Network Working Group
Request for Comments: 3417
STD: 62
Obsoletes: 1906
Category: Standards Track

Editor of this version:
R. Presuhn
BMC Software, Inc.
Authors of previous version:
J. Case
SNMP Research, Inc.
K. McCloghrie
Cisco Systems, Inc.
M. Rose
Dover Beach Consulting, Inc.
S. Waldbusser
International Network Services

December 2002

Transport Mappings for the Simple Network Management Protocol (SNMP)

Status of this Memo

Copyright Notice

Abstract

This document defines the transport of Simple Network Management Protocol (SNMP) messages over various protocols. This document obsoletes RFC 1906.

Table of Contents

1. Introduction

For a detailed overview of the documents that describe the current Internet-Standard Management Framework, please refer to section 7 of RFC 3410 [RFC3410].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. MIB objects are generally accessed through the Simple Network Management Protocol (SNMP). Objects in the MIB are defined using the mechanisms defined in the Structure of Management Information (SMI). This memo specifies a MIB

module that is compliant to the SMIv2, which is described in STD 58, RFC 2578 [RFC2578], STD 58, RFC 2579 [RFC2579] and STD 58, RFC 2580 [RFC2580].

This document, Transport Mappings for the Simple Network Management Protocol, defines how the management protocol [RFC3416] may be carried over a variety of protocol suites. It is the purpose of this document to define how the SNMP maps onto an initial set of transport domains. At the time of this writing, work was in progress to define an IPv6 mapping, described in [RFC3419]. Other mappings may be defined in the future.

Although several mappings are defined, the mapping onto UDP over IPv4 is the preferred mapping for systems supporting IPv4. Systems implementing IPv4 MUST implement the mapping onto UDP over IPv4. To maximize interoperability, systems supporting other mappings SHOULD also provide for access via the UDP over IPv4 mapping.

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 BCP 14, RFC 2119 [RFC2119].

2. Definitions

```
SNMPv2-TM DEFINITIONS ::= BEGIN

IMPORTS
    MODULE-IDENTITY, OBJECT-IDENTITY,
    snmpModules, snmpDomains, snmpProxys
        FROM SNMPv2-SMI
    TEXTUAL-CONVENTION
        FROM SNMPv2-TC;

snmpv2tm MODULE-IDENTITY
    LAST-UPDATED "200210160000Z"
    ORGANIZATION "IETF SNMPv3 Working Group"
    CONTACT-INFO
            "WG-EMail:   snmpv3@lists.tislabs.com
             Subscribe:  snmpv3-request@lists.tislabs.com

             Co-Chair:   Russ Mundy
                         Network Associates Laboratories
             postal:     15204 Omega Drive, Suite 300
                         Rockville, MD 20850-4601
                         USA
             EMail:      mundy@tislabs.com
             phone:      +1 301 947-7107
```

```
                Co-Chair:   David Harrington
                            Enterasys Networks
                postal:     35 Industrial Way
                            P. O. Box 5005
                            Rochester, NH 03866-5005
                            USA
                EMail:      dbh@enterasys.com
                phone:      +1 603 337-2614

                Editor:     Randy Presuhn
                            BMC Software, Inc.
                postal:     2141 North First Street
                            San Jose, CA 95131
                            USA
                EMail:      randy_presuhn@bmc.com
                phone:      +1 408 546-1006"
        DESCRIPTION
                "The MIB module for SNMP transport mappings.

                 Copyright (C) The Internet Society (2002). This
                 version of this MIB module is part of RFC 3417;
                 see the RFC itself for full legal notices.
                "
        REVISION     "200210160000Z"
        DESCRIPTION
                "Clarifications, published as RFC 3417."
        REVISION    "199601010000Z"
        DESCRIPTION
                "Clarifications, published as RFC 1906."
        REVISION    "199304010000Z"
        DESCRIPTION
                "The initial version, published as RFC 1449."
        ::= { snmpModules 19 }

    -- SNMP over UDP over IPv4

    snmpUDPDomain  OBJECT-IDENTITY
        STATUS     current
        DESCRIPTION
                "The SNMP over UDP over IPv4 transport domain.
                The corresponding transport address is of type
                SnmpUDPAddress."
        ::= { snmpDomains 1 }
```

```
   SnmpUDPAddress ::= TEXTUAL-CONVENTION
       DISPLAY-HINT "1d.1d.1d.1d/2d"
       STATUS       current
       DESCRIPTION
               "Represents a UDP over IPv4 address:

                  octets   contents        encoding
                   1-4     IP-address      network-byte order
                   5-6     UDP-port        network-byte order
               "
       SYNTAX       OCTET STRING (SIZE (6))

   -- SNMP over OSI

   snmpCLNSDomain OBJECT-IDENTITY
       STATUS     current
       DESCRIPTION
               "The SNMP over CLNS transport domain.
               The corresponding transport address is of type
               SnmpOSIAddress."
       ::= { snmpDomains 2 }

   snmpCONSDomain OBJECT-IDENTITY
       STATUS     current
       DESCRIPTION
               "The SNMP over CONS transport domain.
               The corresponding transport address is of type
               SnmpOSIAddress."
       ::= { snmpDomains 3 }

   SnmpOSIAddress ::= TEXTUAL-CONVENTION
       DISPLAY-HINT "*1x:/1x:"
       STATUS       current
       DESCRIPTION
               "Represents an OSI transport-address:

             octets   contents           encoding
                1     length of NSAP     'n' as an unsigned-integer
                                           (either 0 or from 3 to 20)
             2..(n+1) NSAP                concrete binary representation
             (n+2)..m TSEL                string of (up to 64) octets
               "
       SYNTAX       OCTET STRING (SIZE (1 | 4..85))
```

```
   -- SNMP over DDP

   snmpDDPDomain  OBJECT-IDENTITY
       STATUS     current
       DESCRIPTION
               "The SNMP over DDP transport domain.  The corresponding
               transport address is of type SnmpNBPAddress."
       ::= { snmpDomains 4 }

   SnmpNBPAddress ::= TEXTUAL-CONVENTION
       STATUS       current
       DESCRIPTION
               "Represents an NBP name:

            octets        contents          encoding
               1          length of object  'n' as an unsigned integer
             2..(n+1)     object            string of (up to 32) octets
              n+2         length of type    'p' as an unsigned integer
         (n+3)..(n+2+p)   type              string of (up to 32) octets
             n+3+p        length of zone    'q' as an unsigned integer
       (n+4+p)..(n+3+p+q) zone              string of (up to 32) octets

               For comparison purposes, strings are
               case-insensitive. All strings may contain any octet
               other than 255 (hex ff)."
       SYNTAX       OCTET STRING (SIZE (3..99))

   -- SNMP over IPX

   snmpIPXDomain  OBJECT-IDENTITY
       STATUS     current
       DESCRIPTION
               "The SNMP over IPX transport domain.  The corresponding
               transport address is of type SnmpIPXAddress."
       ::= { snmpDomains 5 }

   SnmpIPXAddress ::= TEXTUAL-CONVENTION
       DISPLAY-HINT "4x.1x:1x:1x:1x:1x:1x.2d"
       STATUS       current
       DESCRIPTION
               "Represents an IPX address:

                  octets   contents            encoding
                   1-4     network-number      network-byte order
                   5-10    physical-address    network-byte order
                  11-12    socket-number       network-byte order
               "
       SYNTAX       OCTET STRING (SIZE (12))
```

```
   -- for proxy to SNMPv1 (RFC 1157)

   rfc1157Proxy   OBJECT IDENTIFIER ::= { snmpProxys 1 }

   rfc1157Domain  OBJECT-IDENTITY
       STATUS     deprecated
       DESCRIPTION
               "The transport domain for SNMPv1 over UDP over IPv4.
               The corresponding transport address is of type
               SnmpUDPAddress."
       ::= { rfc1157Proxy 1 }

   --  ::= { rfc1157Proxy 2 }            this OID is obsolete

   END
```

3. SNMP over UDP over IPv4

This is the preferred transport mapping.

3.1. Serialization

Each instance of a message is serialized (i.e., encoded according to the convention of [BER]) onto a single UDP [RFC768] over IPv4 [RFC791] datagram, using the algorithm specified in Section 8.

3.2. Well-known Values

It is suggested that administrators configure their SNMP entities supporting command responder applications to listen on UDP port 161. Further, it is suggested that SNMP entities supporting notification receiver applications be configured to listen on UDP port 162.

When an SNMP entity uses this transport mapping, it must be capable of accepting messages up to and including 484 octets in size. It is recommended that implementations be capable of accepting messages of up to 1472 octets in size. Implementation of larger values is encouraged whenever possible.

4. SNMP over OSI

This is an optional transport mapping.

4.1. Serialization

Each instance of a message is serialized onto a single TSDU [IS8072] [IS8072A] for the OSI Connectionless-mode Transport Service (CLTS), using the algorithm specified in Section 8.

4.2. Well-known Values

It is suggested that administrators configure their SNMP entities supporting command responder applications to listen on transport selector "snmp-l" (which consists of six ASCII characters), when using a CL-mode network service to realize the CLTS. Further, it is suggested that SNMP entities supporting notification receiver applications be configured to listen on transport selector "snmpt-l" (which consists of seven ASCII characters, six letters and a hyphen) when using a CL-mode network service to realize the CLTS. Similarly, when using a CO-mode network service to realize the CLTS, the suggested transport selectors are "snmp-o" and "snmpt-o", for command responders and notification receivers, respectively.

When an SNMP entity uses this transport mapping, it must be capable of accepting messages that are at least 484 octets in size. Implementation of larger values is encouraged whenever possible.

5. SNMP over DDP

This is an optional transport mapping.

5.1. Serialization

Each instance of a message is serialized onto a single DDP datagram [APPLETALK], using the algorithm specified in Section 8.

5.2. Well-known Values

SNMP messages are sent using DDP protocol type 8. SNMP entities supporting command responder applications listen on DDP socket number 8, while SNMP entities supporting notification receiver applications listen on DDP socket number 9.

Administrators must configure their SNMP entities supporting command responder applications to use NBP type "SNMP Agent" (which consists of ten ASCII characters) while those supporting notification receiver applications must be configured to use NBP type "SNMP Trap Handler" (which consists of seventeen ASCII characters).

The NBP name for SNMP entities supporting command responders and notification receivers should be stable - NBP names should not change any more often than the IP address of a typical TCP/IP node. It is suggested that the NBP name be stored in some form of stable storage.

When an SNMP entity uses this transport mapping, it must be capable of accepting messages that are at least 484 octets in size. Implementation of larger values is encouraged whenever possible.

5.3. Discussion of AppleTalk Addressing

The AppleTalk protocol suite has certain features not manifest in the TCP/IP suite. AppleTalk's naming strategy and the dynamic nature of address assignment can cause problems for SNMP entities that wish to manage AppleTalk networks. TCP/IP nodes have an associated IP address which distinguishes each from the other. In contrast, AppleTalk nodes generally have no such characteristic. The network-level address, while often relatively stable, can change at every reboot (or more frequently).

Thus, when SNMP is mapped over DDP, nodes are identified by a "name", rather than by an "address". Hence, all AppleTalk nodes that implement this mapping are required to respond to NBP lookups and confirms (e.g., implement the NBP protocol stub), which guarantees that a mapping from NBP name to DDP address will be possible.

In determining the SNMP identity to register for an SNMP entity, it is suggested that the SNMP identity be a name which is associated with other network services offered by the machine.

NBP lookups, which are used to map NBP names into DDP addresses, can cause large amounts of network traffic as well as consume CPU resources. It is also the case that the ability to perform an NBP lookup is sensitive to certain network disruptions (such as zone table inconsistencies) which would not prevent direct AppleTalk communications between two SNMP entities.

Thus, it is recommended that NBP lookups be used infrequently, primarily to create a cache of name-to-address mappings. These cached mappings should then be used for any further SNMP traffic. It is recommended that SNMP entities supporting command generator applications should maintain this cache between reboots. This caching can help minimize network traffic, reduce CPU load on the network, and allow for (some amount of) network trouble shooting when the basic name-to-address translation mechanism is broken.

5.3.1. How to Acquire NBP names

An SNMP entity supporting command generator applications may have a pre-configured list of names of "known" SNMP entities supporting command responder applications. Similarly, an SNMP entity supporting command generator or notification receiver applications might interact with an operator. Finally, an SNMP entity supporting command generator or notification receiver applications might communicate with all SNMP entities supporting command responder or notification originator applications in a set of zones or networks.

5.3.2. When to Turn NBP names into DDP addresses

When an SNMP entity uses a cache entry to address an SNMP packet, it should attempt to confirm the validity mapping, if the mapping hasn't been confirmed within the last T1 seconds. This cache entry lifetime, T1, has a minimum, default value of 60 seconds, and should be configurable.

An SNMP entity supporting a command generator application may decide to prime its cache of names prior to actually communicating with another SNMP entity. In general, it is expected that such an entity may want to keep certain mappings "more current" than other mappings, e.g., those nodes which represent the network infrastructure (e.g., routers) may be deemed "more important".

Note that an SNMP entity supporting command generator applications should not prime its entire cache upon initialization - rather, it should attempt resolutions over an extended period of time (perhaps in some pre-determined or configured priority order). Each of these resolutions might, in fact, be a wildcard lookup in a given zone.

An SNMP entity supporting command responder applications must never prime its cache. When generating a response, such an entity does not need to confirm a cache entry. An SNMP entity supporting notification originator applications should do NBP lookups (or confirms) only when it needs to send an SNMP trap or inform.

5.3.3. How to Turn NBP names into DDP addresses

If the only piece of information available is the NBP name, then an NBP lookup should be performed to turn that name into a DDP address. However, if there is a piece of stale information, it can be used as a hint to perform an NBP confirm (which sends a unicast to the network address which is presumed to be the target of the name lookup) to see if the stale information is, in fact, still valid.

An NBP name to DDP address mapping can also be confirmed implicitly using only SNMP transactions. For example, an SNMP entity supporting command generator applications issuing a retrieval operation could also retrieve the relevant objects from the NBP group [RFC1742] for the SNMP entity supporting the command responder application. This information can then be correlated with the source DDP address of the response.

5.3.4. What if NBP is broken

Under some circumstances, there may be connectivity between two SNMP entities, but the NBP mapping machinery may be broken, e.g.,

- the NBP FwdReq (forward NBP lookup onto local attached network) mechanism might be broken at a router on the other entity's network; or,

- the NBP BrRq (NBP broadcast request) mechanism might be broken at a router on the entity's own network; or,

- NBP might be broken on the other entity's node.

An SNMP entity supporting command generator applications which is dedicated to AppleTalk management might choose to alleviate some of these failures by directly implementing the router portion of NBP. For example, such an entity might already know all the zones on the AppleTalk internet and the networks on which each zone appears. Given an NBP lookup which fails, the entity could send an NBP FwdReq to the network in which the SNMP entity supporting the command responder or notification originator application was last located. If that failed, the station could then send an NBP LkUp (NBP lookup packet) as a directed (DDP) multicast to each network number on that network. Of the above (single) failures, this combined approach will solve the case where either the local router's BrRq-to-FwdReq mechanism is broken or the remote router's FwdReq-to-LkUp mechanism is broken.

6. SNMP over IPX

This is an optional transport mapping.

6.1. Serialization

Each instance of a message is serialized onto a single IPX datagram [NOVELL], using the algorithm specified in Section 8.

6.2. Well-known Values

SNMP messages are sent using IPX packet type 4 (i.e., Packet Exchange Protocol).

It is suggested that administrators configure their SNMP entities supporting command responder applications to listen on IPX socket 36879 (900f hexadecimal). Further, it is suggested that those supporting notification receiver applications be configured to listen on IPX socket 36880 (9010 hexadecimal).

When an SNMP entity uses this transport mapping, it must be capable of accepting messages that are at least 546 octets in size. Implementation of larger values is encouraged whenever possible.

7. Proxy to SNMPv1

Historically, in order to support proxy to SNMPv1, as defined in [RFC2576], it was deemed useful to define a transport domain, rfc1157Domain, which indicates the transport mapping for SNMP messages as defined in [RFC1157].

8. Serialization using the Basic Encoding Rules

When the Basic Encoding Rules [BER] are used for serialization:

(1) When encoding the length field, only the definite form is used; use of the indefinite form encoding is prohibited. Note that when using the definite-long form, it is permissible to use more than the minimum number of length octets necessary to encode the length field.

(2) When encoding the value field, the primitive form shall be used for all simple types, i.e., INTEGER, OCTET STRING, and OBJECT IDENTIFIER (either IMPLICIT or explicit). The constructed form of encoding shall be used only for structured types, i.e., a SEQUENCE or an IMPLICIT SEQUENCE.

(3) When encoding an object whose syntax is described using the BITS construct, the value is encoded as an OCTET STRING, in which all the named bits in (the definition of) the bitstring, commencing with the first bit and proceeding to the last bit, are placed in bits 8 (high order bit) to 1 (low order bit) of the first octet, followed by bits 8 to 1 of each subsequent octet in turn, followed by as many bits as are needed of the final subsequent octet, commencing with bit 8. Remaining bits, if any, of the final octet are set to zero on generation and ignored on receipt.

These restrictions apply to all aspects of ASN.1 encoding, including the message wrappers, protocol data units, and the data objects they contain.

8.1. Usage Example

As an example of applying the Basic Encoding Rules, suppose one wanted to encode an instance of the GetBulkRequest-PDU [RFC3416]:

```
  [5] IMPLICIT SEQUENCE {
          request-id      1414684022,
          non-repeaters   1,
          max-repetitions 2,
          variable-bindings {
              { name sysUpTime,
                value { unSpecified NULL } },
              { name ipNetToMediaPhysAddress,
                value { unSpecified NULL } },
              { name ipNetToMediaType,
                value { unSpecified NULL } }
          }
      }
```

Applying the BER, this may be encoded (in hexadecimal) as:

```
[5] IMPLICIT SEQUENCE          a5 82 00 39
    INTEGER                    02 04 54 52 5d 76
    INTEGER                    02 01 01
    INTEGER                    02 01 02
    SEQUENCE (OF)              30 2b
        SEQUENCE               30 0b
            OBJECT IDENTIFIER  06 07 2b 06 01 02 01 01 03
            NULL               05 00
        SEQUENCE               30 0d
            OBJECT IDENTIFIER  06 09 2b 06 01 02 01 04 16 01 02
            NULL               05 00
        SEQUENCE               30 0d
            OBJECT IDENTIFIER  06 09 2b 06 01 02 01 04 16 01 04
            NULL               05 00
```

Note that the initial SEQUENCE in this example was not encoded using the minimum number of length octets. (The first octet of the length, 82, indicates that the length of the content is encoded in the next two octets.)

9. Notice on Intellectual Property

10. Acknowledgments

This document is the product of the SNMPv3 Working Group. Some special thanks are in order to the following Working Group members:

Randy Bush
Jeffrey D. Case
Mike Daniele
Rob Frye
Lauren Heintz
Keith McCloghrie
Russ Mundy
David T. Perkins
Randy Presuhn
Aleksey Romanov
Juergen Schoenwaelder
Bert Wijnen

This version of the document, edited by Randy Presuhn, was initially based on the work of a design team whose members were:

Jeffrey D. Case
Keith McCloghrie
David T. Perkins
Randy Presuhn
Juergen Schoenwaelder

The previous versions of this document, edited by Keith McCloghrie, was the result of significant work by four major contributors:

Jeffrey D. Case
Keith McCloghrie
Marshall T. Rose
Steven Waldbusser

Additionally, the contributions of the SNMPv2 Working Group to the previous versions are also acknowledged. In particular, a special thanks is extended for the contributions of:

Alexander I. Alten
Dave Arneson
Uri Blumenthal
Doug Book
Kim Curran
Jim Galvin
Maria Greene
Iain Hanson
Dave Harrington
Nguyen Hien
Jeff Johnson
Michael Kornegay
Deirdre Kostick
David Levi
Daniel Mahoney
Bob Natale
Brian O'Keefe
Andrew Pearson
Dave Perkins
Randy Presuhn
Aleksey Romanov
Shawn Routhier
Jon Saperia
Juergen Schoenwaelder
Bob Stewart
Kaj Tesink
Glenn Waters
Bert Wijnen

11. IANA Considerations

The SNMPv2-TM MIB module requires the allocation of a single object identifier for its MODULE-IDENTITY. IANA has allocated this object identifier in the snmpModules subtree, defined in the SNMPv2-SMI MIB module.

12. Security Considerations

SNMPv1 by itself is not a secure environment. Even if the network itself is secure (for example by using IPSec), even then, there is no control as to who on the secure network is allowed to access and GET/SET (read/change) the objects accessible through a command responder application.

It is recommended that the implementors consider the security features as provided by the SNMPv3 framework. Specifically, the use of the User-based Security Model STD 62, RFC 3414 [RFC3414] and the View-based Access Control Model STD 62, RFC 3415 [RFC3415] is recommended.

It is then a customer/user responsibility to ensure that the SNMP entity giving access to a MIB is properly configured to give access to the objects only to those principals (users) that have legitimate rights to indeed GET or SET (change) them.

13. References

13.1. Normative References

[BER] Information processing systems - Open Systems Interconnection - Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1), International Organization for Standardization. International Standard 8825, December 1987.

[IS8072] Information processing systems - Open Systems Interconnection - Transport Service Definition, International Organization for Standardization. International Standard 8072, June 1986.

[IS8072A] Information processing systems - Open Systems Interconnection - Transport Service Definition - Addendum 1: Connectionless-mode Transmission, International Organization for Standardization. International Standard 8072/AD 1, December 1986.

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

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

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

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

[RFC2579] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.

[RFC2580] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.

[RFC3414] Blumenthal, U. and B. Wijnen, "The User-Based Security Model (USM) for Version 3 of the Simple Network Management Protocol (SNMPv3)", STD 62, RFC 3414, December 2002.

[RFC3415] Wijnen, B., Presuhn, R. and K. McCloghrie, "View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3415, December 2002.

[RFC3416] Presuhn, R., Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3416, December 2002.

13.2. Informative References

[APPLETALK] Sidhu, G., Andrews, R. and A. Oppenheimer, Inside AppleTalk (second edition). Addison-Wesley, 1990.

[NOVELL] Network System Technical Interface Overview. Novell, Inc., June 1989.

[RFC1157] Case, J., Fedor, M., Schoffstall, M. and J. Davin, "Simple Network Management Protocol", STD 15, RFC 1157, May 1990.

[RFC1742] Waldbusser, S. and K. Frisa, "AppleTalk Management Information Base II", RFC 1742, January 1995.

[RFC2576] Frye, R., Levi, D., Routhier, S. and B. Wijnen, "Coexistence between Version 1, Version 2, and Version 3 of the Internet-Standard Network Management Framework", RFC 2576, March 2000.

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

[RFC3419] Daniele, M. and J. Schoenwaelder, "Textual Conventions for Transport Addresses", RFC 3419, November 2002.

14. Changes from RFC 1906

This document differs from RFC 1906 only in editorial improvements. The protocol is unchanged.

15. Editor's Address

Randy Presuhn
BMC Software, Inc.
2141 North First Street
San Jose, CA 95131
USA

Phone: +1 408 546-1006
EMail: randy_presuhn@bmc.com

16. Full Copyright Statement

Acknowledgement

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