Network Working Group
Request for Comments: 3269
Category: Informational

R. Kermode
Motorola
L. Vicisano
Cisco

April 2002

# Author Guidelines for Reliable Multicast Transport (RMT) Building Blocks and Protocol Instantiation documents

## Status of this Memo



## Copyright Notice



## Abstract

This document provides general guidelines to assist the authors of Reliable Multicast Transport (RMT) building block and protocol instantiation definitions. The purpose of these guidelines is to ensure that any building block and protocol instantiation definitions produced contain sufficient information to fully explain their operation and use. In addition these guidelines provide directions to specify modular and clearly defined RMT building blocks and protocol instantiations that can be refined and augmented to safely create new protocols for use in new scenarios for which any existing protocols were not designed.

## Table of Contents

## 1. Introduction

Reliable Multicast Transport (RMT) protocols can be constructed in a variety of ways, some of which will work better for certain situations than others. It is believed that the requirements space for reliable multicast transport is sufficiently diverse that no one protocol can meet all the requirements [RFC2887]. However, it is also believed that there is sufficient commonality between the various approaches that it should be possible to define a number of building blocks [RFC3048] from which the various RMT protocols can be constructed.

One key benefit of this approach is that the same building block can be used multiple times in different protocol instantiations. Another key benefit is that building blocks may be upgraded as experience and understanding is gained. For this operation to be possible the building block needs to be clearly defined in terms of what it does, how it interacts with other building blocks, and how it fits into the overall architecture of a protocol instantiation. This description should also be sufficiently detailed so that those wishing to improve upon a particular building block or protocol instantiation can do so with a full understanding of the design decisions and tradeoffs that were made earlier.

The building block approach also presents some dangers that must be well understood in order to avoid potential specification flaws.

The most important danger is related to inappropriate usage of building blocks. Although efforts should be made in order to produce a modular and reusable specification of building blocks, for practical reasons this goal is not always fully achievable. This results in the specification of building blocks whose applicability is context dependent, which in turn creates the potential for the risk of co-dependence incompatibilities between building blocks. An example of such an incompatibility would be situation where the

combinations of building blocks A and B works, the combination of building blocks B and C works, however the combination of building blocks A, B, and C does not work.

In order to avoid misusage of and incompatibilities between building blocks, any external dependency must be highlighted in the building block specification. Furthermore, the specification must contain a precise applicability statement for the building block. Conversely, any protocol instantiation specification must state how any building block being used in it meets the protocol instantiation's applicability requirements. These guidelines are not intended to replace the common practice of Internet specification writing, but to augment them in a manner that better fits the RMT framework.

### 1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

## 2. The Guidelines

This document provides guidelines for authors of the two main kinds of RMT documents; building block documents and protocol instantiation documents. The guidelines for each are as follows.

### 2.1. Building Block Document Guidelines

All RMT Building block documents MUST contain sections that cover the following.

#### 2.1.1. Rationale

Individual building blocks SHOULD be reusable within multiple protocols and MUST provide functionality not present within other building blocks. If a building block is currently used in a single protocol instantiation, then it MUST specify some functionality that is likely to be reused in another (future) protocol instantiation.

The rationale section of a building block document must clearly define why the particular level of granularity for the functional decomposition resulted in that building block being chosen. If the granularity is too small it is highly likely that the building blocks will be trivial, and therefore require excessive additional effort to realize a working protocol. Conversely, if the level of granularity is too large, building blocks will only be usable within a single protocol instantiation. The rationale section MUST show that the level of granularity is appropriate so that neither problem occurs.

### 2.1.2. Functionality

The functionality section within a building block document MUST describe all algorithms and functions contained within the building block. In addition, the external interfaces for accessing these algorithms and functions MUST be fully specified so that the building block can be combined with other building blocks and any additional functionality specified within a protocol instantiation document to realize a working protocol.

### 2.1.3. Applicability Statement

One of the most important sections of a building block document will be the Applicability Statement. The purpose of this section is to provide sufficient details about the intended use of the building block so that potential authors of protocol instantiations will be able to use the building block in conformance to its applicability constraints. Also the Applicability Statement section will enable future building block document authors to quickly determine whether or not their particular need can be met with an existing building block. For this to be possible the Applicability Statement MUST describe:

- Intended scenarios for the building block's use.
- The building block's known failure modes, why they occur, and how they can be detected.
- A list of environmental considerations that includes but is not limited to whether the building block requires multi-source multicast or can be used in single-source only multicast networks, satellite networks, asymmetric networks, and wireless networks.
- A list of potential areas of conflict or incompatibilities with other building blocks.

### 2.1.4. Packet-Header Fields

If a building block implements a functionality whose realization requires an exchange of protocol messages between multiple agents, then the building block specification MUST state what kind of information is required and how the exchanged occurs. This includes detailed description of the data format and various communication requirements, such as timing constraints, and network requirements (e.g., multicast vs. unicast delivery).

Typically the data format specification is at the level of "generic header fields" without a full bit-level header specification. Generic header fields MAY specify additional requirements, such as representation precision or preferred position within the packet header (this last constraint might be dictated by efficiency concerns).

A building block specification MAY specify "abstract messages" that carry particular information for exclusive use within the building block, however, more frequently, it will rely on the protocol messages specified in the protocol instantiation to carry the information it needs.

The building block that provides Generic Router Assist functionality is an exception to the rule stated above. For efficiency reasons, this building block may fully specify header fields and positions of these fields within the packet-header.

### 2.1.5. Requirements from other Building Blocks

Each building block will specify a well defined piece of functionality that is common to multiple protocol instantiations. However, this does not mean that building block definitions will be generated in isolation from other building blocks. For example, a congestion control building block will have specific requirements regarding loss notification from either a NACK or ACK building block. The "Requirements from other Building Blocks" section is included to capture these requirements so that the authors of related building blocks can determine what functionality they need to provide in order to use a particular building block.

Specifically, the "Requirements from other Building Blocks section" MUST provide a complete and exhaustive enumeration of all the requirements that will be made upon other building blocks in order for the building block being specified to operate in its intended manner. Requirements that SHOULD be enumerated include but are not limited to:

- Event generation for and responses to other building blocks.
- Message ordering relative to messages from other building blocks.

### 2.1.6. Security Considerations

Protocol instantiations have the ultimate responsibility of addressing security requirements, in conformance to RFC 2357. Security considerations may not be applicable to generic building blocks other than a specific "security" building block. Some

building blocks, however, may raise special security issues, either due to the nature of communication required by the building block or due to the intended usage of the building block in a protocol instantiation. When special security issues are present in a building block, its specification MUST address them explicitly.

An example of this might be a building block that involves exchange of data that is particularly sensitive to security attacks.

### 2.1.7. Codepoint Considerations

Certain Building Blocks will specify general frameworks for describing functionality while leaving the detail open for implementation specific algorithms. One example of such a building block is the Forward Error Correction (FEC) building block which describes the framing aspects for FEC message fragments but not the algorithms used to generate the redundant data.

### 2.1.8. Summary Checklist

Rationale
- _ Provide justification for the building block's existence
- _ Provide rationale for the building block's granularity

Functionality
- _ Functionality contained within the building block
- _ External interfaces

Applicability Statement
- _ Intended usage
- _ Failure modes (including means of detection if known)
- _ Environmental considerations
- _ Incompatibilities / Conflicts with other building blocks

Packet Header Fields
- _ Specification of logical packet-header fields (*)
- _ Abstract messages specifications (*)

Requirements from other building blocks;
- _ Mandatory needs from other building blocks

Security Considerations
- _ Specify as much as possible (with respect to procedures, algorithms and data encoding), without affecting the general applicability of the building block.

(*) May not be applicable to some building blocks.

### 2.2. Protocol Instantiation Document Guidelines

Protocol Instantiation documents have one purpose: to specify how one can combine multiple building blocks to construct a new fully specified working protocol. To that end RMT Protocol Instantiation documents MUST contain the following four sections.

#### 2.2.1. Applicability Statement

The applicability statement's purpose is to frame the design space in which the fully realized protocol will operate and to thereby enable subsequent would-be RMT protocol designers to determine whether or not an existing protocol already meets their needs. For this to be possible the applicability statement MUST adhere to the following guidelines:

1) The target application space for which the protocol is intended MUST be clearly identified. For example; is the protocol to be used for real-time delivery, or non-real time file transfer?

2) The target scale, in terms of maximum number of receivers per session, for which the protocol is intended MUST be clearly specified. If the protocol has an architectural limitation resulting from the optimization of another feature, such as per packet acknowledgment, this SHOULD be included.

3) The applicability statement MUST identify the intended environments for the protocol's use AND list any environments in which the protocol should not be used. Example environments that should be considered include asymmetric networks, wireless networks, and satellite networks.

4) Finally, all protocols have inherent weaknesses that stem from the optimization for a specific feature. These weaknesses can manifest in spectacular failure modes when certain conditions occur. When known, these conditions and the nature of how the subsequent failure can be detected MUST be included in the applicability statement.

#### 2.2.2. Architecture Definition

Protocol Instantiations define how to combine one or more building blocks to create a working protocol. The Architecture Definition lays out the framework for how this take place. For this framework to be complete, it MUST contain the following information:

1) An overview of the major facets of the protocol's operation.

2) Full enumeration and overview of which Building Blocks are used with explicit references to their documents that define them.

3) An overview of how the aforementioned building blocks are to be joined.

4) A discussion of the design tradeoffs made in the selection of the chosen architecture.

#### 2.2.3. Conformance Statement

The conformance statement below MUST be included and adhered to:

> "This Protocol Instantiation document, in conjunction with the following Building Block documents identified in [list of relevant building block references] completely specifies a working reliable multicast transport protocol that conforms to the requirements described in RFC 2357."

Protocol instantiation document authors are specifically reminded that RFC 2357 requires that any RMT protocol put forward for standardization with the IETF is required to protect the network in as much as is possible. This does not mean that RMT protocols will be held to a higher standard than unicast transport protocols, merely that they should be designed to perform at least as well as unicast transport protocols when it comes to the possibility of protocol failure.

#### 2.2.4. Functionality Definition

Building Block documents will be incomplete in that they will specify an abstract framework of a building block's functionality. Complete algorithmic specifications for each building block along with any additional functionality MUST be provided within the Protocol Instantiation document's functionality definition. Furthermore, this description must show that each building block is used in accordance with its respective applicability statement. Finally the functionality description must provide a description of the abstract programming interface for interfacing the protocol instantiation with the applications that will use it.

### 2.2.5. Packet Formats

Once all the functionality has been fully defined, the Protocol Instantiation document must define the packet formats that will be used by the protocol. Each message part and the rules for their concatenation MUST be specified for both IPv4 [RFC791] and IPv6 [RFC2460]. Support for IPSEC [RFC2401] MUST be explicitly shown.

In recognition of the fact that protocols will evolve and that IP protocol numbers are a scarce resource, protocol instantiations MUST initially define packet formats for use over UDP [RFC768]. Whether or not a particular Reliable Multicast Transport protocol instantiation becomes sufficiently popular to warrant its own protocol number is an issue which will be deferred until such time that the protocol has been sufficiently widely deployed and understood.

### 2.2.6. Summary Checklist

Applicability Statement
- _ Target application space
- _ Target scale
- _ Intended environment
- _ Weaknesses and known failure modes

Architecture Definition
- _ Operational overview
- _ Building blocks used
- _ Details on how building blocks are joined

Conformance Statement
- _ Inclusion of mandatory paragraph

Functionality Definition
- _ Building block algorithmic specification
- _ Addition functionality specification
- _ Compliance with building block applicability statements
- _ Abstract program interface

Packet Formats
- _ IPv4 message parts
- _ IPv6 message parts
- _ IPSEC support
- _ Message ordering

## 3. IANA Considerations

There are no explicit IANA considerations for this document.

## 4. Acknowledgements

This document represents an overview of the mandatory elements required for the specification of building blocks and protocol instantiations within the RMT working group. The requirements presented are a summarization of discussions held between the RMT Working Group chairs and the participants in the IRTF Reliable Multicast Research Group. Although the name of these participants are too numerous to list here, the Working Group chairs would like to thank everyone who has participated in these discussions for their contributions.

## 5. References

[RFC768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980.

[RFC791] Postel, J., "Darpa Internet Protocol Specification", STD 5, RFC 791, September 1981.

[RFC2401] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC2460, December 1998.

[RFC2887] Handley, M., Floyd, S., Whetten, B., Kermode, R., Vicisano, L. and M. Luby, "The Reliable Multicast Design Space for Bulk Data Transfer", RFC 2887, August 2000.

[RFC3048] Whetten, B., Vicisano, L., Kermode, R., Handley, M., Floyd, S. and M. Luby, "Reliable Multicast Transport Building Blocks for One-to-Many Bulk-Data Transfer", RFC 3048, January 2001.

## 6. Authors' Addresses

Roger Kermode
Motorola Australian Research Centre
Locked Bag 5028
Botany NSW 1455,
Australia.

EMail: Roger.Kermode@motorola.com

Lorenzo Vicisano
Cisco Systems,
170 West Tasman Dr.
San Jose, CA 95134, USA

EMail: lorenzo@cisco.com

## 7. Full Copyright Statement



## Acknowledgement

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