current drafts indicates that the specification costs for real-life devices will not be low.

RFC 3535 also outlines some of the reasons SNMP has not reached its goals and why a replacement is needed. Many of these reasons are traced back to fundamental problems or missing features in the SMI schema language. A lack of commonly accepted MIBs, lags in creating MIBs for new functionality and a lack of configuration information in these MIBs are the consequences. Netconf still lacks a standard schema and needs to significantly improve on the SMI if the protocol is to replace SNMP as the main management interface.

It is unfortunate that in the history of modelling languages, many standards and drafts have had to be abandoned due to their inadequacies despite, in many cases, strong commercial and technical support from the major stakeholders in the industry.

A particular example is the work done in 2001 / 2002 by the IETF working group on SMI NG (Structure of Management Information Next Generation). This was originally meant to provide a successor to the well established SMIv2 within the SNMP world. The starting point was work done by the Internet Research Task Force (IRTF) and the new standard was meant to also incorporate an existing standard called Structure of Policy Provisioning Information (SPPI, RFC 3159). One of the main improvements planned was to provide the benefits of object oriented design to the modelling of network management information [4]. Despite significant industry involvement and the proven, supported nature of the underlying SNMP technology the initiative ultimately failed to produce a successor and was wound down.

One of the fundamental dynamics involved in the definition of a schema language for the device management interface is the interaction with device modelling languages (DMLs). To meet their requirements in supporting implementation of the device, DMLs require more complex structures and richer expression of the device functionality. Unlike the interface schema this should not be exposed outside the device. Good design at any level requires clear separation of interface and implementation and therefore a separation between the interface schema and the DML. The challenge to new interface schema definition initiatives is to avoid trying to model too much yet still provide significant improvements on SNMP.

5.3 Potential for SMI reuse in Netconf ?

Provided standards for Netconf content definition format can be agreed (ie an XML equivalent to SMI but more powerful), work can then begin on the real task of writing the equivalent of SNMP MIBs. The exponentially growing range of device functionality available means this will be a large task for the industry. Unfortunately there are few shortcuts.

Work done at the L3S Research Centre (“Generating skeleton code for netconf modules from smi mib module definitions” [5]) showed that transformation from SMI MIBs is possible but stated:

“It is questionable whether this transformation is valuable on the long run.”

Two of the serious issues with SNMP MIBs pointed out in Chapter 4 of RFC3535 are as follows:

1. It is usually not possible to retrieve complete device configurations via SNMP so that they can be compared with previous configurations or checked for consistency across devices. There is usually only incomplete coverage of device features via the SNMP interface, and there is a lack of differentiation between configuration data and operational state data for many features.

and

17. There is a semantic mismatch between the low-level data-oriented abstraction level of MIB modules and the task-oriented abstraction level desired by network operators. Bridging the gap with tools is in principle possible, but in general it is expensive as it requires some serious development and programming efforts.

It is tempting to think that some sort of automatic translation or conversion of the existing body of standardized and proprietary MIBs will allow the industry to avoid the large task of writing useful content definitions or models. These two observations in the RFC show the flaw in that line of reasoning. If the SNMP MIB does not even address configuration data then it cannot be added in by automatic translation.

Even where the configuration data is addressed it is usually done at too low a level to be useful. Observation 13 says it best:

13. MIB modules often lack a description of how the various objects can be used to achieve certain management functions. (MIB modules can often be characterized as a list of ingredients without a recipe.)

Automated translation of SNMP MIBs provides a useful starting point for some devices in some cases but no more. Crucial information needed for configuration, not to mention validation, is missing.

The network industry has a large task ahead if it is to provide a standard description of all the devices it wants to manage in a language or format which does not yet exist.

5.4 Netconf Transactions

Transaction-based configuration is the holy grail of networked device management. Being able to treat a configuration change as a transaction has the major benefit of assisting the operator avoid badly configured devices. Bad configurations are simply rejected and the device rolls back to the previous state.

Abstracting even further, a network-wide transaction can ensure that a configuration change to a group of devices either proceeds as a whole or does not proceed at all, with all devices automatically rolling back to the previous state if there are any failures. Such functionality is essential when the network itself is seen as a single configurable entity.

From a network manager's point of view this is ideal. Provided bad configurations can be recognized as such, either before sending or at least before they do any irreversible damage on the device, then this guarantees a dramatically higher level of protection against the problems of misconfiguration than the present approaches.

Leaving aside for now the ability to recognise bad configurations, the key point to appreciate is that the transactions mentioned in transaction-based configuration are all distributed. Distributed transactions require two things, a transaction manager and transactional resources.

Distributed transactions are a very complex algorithmic area. Blocking protocols such as two phase commit and non-blocking such as three phase commit can be used between the transactional resources and the transaction manager. Transactional resources in real systems are almost always advanced relational database management systems.

At a network device level the actual resources include hardware chassis, line cards, operating systems and software stacks. At the higher network level the resources are the network devices themselves. Obviously real life items such as hardware cannot roll back to a previous state and irreversible failures can occur.

More seriously, the software running on current networked devices (ie protocol stacks, operating systems and third party applications) has not been designed to be transactional and therefore cannot function as a transactional resource. Even if it contains only one non-transactional configurable resource the device itself cannot be described as transactional.

Using a total restart of an application, or a device, with a clean and known configuration as the ultimate rollback approach is not an option for real networks.

The end result is that, to provide real transaction-based configuration across a network, a ground up redesign and rewrite of all software involved has to be considered. This is the real reason that distributed transaction implementations exist almost exclusively inside advanced non-realtime software environments like J2EE and Microsoft Transaction Server rather than on more constrained networked devices.

Given that a large proportion of the configurable software on networked devices has been hardened through years of use and that the functionality is often standards-based (ie OSPF) the likelihood of a redesign to accommodate new requirements is effectively zero. A Netconf manager would have been of limited utility in preventing the AT&T scenario mentioned at the beginning unless the OSPF stack was rewritten to be transactional or the expense of restarting it from scratch with correct parameters could be borne.

The way in which Netconf introduces transactions is also questionable. Existing transactional systems, mainly databases, do not explicitly name intermediate states such as candidate configurations. Transactional systems also comprehensively manage the need for isolation across different sessions. Netconf appears to ignore this difficult question leaving the way open for deficient or incompatible solutions.

The Netconf protocol is very frequently mis-interpreted as providing transactional configuration of both networked devices and whole networks and this is a very frequently touted benefit of the new technology. Unfortunately delivering real distributed transaction functionality is much harder than writing a protocol which can declare that a device possesses this desired property.

Even if the software handling the Netconf protocol correctly implements a transaction manager the underlying resources still need to be transactional. Currently they are not. The Netconf protocol introduces the powerful concept of transactions to the network management vocabulary but does little to actually implement it.

This implied need for transactional resources can be seen as part of the fine print of Netconf.

5.5 Netconf Candidate Capability

The Netconf base protocol defines various basic operations such as get, (<get>) and edit config (<edit-config>). In addition to these there is a range of capabilities which devices can choose to support. Different capabilities modify the definitions of the basic operations and add new operations.

A notable capability is the optional candidate configuration capability which meets the RFC 3535 recommendation that devices should be able to store configurations.

Candidate configuration is meant to act as a store for configuration changes which can be made in advance and then committed in one transaction to the running configuration. A new commit operation is provided for this. To implement the candidate configuration capability the device must be able to explicitly guarantee that the candidate configuration can be implemented either completely or not at all. The candidate configuration spans the entire device and all configurable applications under management. In practice and using multiple real applications it is impossible to give that guarantee without at least a major rewrite to add transactional functionality.

Lock and unlock operations are provided in the Netconf protocol to avoid other sessions interfering while candidate configurations are being changed. The use of these lock and unlock operations is not mandatory but prudent. Because lock and unlock cover the entire configuration different parts of the candidate cannot be modified independently. A newly proposed standard entitled partial lock is now being drafted to deal with this deficiency.

Implementation of the lock and unlock operations in the protocol requires the use of system resources to restrict access. The unlock operation additionally requires that storage be able to rollback the candidate configuration to the pre-locked state but based on subsequent working group discussions this may also require an implicit copy from the running configuration. This is potentially contradictory and clarification to the standard is required here.

These are non-trivial requirements and they block real implementation of the vital candidate configuration capability.

With the applications currently deployed on network devices there can be no guarantees of a transactional commit and therefore candidate configuration cannot be offered as a capability. Without candidate configuration capability the device has to rely on direct configuration over XML which, while a significant improvement over alternative options such as SNMP, is a long way from the promised potential of Netconf.

It is also doubtful that the technical improvement over widely deployed SNMP would justify the effort involved.

If any devices advertise this candidate configuration capability yet then proceed on the traditional best-effort basis required by use of existing software stacks then the contract is broken. The promise of Netconf to deliver real value in this area is then destroyed with major risk implications for network operators.

5.6 Minimal Compliance

RFC 4741 defines a minimum mandatory set of capabilities for Netconf agents. Effectively the protocol requires a level of discovery in the handshaking, provides an XML wrapper for use over SSH and leaves the real functional gains over previous management protocols to capabilities which are completely optional.

At least one major vendor has released Netconf - enabled devices which use Netconf to merely wrap the existing CLI commands. This does not add significant value to device management.

Past experience indicates that acceptance of this next generation protocol by device vendors will be limited to minimal compliance for a long time. The costlier parts of this protocol which are the ones that should deliver the significant benefits will not appear for some time, if at all.

For previous generations of network management protocols the existence of devices in a heterogeneous network (ie most networks in question) which only achieve minimal compliance would not be an issue. SNMPv1 devices can exist (albeit insecurely unless receiving additional assistance) within networks which also contain SNMP v2c and SNMP v3 devices without drastically affecting the overall network. Unfortunately the same is not true for Netconf. The presence of a single minimal compliance device brings down the functionality which the network can offer to the lowest common denominator.

For the foreseeable future, ie until all current devices disappear from use, network operators will have to deal with traditional non-transactional best-effort devices which at best offer a minimally compliant Netconf subset of management operations.

This eliminates the Netconf promise of managing the entire network as a single resource from a configuration point of view.

The desirable situation described in RFC 3535; “It is necessary for operators to concentrate on the configuration of the network as a whole.” , may require new networks.

5.7 EMS / NMS /OSS / BSS Support

RFC 3535 noted the consensus among network operators on the need for standardized XML-based network management technologies. Netconf is an XML-based protocol defined for management of networked devices and as such has to be used by a manager to manage a device.

To provide value, Netconf management needs to be harnessed to the needs of EMS/NMS/OSS/BSS (Element Management Systems / Network Management Systems / Operations Support Systems / Business Support Systems) systems in order to implement the requirements of the network operator in fault management, network provisioning, inventory checking, configuration, access policy and other network device management areas. This is currently a missing piece of the puzzle, managers (ie EMS/NMS/BSS/OSS systems) relying purely on Netconf standards do not exist and cannot exist until the Netconf standard is a complete system.

Various mixes of proprietary technologies and Netconf do exist but are not sufficiently widely deployed. Even among these small minority of network management systems most use an absolutely minimally compliant set of Netconf capabilities which are essentially only “thinly wrapped” XML.

Using the real power of Netconf means the managing system has to be able to make use of optional capabilities and extensions. These can come in many different combinations and custom capabilities are an additional complicating factor. Adding real Netconf power to these systems will require significant effort.

Far more seriously, even if managers offering the current Netconf standards were made available they would be of little use. Without knowing the model at design time managers cannot do much to hide the underlying data structure from the operators. What operators need is a task-oriented model of the devices under management. Providing this first requires an accepted and sufficiently powerful model format and that is not yet available.

Until management systems routinely incorporate Netconf protocol handling to the extent that vendors can prove wide interoperability, with the accompanying certification requirements, then the real business value promised by Netconf will be extremely limited.

5.8 Burdens and Validation

Netconf promises to solve many very difficult problems at the network management layer. Partly it does this through shifting or delegating the burden down from the manager layer onto the managed networked devices.

One particular burden is validation of configurations. This optional capability requires the device to be able to validate incoming configuration data. To provide this capability the device should be able to check for syntax errors. Given that the data is being received in XML written according to some schema which is assumed to be available, then a validating parser, although expensive in terms of code size and processing resources, can be used.

More usefully the device can check for semantic errors. Within RFC 4741 the possibility to validate for non syntax errors is passed over quickly (the word semantic appears once in the capability definition). In practice, properly checking a configuration for semantic errors is non-trivial and requires the checker to have intimate knowledge of the complex relationships between the data or alternatively to delegate this to a new class of underlying applications which can validate a device-wide configuration.

In the AT&T example earlier, a useful validation would have had to recognize that while syntactically and also semantically correct, shutting down the broadband access across a metro area was unlikely to be the intended meaning of the configuration change. Encoding such intelligence is non-trivial. Computers do not respond well to the “do what I mean, not what I say” method of communication.

As syntactical correctness can be nearly assumed given the use of valid XML documents and as there are no other strict requirements most Netconf enabled devices will take the opportunity to advertise a validate capability without doing anything much and without contributing any value to the network management process.

5.9 Protocol-Centric Management

Despite all the issues with SNMP, RFC 3535 states in Chapter 5:

“10. Eliminating SNMP altogether is not an option.”

Given the possibility of stacking the Netconf protocol it is a tempting design choice to attempt to use Netconf as a backbone technology in a protocol centric management layer embedded within the device and to somehow derive SNMP, Command Line Interface (CLI), HTTP and future or custom management interfaces from this.

Partly because SNMP is also ”stackable” the same design choice was occasionally taken in the past but instead using SNMP as a backbone technology. This severely limited the functionality of the derived interfaces and necessitated complex workarounds and blatant hacks. To implement newer features of network management these SNMP-based management systems have to be completely replaced or bypassed at very significant cost.

In particular, as SNMP lacks the concepts and mechanisms for configurations and transactions, it is extremely difficult and usually impossible in practice to implement a non-minimal Netconf system on top of an SNMP-based layer.

Building a unified management layer on a backbone of a single management protocol ignores the experience of the past two decades of device management. Using a very recent incomplete and unproven protocol combined with proprietary and non-standard elements adds further risk.

5.10 Devices as Documents

Netconf approaches the problems of device management from a top down design perspective. Devices under management are seen as black boxes and all state information can be represented as XML documents whose costs of production can be ignored.

Devices in real systems differ greatly, some have little state which is inexpensive to query, others are large and expensive. Netconf deliberately does not distinguish between them. Depending on the implementation, simple operations like <unlock> on a candidate configuration may require a running candidate configuration XML document to be created and copied. This can have significant costs depending on the underlying implementations, the majority of which will be legacy for the foreseeable future.

Netconf queries or filters are another example of the difficulties introduced by ignoring the device implementation. Even basic subtree filtering which is part of the base protocol imposes large and unknown costs on a device which either needs to introduce device specific optimisations to the generic XML algorithms or must give the filter processor the power to access arbitrary parts of the device again at potentially large processing cost. Complex managed devices become vulnerable to complex queries whose cost the manager does not know in advance. These are different to the simple and inexpensive queries of SNMP.

On a theoretical level everything is straightforward, the device simply captures all current state in a single XML document and submits it for generic processing either using the basic filter mechanism or using XPath. With real life devices this can be prohibitively expensive and impractical.

In contrast, Observation 16 of RFC 3535 is clear:

“16. The implementation costs have to be low on devices.”

6 Conclusion

Netconf is an extremely ambitious project and holds out the promise of truly configurable networks using the very powerful paradigms of transactional processing. Lessons learned by the industry with previous management protocols, both successful and less successful, are being incorporated.

Some of the XML-related technology employed in the protocol is both new and unproven in the embedded device context but this is not likely to be insurmountable in the medium term. More seriously there are conceptual problems regarding transactional implementation and the “device as document” metaphor which threaten to undermine or render too costly many of the promised benefits.

Netconf is also not a complete answer to the issues raised in RFC3535. Access control and co-existence of multiple management protocols are among the areas not addressed. The ISO Telecommunications Management Network model defines five categories of management tasks listed in the acronym FCAPS (Fault, Configuration, Accounting, Performance and Security). Current Netconf standards are mainly focussed on the configuration category.

Unfortunately the Netconf initiative has only just started and required standards are incomplete. Once the standards are agreed, work can begin on content definition and implementation. These phases can be costly and time consuming. It remains to be seen if the industry will value this enough to justify the investment.

7 Recommendations

7.1 Wait and See

Despite being a leading vendor of embedded Netconf technology with involvement in the standardization process we currently recommend a “wait and see” approach to this new management protocol initiative.

Netconf aims to solve major issues in device management and it is realistic to expect this to take time. Network equipment providers should consider placing Netconf on their long term roadmap and also keep it in mind when making architectural, design and commercial choices.

7.2 If necessary, first aim for minimal compliance

If Netconf functionality is immediately required (ie is on the Market Requirements Document) then implementation should avoid second guessing the proposed standards or over-committing.
Instead aim for the minimal levels of compliance initially.

7.3 Avoid vendor lock-in through proprietary technologies

All non-standard proprietary technology should obviously be avoided. Vendor lock-in and lack of common tools are major risks for network equipment providers in all areas.

In particular any effort spent learning and expressing device functionality in non-standard or proprietary content definition formats is vulnerable to the unpredictable standardization process. This is a major potential waste of resources and is not where network equipment providers should be investing effort.

7.4 Examine claimed benefits closely

Netconf standards combined with proprietary systems can be used by product marketing to provide reasonably convincing small demonstrations and leverage hype around new standardization initiatives.

Network equipment providers should beware of the pitfalls of early adoption and request applicability to their own needs, proven deployment references and close examination of the design choices by vendors. Will the claimed benefits really apply to your device and what are the real costs and risks involved ?

7.5 Focus on future-proof solutions

Due to the high costs of software development and maintenance every effort should be made to avoid importing the limitations of specific management protocol approaches into the device management layer.

Network equipment providers should instead focus on providing future proof solutions.

7.6 Install a powerful management layer

Regardless of standardisation status or direction, the motivations behind current efforts are well-founded. Device management needs to be improved.

In layered network device architectures the embedded management layer should be sufficiently powerful to accommodate current and future variations on Netconf as well as current industry standard management protocols such as SNMP and arbitrary custom management protocol variations.

References

[1] BGPexpert newsletter, 2002-10-15;
http://www.bgpexpert.com/archive2002q4.php
http://www.computerweekly.com/Articles/2002/08/29/189319/att-customers-suffer-broadband-failure.htm

[2] RFC3535
http://www.ietf.org/rfc/rfc3535.txt

[3] Netconf charter
http://www.ietf.org/html.charters/netconf-charter.html
Netconf working group protocol and documents
http://trac.tools.ietf.org/wg/netconf/trac/wiki

[4] SMI NG Charter
http://www.ietf.org/html.charters/OLD/sming-charter.html

[5] Torsten Klie
Generating skeleton code for netconf modules from smi mib module definitions
http://www.ibr.cs.tu-bs.de/papers/klie_ICWI2005.pdf

[6] RFC 4741
http://www.ietf.org/rfc/rfc4741.txt

[7] XML based Network Management
http://www.juniper.net/solutions/literature/white_papers/200017.pdf

[8]
Netmod working group charter (goals and milestones)
http://www.ietf.org/html.charters/netmod-charter.html

About Silicon & Software Systems Ltd. (S3):

S3 is the leading vendor of embedded management software for networked devices to the major telecom and datacom equipment manufacturers worldwide. Founded and headquartered in Dublin, Ireland in 1986, S3 also has operations in the U.S., Poland, Czech Republic and Portugal with representatives globally. For further information please visit www.s3group.com

***embedded*MIND**™ software enables manufacturers of networked equipment, as well as hardware and software component providers, to provide advanced multi-protocol functionality supporting Netconf, command line (CLI), SNMP, XML and Web management interfaces. ***embedded*MIND**™ provides a radical improvement over traditional architectures for management software and delivers major benefits in terms of cutting development cost and risk while improving software maintainability.

For further information please visit www.embeddedmind.com

About the Author:

Before joining S3 as VP Product Management for the ***embedded*MIND**™ product line, David French worked with a range of Tier 1 telecommunications providers including Lucent, Philips Communications Industries and Alcatel in The Netherlands, the United Kingdom and Germany. Since the early 90's he has worked on GSM base stations, SDH/Sonet ROADMs, GMPLS and OSS systems in project management, architectural, development and testing roles. Technologies used include CMIS/CMIP, XML, UNIX and C/C++.

Legal Disclaimer:

This document is for discussion purposes only and does not and shall not constitute or imply any inference, promise or intention to enter into any binding, contractual or other business relationship. S3, Silicon & Software Systems does not make any express or implied representation or warranty as to the accuracy or completeness of the information supplied hereunder. All such information provided is provided 'AS IS'.

Any trademarks contained herein (whether registered or not) and all associated rights are recognized.