Network Working Group
Internet-Draft
Obsoletes: 5706 (if approved)
Intended status: Best Current Practice
Expires: 17 May 2026

B. Claise
Everything OPS
J. Clarke
Cisco
A. Farrel
Old Dog Consulting
S. Barguil
Nokia
C. Pignataro
Blue Fern Consulting
R. Chen
ZTE

13 November 2025

Guidelines for Considering Operations and Management in IETF Specifications

draft-ietf-opsawg-rfc5706bis-00

Abstract

New Protocols or Protocol Extensions are best designed with due consideration of the functionality needed to operate and manage them. Retrofitting operations and management considerations is suboptimal. The purpose of this document is to provide guidance to authors and reviewers on what operational and management aspects should be addressed when defining New Protocols or Protocol Extensions.

This document obsoletes RFC 5706, replacing it completely and updating it with new operational and management techniques and mechanisms. It also introduces a requirement to include an "Operational Considerations" section in new RFCs in the IETF Stream.

Status of This Memo

This Internet-Draft will expire on 17 May 2026.

Table of Contents

1. Introduction

Often, when New Protocols or Protocol Extensions are developed, not enough consideration is given to how they 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 difficult. To ensure deployability, the operational environment and manageability must be considered during design.

This document provides guidelines to help Protocol Designers and working groups (WGs) consider the operations and management functionality for their New Protocol or Protocol Extension at an early phase in the design process.

This document obsoletes [RFC5706] and fully updates its content with new operational and management techniques and mechanisms. It also introduces a requirement for an "Operational Considerations" section in new RFCs in the IETF Stream. This document also removes outdated references and aligns with current practices, protocols, and technologies used in operating and managing devices, networks, and services. See Appendix A for more details.

1.1. This Document

This document provides a set of guidelines for considering operations and management in an IETF technical specification with an eye toward being flexible while also striving for interoperability.

Entirely New Protocols may require significant consideration of expected operations and management, while Protocol Extensions to existing, widely deployed protocols may have established de facto operations and management practices that are already well understood. This document does not mandate a comprehensive inventory of all operational considerations. Instead, it guides authors to focus on key aspects that are essential for the technology's deployability, operation, and maintenance.

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, given the IETF's role in developing technologies, New Protocols, and Protocol Extensions to be deployed and operated in the real-world Internet.

A WG may decide that its protocol does not need interoperable management or a standardized Data Model, but this should be a deliberate and documented decision, not the result of omission. This document provides some guidelines for those considerations.

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

1.2. Audience

The guidelines are intended to be useful to authors writing protocol specifications. They outline what to consider for management and deployment, how to document those aspects, and how to present them in a consistent format. This document is intended to offer a flexible set of guiding principles applicable to various circumstances. It provides a framework for working groups to ensure that manageability considerations are an integral part of the protocol design process, and its use should not be misinterpreted as imposing new hurdles on work in other areas.

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 WG that considers which security threats are relevant to their protocol, documents (in the required Security Considerations section, per Guidelines for Writing RFC Text on Security Considerations [BCP72]) how threats should be mitigated, and then suggests appropriate standard protocols that could mitigate the threats.

A core principle of this document is to encourage early on discussions rather than mandating any specific solution. It does not impose a specific management or operational 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 Interfaces or that it does not need any structured or standardized Data Model, 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 deferring manageability to a later phase of the development of the specification.

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, operated, and managed. The considerations do not need to be comprehensive and exhaustive; focus should be on key aspects. 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.

The Ops Directorate (OpsDir) can use this document to inform their reviews. A list of guidelines and a checklist of questions to consider, which a reviewer can use to evaluate whether the protocol and documentation address common operations and management needs, is provided in [CHECKLIST]. Ultimately, the decision to incorporate this document’s advice into their work remains with Protocol Designers and working groups themselves.

This document is also of interest to the broader community, who wants to understand, contribute to, and review Internet-Drafts, taking operational considerations into account.

2. Terminology

This document does not describe interoperability requirements. As such, it does not use the capitalized keywords defined in [BCP14].

This section defines key terms used throughout the document to ensure clarity and consistency. Some terms are drawn from existing RFCs and IETF Internet-Drafts, while others are defined here for the purposes of this document. Where appropriate, references are provided for further reading or authoritative definitions.

* Anomaly: See [I-D.ietf-nmop-terminology].

* Cause: See [I-D.ietf-nmop-terminology].

* CLI: Command Line Interface. A human-oriented interface, typically a Proprietary Interface, to hardware or software devices (e.g., hosts, routers, or operating systems). The commands, their syntax, and the precise semantics of the parameters may vary considerably between different vendors, between products from the same vendor, and even between different versions or releases of a single product. No attempt at standardizing CLIs has been made by the IETF.

* Data Model: A set of mechanisms for representing, organizing, storing, and handling data within a particular type of data store or repository. This usually comprises a collection of data structures such as lists, tables, relations, etc., a collection of operations that can be applied to the structures such as retrieval, update, summation, etc., and a collection of integrity rules that define the legal states (set of values) or changes of state (operations on values). A Data Model may be derived by mapping the contents of an Information Model or may be developed ab initio. Further discussion of Data Models can be found in [RFC3444], Section 5.2, and Section 5.3.

* Fault: See [I-D.ietf-nmop-terminology].

* Fault Management: The process of interpreting fault notifications and other alerts and alarms, isolating faults, correlating them, and deducing underlying Causes. See Section 5.4 for more information.

* 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. The model is independent of any specific software usage, protocol, or platform [RFC3444]. See Sections 5.2 and 5.3.1 for further discussion of Information Models.

* New Protocol and Protocol Extension: These terms are used in this document to identify entirely new protocols, new versions of existing protocols, and extensions to protocols.

* OAM: Operations, Administration, and Maintenance [RFC6291] [I-D.ietf-opsawg-oam-characterization] is the term given to the combination of:

  1. Operation activities that are undertaken to keep the network running as intended. They include monitoring of the network.

  2. Administration activities that keep track of resources in the network and how they are used. They include the bookkeeping necessary to track networking resources.

  3. Maintenance activities focused on facilitating repairs and upgrades. They also involve corrective and preventive measures to make the managed network run more effectively.

  The broader concept of "operations and management" that is the subject of this document encompasses OAM, in addition to other management and provisioning tools and concepts.

* Probable Root Cause: See [I-D.ietf-nmop-network-incident-yang]

* Problem: See [I-D.ietf-nmop-terminology].

* Proprietary Interface: An interface to manage a network element that is not standardized. As such, the user interface, syntax, and semantics typically vary significantly between implementations. Examples of proprietary interfaces include Command Line Interface (CLI), management web portal and Browser User Interface (BUI), Graphical User Interface (GUI), and vendor-specific application programming interface (API).

* Protocol Designer: An individual, a group of people, or an IETF WG involved in the development and specification of New Protocols or Protocol Extensions.

3. Documentation Requirements for IETF Specifications

3.1. "Operational Considerations" Section

All Internet-Drafts that document a technical specification and are advanced for publication as IETF RFCs are required to include an "Operational Considerations" section. Internet-Drafts that do not document technical specifications such as process, policy, or administrative Internet-Drafts are not required to include such a section.

After evaluating the operational (Section 4) and manageability aspects (Section 5) of a New Protocol, a Protocol Extension, or an architecture, the resulting practices and requirements should be documented in an "Operational Considerations" section within a specification. Since protocols are intended for operational deployment and management within real networks, it is expected that such considerations will be present.

It is also recommended that operational and manageability considerations be addressed early in the protocol design process. Consequently, early revisions of Internet-Drafts are expected to include an "Operational Considerations" section.

An "Operational 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 include a simple statement explaining why the topic is not relevant or applicable for the New Protocol or Protocol Extension. Of course, additional relevant operational and manageability topics should be included as well.

3.2. "Operational Considerations" Section Boilerplate When No New Considerations Exist

After a Protocol Designer has considered the manageability requirements of a New Protocol or Protocol Extension, they may determine that no management functionality or operational best-practice clarifications are needed. It would be helpful to reviewers, those who may update or write extensions to the protocol in the future, or to those deploying the protocol, to know the rationale regarding the decisions on manageability of the protocol at the time of its design.

If there are no new manageability or deployment considerations, the "Operational Considerations" section must contain the following simple statement, followed by a brief explanation of why that is the case.

"There are no new operations or manageability requirements introduced by this document.

Explanation: [brief rationale goes here]"

The presence of such a section would indicate to the reader that due consideration has been given to manageability and operations.

In cases where the specification is a Protocol Extension and the base protocol already addresses the relevant operational and manageability considerations, it is helpful to reference the considerations section in the base document.

3.3. Placement of the "Operational Considerations" Section

It is recommended that the section be placed immediately before the Security Considerations section. Reviewers interested in such sections will find it easily, and this placement could simplify the development of tools to detect the presence of such a section.

4. How Will the New Protocol or Protocol Extension Fit into the Current Environment?

Designers of a New Protocol or Protocol Extension 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. [RFC5218] provides a more detailed discussion on what makes for a successful protocol.

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.

4.1. Operations

Protocol Designers can analyze the operational environment and mode of work in which the New Protocol and Protocol 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 both a human interface (e.g., for troubleshooting) and a programmatic interface (e.g., for automated monitoring and Cause analysis). The application programming interfaces (APIs) and the human interfaces might benefit from being similar to ensure that the information exposed by both is consistent when presented to an operator. It is also relevant to identify consistent methods for determining information, such as what is counted in specific counters.

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

4.2. 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 or service infrastructure as a whole rather than on individual devices. Of course, how one accomplishes this is the hard part.

Protocol Designers should explain the background of chosen default values and provide the rationale, especially when those choices may affect operations. 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 cryptographic 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.

Default values should generally favor the conservative side over the "optimizing performance" side (e.g., the initial Round-Trip Time (RTT) and Round-Trip Time Variance (RTTVAR) values of a TCP connection [RFC6298]).

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.

For example, where protocols involve cryptographic keys, Protocol Designers should consider not only key generation and validation mechanisms but also the format in which private keys are stored, transmitted, and restored. Designers should specify any expected consistency checks (e.g., recomputing an expanded key from the seed) that help verify correctness and integrity. Additionally, guidance should be given on data retention, restoration limits, and cryptographic module interoperability when importing/exporting private key material. See [I-D.ietf-lamps-dilithium-certificates] for an example of how such considerations are incorporated.

4.3. Migration Path

If the New Protocol or Protocol Extension 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 or Protocol Extensions. This should include coexistence with previously deployed protocols and/or previous versions of the same protocol, management of incompatibilities between versions, translation between versions, and consideration of potential side effects. A key question becomes: Are older protocols or versions disabled, or do they coexist in the network with the New Protocol or Protocol Extension?

Many protocols benefit from being incrementally deployable -- operators may deploy aspects of a protocol before deploying the protocol fully. In those cases, the design considerations should also specify whether the New Protocol or Protocol Extension requires any changes to the existing infrastructure, particularly the network. If so, the protocol specification should describe the nature of those changes, where they are required, and how they can be introduced in a manner that facilitates deployment.

Incentivizing good security operation practices when migrating to the New Protocol or Protocol Extension should be encouraged. For example, patching is fundamental for security operations and can be incentivized if Protocol Designers consider supporting cheap and fast connection hand-offs and reconnections.

When Protocol Designers are considering how deployments should transition to the New Protocol or Protocol Extension, impacts to current techniques employed by operators should be documented and mitigations included, where possible, so that consistent security operations and management can be achieved. Refer to [RFC8170] for a detailed discussed on transition versus coexistence.

4.4. 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 components 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.

4.5. Impact on Network Operation

The introduction of a New Protocol or Protocol Extensions may have an impact on the operation of existing networks. As discussed in Section 2.1 of [RFC6709] major extensions may have characteristics leading to a risk of operational problems. Protocol Designers should outline such operational impacts (which may be positive), including scaling benefits or concerns, and interactions with other protocols. Protocol Designers should describe the scenarios in which the New Protocol or its extensions are expected to be applicable or beneficial. This includes any relevant deployment environments, network topologies, usage constraints such as limited domains [RFC8799], or use cases that justify or constrain adoption. For example, a New Protocol or Protocol Extension that doubles the number of active, reachable addresses in a network might have implications for the scalability of interior gateway protocols, and such impacts should be evaluated accordingly.

If the protocol specification requires changes to end hosts, it should also indicate whether safeguards exist to protect networks from potential overload. For instance, a congestion control algorithm must comply with [BCP133] to prevent congestion collapse and ensure network stability.

A protocol could send active monitoring packets on the wire. Without careful consideration, active monitoring might achieve high accuracy at the cost of generating an excessive number of monitoring packets.

Protocol Designers 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.

Protocol Designers 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 of New Protocols or Protocol Extensions, and the results of new OAM tools developed for the New Protocols or Protocol Extensions, must be considered with respect to the performance of traffic delivery and the availability of ongoing manageability. For example, it must be noted if the New Protocols or Protocol Extensions or the OAM tools cause increased delay or jitter in real-time traffic applications, or increased response time in client-server applications. Further, if the additional traffic caused by OAM tools and data collection could result in the management plane becoming overwhelmed, then this must be called out, and suitable mechanisms to rate limit the OAM must be considered. Potential options include: document the limitations, propose solution track(s), include an optional rate limiting feature in the specifications, or impose a rate limiting feature in the specifications. Consider three examples: in Bidirectional Forwarding Detection for MPLS [RFC5884] it is possible to configure very rapid BFD transmissions (of the order of 3ms) on a very large number of parallel Label Switched Paths (LSPs) with the result that the management systems and end nodes may become overwhelmed -- this can be protected by applying limits to the number of LSPs that may be tested at once; notifications or logs from systems (through YANG or other means) should be rate-limited so that they do not flood the receiving management station; the application of sophisticated encryption or filtering rules need to be considered in the light of the additional processing they may impose on the hardware forwarding path for traffic.

New metrics may be required to assess traffic performance. Protocol Designers may refer to [RFC6390] for guidelines for considering new performance metrics.

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.

4.6. Impact on Security Operations

Security Operations (SecOps) is a collaborative approach that combines security and operational teams to improve the ability of operators to protect and manage the network effectively and efficiently[SECOPS]. Security operators detect malicious activity and respond to threats and are a crucial part of defending against attacks alongside the management and operation of the network.

Protocol Designers should consider the impacts of a New Protocol or Protocol Extension on Security Operations in networks that the protocol will be deployed in.

Security operators extensively rely upon Indicators of Compromise (IoCs) [RFC9424]. The deployment of New Protocol or Protocol Extension may change the type, locations, or availability of IoCs. Protocol Designers should outline such changes to ensure operators can manage and defend their network consistently. Consider the operators' requirement for digital forensics from the network or endpoints with critical information found in logs. Logging events schema and guidance for operators should be considered when designing a New Protocol or Protocol Extension to ensure that operators have the information they need. [I-D.ietf-quic-qlog-main-schema] is an example of extensible structured logging.

Tooling required by security operators should be documented in the design and deployment of a New Protocol or Protocol Extension. Operators may require new tooling or methods for managing network traffic in response to protocol changes to ensure consistent availability and performance of networks. Similarly, updating and augmenting existing forensic tools such as protocol dissectors is expected when a New Protocol is deployed, but having to completely rebuild such tooling would greatly reduce the effectiveness of security operators, so protocol extensibility should be considered.

4.7. Verifying Correct Operation

Protocol Designers 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 (a.k.a., active monitoring). Protocol Designers should consider how the correct end- to-end operation of the New Protocol or Protocol Extension 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 or Protocol Extension. Which metrics are of interest?

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

4.8. Message Formats

Where protocol specifications result in messages (such as errors or warnings) being carried as text strings or output for consumption by human operators, consideration should be given to making it possible for implementations to be configured so that the messages can be viewed in the local language. In such cases, it may be helpful to transmit a specific message code (i.e., a number) along with the default English language message, so that implementations may easily map the code to a local text string.

Further discussion of Internationalization issues may be found in [BCP166].

5. 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 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 scenario reflects a common operational concern: the inability to manage both ends of a connection effectively. As noted in [RFC3535], "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 or Protocol Extension will be managed in different deployment scales. It might be sensible to use a local management interface to manage the New Protocol or Protocol Extension 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 or Protocol Extensions.

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 or Protocol Exension. 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 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.

5.1. Available Management Technologies

The IETF provides several standardized management protocols suitable for various operational purposes, for example as outlined in [RFC6632].

Readers seeking more in-depth definitions or explanations should consult the referenced materials.

5.2. Interoperability

Management interoperability is critical for enabling information exchange and operations across diverse network devices and management applications, regardless of vendor, model, or software release. It facilitates the use of third-party applications and outsourced management services.

While individual device management via Proprietary Interfaces may suffice for small deployments, large-scale networks comprising equipment from multiple vendors necessitate consistent, automated management. Relying on vendor- and model-specific interfaces for extensive deployments, such as hundreds of branch offices, severely impedes scalability and automation of operational processes. The primary goal of management interoperability is to enable the scalable deployment and lifecycle management of new network functions and services, while ensuring a clear understanding of their operational impact and total cost of ownership.

Achieving universal agreement on a single management syntax and protocol is challenging. However, the IETF has significantly evolved its approach to network management, moving beyond SMIv2 and SNMP. Modern IETF management solutions primarily leverage YANG [RFC7950] for Data Modeling and NETCONF [RFC6241] or RESTCONF [RFC8040] for protocol interactions. This shift, as further elaborated in [RFC6632], emphasizes structured Data Models and programmatic interfaces to enhance automation and interoperability. Other protocols, such as IPFIX [RFC7011] for flow accounting and syslog [RFC5424] for logging, continue to play specific roles in comprehensive network management.

Interoperability must address both syntactic and semantic aspects. While syntactic variations across implementations can often be handled through adaptive processing, semantic differences pose a greater challenge, as the meaning of data is intrinsically tied to the managed entity.

Information Models (IMs) enable and provide the foundation for semantic interoperability. An IM defines the conceptual understanding of managed information, independent of specific protocols or vendor implementations. This allows for consistent interpretation and correlation of data across different data models (and hence management protocols), such as a YANG Data Model and IPFIX Information Elements concerning the same event. For instance, an IM can standardize how error conditions are counted, ensuring that a counter has the same meaning whether collected via NETCONF or exported via IPFIX.

Protocol Designers should consider developing an IM, when multiple Data Model (DM) representations (e.g., YANG and/or IPFIX) are required, to ensure lossless semantic mapping. IMs are also beneficial for complex or numerous DMs. As illustrated in Figure 1, an IM serves as a conceptual blueprint for designers and operators, from which concrete DMs are derived for implementers. [RFC3444] provides further guidance on distinguishing IMs from DMs.

```
                 IM                --> conceptual/abstract model
                 |                     for designers & operators
      +----------+---------+
      |          |         |
      DM         DM        DM      --> concrete/detailed model
                                          for implementers
```

Figure 1: Information Models (IMs) and Data Models (DMs)

Protocol Designers must identify the essential operational, configuration, state, and statistical information required for effective monitoring, control, and troubleshooting of New Protocols or Protocol Extensions and their extensions. This includes defining relevant parameters, performance metrics, error indicators, and contextual data crucial for diagnostics and lifecycle management.

To ensure interoperability, management protocol and Data Model standards should incorporate clear compliance clauses, specifying the expected level of support.

5.3. Management Information

Languages used to describe an Information Model can influence the nature of the model. Using a particular data modeling language, such as YANG, influences the model to use certain types of structures, for example, hierarchical trees, groupings, and reusable types. YANG, as described in [RFC6020] and [RFC7950], provides advantages for expressing network information, including clear separation of configuration data and operational state, support for constraints and dependencies, and extensibility for evolving requirements. Its ability to represent relationships and dependencies in a structured and modular way makes it an effective choice for defining management information models.

Although this document recommends using English text (the official language for IETF specifications) to describe an Information Model, including a complementary YANG module helps translate abstract concepts into implementation-specific Data Models. This ensures consistency between the high-level design and practical deployment.

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 instances in lists) share fate? For example, if an instance in list A must exist before a related instance in list B can be created, what happens to the instance in list B if the related instance in list A is deleted? Does the existence of relationships between objects have an impact on fate sharing? YANG's relationships and constraints can help express and enforce these relationships.

5.3.1. 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 make additions only as further objects are needed with the objective of keeping the absolute number of objects as small as possible while still delivering the required function such that there is no duplication between objects and where one piece of information can be derived from the other pieces of information, it is not itself represented as an object.

2. Require that all objects be essential for management.

3. Consider evidence of current use of the managed protocol, and the perceived utility of objects added to the Information Model.

4. Exclude objects that can be derived from others in this or other information models.

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

6. When defining an Information Model using YANG Data Structure Extensions [RFC8791] (thereby keeping it abstract and implementation-agnostic per [RFC3444]) ensure that the Information Model remains simple, modular, and clear by following the authoring guidelines in [I-D.ietf-netmod-rfc8407bis].

7. When illustrating the abstract Information Model, use YANG Tree Diagrams [RFC8340] to provide a simple, standardized, and implementation-neutral model structure.

5.3.2. YANG Data Model Considerations

When considering YANG Data Models for a new specification, there are multiple types of Data Models that may be applicable. The hierarchy and relationship between these types is described in Section 3.5.1 of [I-D.ietf-netmod-rfc8407bis]. A new specification may require or benefit from one or more of these YANG Data Model types.

* Device Models - Also called Network Element Models, represent the configuration, operational state, and notifications of individual devices. These models are designed to distinguish between these types of data and support querying and updating device-specific parameters. Consideration should be given to how device-level models might fit with broader network and service Data Models.

* Network Models - Also called Network Service Models, define abstractions for managing the behavior and relationships of multiple devices and device subsystems within a network. As described in [RFC8199], these models are used to manage network-wide. These abstractions are useful to network operators and applications that interface with network controllers. Examples of network models include the L3VPN Network Model (L3NM) [RFC9182] and the L2VPN Network Model (L2VPN) [RFC9291].

* Service Models - Also called Customer Service Models, defined in [RFC8309], are designed to abstract the customer interface into a service. They consider customer-centric parameters such as Service Level Agreement (SLA) and high-level policy (e.g., network intent). Given that different operators and different customers may have widely-varying business processes, these models should focus on common aspects of a service with strong multi-party consensus. Examples of service models include the L3VPN Service Model (L3SM) [RFC8299] and the L2VPN Service Model (L2SM) [RFC8466].

A common challenge in YANG Data Model development lies in defining the relationships between abstract service or network constructs and the underlying device models. Therefore, when designing YANG modules, it is important to go beyond simply modeling configuration and operational data (i.e., leaf nodes), and also consider how the status and relationships of abstract or distributed constructs can be reflected based on parameters available in the network.

For example, the status of a service may depend on the operational state of multiple network elements to which the service is attached. In such cases, the YANG Data Model (and its accompanying documentation) should clearly describe how service-level status is derived from underlying device-level information. Similarly, it is beneficial to define events (and relevant triggered notifications) that indicate changes in an underlying state, enabling reliable detection and correlation of service-affecting conditions. Including such mechanisms improves the robustness of integrations and helps ensure consistent behavior across implementations.

Specific guidelines to consider when authoring any type of YANG modules are described in [I-D.ietf-netmod-rfc8407bis].

5.4. Fault Management

Protocol Designers should identify and document essential Faults, health indicators, alarms, and events that must be propagated to management applications or exposed through a Data Model. It is also recommended to describe how the Protocol Extension will affect the existing alarms and notification structure of the base Protocol, and to outline the potential impact of misconfigurations of the Protocol Extensions.

Protocol Designers 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 Protocol Designers 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 or Protocol Extension 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?

5.4.1. 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.

5.4.2. 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?

5.4.3. Probable Root Cause Analysis

Probable Root Cause analysis is about working out where the foundational Fault or Problem might be. Since one Fault may give rise to another Fault or Problem, a probable root cause is commonly meant to describe the original, source event or combination of circumstances that is the foundation of all related Faults.

For example, if end-to-end data delivery is failing (e.g., reported by a notification), Probable Root Cause analysis can help find the failed link or node, or mis-configuration, within the end-to-end path.

5.4.4. 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.

5.5. Configuration Management

A Protocol Designer should document the basic configuration parameters that need to be instrumented for a New Protocol or Protocol Extensions, 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 extends beyond simple multi-device push or pull operations. It also involves ensuring that the configurations being pushed are semantically compatible across devices and that the resulting behavior of all involved devices corresponds to the intended behavior. Is the attachment between them configured compatibly on both ends? Is the IS-IS metric the same? ... Now answer those questions for 1,000 devices.

Several 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 preferable 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 or NMS is configured with adequate rules and guided by administrators with extensive experience, reordering ACLs can introduce significant security risks.

Network-wide configurations may be stored in central master databases and transformed into readable formats that can be pushed to devices, either by generating sequences of CLI commands or complete textual 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 operationally beneficial 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 or service as a whole, rather than individual devices. Support for configuration transactions across several 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 highly beneficial.

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.

5.5.1. 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.

5.6. 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.

5.7. 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 more important to report the time spent in a state rather than just the current state. Snapshots alone are typically of less value.

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). Hence, it is recommended that, if the implementation of the New Protocol or Protocol Extension has any hardware/software performance implications (e.g., increased CPU utilization, memory consumption, or forwarding performance degradation), the Protocol Designers should clearly describe these impacts in the specification, along with any conditions under which they may occur and possible mitigation strategies.

5.7.1. 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 it, or should they rollover? Guidance should explain how rollovers are detected, including multiple occurrences.

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 specification should clearly define any inherent limitations and describe expected behavior when those limits are exceeded. These considerations should be made independently of any specific management protocol or data modeling language. In other words, focus on what makes sense for the protocol being managed, not the protocol used for management. If a constraint is not specific to a management protocol, then it should be left to Data Model designers of that protocol to determine how to handle it. For example, VLAN identifiers are defined by standard to range from 1 to 4094. Therefore, a YANG "vlan-id" definition representing the 12-bit VLAN ID used in the VLAN Tag header uses a range of "1..4094".

5.7.2. 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 provide runtime information about the maximum supported instances, the current number of instances, and expected behavior when capacity is exceeded.

5.7.3. 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 considered when measuring the operational performance of a network built using the protocol? Is it important to measure setup times, end-to-end connectivity, hop-by-hop connectivity, or network throughput?

5.7.4. Monitoring the Service

What are the principal performance factors that need to be considered 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, or user experience (UX)?

5.8. Security Management

Protocol Designers should consider how to monitor and manage security aspects and vulnerabilities of the New Protocol or Protocol Extension.

There will be security considerations related to the New Protocol or Protocol Extension. 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?

Security-oriented manageability topics may include risks of insufficient monitoring, regulatory issues with missing audit trails, log capacity limits, and security exposures in 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 control protocol or management protocol, 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 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 (SSH) Protocol [RFC4251] 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.

6. Operational and Management Tooling Considerations

The operational community's ability to effectively adopt and use new specifications is significantly influenced by the availability and adaptability of appropriate tooling. In this context, "tools" refers to software systems or utilities used by network operators to deploy, configure, monitor, troubleshoot, and manage networks or network protocols in real-world operational environments. While the introduction of a new specification does not automatically mandate the development of entirely new tools, careful consideration must be given to how existing tools can be leveraged or extended to support the management and operation of these new specifications.

The [NEMOPS] workshop highlighted a consistent theme applicable beyond network management protocols: the "ease of use" and adaptability of existing tools are critical factors for successful adoption. Therefore, a new specification should provide examples using existing, common tooling, or running code that demonstrate how to perform key operational tasks.

Specifically, the following tooling-related aspects should be considered, prioritizing the adaptation of existing tools:

* Leveraging Existing Tooling: Before considering new tools, assess whether existing tooling, such as monitoring systems, logging platforms, configuration management systems, and/or orchestration frameworks, can be adapted to support the new specification. This may involve developing plugins, modules, or drivers that enable these tools to interact with the new specification.

* Extending Existing Tools: Identify areas where existing tools can be extended to provide the necessary visibility and control over the new specification. For example, if a new transport protocol is introduced, consider whether existing network monitoring tools can be extended to track its performance metrics or whether existing security tools can be adapted to analyze its traffic patterns.

* New Tools: Only when existing tools are demonstrably inadequate for managing and operating the elements of the new specification should the development of new tools be considered. In such cases, carefully define the specific requirements for these new tools, focusing on the functionalities that cannot be achieved through adaptation or extension of existing solutions.

* IETF Hackathons for Manageability Testing: IETF Hackathons [IETF-HACKATHONS] provide an opportunity to test the functionality, interoperability, and manageability of New

Protocols or Protocol Extensions. These events can be specifically leveraged to assess the operational (including manageability) implications of a New Protocol or Protocol Extension by focusing tasks on:

- Adapting existing tools to interact with the new specification.

- Developing example management scripts or modules for existing management platforms.

- Testing the specification’s behavior under various operational conditions.

- Identifying potential tooling gaps and areas for improvement.

- Creating example flows and use cases for manageability.

* Open-Source for Tooling: If new tools are deemed necessary, or if significant adaptations to existing tools are required, prioritize open-source development with community involvement. Open-source tools lower the barrier to entry, encourage collaboration, and provide operators with the flexibility to customize and extend the tools to meet their specific needs.

6.1. AI Tooling Considerations

With the increasing adoption of Artificial Intelligence (AI) in network operations, Protocol Designers must consider the implication such functions may have on New Protocols and Protocol Extensions. AI models often require extensive and granular data for training and inference, requiring efficient, scalable, and secure mechanisms for telemetry, logging, and state information collection. Protocol Designers should anticipate that AI-powered management tools may generate frequent and potentially aggressive querying patterns on network devices and controllers. Therefore, protocols should be designed with Data Models and mechanisms intended to prevent overload from automated interactions, while also accounting for AI-specific security considerations such as data integrity and protection against adversarial attacks on management interfaces. These considerations are also relevant to Performance Management (Section 5.7) and Security Management (Section 5.8).

7. IANA Considerations

This document does not have any IANA actions required.

8. Operational Considerations

Although this document focuses on operations and manageability guidance, it does not define a New Protocol, a Protocol Extension, or an architecture. As such, there are no new operations or manageability requirements introduced by this document.

9. Security Considerations

This document 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.

While a standard description of a protocol’s manageable parameters facilitates legitimate operation, it may also inadvertently simplify an attacker’s efforts to understand and manipulate the protocol.

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 5.8).

10. Informative References

[BCP133] Best Current Practice 133, <https://www.rfc-editor.org/info/bcp133>.
At the time of writing, this BCP comprises the following:

Duke, M., Ed. and G. Fairhurst, Ed., "Specifying New Congestion Control Algorithms", BCP 133, RFC 9743, DOI 10.17487/RFC9743, March 2025, <https://www.rfc-editor.org/info/rfc9743>.

[BCP14] Best Current Practice 14, <https://www.rfc-editor.org/info/bcp14>. At the time of writing, this BCP comprises the following:

Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[BCP166] Best Current Practice 166, <https://www.rfc-editor.org/info/bcp166>. At the time of writing, this BCP comprises the following:

Hoffman, P. and J. Klensin, "Terminology Used in Internationalization in the IETF", BCP 166, RFC 6365, DOI 10.17487/RFC6365, September 2011, <https://www.rfc-editor.org/info/rfc6365>.

[BCP72] Best Current Practice 72, <https://www.rfc-editor.org/info/bcp72>. At the time of writing, this BCP comprises the following:

Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, DOI 10.17487/RFC3552, July 2003, <https://www.rfc-editor.org/info/rfc3552>.

Gont, F. and I. Arce, "Security Considerations for Transient Numeric Identifiers Employed in Network Protocols", BCP 72, RFC 9416, DOI 10.17487/RFC9416, July 2023, <https://www.rfc-editor.org/info/rfc9416>.

[CHECKLIST] "Operations and Management Review Checklist", 2025, <https://github.com/IETF-OPS-DIR/Review-Template/tree/main>.

[I-D.ietf-lamps-dilithium-certificates]
Massimo, J., Kampanakis, P., Turner, S., and B. Westerbaan, "Internet X.509 Public Key Infrastructure - Algorithm Identifiers for the Module-Lattice-Based Digital Signature Algorithm (ML-DSA)", Work in Progress, Internet-Draft, draft-ietf-lamps-dilithium-certificates-13, 30 September 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-lamps-dilithium-certificates-13>.

[I-D.ietf-netmod-rfc8407bis]
Bierman, A., Boucadair, M., and Q. Wu, "Guidelines for Authors and Reviewers of Documents Containing YANG Data Models", Work in Progress, Internet-Draft, draft-ietf-netmod-rfc8407bis-28, 5 June 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-netmod-rfc8407bis-28>.

[I-D.ietf-nmop-network-incident-yang]
Hu, T., Contreras, L. M., Wu, Q., Davis, N., and C. Feng, "A YANG Data Model for Network Incident Management", Work in Progress, Internet-Draft, draft-ietf-nmop-network-incident-yang-06, 12 October 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-nmop-network-incident-yang-06>.

[I-D.ietf-nmop-terminology]
Davis, N., Farrel, A., Graf, T., Wu, Q., and C. Yu, "Some Key Terms for Network Fault and Problem Management", Work in Progress, Internet-Draft, draft-ietf-nmop-terminology-23, 18 August 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-nmop-terminology-23>.

[I-D.ietf-opsawg-oam-characterization]
Pignataro, C., Farrel, A., and T. Mizrahi, "Guidelines for Characterizing "OAM"", Work in Progress, Internet-Draft, draft-ietf-opsawg-oam-characterization-13, 20 October 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-opsawg-oam-characterization-13>.

[I-D.ietf-quic-qlog-main-schema]
Marx, R., Niccolini, L., Seemann, M., and L. Pardue, "qlog: Structured Logging for Network Protocols", Work in Progress, Internet-Draft, draft-ietf-quic-qlog-main-schema-13, 20 October 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-quic-qlog-main-schema-13>.

[IESG-STATEMENT]
IESG, "Writable MIB Module IESG Statement", 2 March 2014, <https://datatracker.ietf.org/doc/statement-iesg-writable-mib-module-iesg-statement-20140302/>.

[IETF-HACKATHONS]
IETF, "IETF Hackathons", 1 May 2025, <https://www.ietf.org/meeting/hackathons/>.

[IETF-OPS-Dir]
"Ops Directorate (opsdir)", 2025, <https://datatracker.ietf.org/group/opsdir/about/>.

[NEMOPS] Hardaker, W. and D. Dhody, "Report from the IAB Workshop on the Next Era of Network Management Operations (NEMOPS)", Work in Progress, Internet-Draft, draft-iab-nemops-workshop-report-04, 29 August 2025, <https://datatracker.ietf.org/doc/html/draft-iab-nemops-workshop-report-04>.

[NEMOPS-WORKSHOP]
IAB, "IAB workshop on the Next Era of Network Management Operations", December 2024, <https://datatracker.ietf.org/group/nemopsws/about/>.

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <https://www.rfc-editor.org/rfc/rfc1034>.

[RFC1958] Carpenter, B., Ed., "Architectural Principles of the Internet", RFC 1958, DOI 10.17487/RFC1958, June 1996, <https://www.rfc-editor.org/rfc/rfc1958>.

[RFC2113] Katz, D., "IP Router Alert Option", RFC 2113, DOI 10.17487/RFC2113, February 1997, <https://www.rfc-editor.org/rfc/rfc2113>.

[RFC2205] Braden, R., Ed., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, DOI 10.17487/RFC2205, September 1997, <https://www.rfc-editor.org/rfc/rfc2205>.

[RFC2439] Villamizar, C., Chandra, R., and R. Govindan, "BGP Route Flap Damping", RFC 2439, DOI 10.17487/RFC2439, November 1998, <https://www.rfc-editor.org/rfc/rfc2439>.

[RFC2711] Partridge, C. and A. Jackson, "IPv6 Router Alert Option", RFC 2711, DOI 10.17487/RFC2711, October 1999, <https://www.rfc-editor.org/rfc/rfc2711>.

[RFC2975] Aboba, B., Arkko, J., and D. Harrington, "Introduction to Accounting Management", RFC 2975, DOI 10.17487/RFC2975, October 2000, <https://www.rfc-editor.org/rfc/rfc2975>.

[RFC3139] Sanchez, L., McCloghrie, K., and J. Saperia, "Requirements for Configuration Management of IP-based Networks", RFC 3139, DOI 10.17487/RFC3139, June 2001, <https://www.rfc-editor.org/rfc/rfc3139>.

[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, DOI 10.17487/RFC3198, November 2001, <https://www.rfc-editor.org/rfc/rfc3198>.

[RFC3444] Pras, A. and J. Schoenwaelder, "On the Difference between Information Models and Data Models", RFC 3444, DOI 10.17487/RFC3444, January 2003, <https://www.rfc-editor.org/rfc/rfc3444>.

[RFC3535] Schoenwaelder, J., "Overview of the 2002 IAB Network Management Workshop", RFC 3535, DOI 10.17487/RFC3535, May 2003, <https://www.rfc-editor.org/rfc/rfc3535>.

[RFC4251] Ylonen, T. and C. Lonvick, Ed., "The Secure Shell (SSH) Protocol Architecture", RFC 4251, DOI 10.17487/RFC4251, January 2006, <https://www.rfc-editor.org/rfc/rfc4251>.

[RFC5218] Thaler, D. and B. Aboba, "What Makes for a Successful Protocol?", RFC 5218, DOI 10.17487/RFC5218, July 2008, <https://www.rfc-editor.org/rfc/rfc5218>.

[RFC5321] Klensin, J., "Simple Mail Transfer Protocol", RFC 5321, DOI 10.17487/RFC5321, October 2008, <https://www.rfc-editor.org/rfc/rfc5321>.

[RFC5424] Gerhards, R., "The Syslog Protocol", RFC 5424, DOI 10.17487/RFC5424, March 2009, <https://www.rfc-editor.org/rfc/rfc5424>.

[RFC5706] Harrington, D., "Guidelines for Considering Operations and Management of New Protocols and Protocol Extensions", RFC 5706, DOI 10.17487/RFC5706, November 2009, <https://www.rfc-editor.org/rfc/rfc5706>.

[RFC5884] Aggarwal, R., Kompella, K., Nadeau, T., and G. Swallow, "Bidirectional Forwarding Detection (BFD) for MPLS Label Switched Paths (LSPs)", RFC 5884, DOI 10.17487/RFC5884, June 2010, <https://www.rfc-editor.org/rfc/rfc5884>.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <https://www.rfc-editor.org/rfc/rfc6020>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/rfc/rfc6241>.

[RFC6291] Andersson, L., van Helvoort, H., Bonica, R., Romascanu, D., and S. Mansfield, "Guidelines for the Use of the "OAM" Acronym in the IETF", BCP 161, RFC 6291, DOI 10.17487/RFC6291, June 2011, <https://www.rfc-editor.org/rfc/rfc6291>.

[RFC6298] Paxson, V., Allman, M., Chu, J., and M. Sargent, "Computing TCP’s Retransmission Timer", RFC 6298, DOI 10.17487/RFC6298, June 2011, <https://www.rfc-editor.org/rfc/rfc6298>.

[RFC6390] Clark, A. and B. Claise, "Guidelines for Considering New Performance Metric Development", BCP 170, RFC 6390, DOI 10.17487/RFC6390, October 2011, <https://www.rfc-editor.org/rfc/rfc6390>.

[RFC6632] Ersue, M., Ed. and B. Claise, "An Overview of the IETF Network Management Standards", RFC 6632, DOI 10.17487/RFC6632, June 2012, <https://www.rfc-editor.org/rfc/rfc6632>.

[RFC6709] Carpenter, B., Aboba, B., Ed., and S. Cheshire, "Design Considerations for Protocol Extensions", RFC 6709, DOI 10.17487/RFC6709, September 2012, <https://www.rfc-editor.org/rfc/rfc6709>.

[RFC7011] Claise, B., Ed., Trammell, B., Ed., and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information", STD 77, RFC 7011, DOI 10.17487/RFC7011, September 2013, <https://www.rfc-editor.org/rfc/rfc7011>.

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <https://www.rfc-editor.org/rfc/rfc7950>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/rfc/rfc8040>.

[RFC8170] Thaler, D., Ed., "Planning for Protocol Adoption and Subsequent Transitions", RFC 8170, DOI 10.17487/RFC8170, May 2017, <https://www.rfc-editor.org/rfc/rfc8170>.

[RFC8199] Bogdanovic, D., Claise, B., and C. Moberg, "YANG Module Classification", RFC 8199, DOI 10.17487/RFC8199, July 2017, <https://www.rfc-editor.org/rfc/rfc8199>.

[RFC8299] Wu, Q., Ed., Litkowski, S., Tomotaki, L., and K. Ogaki, "YANG Data Model for L3VPN Service Delivery", RFC 8299, DOI 10.17487/RFC8299, January 2018, <https://www.rfc-editor.org/rfc/rfc8299>.

[RFC8309] Wu, Q., Liu, W., and A. Farrel, "Service Models Explained", RFC 8309, DOI 10.17487/RFC8309, January 2018, <https://www.rfc-editor.org/rfc/rfc8309>.

[RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <https://www.rfc-editor.org/rfc/rfc8340>.

[RFC8466] Wen, B., Fioccola, G., Ed., Xie, C., and L. Jalil, "A YANG Data Model for Layer 2 Virtual Private Network (L2VPN) Service Delivery", RFC 8466, DOI 10.17487/RFC8466, October 2018, <https://www.rfc-editor.org/rfc/rfc8466>.

[RFC8791] Bierman, A., Bj旦rklund, M., and K. Watsen, "YANG Data Structure Extensions", RFC 8791, DOI 10.17487/RFC8791, June 2020, <https://www.rfc-editor.org/rfc/rfc8791>.

[RFC8799] Carpenter, B. and B. Liu, "Limited Domains and Internet Protocols", RFC 8799, DOI 10.17487/RFC8799, July 2020, <https://www.rfc-editor.org/rfc/rfc8799>.

[RFC9182] Barguil, S., Gonzalez de Dios, O., Ed., Boucadair, M., Ed., Munoz, L., and A. Aguado, "A YANG Network Data Model for Layer 3 VPNs", RFC 9182, DOI 10.17487/RFC9182, February 2022, <https://www.rfc-editor.org/rfc/rfc9182>.

[RFC9291] Boucadair, M., Ed., Gonzalez de Dios, O., Ed., Barguil, S., and L. Munoz, "A YANG Network Data Model for Layer 2 VPNs", RFC 9291, DOI 10.17487/RFC9291, September 2022, <https://www.rfc-editor.org/rfc/rfc9291>.

[RFC9424] Paine, K., Whitehouse, O., Sellwood, J., and A. Shaw, "Indicators of Compromise (IoCs) and Their Role in Attack Defence", RFC 9424, DOI 10.17487/RFC9424, August 2023, <https://www.rfc-editor.org/rfc/rfc9424>.

[SECOPS] "NICCS Glossary", August 2025, <https://niccs.cisa.gov/resources/glossary>.

Appendix A. Changes Since RFC 5706

The following changes have been made to the guidelines published in [RFC5706]:

* Change intended status from Informational to Best Current Practice

* Move the "Operational Considerations" Appendix A to a Checklist [CHECKLIST] maintained in GitHub

* Add a requirement for an "Operational Considerations" section in all new Standard Track RFCs, along with specific guidance on its content.

* Update the operational and manageability-related technologies to reflect over 15 years of advancements

  - Provide focus and details on YANG-based standards, deprioritizing MIB Modules.

  - Add a "YANG Data Model Considerations" section

  - Update the "Available Management Technologies" landscape

* Add an "Operational and Management Tooling Considerations" section

A.1. TO DO LIST

See the list of open issues at https://github.com/IETF-OPSAWG-WG/draft-opsarea-rfc5706bis/issues

Acknowledgements

The authors thank the following individuals and groups, whose efforts have helped to improve this document:

The IETF Ops Directorate (OpsDir): The IETF OpsDir [IETF-OPS-Dir] reviewer team, which has been providing document reviews for more than a decade, and its Chairs past and present: Gunter Van de Velde, Carlos Pignataro, Bo Wu, and Daniele Ceccarelli.

The AD championing the update: Med Boucadair, who initiated and championed the effort to refresh RFC 5706, 15 years after its publication, building on an idea originally suggested by Carlos Pignataro.

Reviewers of this document, in chronological order: Mahesh Jethanandani, Chongfeng Xie, Alvaro Retana, Michael P., Scott Hollenbeck, Ron Bonica, Italo Busi, Brian Trammel, Aijun Wang, and Richard Shockey for their review, comments, and contributions.

The author of RFC 5706: David Harrington

Acknowledgments from RFC 5706: 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, J端rgen Sch旦nw辰lder, Andy Bierman, and David Harrington.

Thanks to reviewers who helped fashion this document, including Harald Alvestrand, Ron Bonica, Brian Carpenter, Beno樽t Claise, Adrian Farrel, David Kessens, Dan Romascanu, Pekka Savola, J端rgen Sch旦nw辰lder, Bert Wijnen, Ralf Wolter, and Lixia Zhang.

Contributors

Thomas Graf
Swisscom
Email: thomas.graf@swisscom.com

Authors’ Addresses

Benoit Claise
Everything OPS
Email: benoit@everything-ops.net

Joe Clarke
Cisco
Email: jclarke@cisco.com

Adrian Farrel
Old Dog Consulting
Email: adrian@olddog.co.uk

Samier Barguil
Nokia
Email: samier.barguil_giraldo@nokia.com

Carlos Pignataro
Blue Fern Consulting
Email: carlos@bluefern.consulting, cpignata@gmail.com
URI: https://bluefern.consulting

Ran Chen
ZTE
Email: chen.ran@zte.com.cn