Network Working Group G. Scott
INTERNET DRAFT Defense Information Systems Agency
November 1996

Guide for Internet Standards Writers

<draft-ietf-stdguide-ops-01.txt>

Status of this Document

Abstract

This document is a guide for Internet standard writers. It defines those characteristics that make standards coherent, unambiguous, and easy to interpret. Also, it singles out usage believed to have led to unclear specifications, resulting in non-interoperable interpretations in the past. These guidelines are to be used with |
RFC 1543, "Instructions to RFC Authors." |

This version of the document is a draft. It is intended to generate further discussion and addition by the STDGUIDE working group. Please send comments to stdguide@midnight.com.

Table of Contents

1 Introduction

This document is a guide for Internet standard writers. It offers guidelines on how to write a protocol specification with clarity, precision, and completeness. These guidelines are based on both prior successful and unsuccessful IETF specification experiences. These guidelines are to be used with RFC 1543, "Instructions to RFC Authors," or its update. Note that some guidelines may not apply in certain situations. |

The goal is to increase the possibility that multiple implementations of a protocol will interoperate. Writing specifications to these guidelines will not guarantee interoperability. However, a recognized barrier to the creation of interoperable protocol implementations is unclear specifications.

Many will benefit from having well-written protocol specifications. Implementors will have a better chance to conform to the protocol specification. Protocol testers can use the specification to derive unambiguous testable statements. Purchasers and users of the protocol will have a better understanding of its capabilities.

2 General Guidelines

It is important that multiple readers and implementors of a standard have the same understanding of a document. To this end, information should be orderly and detailed. The following are general guidelines intended to help in the production of such a document.

2.1 Protocol Description

Standards track documents must include a description of the purpose | or context of the protocol's use. The author of a protocol specification will have a great deal of knowledge as to the purpose of the protocol. However, the reader is more likely to have general networking knowledge and experience, rather than expertise in a particular protocol. An explanation of the purpose will | give the reader a reference point for understanding the protocol | and where it fits in the Internet. The Draft Standard RFC 1583 was | recommended to the STDGUIDE working guide as providing a good | example of this in it "Protocol Overview" section. |

The protocol's general description should also provide information | on the relationship between the different parties to the protocol. | This can be done by showing typical packet sequences. |

This also applies to the algorithms used by a protocol. A detailed description of the algorithms or citation of readily available references that give such a description is necessary.

2.2 Discussion of Security

If the Internet is to achieve its full potential in commercial, governmental, and personal affairs, it must assure users that deliveries of their information transfers are free from tampering or compromise. Well-written security sections in standard protocol documents can do much to achieve that condition. Implementors will find it easier to comply and do security. Users can understand the security measures in place, and so have faith in the Internet.

The security section should address several topics. Every | standards track document must discuss the security risks inherent | in the protocol being specified. After the document's author has | set out the security risks the protocol is open to, he then must | discuss the remedies offered. Additionally, the effects the | security measures have on the protocol's use and performance. If possible, the discussion should address how much insurance the implementation of the security measures achieves.

When no security measures are offered, the author must provide a | detailed explanation why. This discussion could present the | reasons why the security issues are unresolvable at this time. Alternatively, the author could present a case why security is unneeded when using the protocol.

These security sections should be complete and separate. If | security measures are part of the general protocol text, they will be difficult to find. If the security measures are not clear they may not be implemented, nor will a user be assured that they exist.

Finally, it is no longer acceptable that security sections consist solely of statements similar to: "Security issues are not discussed in this RFC."

2.3 Level of Detail

The author should consider what level of descriptive detail best | conveys the protocol's intent. Concise text has several advantages. It makes the document easier to read. Such text reduces the chance for conflict between different portions of the specification. The reader can readily identify the required protocol mechanisms in the standard. Also, it makes it easier to identify the requirements for protocol implementation. A disadvantage of concise descriptions is that a reader may not fully comprehend the reasoning behind the protocol, and thus make assumptions that will lead to implementation errors.

Longer descriptions may be necessary to explain purpose, | background, rationale, implementation experience, or to provide | tutorial information. This helps the reader understand the | protocol. Yet several dangers exist with lengthy text. Finding the protocol requirements in the text is difficult or confusing. The same mechanism may have multiple descriptions, which leads to | misinterpretations or conflict. Lengthy text is a challenge to the attention span of some readers. Finally, it is more difficult to comprehend, a consideration as English is not the native language of the many worldwide readers of IETF standards.

One approach is to divide the standard into sections: one describing the protocol concisely, while another section consists of explanatory text. The STD 3/RFC 1122/RFC 1123 and Draft | Standard RFC 1583 provides examples of this method. |

2.4 Protocol Versions

Often the standard is specifying a new version of an existing protocol. In such a case, the authors should detail the differences between the previous version and the new version. This should include the rationale for the changes, for example, implementation experience, changes in technology, responding to user demand, etc.

2.5 Decision History

In standards development, reaching consensus requires making difficult choices. By including a discussion history and | rationales for a decision, the author can prevent future revisiting | of these disagreements later, when the original parties have moved on. Also, the knowledge of the "why" is as useful to an | implementor as the description of "how." For example, the | alternative not taken may have been simpler to implement, so including the logic behind the choice may prevent future implementors from taking nonstandard shortcuts.

2.6 Response to Out of Specification Behavior

The STDGUIDE working group recommends that detail description of |
the actions taken in case of behavior that is deviant from or
exceeds the specification be included. This is an area where

implementors often differ in opinion as to the appropriate response. By specifying a common response, the standard author can reduce the risk that different inplementations will come in to conflict. |

The standard should describe responses to behavior explicitly forbidden or out of the boundaries defined by the specification. Two possible approaches to such cases are discarding, or invoking error-handling mechanisms. If discarding is chosen, detailing the disposition may be necessary. For instance, treat dropped frames as if they were never received, or reset an existing connection or adjacency state.

The specification should describe actions taken when critical resource or performance scaling limits are exceeded. This is not necessary for every case. It is necessary for cases where a risk of network degradation or operational failure exists. In such cases, a consistent behavior between implementations is necessary.

2.7 The Liberal/Conservative Rule

A rule, first stated in RFC 791, recognized as having benefits in implementation robustness and interoperability is:

"Be liberal in what you accept, and
conservative in what you send."

Or establish restrictions on what a protocol transmits, but be able to deal with every conceivable error received. Caution is urged in applying this approach in standards track protocols. It has in the past lead to conflicts between vendors when interoperability fails. The sender accuses the receiver of failing to be liberal enough, and the receiver accuses the sender of not being conservative enough. Therefore, the author is obligated to provide extensive detail on send and receive behavior. |

To avoid any confusion between the two, recommend that standard authors specify send and receive behavior separately. The description of reception will require the most detailing. For | implementations will be expected to accept any packet from the network without failure or malfunction. Therefore, the actions taken to achieve that result, need to be laid out in the protocol specification. Standard authors should consider not just how to survive on the network, but achieve the highest level of cooperation possible to limit the amount of network disruption. The appearance of undefined information or conditions must not cause a network or host failure. This requires specification on how to attempt acceptance of most of the packets. Two approaches are available, either using as much of the packet's content as

possible, or invoking error procedures. The author should specify |
a dividing line on when to take which approach.

A case for consideration is that of a routing protocol, where acceptance of flawed information can cause network failure. For protocols such as this, the specification should identify packets

that could have differing interpretations and mandate that they be |
either rejected completely or the nature of the attempt to recover |
some information from them. For example, routing updates that |
contain more data than the tuple count shows. The protocol authors |
should consider whether some trailing data can be accepted as |
additional routes, or to reject the entire packet as suspect |
because it is non-conformant. |

2.8 Handling of Protocol Options

Standards with many optional features increase the chance of non-interoperable implementations. The danger is that different protocol implementations may specify some optional combinations that are unable to interoperate with each other. Ideally, implementation experience purges options from the protocol while the document moves along the standard track.

Therefore, options should only be present in a protocol to |
support a particular market, e.g., the financial industry, or |
network environment, e.g., a network constrained by limited |
bandwidth. The protocol specification must explain the full |
implications of either using the option or not, and the case for
choosing either course. As part of this, the author needs to |
consider and describe how the options are intended to be used |
alongside other protocols. However, omission of the optional item |
should have no interoperability consequences for the implementation
that does so.

Certain cases will require the specifying of mutually exclusive options within a protocol. That is, the implementation of an optional feature precludes the implementation of the other optional
feature. For clarity, the author needs to state when to implement |
one or the other, what the effect of choosing one over the other is, and what problems the implementor or user may face. The choice of one or the other options should have no interoperability consequences between multiple implementations.

2.9 Indicating Requirement Levels |

The Internet-Draft draft-bradner-key-words-03.txt, "Key words for |
use in RFCs to Indicate Requirement Levels," defines several words |
that can be used in many standards track documents to signify the |
mandatory protocol features from the optional features of the |
specification. The definitions provided are as they should be |
interpreted in implementing IETF standards. Note that the force of |
these words is modified by the requirement level of the document in |
which they are used. |

Some authors of existing IETF standards have chosen to capitalize |

these words to clarify or stress their intent, but this is not |
required. What is necessary, is that these words are used |
consistently throughout the document. That is, every mandatory or |
optional protocol requirement shall be identified by the authors |
and documented by these words. If a requirement is not identified |

in this manner, it will not be considered an equal part of the protocol and be likely passed over by the implementor.

2.10 Notational Conventions

Formal syntax notations can be used to define complicated protocol concepts or data types, and to specify values of these data types. This permits the protocol to be written without concern on how the implementation is constructed, or how the data type is represented during transfer. The specification is simplified because it can be presented as "axioms" that will be proven by implementation.

The formal specification of the syntax used should be referenced in the text of the standard. Any extensions, subsets, alterations, or exceptions to the formal syntax should be defined.

The STD 11/RFC 822 provides an example of this. In RFC 822 (Section 2 and Appendix D) the Backus-Naur Form (BNF) meta-language was extended to make its representation smaller and easier to understand. Note, that the Internet-Draft draft-ietf-drums-abnf-01.txt, "Augmented BNF for Syntax Specifications: ABNF," captures RFC 822's definition so that it can be used as a reference. Another example is STD 16/RFC 1155 (Section 3.2) where a subset of the Abstract Syntax Notation One (ASN.1) is defined.

The author of a standards track protocol needs to consider several things before they use a formal syntax notation. Is the formal specification language being used parseable by an existing machine? If no parser exists, is there enough information provided in the specification to permit the building of a parser? If not, it is likely the reader will not have enough information to decide what the notation means. Also, the author should remember machine parseable syntax is often unreadable by humans, and can make the specification excessive in length. Therefore, syntax notations cannot in place of a clearly written protocol description.

2.11 Implementation Experience

For a protocol to be designated a standard, it must go through the rigors of actual implementation. This implementation experience should be captured in the final document. For example, lessons learned from bake-offs between multiple vendors.

2.12 Glossary

Every standards track RFC should have a glossary, as words can have many meanings. By defining any new words introduced, the author can avoid confusing or misleading the implementer. The definition should appear on the word's first appearance within the text of the protocol specification, and in a separate glossary section. |

It is likely that definition of the protocol will rely on many |
words frequently used in IETF documents. All authors must be |
knowledgeable of the common accepted definitions of these |

frequently used words. FYI 18/RFC 1983, "Internet Users' Glossary," provides definitions that are specific to the Internet. Any deviation from these definitions by authors is strongly discouraged. If circumstances require deviation, an author should state that he is altering the commonly accepted definition, and provide rationale as to the necessity of doing so. The altered definition must be included in the Glossary section. |

If the author uses the word as commonly defined, she does not have to include the definition in the glossary. As a minimum, FYI 18/RFC 1983 should be referenced as a source. |

2.13 Protocol Parameter Assignment |

The common use of the Internet standard track protocols by the Internet community requires that the unique values be assigned to the parameter fields. The Internet Assigned Numbers Authority (IANA) is the central coordinator for the assignment of unique parameter values for Internet protocols. The authors of a developing protocol that use a link, socket, port, protocol, etc., need to contact the IANA to receive a number assignment. For further information on parameter assignment and current assignments, authors should reference STD 2/RFC 1700, "Assigned Numbers." |

3 Specific Guidelines

The following are guidelines on how to present specific technical information in standards.

3.1 Packet Diagrams

Most link, network, and transport layer protocols have packet descriptions.   The STDGUIDE working group recommends that packet diagrams be included in the standard, as they are very helpful to the reader. The preferred form for packet diagrams is a sequence of long words in network byte order, with each word horizontal on the page and bit numbering at the top: |

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version| Prio. |                   Flow Label                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

In cases where a packet is strongly byte-aligned rather than word-aligned (e.g., when byte-boundary variable-length fields are used), display packet diagrams in a byte-wide format. The author |

can use different height boxes for short and long words, and broken |
boxes for variable-length fields:

```
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|    Length N   |
+-+-+-+-+-+-+-+-+
|               |
+    Address    +
      ...
+   (N bytes)   +
|               |
+-+-+-+-+-+-+-+-+
|               |
+  2-byte field +
|               |
+-+-+-+-+-+-+-+-+
```

3.2 Summary Tables

The specifications of some protocols are particularly lengthy, sometimes covering a hundred pages or more. In such cases the inclusion of a summary table can reduce the risk of conformance failure by an implementation through oversight. A summary table itemizes what in a protocol is mandatory, optional, or prohibited. Summary tables do not guarantee conformance, but serve to assist an implementor in checking that they have addressed all protocol features.

The summary table will consist of, as a minimum, four (4) columns: Protocol Feature, Section Reference, Status, and References/Footnotes. The author may add columns if they further |
explain or clarify the protocol. |

In the Protocol Feature column describe the feature, for example, a command word. We recommend grouping series of related |
transactions under descriptive headers, for example, RECEPTION.

Section reference directs the implementor to the section, paragraph, or page that describes the protocol feature in detail.

Status indicates whether the feature is mandatory, optional, or prohibited. The author can either use a separate column for each |
possibility, or a single column with appropriate codes. These codes need to be defined at the start of the summary table to avoid confusion. Possible status codes:

M - must
M - mandatory
MN - must not
O - optional

S - should
X - prohibited
SN - should not

In the References/Footnotes column authors can point to other |
RFCs that are necessary to consider in implementing this protocol

feature, or any footnotes necessary to explain the implementation further.

The STD 3/RFC 1122/RFC 1123 provides examples of summary tables.

3.3 State Machine Descriptions

A convenient method of presenting a protocol's behavior is as a state-machine model.   That is, a protocol can be described by a series of states resulting from a command, operation, or transaction. State-machine models define the variables and constants that establish a state, the events that cause state transitions, and the actions that result from those transitions. Through these models, an understanding of the protocol's dynamic operation as sequence of state transitions that occur for any given event is possible.   State transitions can be detailed by diagrams, tables, or time lines.

Note that state-machine models are never to take the place of detailed text description of the specification. They are adjuncts to the text. The protocol specification shall always take precedence in the case of a conflict.

When using a state transition diagram, show each possible protocol state as a box connected by state transition arcs. The author should label each arc with the event that causes the transition, and, in parentheses, any actions taken during the transition. The STD 5/RFC 1112 provides an example of such a diagram. As ASCII text is the preferred storage format for RFCs, only simple diagrams are possible. Tables can summarize more complex or extensive state transitions.

In a state transition table, read events vertically and states horizontally.   The form, action/new state, represents state transitions and actions. Commas separate multiple actions, and succeeding lines are used as required. The authors should present multiple actions in the order they must be executed, if relevant. Letters that follow the state indicate an explanatory footnote. The dash ('-') indicates an illegal transition. The STD 51/RFC 1661 provides an example of such a state transition table. The initial columns and rows of that table are below as an example:

	State					
	0	1	2	3	4	5
Events	Initial	Starting	Closed	Stopped	Closing	Stopping
Up	2	irc,scr/6	-	-	-	-
Down	-	-	0	tls/1	0	1
Open	tls/1	1	irc,scr/6	3r	5r	5r

```
    Close|     0        tlf/0       2         2         4         4
         |
     TO+ |     -          -         -         -       str/4     str/5
     TO- |     -          -         -         -       tlf/2     tlf/3
```

The STD 18/RFC 904 also presents state transitions in table format. However, it lists transitions in the form n/a, where n is the next state and a represents the action. The method in RFC 1661 is preferred as new-state logically follows action. Also, this RFC's Appendix C models transitions as the Cartesian product of two state machines. This is a more complex representation that may be difficult to comprehend for those readers that are unfamiliar with the format. The working group recommends that authors present tables as defined in the previous paragraph.

A final method of representing state changes is by a timeline. The two sides of the timeline represent the machines involved in the exchange. The author lists the states the machines enter as time progresses (downward) along the outside of timeline. Within the timeline, show the actions that cause the state transitions. An example:

```
         client                                    server

            |                                         |
            |                                         |   LISTEN
SYN_SENT    |-----------------------                  |
            |                       \ syn j           |
            |                        ---------------->|   SYN_RCVD
            |                                         |
            |                        -----------------|
            |        syn k, ack j+1 /                 |
ESTABLISHED |<----------------------                  |
            |                                         |
```

4 Document Checklist

The following is a checklist based on these guidelines that can be applied to a document:

- o Does it explain the purpose of the protocol?
- o Does it reference or explain the algorithms used in the protocol?
- o Does it give packet diagrams in recommended form, if applicable?
- o Does it use the recommended Internet meanings for any terms use to specify the protocol?
- o Are new or altered definitions for terms given in a glossary?
- o Does it separate explanatory portions of the document from requirements?
- o Does it describe differences from previous versions, if applicable?
- o Does it give examples of protocol operation?
- o Does it specify behavior in the face of incorrect operation by

other implementations?

- o Does it delineate which packets should be accepted for processing and which should be ignored?
- o Does it consider performance and scaling issues?
- o How many optional features does it specify? If more than [X], does it separate them into option classes?

- o Have all combinations of options or option classes been examined for incompatibility?
- o Does it explain the rational and use of options?
- o If multiple descriptions of a requirement are given, does it identify one as binding?
- o Have all mandatory and optional requirements be identified and documented by the accepted key words that define Internet requirement levels? |

5. Security Considerations |

This document does not define any security service or mechanism. It does call on IETF standards authors to define clearly the way the protocol they are specifying does or does not provide security assurances to the user. |

6 Working Group Chair's Address

Gregor D. Scott
Director, Defense Information Systems Agency
ATTN: JIEO-JEBBD
Ft. Monmouth, NJ 07703-5613
USA

Phone: (908) 427-6856 |
Fax: (908) 532-0853 |
EMail: scottg@ftm.disa.mil

7 References

RFC 791   "Internet Protocol (IP)," J. Postel, September 1981.

RFC 904   "Exterior Gateway Protocol formal specification," D. Mills, April 1984

RFC 1112 "Host extensions for IP multicasting," S. Deering, August 1989

RFC 1122 "Requirements for Internet Hosts -- Communication Layers," October 1989

RFC 1123 "Requirements for Internet hosts -- Application and Support," October 1989

RFC 1311 "Introduction to the STD Notes"

RFC 1583 "OSPF Version 2" |

RFC 1602 "The Internet Standards Process - Revision 2"

RFC 1661 "The Point-to-Point Protocol (PPP)," W. Simpson, July 1994

RFC 1700 "Assigned Numbers," J. Reynolds, J. Postel, October 1994 |

RFC 1983 "Internet Users' Glossary" |

draft-ietf-drums-abnf-01.txt, "Augmented BNF for Syntax Specifications: ABNF," D. Crocker |

draft-bradner-key-words-03.txt, "Key words for use in RFCs to Indicate Requirement Levels," S. Bradner |

CHANGES FROM PREVIOUS DRAFT

Changes are marked by "|" along the right margin. Many of the changes are editorial in nature. Some were rewriting sentences for clarity. Others are noted as follows:

1. A reference to RFC 1543 was added to the Abstract and Introduction so that authors would know that this was not a stand alone document. That they had to comply to RFC 1543 as well.

2. In section 2.1, text recommending a "Protocol Overview" and a description of how the parties to the protocol relate was added. Reference to Draft Standard RFC 1583 was added.

3. In section 2.2, text was added calling for discussion of all the security risks a protocol faces, rather than just the security problems the protocol solves.

4. In section 2.7, cautionary text regarding the use of the liberal/conservative rule was added.

5. In section 2.8, text calling on authors to consider how protocol options are used with other protocols was added.

6. A new section, "2.9 Indicating Requirement Levels," was added to discuss the use of key words to identify protocol mandatory and option features.

7. In section 2.10, a reference to DRUMS work in defining ABNF, and cautionary text on using formal syntax notation was added.

8. A new section, "2.12 Glossary," was added calling on standards track protocol authors to include a glossary of new or revised terms.

9. A new section, "2.13 Protocol Parameter Assignment," calls on authors to get such assignments from IANA.

10. In section 3.3, a statement that text takes precedence over state machine models was added.

11. The previous draft's section 4, "Glossary," was deleted. In its place, a reference to draft-bradner-key-words-03.txt is made in the new section 2.9.

12. New items were added to section 4, "Document Checklist," to reflect changes above.

13. A new section 5, "Security Considerations," was added.