Network Working Group
Internet-Draft
Obsoletes: 5706 (if approved)
Intended status: Best Current Practice
Expires: 8 February 2026

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

7 August 2025

Guidelines for Considering Operations and Management in IETF Specifications
draft-opsarea-rfc5706bis-04

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 they should consider and include in their documents.

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 is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 8 February 2026.

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 difficult. In order to make sure that protocols can be deployed and used, the operational environment and manageability of a protocol should be considered when New Protocols or Protocol Extensions are designed.

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 and protocols 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 the protocol within a network, even if there were 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 specification, providing guidance about what to consider when thinking about the management and deployment of a new protocol, to provide guidance about documenting those considerations, and helping them provide a reasonably consistent format for such documentation.

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.

This document 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 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].

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., 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.

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.

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, "Operations 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. <-- Insert a brief explanation 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 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.

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 -- such as whether write operations will be allowed 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 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.

Default values should generally favor the conservative side over the "optimizing performance" side (e.g., the initial RTT and 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 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?

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 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.

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. Protocol Designers should outline such 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 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.

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.

4.6. 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 (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. Which metrics are of interest?

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

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 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 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]. Broadly, these include core network management protocols, purpose-specific management protocols, and network management Data Models. A non-exhaustive list of such protocols is provided below:

* Remote Authentication Dial In User Service (RADIUS) [RFC2865]
* The Syslog Protocol [RFC5424]
* Packet Sampling (PSAMP) Protocol Specifications [RFC5476]
* Network Configuration Protocol (NETCONF) [RFC6241]
* Diameter Base Protocol [RFC6733]
* Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information [RFC7011]

* BGP Monitoring Protocol (BMP) [RFC7854]

* RESTCONF Protocol [RFC8040]

* Network Telemetry Framework [RFC9232]

The IETF previously also worked on the Simple Network Management Protocol (SNMP) [RFC3410] and the Structure of Management Information (SMI) [STD58], but further use of this management protocol in new IETF documents has been constrained to maintenance of existing MIB modules and development of MIB modules for legacy devices that do not support more resent management protocols [IESG-STATEMENT].

This section is not intended to offer in-depth definitions or explanations; readers seeking more detail should consult the referenced materials.

5.2. 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 protocol support 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 the IETF likes this. The IETF is moving from support for one schema language for modeling the structure of management information (SMIv2) and one simple network management protocol (SNMP) 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-0-20041028]

and

* NETCONF Configuration Protocol [RFC6241]

* the IP Flow Information Export (IPFIX) Protocol [RFC7011] 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 help focus interoperability on the semantic level by defining what information should be gathered and how it might be used, regardless of the underlying management protocol or vendor implementation. 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 YANG Data Model and IPFIX Information Elements about the same event. An Information Model might identify which error conditions should be counted separately, and which error conditions can be recorded together in a single counter. Then, whether the counter is gathered via, e.g., NETCONF or exported via IPFIX, the counter will have the same meaning.

Protocol Designers must consider what operational, configuration, state, or statistical information will be relevant for effectively monitoring, controlling, or troubleshooting a New Protocol and its Protocol Extensions. This includes identifying key parameters that reflect the protocol's behavior, performance metrics, error indicators, and any contextual data that would aid in diagnostic, troubleshooting, or lifecycle management.

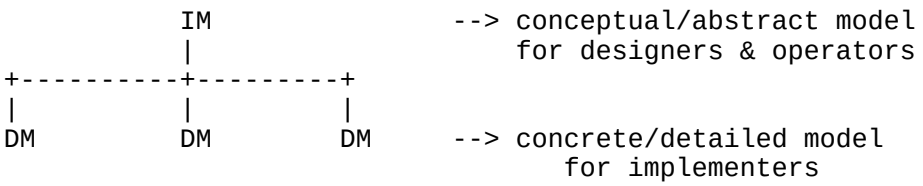

Figure 1: Information Models (IMs) and Data Models (DMs)

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

Protocol Designers may directly develop Data Models without first producing an Information Model. For example, such a decision can be taken when it is given that the data component is not used by distinct protocols (e.g., IPFIX-only).

Alternatively, Protocol Designers may decide to use an Information Model to describe the managed elements in a protocol or Protocol Extension. The protocol Designers then use the Information Model to develop Data Models that will be used for managing the protocol.

Specifically, Protocol Designers should develop an Information Model if multiple Data Model representations (e.g., YANG [RFC6020][RFC7950] and/or IPFIX [RFC7011]) are to be produced, to ensure lossless semantic mapping. Protocol Designers may create an Information Model if the resulting Data Models are complex or numerous.

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

* [RFC3060] - Policy Core Information Model -- Version 1 Specification

* [RFC3290] - An Informal Management Model for Diffserv Routers

* [RFC3460] - Policy Core Information Model (PCIM) Extensions

* [RFC3585] - IPsec Configuration Policy Information Model

* [RFC3644] - Policy Quality of Service (QoS) Information Model

* [RFC3670] - Information Model for Describing Network Device QoS Datapath Mechanisms

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.

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

The Protocol Designer should document the basic Faults and health indicators that need to be instrumented for the New Protocol or Protocol Extension, 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?

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 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 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. 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 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 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.

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).

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 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 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 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.

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. 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 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. These events can be specifically leveraged to assess the operational (including manageability) implications of a New Protocol 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.

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>.

[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-12, 26 June 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-lamps-dilithium-certificates-12>.

[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-05, 6 July 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-nmop-network-incident-yang-05>.

[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-21, 6 August 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-nmop-terminology-21>.

[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-09, 2 July 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-opsawg-oam-characterization-09>.

[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-03, 21 July 2025, <https://datatracker.ietf.org/doc/html/draft-iab-nemops-workshop-report-03>.

[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>.

[RFC2865] Rigney, C., Willens, S., Rubens, A., and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", RFC 2865, DOI 10.17487/RFC2865, June 2000, <https://www.rfc-editor.org/rfc/rfc2865>.

[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>.

[RFC3060] Moore, B., Ellesson, E., Strassner, J., and A. Westerinen, "Policy Core Information Model -- Version 1 Specification", RFC 3060, DOI 10.17487/RFC3060, February 2001, <https://www.rfc-editor.org/rfc/rfc3060>.

[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>.

[RFC3290] Bernet, Y., Blake, S., Grossman, D., and A. Smith, "An Informal Management Model for Diffserv Routers", RFC 3290, DOI 10.17487/RFC3290, May 2002, <https://www.rfc-editor.org/rfc/rfc3290>.

[RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, DOI 10.17487/RFC3410, December 2002, <https://www.rfc-editor.org/rfc/rfc3410>.

[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>.

[RFC3460] Moore, B., Ed., "Policy Core Information Model (PCIM) Extensions", RFC 3460, DOI 10.17487/RFC3460, January 2003, <https://www.rfc-editor.org/rfc/rfc3460>.

[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>.

[RFC3585] Jason, J., Rafalow, L., and E. Vyncke, "IPsec Configuration Policy Information Model", RFC 3585, DOI 10.17487/RFC3585, August 2003, <https://www.rfc-editor.org/rfc/rfc3585>.

[RFC3644] Snir, Y., Ramberg, Y., Strassner, J., Cohen, R., and B. Moore, "Policy Quality of Service (QoS) Information Model", RFC 3644, DOI 10.17487/RFC3644, November 2003, <https://www.rfc-editor.org/rfc/rfc3644>.

[RFC3670] Moore, B., Durham, D., Strassner, J., Westerinen, A., and W. Weiss, "Information Model for Describing Network Device QoS Datapath Mechanisms", RFC 3670, DOI 10.17487/RFC3670, January 2004, <https://www.rfc-editor.org/rfc/rfc3670>.

[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>.

[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>.

[RFC5476] Claise, B., Ed., Johnson, A., and J. Quittek, "Packet Sampling (PSAMP) Protocol Specifications", RFC 5476, DOI 10.17487/RFC5476, March 2009, <https://www.rfc-editor.org/rfc/rfc5476>.

[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>.

[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>.

[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>.

[RFC6733] Fajardo, V., Ed., Arkko, J., Loughney, J., and G. Zorn, Ed., "Diameter Base Protocol", RFC 6733, DOI 10.17487/RFC6733, October 2012, <https://www.rfc-editor.org/rfc/rfc6733>.

[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>.

[RFC7854] Scudder, J., Ed., Fernando, R., and S. Stuart, "BGP Monitoring Protocol (BMP)", RFC 7854, DOI 10.17487/RFC7854, June 2016, <https://www.rfc-editor.org/rfc/rfc7854>.

[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>.

[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., Bjrklund, 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>.

[RFC9232] Song, H., Qin, F., Martinez-Julia, P., Ciavaglia, L., and A. Wang, "Network Telemetry Framework", RFC 9232, DOI 10.17487/RFC9232, May 2022, <https://www.rfc-editor.org/rfc/rfc9232>.

[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>.

[STD58] Internet Standard 58, <https://www.rfc-editor.org/info/std58>.
At the time of writing, this STD comprises the following:

McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, DOI 10.17487/RFC2578, April 1999, <https://www.rfc-editor.org/info/rfc2578>.

McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Textual Conventions for SMIv2", STD 58, RFC 2579, DOI 10.17487/RFC2579, April 1999, <https://www.rfc-editor.org/info/rfc2579>.

McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Conformance Statements for SMIv2", STD 58, RFC 2580, DOI 10.17487/RFC2580, April 1999, <https://www.rfc-editor.org/info/rfc2580>.

[W3C.REC-xmlschema-0-20041028]
Fallside, D., Ed. and P. Walmsley, Ed., "XML Schema Part 0: Primer Second Edition", W3C REC REC-xmlschema-0-20041028, W3C REC-xmlschema-0-20041028, 28 October 2004, <https://www.w3.org/TR/2004/REC-xmlschema-0-20041028/>.

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 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 wish to thank the following individuals and groups.

The IETF Ops Directorate: The IETF Ops Directorate [IETF-OPS-Dir]

reviewer team, who has been providing document reviews for over a decade, and its Chairs, Gunter Van de Velde, Carlos Pignataro, and Bo Wu.

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

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, Jrgen Schnwlder, Andy Bierman, and David Harrington.

Thanks to reviewers who helped fashion this document, including Harald Alvestrand, Ron Bonica, Brian Carpenter, Benot Claise, Adrian Farrel, David Kessens, Dan Romascanu, Pekka Savola, Jrgen Schnwlder, 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