Network Working Group
Request for Comments: 3535
Category: Informational

J. Schoenwaelder
International University Bremen

May 2003

Overview of the 2002 IAB Network Management Workshop

Status of this Memo

Copyright Notice

Abstract

This document provides an overview of a workshop held by the Internet Architecture Board (IAB) on Network Management. The workshop was hosted by CNRI in Reston, VA, USA on June 4 thru June 6, 2002. The goal of the workshop was to continue the important dialog started between network operators and protocol developers, and to guide the IETFs focus on future work regarding network management. This report summarizes the discussions and lists the conclusions and recommendations to the Internet Engineering Task Force (IETF) community.

Table of Contents

1. Introduction

The IETF has started several activities in the operations and management area to develop technologies and standards that aim to help network operators manage their networks. The main network management technologies currently being developed within the IETF are:

- The Simple Network Management Protocol (SNMP) [RFC3410] was created in the late 1980s. The initial version (SNMPv1) is widely deployed, while the latest version (SNMPv3), which addresses security requirements, is just beginning to gain significant deployment.

- The Common Information Model (CIM) [CIM], developed by the Distributed Management Task Force (DMTF), has been extended in cooperation with the DMTF to describe high-level policies as rule sets (PCIM) [RFC3060]. Mappings of the CIM policy extensions to LDAP schemas have been defined and work continues to define specific schema extension for QoS and security policies.

- The Common Open Policy Service (COPS) [RFC2748] protocol has been extended to provision configuration information on devices (COPS-PR) [RFC3084]. Work is underway to define data definitions for specific services such as Differentiated Services (DiffServ).

During 2001, several meetings have been organized at various events (NANOG-22 May 2001, RIPE-40 October 2001, LISA-XV December 2001, IETF-52 December 2001) to start a direct dialog between network operators and protocol developers. During these meetings, several operators have expressed their opinion that the developments in the IETF do not really address their requirements, especially for configuration management. This naturally leads to the question of whether the IETF should refocus resources, and which strategic future activities in the operations and management area should be started.

The Internet Architecture Board (IAB), on June 4 thru June 6, 2002, held an invitational workshop on network management. The goal of the workshop was to continue the important dialog started between network operators and protocol developers, and to guide the IETFs focus on future work regarding network management.

The workshop started with two breakout session to (a) identify a list of technologies relevant for network management together with their strengths and weaknesses, and to (b) identify the most important operator needs. The results of these discussions are documented in Section 2 and Section 3. During the following discussions, many more specific characteristics of the current SNMP framework were identified. These discussions are documented in Section 4. Section 5 defines a combined feature list that was developed during the discussions following the breakout sessions. Section 6 gives concrete recommendations to the IETF.

The following text makes no explicit distinction between different versions of SNMP. For the majority of the SNMP related statements, the protocol version is irrelevant. Nevertheless, some statements are more applicable to SNMPv1/SNMPv2c environments, while other statements (especially those concerned with security) are more applicable to SNMPv3 environments.

2. Network Management Technologies

During the breakout sessions, the protocol developers assembled a list of the various network management technologies that are available or under active development. For each technology, a list of strong (+) and weak (-) points were identified. There are also some characteristics which appear to be neutral (o).

The list does not attempt to be complete. Focus was given to IETF specific technologies (SNMP, COPS-PR, PCIM) and widely used proprietary technologies (CLI, HTTP/HTML, XML). The existence of other generic management technologies (such as TL1, CORBA, CMIP/GDMO,

TMN) or specific management technologies for specific problem domains (such as RADIUS, DHCP, BGP, OSPF) were acknowledged, but were not the focus of discussion.

2.1 SNMP / SMI / MIBs

The SNMP management technology was created in the late 1980s and has since been widely implemented and deployed in the Internet. There is lots of implementational and operational experience, and the characteristics of the technology are thus well understood.

+ SNMP works reasonably well for device monitoring. The stateless nature of SNMP is useful for statistical and status polling.

+ SNMP is widely deployed for basic monitoring. Some core MIB modules, such as the IF-MIB [RFC2863], are implemented on most networking devices.

+ There are many well defined proprietary MIB modules developed by network device vendors to support their management products.

+ SNMP is an important data source for systems that do event correlation, alarm detection, and root cause analysis.

o SNMP requires applications to be useful. SNMP was, from its early days, designed as a programmatic interface between management applications and devices. As such, using SNMP without management applications or smart tools appears to be more complicated.

o Standardized MIB modules often lack writable MIB objects which can be used for configuration, and this leads to a situation where the interesting writable objects exist in proprietary MIB modules.

- There are scaling problems with regard to the number of objects in a device. While SNMP provides reasonable performance for the retrieval of a small amount of data from many devices, it becomes rather slow when retrieving large amounts of data (such as routing tables) from a few devices.

- There is too little deployment of writable MIB modules. While there are some notable exceptions in areas, such as cable modems where writable MIB modules are essential, it appears that router equipment is usually not fully configurable via SNMP.

- The SNMP transactional model and the protocol constraints make it more complex to implement MIBs, as compared to the implementation of commands of a command line interface interpreter. A logical operation on a MIB can turn into a sequence of SNMP interactions

where the implementation has to maintain state until the operation is complete, or until a failure has been determined. In case of a failure, a robust implementation must be smart enough to roll the device back into a consistent state.

- SNMP does not support easy retrieval and playback of configurations. One part of the problem is that it is not easy to identify configuration objects. Another part of the problem is that the naming system is very specific and physical device reconfigurations can thus break the capability to play back a previous configuration.

- There is often a semantic mismatch between the task-oriented view of the world usually preferred by operators and the data-centric view of the world provided by SNMP. Mapping from a task-oriented view to the data-centric view often requires some non-trivial code on the management application side.

- Several standardized MIB modules lack a description of high-level procedures. It is often not obvious from reading the MIB modules how certain high-level tasks are accomplished, which leads to several different ways to achieve the same goal, which increases costs and hinders interoperability.

A more detailed discussion about the SNMP management technology can be found in Section 4.

2.2 COPS-PR / SPPI / PIBs

The COPS protocol [RFC2748] was defined in the late 1990s to support policy control over QoS signaling protocols. The COPS-PR extension allows provision policy information on devises.

+ COPS-PR allows high-level transactions for single devices, including deleting one configuration and replacing it with another.

+ COPS-PRs non-overlapping instance namespace normally ensures that no other manager can corrupt a specific configuration. All transactions for a given instance namespace are required to be executed in-order.

+ Both manager and device states are completely synchronized with one another at all times. If there is a failure in communication, the state is resynchronized when the network is operating properly again and the device’s network configuration is valid.

+ The atomicity of transactions is well-defined. If there is any failure in a transaction, that specific failure is reported to the manager, and the local configuration is supposed to be automatically rolled-back to the state of the last "good" transaction.

+ Capability reporting is part of the framework PIB which must be supported by COPS-PR implementations. This allows management applications to adapt to the capabilities present on a device.

+ The focus of COPS-PR is configuration, and the protocol has been optimized for this purpose (by using for example TCP as a transport mechanism).

o Only a single manager is allowed to have control, at any point in time, for a given subject category on a device. (The subject category maps to a COPS Client-Type.) This single manager assumption simplifies the protocol as it makes it easier to maintain shared state.

o Similar to SNMP, COPS-PR requires applications to be useful since it is also designed as a programmatic interface between management applications and devices.

- As of the time of the meeting, there are no standardized PIB modules.

- Compared to SNMP, there is not yet enough experience to understand the strong and weak aspects of the protocol in operational environments.

- COPS-PR does not support easy retrieval and playback of configurations. The reasons are similar as for SNMP.

- The COPS-PR view of the world is data-centric, similar to SNMP's view of the world. A mapping from the data-centric view to a task-oriented view and vice versa, has similar complexities as with SNMP.

2.3 CIM / MOF / UML / PCIM

The development of the Common Information Model (CIM) [CIM] started in the DMTF in the mid 1990s. The development follows a top-down approach where core classes are defined first and later extended to model specific services. The DMTF and the IETF jointly developed policy extensions of the CIM, known as PCIM [RFC3060].

+ The CIM technology generally follows principles of object-orientation with full support of methods on data objects, which is not available in SNMP or COPS-PR.

+ The MOF format allows representation of instances in a common format. No such common format exists for SNMP or COPS-PR. It is of course possible to store instances in the form of BER encoded ASN.1 sequences, but this is generally not suitable for human readability.

+ There is support for a query facility which allows the locating of CIM objects. However, the query language itself is not yet specified as part of the CIM standards. Implementations currently use proprietary query languages, such as the Windows Management Instrumentation Query Language (WQL).

+ The information modeling work in CIM is done by using Unified Modeling Language (UML) as a graphical notation. This attracts people with a computer science background who have learned to use UML as part of their education.

o The main practical use of CIM schemas today seems to be the definition of data structures used internally by management systems.

- The CIM schemas have rather complex interrelationships that must be understood before one can reasonably extend the set of existing schemas.

- Interoperability between CIM implementations seems to be problematic compared to the number of interoperable SNMP implementations available today.

- So far, CIM schemas have seen limited implementation and usage as an interface between management systems and network devices.

2.4 CLI / TELNET / SSH

Most devices have a builtin command line interface (CLI) for configuration and troubleshooting purposes. Network access to the CLI has traditionally been through the TELNET protocol, while the SSH protocol is gaining momentum to address security issues associated with TELNET. In the following, only CLIs that actually parse and execute commands are considered. (Menu-oriented interfaces are difficult for automation and thus not relevant here.)

+ Command line interfaces are generally task-oriented, which make them easier to use for human operators.

+ A saved sequence of textual commands can easily be replayed. Simple substitutions can be made with arbitrary text processing tools.

+ It is usually necessary to learn at least parts of the command line interface of new devices in order to create the initial configuration. Once people have learned (parts of) the command line interface, it is natural for them to use the same interface and abstractions for automating configuration changes.

+ A command line interface does not require any special purpose applications (telnet and ssh are readily available on most systems today).

+ Most command line interfaces provide context sensitive help that reduces the learning curve.

- Some command line interfaces lack a common data model. It is very well possible that the same command on different devices, even from the same vendor, behaves differently.

- The command line interface is primarily targeted to humans which can adapt to minor syntax and format changes easily. Using command line interfaces as a programmatic interface is troublesome because of parsing complexities.

- Command line interfaces often lack proper version control for the syntax and the semantics. It is therefore time consuming and error prone to maintain programs or scripts that interface with different versions of a command line interface.

- Since command line interfaces are proprietary, they can not be used efficiently to automate processes in an environment with a heterogenous set of devices.

- The access control facilities, if present at all, are often ad-hoc and sometimes insufficient.

2.5 HTTP / HTML

Many devices have an embedded web server which can be used to configure the device and to obtain status information. The commonly used protocol is HTTP, and information is rendered in HTML. Some devices also expect that clients have facilities such as Java or Java Script.

+ Embedded web servers for configuration are end-user friendly and solution oriented.

+ Embedded web servers are suitable for configuring consumer devices by inexperienced users.

+ Web server configuration is widely deployed, especially in boxes targeted to the consumer market.

+ There is no need for specialized applications to use embedded web servers since web browsers are commonly available today.

- Embedded web servers are management application hostile. Parsing HTML pages to extract useful information is extremely painful.

- Replay of configuration is often problematic, either because the web pages rely on some active content or because different versions of the same device use different ways to interact with the user.

- The access control facilities, if present at all, are often ad-hoc and sometimes insufficient.

2.6 XML

In the late 1990's, some vendors started to use the Extensible Markup Language (XML) [XML] for describing device configurations and for protocols that can be used to retrieve and manipulate XML formatted configurations.

+ XML is a machine readable format which is easy to process and there are many good off the shelf tools available.

+ XML allows the description of structured data of almost arbitrary complexity.

+ The basic syntax rules behind XML are relatively easy to learn.

+ XML provides a document-oriented view of configuration data (similar to many proprietary configuration file formats).

+ XML has a robust schema language XSD [XSD] for which many good off the shelf tools exist.

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

- XML is rather verbose. This either increases the bandwidth required to move management information around (which is an issue in e.g., wireless or asymmetric cable networks) or it requires that the systems involved have the processing power to do on the fly compression/decompression.

- There is a lack of commonly accepted standardized management specific XML schemas.

3. Operator Requirements

During the breakout session, the operators were asked to identify needs that have not been sufficiently addressed. The results produced during the breakout session were later discussed and resulted in the following list of operator requirements.

1. Ease of use is a key requirement for any network management technology from the operators point of view.

2. It is necessary to make a clear distinction between configuration data, data that describes operational state and statistics. Some devices make it very hard to determine which parameters were administratively configured and which were obtained via other mechanisms such as routing protocols.

3. It is required to be able to fetch separately configuration data, operational state data, and statistics from devices, and to be able to compare these between devices.

4. It is necessary to enable operators to concentrate on the configuration of the network as a whole rather than individual devices.

5. Support for configuration transactions across a number of devices would significantly simplify network configuration management.

6. Given configuration A and configuration B, it should be possible to generate the operations necessary to get from A to B with minimal state changes and effects on network and systems. It is important to minimize the impact caused by configuration changes.

7. A mechanism to dump and restore configurations is a primitive operation needed by operators. Standards for pulling and pushing configurations from/to devices are desirable.

8. It must be easy to do consistency checks of configurations over time and between the ends of a link in order to determine the changes between two configurations and whether those configurations are consistent.

9. Network wide configurations are typically stored in central master databases and transformed into formats that can be pushed to devices, either by generating sequences of CLI commands or complete configuration files that are pushed to devices. There is no common database schema for network configuration, although the models used by various operators are probably very similar. It is desirable to extract, document, and standardize the common parts of these network wide configuration database schemas.

10. It is highly desirable that text processing tools such as diff, and version management tools such as RCS or CVS, can be used to process configurations, which implies that devices should not arbitrarily reorder data such as access control lists.

11. The granularity of access control needed on management interfaces needs to match operational needs. Typical requirements are a role-based access control model and the principle of least privilege, where a user can be given only the minimum access necessary to perform a required task.

12. It must be possible to do consistency checks of access control lists across devices.

13. It is important to distinguish between the distribution of configurations and the activation of a certain configuration. Devices should be able to hold multiple configurations.

14. SNMP access control is data-oriented, while CLI access control is usually command (task) oriented. Depending on the management function, sometimes data-oriented or task-oriented access control makes more sense. As such, it is a requirement to support both data-oriented and task-oriented access control.

So far, there is no published document that clearly defines the requirements of the operators.

4. SNMP Framework Discussions

During the discussions, many properties of the SNMP framework were identified.

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.

2. The quality of SNMP instrumentations is sometimes disappointing. SNMP access sometimes crashes systems or returns wrong data.

3. MIB modules and their implementations are not available in a timely manner (sometimes MIB modules lag years behind) which forces users to use the CLI.

4. Operators view current SNMP programming/scripting interfaces as being too low-level and thus too time consuming and inconvenient for practical use.

5. Lexicographic ordering is sometimes artificial with regard to internal data structures and causes either significant runtime overhead, or increases implementation costs or implementation delay or both.

6. Poor performance for bulk data transfers. The typical examples are routing tables.

7. Poor performance on query operations that were not anticipated during the MIB design. A typical example is the following query: Which outgoing interface is being used for a specific destination address?

8. The SNMP credentials and key management are considered complex, especially since they do not integrate well with other existing credential and key management systems.

9. The SMI language is hard to deal with and not very practical.

10. MIB modules are often over-engineered in the sense that they contain lots of variables that operators do not look at.

11. SNMP traps are used to track state changes but often syslog messages are considered more useful since they usually contain more information to describe the problem. SNMP traps usually require subsequent get operations to figure out what the trap really means.

12. Device manufacturers find SNMP instrumentations inherently difficult to implement, especially with complex table indexing schemes and table interrelationships.

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.)

14. The lack of structured types and various RPC interactions (methods) make MIB modules much more complex to design and implement.

15. The lack of query and aggregation capabilities (reduction of data) causes efficiency and scalability problems.

16. The SNMP protocol was simplified in terms of the number of protocol operations and resource requirements on managed devices. It was not simplified in terms of usability by network operators or instrumentation implementors.

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.

18. SNMP seems to work reasonably well for small devices which have a limited number of managed objects and where end-user management applications are shipped by the vendor. For more complex devices, SNMP becomes too expensive and too hard to use.

19. There is a disincentive for vendors to implement SNMP equivalent MIB modules for all their CLI commands because they do not see a valued proposition. This undermines the value of third party standard SNMP solutions.

20. Rapid feature development is in general not compatible with the standardization of the configuration interface.

5. Consolidated Observations

1. Programmatic interfaces have to provide full coverage otherwise they will not be used by network operators since they have to revert to CLIs anyway.

2. Operators perceive that equipment vendors do not implement MIB modules in a timely manner. Neither read-only nor read-write MIB modules are available on time today.

3. The attendees perceive that right now it is too hard to implement useful MIB modules within network equipment.

4. Because of the previous items, SNMP is not widely used today for network device configuration, although there are notable exceptions.

5. It is necessary to clearly distinguish between configuration data and operational data.

6. It would be nice to have a single data definition language for all programmatic interfaces (in case there happen to be multiple programmatic interfaces).

7. In general, there is a lack of input from the enterprise network space. Those enterprises who provided input tend to operate their networks like network operators.

8. It is required to be able to dump and reload a device configuration in a textual format in a standard manner across multiple vendors and device types.

9. It is desirable to have a mechanism to distribute configurations to devices under transactional constraints.

10. Eliminating SNMP altogether is not an option.

11. Robust access control is needed. In addition, it is desirable to be able to enable/disable individual MIB modules actually implemented on a device.

12. Textual configuration files should be able to contain international characters. Human-readable strings should utilize the least-bad internationalized character set and encoding, which this year almost certainly means UTF-8. Protocol elements should be in case insensitive ASCII.

13. The deployed tools for event/alarm correlation, root cause analysis and logging are not sufficient.

14. There is a need to support a human interface and a programmatic interface.

15. The internal method routines for both interfaces should be the same to ensure that data exchanged between these two interfaces is always consistent.

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

17. The implementation costs have to be low on managers.

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

19. Standardization costs for data models have to be low.

20. There should be a single data modeling language with a human friendly syntax.

21. The data modeling language must support compound data types.

22. There is a need for data aggregation capabilities on the devices.

23. There should be a common data interchange format for instance data that allows easy post-processing and analysis.

24. There is a need for a common data exchange format with single and multi-system transactions (which implies rollback across devices in error situations).

25. There is a need to reduce the semantic mismatch between current data models and the primitives used by operators.

26. It should be possible to perform operations on selected subsets of management data.

27. It is necessary to discover the capabilities of devices.

28. There is a need for a secure transport, authentication, identity, and access control which integrates well with existing key and credential management infrastructure.

29. It must be possible to define task oriented views and access control rules.

30. The complete configuration of a device should be doable with a single protocol.

31. A configuration protocol must be efficient and reliable and it must scale in the number of transactions and devices.

32. Devices must be able to support minimally interruptive configuration deltas.

33. A solution must support function call semantics (methods) to implement functions, such as a longest prefix match on a routing table.

6. Recommendations

1. The workshop recommends that the IETF stop forcing working groups to provide writable MIB modules. It should be the decision of the working group whether they want to provide writable objects or not.

2. The workshop recommends that a group be formed to investigate why current MIB modules do not contain all the objects needed by operators to monitor their networks.

3. The workshop recommends that a group be formed to investigate why the current SNMP protocol does not satisfy all the monitoring requirements of operators.

4. The workshop recommends, with strong consensus from both protocol developers and operators, that the IETF focus resources on the standardization of configuration management mechanisms.

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).

6. The workshop recommends, with strong consensus from the operators and rough consensus from the protocol developers, that the IETF/IRTF should not spend resources on developing HTML-based or HTTP-based methods for configuration management.

7. The workshop recommends, with rough consensus from the operators and strong consensus from the protocol developers, that the IETF should continue to spend resources on the evolution of the SMI/SPPI data definition languages as being done in the SMIng working group.

8. The workshop recommends, with split consensus from the operators and rough consensus from the protocol developers, that the IETF should spend resources on fixing the MIB development and standardization process.

The workshop also discussed the following items and achieved rough consensus, but did not make a recommendation.

1. The workshop had split consensus from the operators and rough consensus from the protocol developers, that the IETF should not focus resources on CIM extensions.

2. The workshop had rough consensus from the protocol developers that the IETF should not spend resources on COPS-PR development. So far, the operators have only very limited experience with COPS-PR. In general, however, they felt that further development of COPS-PR might be a waste of resources as they assume that COPS-PR does not really address their requirements.

3. The workshop had rough consensus from the protocol developers that the IETF should not spend resources on SPPI PIB definitions. The operators had rough consensus that they do not care about SPPI PIBs.

7. Security Considerations

This document is a report of an IAB Network Management workshop. As such, it does not have any direct security implications for the Internet.

8. Acknowledgments

The editor would like to thank Dave Durham, Simon Leinen and John Schnizlein for taking detailed minutes during the workshop.

Normative References

[RFC3410] Case, J., Mundy, R., Partain, D. and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, December 2002.

[CIM] Distributed Management Task Force, "Common Information Model (CIM) Specification Version 2.2", DSP 0004, June 1999.

[RFC3060] Moore, B., Ellesson, E., Strassner, J. and A. Westerinen, "Policy Core Information Model -- Version 1 Specification", RFC 3060, February 2001.

[RFC2748] Durham, D., Boyle, J., Cohen, R., Herzog, S., Rajan, R. and A. Sastry, "The COPS (Common Open Policy Service) Protocol", RFC 2748, January 2000.

[RFC3084] Chan, K., Seligson, J., Durham, D., Gai, S., McCloghrie, K., Herzog, S., Reichmeyer, F., Yavatkar, R. and A. Smith, "COPS Usage for Policy Provisioning (COPS-PR)", RFC 3084, March 2001.

[XML] Bray, T., Paoli, J. and C. Sperberg-McQueen, "Extensible Markup Language (XML) 1.0", W3C Recommendation, February 1998.

Informative References

[RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, June 2000.

[XSD] David, D., "XML Schema Part 0: Primer", W3C Recommendation, May 2001.

Appendix - Participants

Ran Atkinson	Extreme Networks
Rob Austein	InterNetShare
Andy Bierman	Cisco Systems
Steve Bellovin	AT&T
Randy Bush	AT&T
Leslie Daigle	VeriSign
David Durham	Intel
Vijay Gill	
Wes Hardaker	Network Associates Laboratories
Ed Kern	
Simon Leinen	Switch
Ken Lindahl	University of California Berkeley
David Partain	Ericsson
Andrew Partan	UUnet/Verio/MFN
Vern Paxson	ICIR
Aiko Pras	Univeristy of Twente
Randy Presuhn	BMC Software
Juergen Schoenwaelder	University of Osnabrueck
John Schnizlein	Cisco Systems
Mike St. Johns	
Ruediger Volk	Deutsche Telekom
Steve Waldbusser	
Margaret Wassermann	Windriver
Glen Waters	Nortel Networks
Bert Wijnen	Lucent

Author’s Address

Comments should be submitted to the <nm-ws@ops.ietf.org> mailing list.

Juergen Schoenwaelder
International University Bremen
P.O. Box 750 561
28725 Bremen
Germany

Phone: +49 421 200 3587
EMail: j.schoenwaelder@iu-bremen.de

Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.