Network Working Group
Request for Comments: 5706
Category: Informational

D. Harrington
HuaweiSymantec USA

November 2009

Guidelines for Considering Operations and Management of New Protocols and Protocol Extensions

Abstract

New protocols or protocol extensions are best designed with due consideration of the functionality needed to operate and manage the protocols. Retrofitting operations and management is sub-optimal. The purpose of this document is to provide guidance to authors and reviewers of documents that define new protocols or protocol extensions regarding aspects of operations and management that should be considered.

Status of This Memo

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1. Introduction

Often when new protocols or protocol extensions are developed, not enough consideration is given to how the protocol will be deployed, operated, and managed. Retrofitting operations and management mechanisms is often hard and architecturally unpleasant, and certain protocol design choices may make deployment, operations, and management particularly hard. This document provides guidelines to help protocol designers and working groups consider the operations and management functionality for their new IETF protocol or protocol extension at an earlier phase.

1.1. Designing for Operations and Management

The operational environment and manageability of the protocol should be considered from the start when new protocols are designed.

Most of the existing IETF management standards are focused on using Structure of Management Information (SMI)-based data models (MIB modules) to monitor and manage networking devices. As the Internet has grown, IETF protocols have addressed a constantly growing set of needs, such as web servers, collaboration services, and applications. The number of IETF management technologies has been expanding and the IETF management strategy has been changing to address the emerging management requirements. The discussion of emerging sets of management requirements has a long history in the IETF. The set of management protocols you should use depends on what you are managing.

Protocol designers should consider which operations and management needs are relevant to their protocol, document how those needs could be addressed, and suggest (preferably standard) management protocols and data models that could be used to address those needs. This is similar to a working group (WG) that considers which security threats are relevant to their protocol, documents how threats should be mitigated, and then suggests appropriate standard protocols that could mitigate the threats.

When a WG considers operation and management functionality for a protocol, the document should contain enough information for readers to understand how the protocol will be deployed and managed. The WG should expect that considerations for operations and management may need to be updated in the future, after further operational experience has been gained.

1.2. This Document

This document makes a distinction between "Operational Considerations" and "Management Considerations", although the two are closely related. The section on manageability is focused on management technology, such as how to utilize management protocols and how to design management data models. The operational considerations apply to operating the protocol within a network, even if there were no management protocol actively being used.

The purpose of this document is to provide guidance about what to consider when thinking about the management and deployment of a new protocol, and to provide guidance about documenting the considerations. The following guidelines are designed to help writers provide a reasonably consistent format for such documentation. Separate manageability and operational considerations sections are desirable in many cases, but their structure and location is a decision that can be made from case to case.

This document does not impose a solution, imply that a formal data model is needed, or imply that using a specific management protocol is mandatory. If protocol designers conclude that the technology can be managed solely by using proprietary command line interfaces (CLIs) and that no structured or standardized data model needs to be in place, this might be fine, but it is a decision that should be explicit in a manageability discussion -- that this is how the protocol will need to be operated and managed. Protocol designers should avoid having manageability pushed for a later phase of the development of the standard.

In discussing the importance of considering operations and management, this document sets forth a list of guidelines and a checklist of questions to consider (see Appendix A), which a protocol designer or reviewer can use to evaluate whether the protocol and documentation address common operations and management needs. Operations and management are highly dependent on their environment, so most guidelines are subjective rather than objective.

1.3. Motivation

For years the IETF community has used the IETF Standard Management Framework, including the Simple Network Management Protocol [RFC3410], the Structure of Management Information [RFC2578], and MIB data models for managing new protocols. As the Internet has evolved, operators have found the reliance on one protocol and one schema language for managing all aspects of the Internet inadequate. The IESG policy to require working groups to write a MIB module to

provide manageability for new protocols is being replaced by a policy that is more open to using a variety of management protocols and data models designed to achieve different goals.

This document provides some initial guidelines for considering operations and management in an IETF Management Framework that consists of multiple protocols and multiple data-modeling languages, with an eye toward being flexible while also striving for interoperability.

Fully new protocols may require significant consideration of expected operations and management, while extensions to existing, widely deployed protocols may have established de facto operations and management practices that are already well understood.

Suitable management approaches may vary for different areas, working groups, and protocols in the IETF. This document does not prescribe a fixed solution or format in dealing with operational and management aspects of IETF protocols. However, these aspects should be considered for any IETF protocol because we develop technologies and protocols to be deployed and operated in the real-world Internet. It is fine if a WG decides that its protocol does not need interoperable management or no standardized data model, but this should be a deliberate decision, not the result of omission. This document provides some guidelines for those considerations.

1.4. Background

There have been a significant number of efforts, meetings, and documents that are related to Internet operations and management. Some of them are mentioned here to help protocol designers find documentation of previous efforts. Hopefully, providing these references will help the IETF avoid rehashing old discussions and reinventing old solutions.

In 1988, the IAB published "IAB Recommendations for the Development of Internet Network Management Standards" [RFC1052], which recommended a solution that, where possible, deliberately separates modeling languages, data models, and the protocols that carry data. The goal is to allow standardized information and data models to be used by different protocols.

In 2001, Operations and Management Area design teams were created to document requirements related to the configuration of IP-based networks. One output was "Requirements for Configuration Management of IP-based Networks" [RFC3139].

In 2003, the Internet Architecture Board (IAB) held a workshop on Network Management [RFC3535] that discussed the strengths and weaknesses of some IETF network management protocols and compared them to operational needs, especially configuration.

One issue discussed was the user-unfriendliness of the binary format of SNMP [RFC3410] and Common Open Policy Service (COPS) Usage for Policy Provisioning (COPS-PR) [RFC3084], and it was recommended that the IETF explore an XML-based Structure of Management Information and an XML-based protocol for configuration.

Another conclusion was that the tools for event/alarm correlation and for root cause analysis and logging are not sufficient and that there is a need to support a human interface and a programmatic interface. The IETF decided to standardize aspects of the de facto standard for system-logging security and programmatic support.

In 2006, the IETF discussed whether the Management Framework should be updated to accommodate multiple IETF schema languages for describing the structure of management information and multiple IETF standard protocols for performing management tasks. The IESG asked that a document be written to discuss how protocol designers and working groups should address management in this emerging multi-protocol environment. This document and some planned companion documents attempt to provide some guidelines for navigating the rapidly shifting operating and management environments.

1.5. Available Management Technologies

The IETF has a number of standard management protocols available that are suitable for different purposes. These include:

Simple Network Management Protocol - SNMP [RFC3410]

Syslog [RFC5424]

Remote Authentication Dial-In User Service - RADIUS [RFC2865]

DIAMETER [RFC3588]

Network Configuration Protocol - NETCONF [RFC4741]

IP Flow Information Export - IPFIX [RFC5101]

A planned supplement to this document will discuss these protocol standards, discuss some standard information and data models for specific functionality, and provide pointers to the documents that define them.

1.6. Terminology

This document deliberately does not use the (capitalized) keywords described in RFC 2119 [RFC2119]. RFC 2119 states the keywords must only be used where it is actually required for interoperation or to limit behavior which has potential for causing harm (e.g., limiting retransmissions). For example, they must not be used to try to impose a particular method on implementers where the method is not required for interoperability. This informational document is a set of guidelines based on current practices of **some** protocol designers and operators. This document is biased toward router operations and management and some advice may not be directly applicable to protocols with a different purpose, such as application server protocols. This document **does not** describe interoperability requirements, so the capitalized keywords from RFC 2119 do not apply here.

- CLI: Command Line Interface
- Data model: a mapping of the contents of an information model into a form that is specific to a particular type of data store or repository [RFC3444].
- Information model: an abstraction and representation of the entities in a managed environment, their properties, attributes and operations, and the way that they relate to each other. It is independent of any specific repository, software usage, protocol, or platform [RFC3444].
- New protocol: includes new protocols, protocol extensions, data models, or other functionality being designed.
- Protocol designer: represents individuals and working groups involved in the development of new protocols or extensions.

2. Operational Considerations - How Will the New Protocol Fit into the Current Environment?

Designers of a new protocol should carefully consider the operational aspects. To ensure that a protocol will be practical to deploy in the real world, it is not enough to merely define it very precisely in a well-written document. Operational aspects will have a serious impact on the actual success of a protocol. Such aspects include bad interactions with existing solutions, a difficult upgrade path, difficulty of debugging problems, difficulty configuring from a central database, or a complicated state diagram that operations staff will find difficult to understand.

BGP flap damping [RFC2439] is an example. It was designed to block high-frequency route flaps; however, the design did not consider the existence of BGP path exploration / slow convergence. In real operations, path exploration caused false flap damping, resulting in loss of reachability. As a result, many networks turned flap damping off.

2.1. Operations

Protocol designers can analyze the operational environment and mode of work in which the new protocol or extension will work. Such an exercise need not be reflected directly by text in their document, but could help in visualizing how to apply the protocol in the Internet environments where it will be deployed.

A key question is how the protocol can operate "out of the box". If implementers are free to select their own defaults, the protocol needs to operate well with any choice of values. If there are sensible defaults, these need to be stated.

There may be a need to support a human interface, e.g., for troubleshooting, and a programmatic interface, e.g., for automated monitoring and root cause analysis. The application programming interfaces and the human interfaces might benefit from being similar to ensure that the information exposed by these two interfaces is consistent when presented to an operator. Identifying consistent methods of determining information, such as what gets counted in a specific counter, is relevant.

Protocol designers should consider what management operations are expected to be performed as a result of the deployment of the protocol -- such as whether write operations will be allowed on routers and on hosts, or whether notifications for alarms or other events will be expected.

2.2. Installation and Initial Setup

Anything that can be configured can be misconfigured. "Architectural Principles of the Internet" [RFC1958], Section 3.8, states: "Avoid options and parameters whenever possible. Any options and parameters should be configured or negotiated dynamically rather than manually."

To simplify configuration, protocol designers should consider specifying reasonable defaults, including default modes and parameters. For example, it could be helpful or necessary to specify default values for modes, timers, default state of logical control

variables, default transports, and so on. Even if default values are used, it must be possible to retrieve all the actual values or at least an indication that known default values are being used.

Protocol designers should consider how to enable operators to concentrate on the configuration of the network as a whole rather than on individual devices. Of course, how one accomplishes this is the hard part.

It is desirable to discuss the background of chosen default values, or perhaps why a range of values makes sense. In many cases, as technology changes, the values in an RFC might make less and less sense. It is very useful to understand whether defaults are based on best current practice and are expected to change as technologies advance or whether they have a more universal value that should not be changed lightly. For example, the default interface speed might be expected to change over time due to increased speeds in the network, and cryptographical algorithms might be expected to change over time as older algorithms are "broken".

It is extremely important to set a sensible default value for all parameters.

The default value should stay on the conservative side rather than on the "optimizing performance" side (example: the initial RTT and RTTvar values of a TCP connection).

For those parameters that are speed-dependent, instead of using a constant, try to set the default value as a function of the link speed or some other relevant factors. This would help reduce the chance of problems caused by technology advancement.

2.3. Migration Path

If the new protocol is a new version of an existing one, or if it is replacing another technology, the protocol designer should consider how deployments should transition to the new protocol. This should include coexistence with previously deployed protocols and/or previous versions of the same protocol, incompatibilities between versions, translation between versions, and side effects that might occur. Are older protocols or versions disabled or do they coexist in the network with the new protocol?

Many protocols benefit from being incrementally deployable -- operators may deploy aspects of a protocol before deploying the protocol fully.

2.4. Requirements on Other Protocols and Functional Components

Protocol designers should consider the requirements that the new protocol might put on other protocols and functional components and should also document the requirements from other protocols and functional elements that have been considered in designing the new protocol.

These considerations should generally remain illustrative to avoid creating restrictions or dependencies, or potentially impacting the behavior of existing protocols, or restricting the extensibility of other protocols, or assuming other protocols will not be extended in certain ways. If restrictions or dependencies exist, they should be stated.

For example, the design of the Resource ReSerVation Protocol (RSVP) [RFC2205] required each router to look at the RSVP PATH message and, if the router understood RSVP, add its own address to the message to enable automatic tunneling through non-RSVP routers. But in reality, routers cannot look at an otherwise normal IP packet and potentially take it off the fast path! The initial designers overlooked that a new "deep packet inspection" requirement was being put on the functional components of a router. The "router alert" option ([RFC2113], [RFC2711]) was finally developed to solve this problem for RSVP and other protocols that require the router to take some packets off the fast-forwarding path. Yet, router alert has its own problems in impacting router performance.

2.5. Impact on Network Operation

The introduction of a new protocol or extensions to an existing protocol may have an impact on the operation of existing networks. Protocol designers should outline such impacts (which may be positive), including scaling concerns and interactions with other protocols. For example, a new protocol that doubles the number of active, reachable addresses in use within a network might need to be considered in the light of the impact on the scalability of the interior gateway protocols operating within the network.

A protocol could send active monitoring packets on the wire. If we don’t pay attention, we might get very good accuracy, but could send too many active monitoring packets.

The protocol designer should consider the potential impact on the behavior of other protocols in the network and on the traffic levels and traffic patterns that might change, including specific types of traffic, such as multicast. Also, consider the need to install new

components that are added to the network as a result of changes in the configuration, such as servers performing auto-configuration operations.

The protocol designer should consider also the impact on infrastructure applications like DNS [RFC1034], the registries, or the size of routing tables. For example, Simple Mail Transfer Protocol (SMTP) [RFC5321] servers use a reverse DNS lookup to filter out incoming connection requests. When Berkeley installed a new spam filter, their mail server stopped functioning because of overload of the DNS cache resolver.

The impact on performance may also be noted -- increased delay or jitter in real-time traffic applications, or increased response time in client-server applications when encryption or filtering are applied.

It is important to minimize the impact caused by configuration changes. 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.

2.6. Verifying Correct Operation

The protocol designer should consider techniques for testing the effect that the protocol has had on the network by sending data through the network and observing its behavior (aka active monitoring). Protocol designers should consider how the correct end-to-end operation of the new protocol in the network can be tested actively and passively, and how the correct data or forwarding plane function of each network element can be verified to be working properly with the new protocol. Which metrics are of interest?

Having simple protocol status and health indicators on network devices is a recommended means to check correct operation.

3. Management Considerations - How Will the Protocol Be Managed?

The considerations of manageability should start from identifying the entities to be managed, as well as how the managed protocol is supposed to be installed, configured, and monitored.

Considerations for management should include a discussion of what needs to be managed, and how to achieve various management tasks. Where are the managers and what type of management interfaces and protocols will they need? The "write a MIB module" approach to considering management often focuses on monitoring a protocol endpoint on a single device. A MIB module document typically only

considers monitoring properties observable at one end, while the document does not really cover managing the *protocol* (the coordination of multiple ends), and does not even come near managing the *service* (which includes a lot of stuff that is very far away from the box). This is exactly what operators hate -- you need to be able to manage both ends. As [RFC3535] says, "MIB modules can often be characterized as a list of ingredients without a recipe".

The management model should take into account factors such as:

- What type of management entities will be involved (agents, network management systems)?
- What is the possible architecture (client-server, manager-agent, poll-driven or event-driven, auto-configuration, two levels or hierarchical)?
- What are the management operations (initial configuration, dynamic configuration, alarm and exception reporting, logging, performance monitoring, performance reporting, debugging)?
- How are these operations performed (locally, remotely, atomic operation, scripts)? Are they performed immediately or are they time scheduled or event triggered?

Protocol designers should consider how the new protocol will be managed in different deployment scales. It might be sensible to use a local management interface to manage the new protocol on a single device, but in a large network, remote management using a centralized server and/or using distributed management functionality might make more sense. Auto-configuration and default parameters might be possible for some new protocols.

Management needs to be considered not only from the perspective of a device, but also from the perspective of network and service management. A service might be network and operational functionality derived from the implementation and deployment of a new protocol. Often an individual network element is not aware of the service being delivered.

WGs should consider how to configure multiple related/co-operating devices and how to back off if one of those configurations fails or causes trouble. NETCONF [RFC4741] addresses this in a generic manner by allowing an operator to lock the configuration on multiple devices, perform the configuration settings/changes, check that they are OK (undo if not), and then unlock the devices.

Techniques for debugging protocol interactions in a network must be part of the network-management discussion. Implementation source code should be debugged before ever being added to a network, so asserts and memory dumps do not normally belong in management data models. However, debugging on-the-wire interactions is a protocol issue: while the messages can be seen by sniffing, it is enormously helpful if a protocol specification supports features that make debugging of network interactions and behaviors easier. There could be alerts issued when messages are received or when there are state transitions in the protocol state machine. However, the state machine is often not part of the on-the-wire protocol; the state machine explains how the protocol works so that an implementer can decide, in an implementation-specific manner, how to react to a received event.

In a client/server protocol, it may be more important to instrument the server end of a protocol than the client end, since the performance of the server might impact more nodes than the performance of a specific client.

3.1. Interoperability

Just as when deploying protocols that will inter-connect devices, management interoperability should be considered -- whether across devices from different vendors, across models from the same vendor, or across different releases of the same product. Management interoperability refers to allowing information sharing and operations between multiple devices and multiple management applications, often from different vendors. Interoperability allows for the use of third-party applications and the outsourcing of management services.

Some product designers and protocol designers assume that if a device can be managed individually using a command line interface or a web page interface, that such a solution is enough. But when equipment from multiple vendors is combined into a large network, scalability of management may become a problem. It may be important to have consistency in the management interfaces so network-wide operational processes can be automated. For example, a single switch might be easily managed using an interactive web interface when installed in a single-office small business, but when, say, a fast-food company installs similar switches from multiple vendors in hundreds or thousands of individual branches and wants to automate monitoring them from a central location, monitoring vendor- and model-specific web pages would be difficult to automate.

The primary goal is the ability to roll out new useful functions and services in a way in which they can be managed in a scalable manner, where one understands the network impact (as part of the total cost of operations) of that service.

Getting everybody to agree on a single syntax and an associated protocol to do all management has proven to be difficult. So management systems tend to speak whatever the boxes support, whether or not the IETF likes this. The IETF is moving from support for one schema language for modeling the structure of management information (Structure of Management Information Version 2 (SMIv2) [RFC2578]) and one simple network management protocol (Simple Network Management Protocol (SNMP) [RFC3410]) towards support for additional schema languages and additional management protocols suited to different purposes. Other Standard Development Organizations (e.g., the Distributed Management Task Force - DMTF, the Tele-Management Forum - TMF) also define schemas and protocols for management and these may be more suitable than IETF schemas and protocols in some cases. Some of the alternatives being considered include:

- XML Schema Definition [W3C.REC-xmlschema-0-20010502]

and

- NETCONF Configuration Protocol [RFC4741]

- the IP Flow Information Export (IPFIX) Protocol [RFC5101]) for usage accounting

- the syslog protocol [RFC5424] for logging

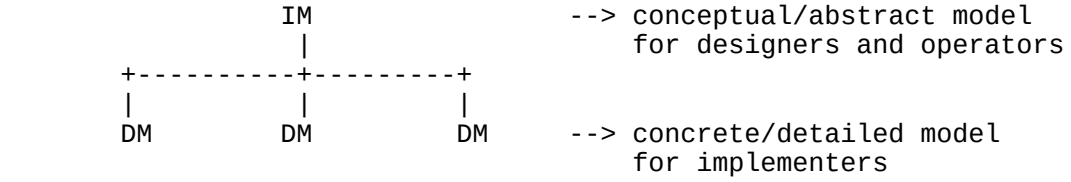
Interoperability needs to be considered on the syntactic level and the semantic level. While it can be irritating and time-consuming, application designers, including operators who write their own scripts, can make their processing conditional to accommodate syntactic differences across vendors, models, or releases of product.

Semantic differences are much harder to deal with on the manager side -- once you have the data, its meaning is a function of the managed entity.

Information models are helpful to try to focus interoperability on the semantic level -- they establish standards for what information should be gathered and how gathered information might be used, regardless of which management interface carries the data or which vendor produces the product. The use of an information model might help improve the ability of operators to correlate messages in different protocols where the data overlaps, such as a syslog message

and an SNMP notification about the same event. An information model might identify which error conditions should be counted separately and which error conditions can be counted together in a single counter. Then, whether the counter is gathered via SNMP, a CLI command, or a syslog message, the counter will have the same meaning.

Protocol designers should consider which information might be useful for managing the new protocol or protocol extensions.



Information Models and Data Models

Figure 1

Protocol designers may decide an information model or data model would be appropriate for managing the new protocol or protocol extensions.

"On the Difference between Information Models and Data Models" [RFC3444] can be helpful in determining what information to consider regarding information models (IMs), as compared to data models (DMs).

Information models should come from the protocol WGs and include lists of events, counters, and configuration parameters that are relevant. There are a number of information models contained in protocol WG RFCs. Some examples:

- [RFC3060] - Policy Core Information Model version 1
- [RFC3290] - An Informal Management Model for Diffserv Routers
- [RFC3460] - Policy Core Information Model Extensions
- [RFC3585] - IPsec Configuration Policy Information Model
- [RFC3644] - Policy Quality of Service Information Model
- [RFC3670] - Information Model for Describing Network Device QoS Datapath Mechanisms
- [RFC3805] - Printer MIB v2 (contains both an IM and a DM)

Management protocol standards and management data model standards often contain compliance clauses to ensure interoperability. Manageability considerations should include discussion of which level of compliance is expected to be supported for interoperability.

3.2. Management Information

Languages used to describe an information model can influence the nature of the model. Using a particular data-modeling language, such as the SMIv2, influences the model to use certain types of structures, such as two-dimensional tables. This document recommends using English text (the official language for IETF specifications) to describe an information model. A sample data model could be developed to demonstrate the information model.

A management information model should include a discussion of what is manageable, which aspects of the protocol need to be configured, what types of operations are allowed, what protocol-specific events might occur, which events can be counted, and for which events an operator should be notified.

Operators find it important to be able to make a clear distinction between configuration data, operational state, and statistics. They need to determine which parameters were administratively configured and which parameters have changed since configuration as the result of mechanisms such as routing protocols or network management protocols. It is important to be able to separately fetch current configuration information, initial configuration information, operational state information, and statistics from devices; to be able to compare current state to initial state; and to compare information between devices. So when deciding what information should exist, do not conflate multiple information elements into a single element.

What is typically difficult to work through are relationships between abstract objects. Ideally, an information model would describe the relationships between the objects and concepts in the information model.

Is there always just one instance of this object or can there be multiple instances? Does this object relate to exactly one other object or may it relate to multiple? When is it possible to change a relationship?

Do objects (such as rows in tables) share fate? For example, if a row in table A must exist before a related row in table B can be created, what happens to the row in table B if the related row in table A is deleted? Does the existence of relationships between objects have an impact on fate sharing?

3.2.1. Information Model Design

This document recommends keeping the information model as simple as possible by applying the following criteria:

1. Start with a small set of essential objects and add only as further objects are needed.

2. Require that objects be essential for management.

3. Consider evidence of current use and/or utility.

4. Limit the total number of objects.

5. Exclude objects that are simply derivable from others in this or other information models.

6. Avoid causing critical sections to be heavily instrumented. A guideline is one counter per critical section per layer.

3.3. Fault Management

The protocol designer should document the basic faults and health indicators that need to be instrumented for the new protocol, as well as the alarms and events that must be propagated to management applications or exposed through a data model.

The protocol designer should consider how fault information will be propagated. Will it be done using asynchronous notifications or polling of health indicators?

If notifications are used to alert operators to certain conditions, then the protocol designer should discuss mechanisms to throttle notifications to prevent congestion and duplications of event notifications. Will there be a hierarchy of faults, and will the fault reporting be done by each fault in the hierarchy, or will only the lowest fault be reported and the higher levels be suppressed? Should there be aggregated status indicators based on concatenation of propagated faults from a given domain or device?

SNMP notifications and syslog messages can alert an operator when an aspect of the new protocol fails or encounters an error or failure condition, and SNMP is frequently used as a heartbeat monitor. Should the event reporting provide guaranteed accurate delivery of the event information within a given (high) margin of confidence? Can we poll the latest events in the box?

3.3.1. Liveness Detection and Monitoring

Protocol designers should always build in basic testing features (e.g., ICMP echo, UDP/TCP echo service, NULL RPCs (remote procedure calls)) that can be used to test for liveness, with an option to enable and disable them.

Mechanisms for monitoring the liveness of the protocol and for detecting faults in protocol connectivity are usually built into protocols. In some cases, mechanisms already exist within other protocols responsible for maintaining lower-layer connectivity (e.g., ICMP echo), but often new procedures are required to detect failures and to report rapidly, allowing remedial action to be taken.

These liveness monitoring mechanisms do not typically require additional management capabilities. However, when a system detects a fault, there is often a requirement to coordinate recovery action through management applications or at least to record the fact in an event log.

3.3.2. Fault Determination

It can be helpful to describe how faults can be pinpointed using management information. For example, counters might record instances of error conditions. Some faults might be able to be pinpointed by comparing the outputs of one device and the inputs of another device, looking for anomalies. Protocol designers should consider what counters should count. If a single counter provided by vendor A counts three types of error conditions, while the corresponding counter provided by vendor B counts seven types of error conditions, these counters cannot be compared effectively -- they are not interoperable counters.

How do you distinguish between faulty messages and good messages?

Would some threshold-based mechanisms, such as Remote Monitoring (RMON) events/alarms or the EVENT-MIB, be usable to help determine error conditions? Are SNMP notifications for all events needed, or are there some "standard" notifications that could be used? Or can relevant counters be polled as needed?

3.3.3. Root Cause Analysis

Root cause analysis is about working out where in the network the fault is. For example, if end-to-end data delivery is failing (reported by a notification), root cause analysis can help find the failed link or node in the end-to-end path.

3.3.4. Fault Isolation

It might be useful to isolate or quarantine faults, such as isolating a device that emits malformed messages that are necessary to coordinate connections properly. This might be able to be done by configuring next-hop devices to drop the faulty messages to prevent them from entering the rest of the network.

3.4. Configuration Management

A protocol designer should document the basic configuration parameters that need to be instrumented for a new protocol, as well as default values and modes of operation.

What information should be maintained across reboots of the device, or restarts of the management system?

"Requirements for Configuration Management of IP-based Networks" [RFC3139] discusses requirements for configuration management, including discussion of different levels of management, high-level policies, network-wide configuration data, and device-local configuration. Network configuration is not just multi-device push or pull. It is knowing that the configurations being pushed are semantically compatible. Is the circuit between them configured compatibly on both ends? Is the IS-IS metric the same? ... Now answer those questions for 1,000 devices.

A number of efforts have existed in the IETF to develop policy-based configuration management. "Terminology for Policy-Based Management" [RFC3198] was written to standardize the terminology across these efforts.

Implementations should not arbitrarily modify configuration data. In some cases (such as access control lists (ACLs)), the order of data items is significant and comprises part of the configured data. If a protocol designer defines mechanisms for configuration, it would be desirable to standardize the order of elements for consistency of configuration and of reporting across vendors and across releases from vendors.

There are two parts to this:

1. A Network Management System (NMS) could optimize ACLs for performance reasons.

2. Unless the device/NMS systems has correct rules / a lot of experience, reordering ACLs can lead to a huge security issue.

Network-wide configurations may be 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. Many operators consider it desirable to extract, document, and standardize the common parts of these network-wide configuration database schemas. A protocol designer should consider how to standardize the common parts of configuring the new protocol, while recognizing that vendors may also have proprietary aspects of their configurations.

It is important to enable operators to concentrate on the configuration of the network as a whole, rather than individual devices. Support for configuration transactions across a number of devices could significantly simplify network configuration management. The ability to distribute configurations to multiple devices, or to modify candidate configurations on multiple devices, and then activate them in a near-simultaneous manner might help. Protocol designers can consider how it would make sense for their protocol to be configured across multiple devices. Configuration templates might also be helpful.

Consensus of the 2002 IAB Workshop [RFC3535] was that textual configuration files should be able to contain international characters. Human-readable strings should utilize UTF-8, and protocol elements should be in case-insensitive ASCII.

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.

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.

A protocol designer should consider the configurable items that exist for the control of function via the protocol elements described in the protocol specification. For example, sometimes the protocol

requires that timers can be configured by the operator to ensure specific policy-based behavior by the implementation. These timers should have default values suggested in the protocol specification and may not need to be otherwise configurable.

3.4.1. Verifying Correct Operation

An important function that should be provided is guidance on how to verify the correct operation of a protocol. A protocol designer could suggest techniques for testing the impact of the protocol on the network before it is deployed as well as techniques for testing the effect that the protocol has had on the network after being deployed.

Protocol designers should consider how to test the correct end-to-end operation of the service or network, how to verify the correct functioning of the protocol, and whether that is verified by testing the service function and/or by testing the forwarding function of each network element. This may be achieved through status and statistical information gathered from devices.

3.5. Accounting Management

A protocol designer should consider whether it would be appropriate to collect usage information related to this protocol and, if so, what usage information would be appropriate to collect.

"Introduction to Accounting Management" [RFC2975] discusses a number of factors relevant to monitoring usage of protocols for purposes of capacity and trend analysis, cost allocation, auditing, and billing. The document also discusses how some existing protocols can be used for these purposes. These factors should be considered when designing a protocol whose usage might need to be monitored or when recommending a protocol to do usage accounting.

3.6. Performance Management

From a manageability point of view, it is important to determine how well a network deploying the protocol or technology defined in the document is doing. In order to do this, the network operators need to consider information that would be useful to determine the performance characteristics of a deployed system using the target protocol.

The IETF, via the Benchmarking Methodology WG (BMWG), has defined recommendations for the measurement of the performance characteristics of various internetworking technologies in a laboratory environment, including the systems or services that are

built from these technologies. Each benchmarking recommendation describes the class of equipment, system, or service being addressed; discusses the performance characteristics that are pertinent to that class; clearly identifies a set of metrics that aid in the description of those characteristics; specifies the methodologies required to collect said metrics; and lastly, presents the requirements for the common, unambiguous reporting of benchmarking results. Search for "benchmark" in the RFC search tool.

Performance metrics may be useful in multiple environments and for different protocols. The IETF, via the IP Performance Monitoring (IPPM) WG, has developed a set of standard metrics that can be applied to the quality, performance, and reliability of Internet data delivery services. These metrics are designed such that they can be performed by network operators, end users, or independent testing groups. The existing metrics might be applicable to the new protocol. Search for "metric" in the RFC search tool. In some cases, new metrics need to be defined. It would be useful if the protocol documentation identified the need for such new metrics. For performance monitoring, it is often important to report the time spent in a state, rather than reporting the current state. Snapshots are of less value for performance monitoring.

There are several parts to performance management to be considered: protocol monitoring, device monitoring (the impact of the new protocol / service activation on the device), network monitoring, and service monitoring (the impact of service activation on the network).

3.6.1. Monitoring the Protocol

Certain properties of protocols are useful to monitor. The number of protocol packets received, the number of packets sent, and the number of packets dropped are usually very helpful to operators.

Packet drops should be reflected in counter variable(s) somewhere that can be inspected -- both from the security point of view and from the troubleshooting point of view.

Counter definitions should be unambiguous about what is included in the count and what is not included in the count.

Consider the expected behaviors for counters -- what is a reasonable maximum value for expected usage? Should they stop counting at the maximum value and retain the maximum value, or should they rollover? How can users determine if a rollover has occurred, and how can users determine if more than one rollover has occurred?

Consider whether multiple management applications will share a counter; if so, then no one management application should be allowed to reset the value to zero since this will impact other applications.

Could events, such as hot-swapping a blade in a chassis, cause discontinuities in counter? Does this make any difference in evaluating the performance of a protocol?

The protocol document should make clear the limitations implicit within the protocol and the behavior when limits are exceeded. This should be considered in a data-modeling-independent manner -- what makes managed-protocol sense, not what makes management-protocol-sense. If constraints are not managed-protocol-dependent, then it should be left for the management-protocol data modelers to decide. For example, VLAN identifiers have a range of 1..4095 because of the VLAN standards. A MIB implementing a VLAN table should be able to support 4096 entries because the content being modeled requires it.

3.6.2. Monitoring the Device

Consider whether device performance will be affected by the number of protocol entities being instantiated on the device. Designers of an information model should include information, accessible at runtime, about the maximum number of instances an implementation can support, the current number of instances, and the expected behavior when the current instances exceed the capacity of the implementation or the capacity of the device.

Designers of an information model should model information, accessible at runtime, about the maximum number of protocol entity instances an implementation can support on a device, the current number of instances, and the expected behavior when the current instances exceed the capacity of the device.

3.6.3. Monitoring the Network

Consider whether network performance will be affected by the number of protocol entities being deployed.

Consider the capability of determining the operational activity, such as the number of messages in and the messages out, the number of received messages rejected due to format problems, and the expected behaviors when a malformed message is received.

What are the principal performance factors that need to be looked at when measuring the operational performance of the network built using the protocol? Is it important to measure setup times? End-to-end connectivity? Hop-to-hop connectivity? Network throughput?

3.6.4. Monitoring the Service

What are the principal performance factors that need to be looked at when measuring the performance of a service using the protocol? Is it important to measure application-specific throughput? Client-server associations? End-to-end application quality? Service interruptions? User experience?

3.7. Security Management

Protocol designers should consider how to monitor and manage security aspects and vulnerabilities of the new protocol.

There will be security considerations related to the new protocol. To make it possible for operators to be aware of security-related events, it is recommended that system logs should record events, such as failed logins, but the logs must be secured.

Should a system automatically notify operators of every event occurrence, or should an operator-defined threshold control when a notification is sent to an operator?

Should certain statistics be collected about the operation of the new protocol that might be useful for detecting attacks, such as the receipt of malformed messages, messages out of order, or messages with invalid timestamps? If such statistics are collected, is it important to count them separately for each sender to help identify the source of attacks?

Manageability considerations that are security-oriented might include discussion of the security implications when no monitoring is in place, the regulatory implications of absence of audit-trail or logs in enterprises, exceeding the capacity of logs, and security exposures present in chosen/recommended management mechanisms.

Consider security threats that may be introduced by management operations. For example, Control and Provisioning of Wireless Access Points (CAPWAP) breaks the structure of monolithic Access Points (APs) into Access Controllers and Wireless Termination Points (WTPs). By using a management interface, internal information that was previously not accessible is now exposed over the network and to management applications and may become a source of potential security threats.

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.

Some operators wish to do consistency checks of access control lists across devices. Protocol designers should consider information models to promote comparisons across devices and across vendors to permit checking the consistency of security configurations.

Protocol designers should consider how to provide a secure transport, authentication, identity, and access control that integrates well with existing key and credential management infrastructure. It is a good idea to start with defining the threat model for the protocol, and from that deducing what is required.

Protocol designers should consider how access control lists are maintained and updated.

Standard SNMP notifications or syslog messages [RFC5424] might already exist, or can be defined, to alert operators to the conditions identified in the security considerations for the new protocol. For example, you can log all the commands entered by the operator using syslog (giving you some degree of audit trail), or you can see who has logged on/off using the Secure SHell Protocol (SSH) and from where; failed SSH logins can be logged using syslog, etc.

An analysis of existing counters might help operators recognize the conditions identified in the security considerations for the new protocol before they can impact the network.

Different management protocols use different assumptions about message security and data-access controls. A protocol designer that recommends using different protocols should consider how security will be applied in a balanced manner across multiple management interfaces. SNMP authority levels and policy are data-oriented, while CLI authority levels and policy are usually command-oriented (i.e., task-oriented). Depending on the management function, sometimes data-oriented or task-oriented approaches make more sense. Protocol designers should consider both data-oriented and task-oriented authority levels and policy.

4. Documentation Guidelines

This document is focused on what a protocol designer should think about and how those considerations might be documented.

This document does not describe interoperability requirements but rather describes practices that are useful to follow when dealing with manageability aspects in IETF documents, so the capitalized keywords from [RFC2119] do not apply here. Any occurrence of words like 'must' or 'should' needs to be interpreted only in the context of their natural, English-language meaning.

4.1. Recommended Discussions

A Manageability Considerations section should include discussion of the management and operations topics raised in this document, and when one or more of these topics is not relevant, it would be useful to contain a simple statement explaining why the topic is not relevant for the new protocol. Of course, additional relevant topics should be included as well.

Existing protocols and data models can provide the management functions identified in the previous section. Protocol designers should consider how using existing protocols and data models might impact network operations.

4.2. Null Manageability Considerations Sections

A protocol designer may seriously consider the manageability requirements of a new protocol and determine that no management functionality is needed by the new protocol. It would be helpful to those who may update or write extensions to the protocol in the future or to those deploying the new protocol to know the thinking of the working group regarding the manageability of the protocol at the time of its design.

If there are no new manageability or deployment considerations, it is recommended that a Manageability Considerations section contain a simple statement such as, "There are no new manageability requirements introduced by this document," and a brief explanation of why that is the case. The presence of such a Manageability Considerations section would indicate to the reader that due consideration has been given to manageability and operations.

In the case where the new protocol is an extension and the base protocol discusses all the relevant operational and manageability considerations, it would be helpful to point out the considerations section in the base document.

4.3. Placement of Operations and Manageability Considerations Sections

If a protocol designer develops a Manageability Considerations section for a new protocol, it is recommended that the section be placed immediately before the Security Considerations section. Reviewers interested in such sections could find it easily, and this placement could simplify the development of tools to detect the presence of such a section.

5. Security Considerations

This document is informational and provides guidelines for considering manageability and operations. It introduces no new security concerns.

The provision of a management portal to a network device provides a doorway through which an attack on the device may be launched. Making the protocol under development be manageable through a management protocol creates a vulnerability to a new source of attacks. Only management protocols with adequate security apparatus, such as authentication, message integrity checking, and authorization, should be used.

A standard description of the manageable knobs and whistles on a protocol makes it easier for an attacker to understand what they may try to control and how to tweak it.

A well-designed protocol is usually more stable and secure. A protocol that can be managed and inspected offers the operator a better chance of spotting and quarantining any attacks. Conversely, making a protocol easy to inspect is a risk if the wrong person inspects it.

If security events cause logs and/or notifications/alerts, a concerted attack might be able to be mounted by causing an excess of these events. In other words, the security-management mechanisms could constitute a security vulnerability. The management of security aspects is important (see Section 3.7).

6. Acknowledgements

This document started from an earlier document edited by Adrian Farrel, which itself was based on work exploring the need for Manageability Considerations sections in all Internet-Drafts produced within the Routing Area of the IETF. That earlier work was produced by Avri Doria, Loa Andersson, and Adrian Farrel, with valuable feedback provided by Pekka Savola and Bert Wijnen.

Some of the discussion about designing for manageability came from private discussions between Dan Romascanu, Bert Wijnen, Juergen Schoenwaelder, Andy Bierman, and David Harrington.

Thanks to reviewers who helped fashion this document, including Harald Alvestrand, Ron Bonica, Brian Carpenter, Benoit Claise, Adrian Farrel, David Kessens, Dan Romascanu, Pekka Savola, Juergen Schoenwaelder, Bert Wijnen, Ralf Wolter, and Lixia Zhang.

7. Informative References

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.

[RFC1052] Cerf, V., "IAB recommendations for the development of Internet network management standards", RFC 1052, April 1988.

[RFC1958] Carpenter, B., "Architectural Principles of the Internet", RFC 1958, June 1996.

[RFC2113] Katz, D., "IP Router Alert Option", RFC 2113, February 1997.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2205] Braden, B., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, September 1997.

[RFC2439] Villamizar, C., Chandra, R., and R. Govindan, "BGP Route Flap Damping", RFC 2439, November 1998.

[RFC2578] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.

[RFC2711] Partridge, C. and A. Jackson, "IPv6 Router Alert Option", RFC 2711, October 1999.

[RFC2865] Rigney, C., Willens, S., Rubens, A., and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", RFC 2865, June 2000.

[RFC2975] Aboba, B., Arkko, J., and D. Harrington, "Introduction to Accounting Management", RFC 2975, October 2000.

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

[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.

[RFC3139] Sanchez, L., McCloghrie, K., and J. Saperia, "Requirements for Configuration Management of IP-based Networks", RFC 3139, June 2001.

[RFC3198] Westerinen, A., Schnizlein, J., Strassner, J., Scherling, M., Quinn, B., Herzog, S., Huynh, A., Carlson, M., Perry, J., and S. Waldbusser, "Terminology for Policy-Based Management", RFC 3198, November 2001.

[RFC3290] Bernet, Y., Blake, S., Grossman, D., and A. Smith, "An Informal Management Model for Diffserv Routers", RFC 3290, May 2002.

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

[RFC3444] Pras, A. and J. Schoenwaelder, "On the Difference between Information Models and Data Models", RFC 3444, January 2003.

[RFC3460] Moore, B., "Policy Core Information Model (PCIM) Extensions", RFC 3460, January 2003.

[RFC3535] Schoenwaelder, J., "Overview of the 2002 IAB Network Management Workshop", RFC 3535, May 2003.

[RFC3585] Jason, J., Rafalow, L., and E. Vyncke, "IPsec Configuration Policy Information Model", RFC 3585, August 2003.

[RFC3588] Calhoun, P., Loughney, J., Guttman, E., Zorn, G., and J. Arkko, "Diameter Base Protocol", RFC 3588, September 2003.

[RFC3644] Snir, Y., Ramberg, Y., Strassner, J., Cohen, R., and B. Moore, "Policy Quality of Service (QoS) Information Model", RFC 3644, November 2003.

[RFC3670] Moore, B., Durham, D., Strassner, J., Westerinen, A., and W. Weiss, "Information Model for Describing Network Device QoS Datapath Mechanisms", RFC 3670, January 2004.

[RFC3805] Bergman, R., Lewis, H., and I. McDonald, "Printer MIB v2", RFC 3805, June 2004.

[RFC4741] Enns, R., "NETCONF Configuration Protocol", RFC 4741, December 2006.

[RFC5101] Claise, B., "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information", RFC 5101, January 2008.

[RFC5321] Klensin, J., "Simple Mail Transfer Protocol", RFC 5321, October 2008.

[RFC5424] Gerhards, R., "The Syslog Protocol", RFC 5424, March 2009.

[W3C.REC-xmlschema-0-20010502]
Fallside, D., "XML Schema Part 0: Primer", World Wide Web Consortium FirstEdition REC-xmlschema-0-20010502, May 2001, <http://www.w3.org/TR/2001/REC-xmlschema-0-20010502>.

Appendix A. Operations and Management Review Checklist

This appendix provides a quick checklist of issues that protocol designers should expect operations and management expert reviewers to look for when reviewing a document being proposed for consideration as a protocol standard.

A.1. Operational Considerations

1. Has deployment been discussed? See Section 2.1.

   * Does the document include a description of how this protocol or technology is going to be deployed and managed?

   * Is the proposed specification deployable? If not, how could it be improved?

   * Does the solution scale well from the operational and management perspective? Does the proposed approach have any scaling issues that could affect usability for large-scale operation?

   * Are there any coexistence issues?

2. Has installation and initial setup been discussed? See Section 2.2.

   * Is the solution sufficiently configurable?

   * Are configuration parameters clearly identified?

   * Are configuration parameters normalized?

   * Does each configuration parameter have a reasonable default value?

   * Will configuration be pushed to a device by a configuration manager, or pulled by a device from a configuration server?

   * How will the devices and managers find and authenticate each other?

3. Has the migration path been discussed? See Section 2.3.

   * Are there any backward compatibility issues?

4. Have the Requirements on other protocols and functional components been discussed? See Section 2.4.

   * What protocol operations are expected to be performed relative to the new protocol or technology, and what protocols and data models are expected to be in place or recommended to ensure for interoperable management?

5. Has the impact on network operation been discussed? See Section 2.5.

   * Will the new protocol significantly increase traffic load on existing networks?

   * Will the proposed management for the new protocol significantly increase traffic load on existing networks?

   * How will the new protocol impact the behavior of other protocols in the network? Will it impact performance (e.g., jitter) of certain types of applications running in the same network?

   * Does the new protocol need supporting services (e.g., DNS or Authentication, Authorization, and Accounting - AAA) added to an existing network?

6. Have suggestions for verifying correct operation been discussed? See Section 2.6.

   * How can one test end-to-end connectivity and throughput?

   * Which metrics are of interest?

   * Will testing have an impact on the protocol or the network?

7. Has management interoperability been discussed? See Section 3.1.

   * Is a standard protocol needed for interoperable management?

   * Is a standard information or data model needed to make properties comparable across devices from different vendors?

8. Are there fault or threshold conditions that should be reported? See Section 3.3.

   * Does specific management information have time utility?

   * Should the information be reported by notifications? Polling? Event-driven polling?

   * Is notification throttling discussed?

* Is there support for saving state that could be used for root cause analysis?

9. Is configuration discussed? See Section 3.4.

* Are configuration defaults and default modes of operation considered?

* Is there discussion of what information should be preserved across reboots of the device or the management system? Can devices realistically preserve this information through hard reboots where physical configuration might change (e.g., cards might be swapped while a chassis is powered down)?

A.2. Management Considerations

Do you anticipate any manageability issues with the specification?

1. Is management interoperability discussed? See Section 3.1.

* Will it use centralized or distributed management?

* Will it require remote and/or local management applications?

* Are textual or graphical user interfaces required?

* Is textual or binary format for management information preferred?

2. Is management information discussed? See Section 3.2.

* What is the minimal set of management (configuration, faults, performance monitoring) objects that need to be instrumented in order to manage the new protocol?

3. Is fault management discussed? See Section 3.3.

* Is Liveness Detection and Monitoring discussed?

* Does the solution have failure modes that are difficult to diagnose or correct? Are faults and alarms reported and logged?

4. Is configuration management discussed? See Section 3.4.

* Is protocol state information exposed to the user? How? Are significant state transitions logged?

5. Is accounting management discussed? See Section 3.5.

6. Is performance management discussed? See Section 3.6.

   * Does the protocol have an impact on network traffic and network devices? Can performance be measured?

   * Is protocol performance information exposed to the user?

7. Is security management discussed? See Section 3.7.

   * Does the specification discuss how to manage aspects of security, such as access controls, managing key distribution, etc.

A.3. Documentation

Is an operational considerations and/or manageability section part of the document?

Does the proposed protocol have a significant operational impact on the Internet?

Is there proof of implementation and/or operational experience?

Author's Address

David Harrington
HuaweiSymantec USA
20245 Stevens Creek Blvd
Cupertino, CA 95014
USA

Phone: +1 603 436 8634
EMail: ietfdbh@comcast.net