

A Joint Standard of AASHTO, ITE, and NEMA

NTCIP 1103 version v01

National Transportation Communications for ITS Protocol Transportation Management Protocols

published in January 2009

Published by

American Association of State Highway and Transportation Officials (AASHTO)

444 North Capitol Street, N.W., Suite 249
Washington, D.C. 20001

Institute of Transportation Engineers (ITE)

1099 14th Street, N.W., Suite 300 West
Washington, D.C. 20005-3438

National Electrical Manufacturers Association (NEMA)

1300 North 17th Street, Suite 1752
Rosslyn, Virginia 22209-3806

© 2008 AASHTO / ITE / NEMA. All rights reserved.

file version 1103 v01.27r rita

NOTICES

Copyright Notice

© 2008 by the American Association of State Highway and Transportation Officials (AASHTO), the Institute of Transportation Engineers (ITE), and the National Electrical Manufacturers Association (NEMA). All intellectual property rights, including, but not limited to, the rights of reproduction, translation and display are reserved under the laws of the United States of America, the Universal Copyright Convention, the Berne Convention, and the International and Pan American Copyright Conventions. Except as licensed or permitted, you may not copy these materials without prior written permission from AASHTO, ITE, or NEMA. Use of these materials does not give you any rights of ownership or claim of copyright in or to these materials.

Visit www.ntcip.org for other copyright information, for instructions to request reprints of excerpts, and to request reproduction that is not granted below.

PDF File License Agreement

To the extent that these materials are distributed by AASHTO / ITE / NEMA in the form of an Adobe® Portable Document Format (PDF) electronic data file (the "PDF file"), AASHTO / ITE / NEMA authorizes each registered PDF file user to view, download, copy, or print the PDF file available from the authorized Web site, subject to the terms and conditions of this license agreement:

- (a) you may download one copy of each PDF file for personal, noncommercial, and intraorganizational use only;
- (b) ownership of the PDF file is not transferred to you; you are licensed to use the PDF file;
- (c) you may make one more electronic copy of the PDF file, such as to a second hard drive or burn to a CD;
- (d) you agree not to copy, distribute, or transfer the PDF file from that media to any other electronic media or device;
- (e) you may print one paper copy of the PDF file;
- (f) you may make one paper reproduction of the printed copy;
- (g) any permitted copies of the PDF file must retain the copyright notice, and any other proprietary notices contained in the file;
- (h) the PDF file license does not include (1) resale of the PDF file or copies, (2) republishing the content in compendiums or anthologies, (3) publishing excerpts in commercial publications or works for hire, (4) editing or modification of the PDF file except those portions as permitted, (5) posting on network servers or distribution by electronic mail or from electronic storage devices, and (6) translation to other languages or conversion to other electronic formats;
- (i) other use of the PDF file and printed copy requires express, prior written consent.

The information in this publication was considered technically sound by the consensus of persons engaged in the development and approval of the document at the time it was developed. Consensus does not necessarily mean that there is unanimous agreement among every person participating in the development of this document.

AASHTO, ITE, and NEMA standards and guideline publications, of which the document contained herein is one, are developed through a voluntary consensus standards development process. This process brings together volunteers and seeks out the views of persons who have an interest in the topic covered by this publication. While AASHTO, ITE, and NEMA administer the process and establish rules to promote fairness in the development of consensus, they do not write the document and they do not independently test, evaluate, or verify the accuracy or completeness of any information or the soundness of any judgments contained in their standards and guideline publications.

AASHTO, ITE, and NEMA disclaim liability for any personal injury, property, or other damages of any nature whatsoever, whether special, indirect, consequential, or compensatory, directly or indirectly resulting from the publication, use of, application, or reliance on this document. AASHTO, ITE, and NEMA disclaim and make no guaranty or warranty, express or implied, as to the accuracy or completeness of any information published herein, and disclaims and makes no warranty that the information in this document will fulfill any of your particular purposes or needs. AASHTO, ITE, and NEMA do not undertake to guarantee the performance of any individual manufacturer or seller's products or services by virtue of this standard or guide.

In publishing and making this document available, AASHTO, ITE, and NEMA are not undertaking to render professional or other services for or on behalf of any person or entity, nor are AASHTO, ITE, and NEMA undertaking to perform any duty owed by any person or entity to someone else. Anyone using this document should rely on his or her own independent judgment or, as appropriate, seek the advice of a competent professional in determining the exercise of reasonable care in any given circumstances. Information and other standards on the topic covered by this publication may be available from other sources, which the user may wish to consult for additional views or information not covered by this publication.

AASHTO, ITE, and NEMA have no power, nor do they undertake to police or enforce compliance with the contents of this document. AASHTO, ITE, and NEMA do not certify, test, or inspect products, designs, or installations for safety or health purposes. Any certification or other statement of compliance with any health or safety-related information in this document shall not be attributable to AASHTO, ITE, or NEMA and is solely the responsibility of the certifier or maker of the statement.

Data Dictionary and MIB Distribution Permission

To the extent that these materials are distributed by AASHTO / ITE / NEMA in the form of a Data Dictionary ("DD") or Management Information Base ("MIB"), AASHTO / ITE / NEMA extend the following permission:

You may make and/or distribute unlimited copies, including derivative works, of the DD or MIB, including copies for commercial distribution, provided that:

- (i) each copy you make and/or distribute contains the citation "Derived from NTCIP 0000 [insert the standard number]. Used by permission of AASHTO / ITE / NEMA.";
- (ii) the copies or derivative works are not made part of the standards publications or works offered by other standards developing organizations or publishers or as works-for-hire not associated with commercial hardware or software products intended for field implementation;
- (iii) use of the DD or MIB is restricted in that the syntax fields may be modified only to reflect a more restrictive subrange or enumerated values;
- (iv) the description field may be modified but only to the extent that: (a) only those bit values or enumerated values that are supported are listed; and (b) the more restrictive subrange is expressed.

These materials are delivered "AS IS" without any warranties as to their use or performance.

AASHTO / ITE / NEMA and their suppliers do not warrant the performance or results you may obtain by using these materials. AASHTO / ITE / NEMA and their suppliers make no warranties, express or implied, as to noninfringement of third party rights, merchantability, or fitness for any particular purpose. In no event will AASHTO / ITE / NEMA or their suppliers be liable to you or any third party for any claim or for any consequential, incidental or special damages, including any lost profits or lost savings, arising from your reproduction or use of these materials, even if an AASHTO / ITE / NEMA representative has been advised of the possibility of such damages.

Some states or jurisdictions do not allow the exclusion or limitation of incidental, consequential or special damages, or the exclusion of implied warranties, so the above limitations may not apply to you.

Use of these materials does not constitute an endorsement or affiliation by or between AASHTO, ITE, or NEMA and you, your company, or your products and services.

If you are unwilling to accept the foregoing restrictions, you should immediately return these materials.

PRL and RTM Distribution Permission

To the extent that these materials are distributed by AASHTO / ITE / NEMA in the form of a Profile Requirements List ("PRL") or a Requirements Traceability Matrix ("RTM"), AASHTO / ITE / NEMA extend the following permission:

- (i) you may make and/or distribute unlimited copies, including derivative works of the PRL (then known as a Profile Implementation Conformance Statement ("PICS")) or the RTM, provided that each copy you make and/or distribute contains the citation "Based on NTCIP 0000 [insert the standard number] PRL or RTM. Used by permission. Original text (C) AASHTO / ITE / NEMA.";
- (ii) you may not modify the PRL or the RTM except for the Project Requirements column, which is the only column that may be modified to show a products' implementation or the project-specific requirements; and
- (iii) if the PRL or RTM excerpt is made from an unapproved draft, add to the citation "PRL (or RTM) excerpted from a draft document containing preliminary information that is subject to change."

The permission is limited to not include reuse in works offered by other standards developing organizations or publishers, and to not include reuse in works-for-hire or compendiums or electronic storage that are not associated with commercial hardware or software products intended for field installation.

A PICS is a Profile Requirements List which is completed to indicate the features that are supported in an implementation. Visit www.ntcip.org for information on electronic copies of the MIBs, PRLs, and RTMs.

Content and Liability Disclaimer

The information in this publication was considered technically sound by the consensus of persons engaged in the development and approval of the document at the time it was developed. Consensus does not necessarily mean that there is unanimous agreement among every person participating in the development of this document.

AASHTO, ITE, and NEMA standards and guideline publications, of which the document contained herein is one, are developed through a voluntary consensus standards development process. This process brings together volunteers and/or seeks out the views of persons who have an interest in the topic covered by this publication. While AASHTO, ITE, and NEMA administer the process and establish rules to promote fairness in the development of consensus, they do not write the document and they do not independently test, evaluate, or verify the accuracy or completeness of any information or the soundness of any judgments contained in their standards and guideline publications.

AASHTO, ITE, and NEMA disclaim liability for any personal injury, property, or other damages of any nature whatsoever, whether special, indirect, consequential, or compensatory, directly or indirectly resulting from the publication, use of, application, or reliance on this document. AASHTO, ITE, and NEMA disclaim and make no guaranty or warranty, express or implied, as to the accuracy or completeness of any information published herein, and disclaims and makes no warranty that the information in this document will fulfill any of your particular purposes or needs. AASHTO, ITE, and NEMA do not undertake to guarantee the performance of any individual manufacturer or seller's products or services by virtue of this standard or guide.

In publishing and making this document available, AASHTO, ITE, and NEMA are not undertaking to

render professional or other services for or on behalf of any person or entity, nor are AASHTO, ITE, and NEMA undertaking to perform any duty owed by any person or entity to someone else. Anyone using this document should rely on his or her own independent judgment or, as appropriate, seek the advice of a competent professional in determining the exercise of reasonable care in any given circumstances. Information and other standards on the topic covered by this publication may be available from other sources, which the user may wish to consult for additional views or information not covered by this publication.

AASHTO, ITE, and NEMA have no power, nor do they undertake to police or enforce compliance with the contents of this document. AASHTO, ITE, and NEMA do not certify, test, or inspect products, designs, or installations for safety or health purposes. Any certification or other statement of compliance with any health or safety-related information in this document shall not be attributable to AASHTO, ITE, or NEMA and is solely the responsibility of the certifier or maker of the statement.

Trademark Notice

NTCIP is a trademark of AASHTO / ITE / NEMA. All other marks mentioned in this standard are the trademarks of their respective owners.

< This page is intentionally left blank. >

ACKNOWLEDGEMENTS

NTCIP 1103 v01 was prepared by the NTCIP Base Standards and Profiles Working Group (BSP2 WG), which is a subdivision of the Joint Committee on the NTCIP. The Joint Committee is organized under a Memorandum of Understanding among the American Association of State Highway and Transportation Officials (AASHTO), the Institute of Transportation Engineers (ITE), and the National Electrical Manufacturers Association (NEMA). The Joint Committee on the NTCIP consists of six representatives from each of the standards organizations, and provides guidance for NTCIP development.

At the time that NTCIP 1103 v01 was prepared, the following individuals were active members of NTCIP BSP WG:

- Robert DeRoche (Chair)
- Bud Kent
- Alex Mousadi
- Nu Rosenbohm
- Ken Vaughn

Other individuals providing input to NTCIP 1103 v01 include:

- Blake Christie
- Michael Forbis
- Bob Force
- Gary Meredith

In addition to the many volunteer efforts, recognition is also given to those organizations who supported the efforts of BSP2 WG by providing comments and funding, including:

- CalTrans
- Econolite
- Iteris, Inc.
- Mastek
- Ministry of Transportation Ontario
- Mitretek
- PB Farradyne Inc.
- PEEK Traffic
- Robert DeRoche Consulting
- Siemens ITS
- Southwest Research Institute
- Texas Department of Transportation
- Trevilon Corporation
- Transcore
- U.S. Department of Transportation, Research and Innovative Technology Administration
- Washington State Department of Transportation

< This page is intentionally left blank. >

FOREWORD

NTCIP 1103 v01, an NTCIP standards publication, defines a composite application layer protocol for the management of transportation equipment. The composite protocol consists of three component protocols: the Internet-standard Simple Network Management Protocol (SNMP), the Simple Fixed Message Protocol (SFMP), and the Simple Transportation Management Protocol (STMP). The protocols are concerned with the procedures for exchanging information, as well as the format in which the information is exchanged. When relayed to the ISO OSI Reference model, these protocols are concerned with the upper three layers (application, session and presentation layer). NTCIP 1103 v01 contains two normative and two informative annexes. NTCIP 1103 v01 uses only metric units.

NTCIP 1103 v01 is also an NTCIP Base Standard. NTCIP Base Standards provide definitions of the procedures and data formats for use within NTCIP systems. For more information about NTCIP standards, visit the NTCIP Web Site at www.ntcip.org.

User Comment Instructions

The term "User Comment" includes any type of written inquiry, comment, question, or proposed revision, from an individual person or organization, about any part of this standard publication's content. A "Request for Interpretation" of this standard publication is also classified as a User Comment. User Comments are solicited at any time. In preparation of this NTCIP standards publication, input of users and other interested parties was sought and evaluated.

All User Comments will be referred to the committee responsible for developing and/or maintaining this standards publication. The committee chairperson, or their designee, may contact the submitter for clarification of the User Comment. When the committee chairperson or designee reports the committee's consensus opinion related to the User Comment, that opinion will be forwarded to the submitter. The committee chairperson may report that action on the User Comment may be deferred to a future committee meeting and/or a future revision of the standards publication. Previous User Comments and their disposition may be available for reference and information at www.ntcip.org.

A User Comment should be submitted to this address:

NTCIP Coordinator
National Electrical Manufacturers Association
1300 North 17th Street, Suite 1752
Rosslyn, Virginia 22209-3806
e-mail: ntcip@nema.org

A User Comment should be submitted in the following form:

Standard Publication number and version:
Page:
Section, Paragraph, or Clause:
Comment:
Editorial or Substantive?:
Suggested Alternative Language:

Please include your name, organization, and address in your correspondence.

Approvals

This standards publication was separately balloted and approved by AASHTO, ITE, and NEMA after recommendation by the Joint Committee on the NTCIP. Each organization has approved this standard as the following standard type, as of the date:

AASHTO—Standard Specification; November 2005
ITE—Software Standard; October 2005
NEMA—Standard; October 2005

History

From 1996 to 2004, parts of NTCIP 1103 v01, including the definition of STMP, were defined in NTCIP 1101:1996 / NEMA TS 3.2. However, in order to provide a more organized and modular set of standards, NTCIP 1101:1996 was separated into three distinct standards: NTCIP 1103 v01, which includes the definition of STMP; NTCIP 1102:2004, which defines the Octet Encoding Rules (OER); and NTCIP 8004 v01, which defines the Structure and Management of Transportation Information (SMI). These three standards completely replace NTCIP 1101 / NEMA TS 3.2.

NTCIP 1103 v01.18, July 2002—Accepted as User Comment Draft by the Joint Committee on the NTCIP.

NTCIP 1103 v01.25, October 2004—Accepted as Recommended Standard by the Joint Committee on the NTCIP.

NTCIP 1103 v01.26, June 2005—Addressed pre-ballot comment on community name and prepared draft for SDO balloting.

NTCIP 1103 v01.27, November 2005 to December 2008—Prepared standard for publication; edited format, front matter, and style.

Compatibility of Versions

To distinguish NTCIP 1103 v01 (as published) from previous drafts, NTCIP 1103 v01 also includes NTCIP 1103 v01.27 on each page header. All NTCIP Standards Publications have a major and minor version number for configuration management. The version number syntax is "v00.00a," with the major version number before the period, and the minor version number and edition letter (if any) after the period.

NTCIP 1103 v01 is designated, and should be cited as, NTCIP 1103 v01. Anyone using NTCIP 1103 v01 should seek information about the version number that is of interest to them in any given circumstance. The MIB, the PRL, and the PICS should all reference the version number of the standards publication that was the source of the excerpted material.

Compliant systems based on later, or higher, version numbers MAY NOT be compatible with compliant systems based on earlier, or lower, version numbers. Anyone using NTCIP 1103 v01 should also consult NTCIP 8004 v01 for specific guidelines on compatibility.

STMP, as defined within NTCIP 1103 v01 is 100% consistent with the definition contained in NTCIP 1101:1996 / NEMA TS 3.2; however, the protocol has now been extended to support the new Simple Fixed Message Protocol. NTCIP 1103 v01 has also been expanded to address several other issues and to incorporate some of the protocol-specific data originally defined in NTCIP 1201:2005.

INTRODUCTION

NTCIP 1103 v01 provides the definition of a composite application layer protocol that consists of three component protocols. All three protocols provide the same base services, but are designed with different needs in mind. NTCIP 1103 v01 also defines a limited number of data elements necessary to manage these protocols. The data is defined according to the rules of NTCIP 8004.

NTCIP 1103 v01 defines requirements that are applicable to all NTCIP environments and also contains optional and conditional clauses that are applicable to specific environments for which they are intended.

The following keywords apply to NTCIP 1103 v01: AASHTO, ITE, NEMA, NTCIP, protocol, message, transportation, simple, TMP, SFMP, STMP.

In 1992, the NEMA 3-TS Transportation Management Systems and Associated Control Devices Section began the effort to develop the NTCIP. The Transportation Section's purpose was to respond to user needs to include standardized systems communication in the NEMA TS 2 standard, *Traffic Controller Assemblies*. Under the guidance of the Federal Highway Administration's NTCIP Steering Group, the NEMA effort was expanded to include the development of communications standards for all transportation field devices that could be used in an Intelligent Transportation Systems (ITS) network.

In September 1996, an agreement was reached among AASHTO, ITE, and NEMA to jointly develop, approve, and maintain NTCIP Standards. In late 1998, the Base Standards and Protocols Working Group was tasked with the effort to develop and maintain base standards for the NTCIP. The first meeting of the BSP WG was in January 1999, and the first draft of NTCIP 1103 v01 was written in May 2000. In late 2003, the Joint Committee on the NTCIP merged the Base Standards and Protocols Working Group with the Profiles Working Group and the new group was designated the Base Standards and Profiles Working Group. The first meeting of the merged working group was held in January 2004.

< This page is intentionally left blank. >

CONTENTS

	Page
SECTION 1 GENERAL.....	1
1.1 Scope	1
1.2 References	1
1.2.1 Normative References.....	1
1.2.2 Other References	2
1.2.3 Contact Information	2
1.3 Terms	3
1.4 Acronyms.....	5
1.5 Layout	5
SECTION 2 TRANSPORTATION MANAGEMENT PROTOCOL	6
2.1 Composition of TMP	7
2.2 Simultaneous Processing.....	7
2.3 Protocol Identification Logic	8
SECTION 3 SIMPLE NETWORK MANAGEMENT PROTOCOL	10
3.1 Overview.....	10
3.2 Definition.....	10
3.2.1 General Rules.....	11
3.2.2 Set Operations on a Read-Only Variable	11
3.2.3 Extra Data Prohibition.....	11
3.2.4 Response Time	11
SECTION 4 SIMPLE FIXED MESSAGE PROTOCOL	12
4.1 Overview.....	12
4.1.1 Data Identification.....	12
4.1.2 Packet Structure	12
4.1.3 Encoding.....	12
4.2 Definition.....	13
4.2.1 Rules.....	13
4.2.2 Elements of Procedure	13
4.2.3 SFMP Data Packet Structures	16
4.2.4 Definitions of Data Structure Fields.....	18
4.3 Examples.....	21
4.3.1 Get an Object Example	21
4.3.2 Get Block Object with Community Name Example	21
4.3.3 SFMP Set Example	22
4.3.4 SFMP Set Block Object Example.....	23
4.3.5 Get Error Example.....	24
4.3.6 Set Error Example	25
SECTION 5 SIMPLE TRANSPORTATION MANAGEMENT PROTOCOL (STMP)	27
5.1 Overview.....	27
5.1.1 Dynamic Objects	27
5.1.2 Other Truncations.....	29
5.2 Definition.....	29
5.2.1 Rules.....	29
5.2.2 Elements of Procedure	29
5.2.3 STMP Data Packet Structure	33
5.2.4 Dynamic Object Configuration.....	34
5.3 Examples.....	36
5.3.1 Configuring a Dynamic Object.....	36
5.3.2 Getting a Dynamic Object	37

5.3.3	Setting a Dynamic Object.....	38
SECTION 6 RESERVED FOR TRAP DEFINITION		39
SECTION 7 LOGICAL NAMES.....		40
SECTION 8 SECURITY		41
8.1	SNMP and SFMP Security	41
8.2	STMP Security.....	41
SECTION 9 CONFORMANCE STATEMENT		43
ANNEX A TRANSPORTATION MANAGEMENT PROTOCOLS MANAGEMENT INFORMATION BASE (NORMATIVE)		44
A.1.	TMP MIB Header	44
A.2.	Type Definitions.....	46
A.3.	Objects for SNMP	46
A.3.1.	Maximum SNMP Packet Size Parameter	46
A.4.	Objects for SFMP	47
A.4.1.	Number of Incoming SFMP Packets	47
A.4.2.	Number of Outgoing SFMP Packets	47
A.4.3.	Number of Incoming SFMP Packets with Bad Version Numbers	47
A.4.4.	Number of Incoming SFMP Packets with Bad Community Names	47
A.4.5.	Number of Incoming SFMP Packets with Bad Use of a Community Name.....	48
A.4.6.	Number of Incoming SFMP Packets with Parsing Errors	48
A.4.7.	Reserved	48
A.4.8.	Number of Incoming SFMP Packets indicating a Too Big Error	48
A.4.9.	Number of Incoming SFMP Packets indicating a No Such Name Error	48
A.4.10.	Number of Incoming SFMP Packets indicating a Bad Value Error	49
A.4.11.	Number of Incoming SFMP Packets indicating a Read-Only Error	49
A.4.12.	Number of Incoming SFMP Packets indicating a General Error.....	49
A.4.13.	Reserved	49
A.4.14.	Reserved	49
A.4.15.	Number of Incoming SFMP Get Requests.....	49
A.4.16.	Reserved	49
A.4.17.	Number of Incoming SFMP Set Requests	50
A.4.18.	Number of Incoming SFMP Get Responses	50
A.4.19.	Reserved	50
A.4.20.	Number of Outgoing SFMP Packets indicating a Too Big Error	50
A.4.21.	Number of Outgoing SFMP Packets indicating a No Such Name Error	50
A.4.22.	Number of Outgoing SFMP Packets indicating a Bad Value Error	51
A.4.23.	Number of Outgoing SFMP Packets indicating a Read-Only Error	51
A.4.24.	Number of Outgoing SFMP Packets indicating a General Error.....	51
A.4.25.	Number of Outgoing SFMP Get Requests.....	51
A.4.26.	Reserved	51
A.4.27.	Number of Outgoing SFMP Set Requests	52
A.4.28.	Number of Outgoing SFMP Get Responses	52
A.4.29.	Reserved	52
A.4.30.	Reserved	52
A.4.31.	Number of Incoming SFMP Set Requests – No Replies	52
A.4.32.	Number of Incoming SFMP Set Responses	52
A.4.33.	Number of Incoming SFMP Error Responses.....	53
A.4.34.	Number of Outgoing SFMP Set Requests – No Replies	53
A.4.35.	Number of Outgoing SFMP Set Responses	53
A.4.36.	Number of Outgoing SFMP Error Responses.....	53
A.5.	Objects for STMP	53
A.5.1.	Dynamic Object Definition Table	53

A.5.2.	Dynamic Object Data.....	55
A.5.3.	Dynamic Object Configuration.....	55
A.5.4.	STMP Statistics	57
A.5.5.	STMP Configuration	63
A.6.	Logical Names.....	64
A.6.1.	Maximum Logical Name Translations	64
A.6.2.	Logical Name Translation Table.....	65
A.7.	Report Parameter Node	66
A.7.1.	Maximum Event Classes Parameter	66
A.7.2.	Event Class Table	67
A.7.3.	Maximum Event Log Configurations Parameter	69
A.7.4.	Event Log Configuration Table.....	69
A.7.5.	Maximum Event Log Size Parameter	74
A.7.6.	Event Log Table	74
A.7.7.	Total Event Log Counter Parameter.....	76
A.8.	Security Objects	77
A.8.1.	Community Name Administrator Parameter	77
A.8.2.	Maximum Community Names Parameter	77
A.8.3.	Community Names Table	77
ANNEX B DEPRECATED OBJECTS (NORMATIVE).....		79
B.1.	Deprecated Type Definitions	79
B.2.	Deprecated Object Types.....	79
B.2.1.	Dynamic Object Definition	79
B.2.2.	Dynamic Object Data.....	80
ANNEX C AN EXPLANATION OF RELATIVE OBJECT IDENTIFIERS (INFORMATIVE)		84
ANNEX D ENTRY STATUS TYPE (INFORMATIVE)		85

< This page is intentionally left blank. >

SECTION 1 GENERAL

1.1 SCOPE

The Transportation Management Protocol (TMP) specifies an NTCIP Application Layer service. Specifically, it defines a set of rules and procedures for exchanging transportation management information between transportation management applications and transportation equipment such that they interoperate with each other. The transportation information that is exchanged with this protocol is defined elsewhere according to the rules defined in NTCIP 8004 v01. Messages conforming to this protocol may be exchanged using any appropriate transport mechanism.

TMP was carefully designed to provide 100% interoperability with the Internet-standard Simple Network Management Protocol (SNMP), but extends this protocol structure to meet the needs of the transportation environment. Analysis of the transportation environment has revealed the need for protocol simplicity, flexibility, and minimal data packet size; however, in many cases these three requirements are at odds.

After a careful review of existing protocols, it was decided to pursue the development of TMP, which combines the capabilities of three component protocols, one of which is the SNMP of the Internet community. Each component protocol has been designed to maximize two of the three requirements at the expense of the third requirement.

1.2 REFERENCES

Normative references contain provisions that, through reference in this text, constitute provisions of NTCIP 1103 v01. Other references in NTCIP 1103 v01 might provide a complete understanding of the entire protocol and the relations between all parts of the protocol. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent editions of the standard listed.

1.2.1 Normative References

IAB Std. 15—RFC 1157	Simple Network Management Protocol
ISO/IEC 8824-1:2002	Information Technology—Abstract Syntax Notation One (ASN.1): Specification of Basic Notation
AASHTO / ITE / NEMA NTCIP 1102:2004	<i>Octet Encoding Rules (OER) Base Protocol</i> published October 2005
AASHTO / ITE / NEMA NTCIP 1201:2005	<i>Global Object (GO) Definitions—version 02</i> published October 2005

1.2.2 Other References

AASHTO / ITE / NEMA NTCIP 1101:1996	<i>Simple Transportation Management Framework</i> published December 2001 NOTE—Also known as NEMA TS 3.2-1996, <i>National Transportation Communications and ITS Protocol – Simple Transportation Management Framework</i> .
--	--

NOTE—NEMA TS 3.2-1996/NTCIP 1101:1997 is an historical reference. It has since been divided into three component standards, each of which contains significant enhancements from the original text. NTCIP 1103 v01 is one of the three component standards; the other two are NTCIP 1102:2004 and NTCIP 8004 v01.

AASHTO / ITE / NEMA NTCIP 2301 v02	<i>Simple Transportation Management Framework Application Profile – Version 02</i> published March 2008
---------------------------------------	--

AASHTO / ITE / NEMA NTCIP 8004 v01	<i>Structure and Identification of Management Information (SMI)</i> published May 2008
---------------------------------------	---

1.2.3 Contact Information

1.2.3.1 NTCIP Standards

For revision information on this NTCIP standard, contact:

NTCIP Coordinator
National Electrical Manufacturers Association
1300 North 17th Street, Suite 1752
Rosslyn, VA 22209-3806

e-mail: ntcip@nema.org

For draft revisions to this NTCIP standard, and recommended revisions of the NTCIP Joint Committee, visit www.ntcip.org.

1.2.3.2 IAB Documents

For Internet Advisory Board (IAB) documents, contact:

Internet Architecture Board (IAB)
www.rfc-editor.org
www.rfc-editor.org/repositories.html

1.2.3.3 ISO/IEC Standards

Members of the ISO maintain registers of currently valid ISO/IEC International Standards. For the USA, the member of ISO is the American National Standards Institute (ANSI), which may be contacted as follows:

ANSI
11 West 42nd Street, 13th Floor
New York, NY 10036
(212) 642-4900
<http://global.ihs.com>

1.2.3.4 NEMA Standards

NEMA standards may be obtained by contacting NEMA at the following address:

National Electrical Manufacturers Association
1300 North 17th Street, Suite 1847
Rosslyn, VA 22209-3801
www.nema.org

1.3 TERMS

For a better understanding of this standard, here are some terms and definitions.

Block Object	An SNMP object with syntax of OerString. An OerString is defined to be an ASN.1 OCTET STRING that contains a Data Frame or Data Element that has already been serialized according to the rules of OER. This allows a structure of data elements to be treated as a single Object.
Data Element	A syntactically formal representation of some single unit of information of interest (such as a fact, proposition, or observation), with a singular instance value at any point in time, about some entity of interest (e.g., a person, place, process, property, object, concept, association, state, event). A data element is considered indivisible in a certain context. NOTE—For example, the data concept known as “globalTime.0” is a data element because it fully defines a single piece of information in a syntactically precise way. See IEEE 1489-1999.
Data Frame	A grouping of data elements primarily for the purpose of referring to the group with a single name, and thereby efficiently reusing groups of data elements that commonly appear together (as an ASN.1 Sequence, Sequence Of, Set, Set Of, or Choice) in a message body specification. This data concept type may, however, be used to specify groups of data elements for other purposes as well. NOTE—See IEEE 1488-2000.
Data Packet	A serialized Message. In other words, a data packet is the byte stream representing a logical structure which is termed a message.
Dynamic Object	A simple grouping of data elements (i.e., the equivalent of an ASN.1 Sequence) defined at run-time for the purpose of referring to the group with a single short identifier and encoded according to the rules of OER, and thereby providing a mechanism to efficiently exchange the group of data within a protocol. This is conceptually similar to a Data Frame, except that the contents of Dynamic Objects are defined at run-time. Dynamic Objects are only accessible using the STMP protocol, and therefore these are not true objects in the SNMP sense.

Fixed Message	A fixed message is any message (i.e., data packet) as used in the Simple Fixed Message Protocol (SFMP). The term fixed emphasizes that the message only contains one Object, versus SNMP with any number of objects, and that the structure of the Object is not dynamic, as is the case in a Dynamic Object.
Interoperate	The ability of two or more systems or components to exchange information and use the information that has been exchanged. NOTE—See IEEE 610.12-1990.
Message	A grouping of data elements and/or data frames, as well as associated message metadata, that is used to convey a complete unit of information. NOTE—See IEEE 1488-2000. In NTCIP 1103 v01, a message consists of one or more Objects (which may be Data Elements or serialized Data Frames) along with associated information (e.g., whether the message is a get or set, the request id, etc.)
Meta	A word denoting a description that is one level of abstraction above the entity being described.
MIB View	A set of objects within a MIB. Different MIB views may be defined for each community name. A set of objects do not need to be confined to a single node on the ISO global naming tree.
Object	A specific instance of an object type that may be managed by SNMP. Thus, an object may be either a data element or a data frame
Object Type	A classification of one or more objects that share a common definition and representational form. Synonymous with the object-oriented term “class.” Managed devices will frequently contain tables of information; if each row is considered a record, then each column would be an object type and each cell would be an object (i.e., an instance of the object type as defined by the column heading). For objects that are not contained in tables, there is a one-to-one relationship between object type and object; the only difference is that the object type is the abstract concept whereas the object is the precise instance. The definition of each object type includes a name and a syntax; the encoding of the object is defined by applying the protocol-specific rules to the logical syntax as defined in ASN.1. NOTE—SNMP rules require that the contents of an object be a simple ASN.1 Type; however, the NTCIP community has defined the concept of a Block Object, which is a serialized Data Frame, to circumvent this limitation of SNMP and to improve encoding efficiencies. Thus, within NTCIP, the term object may either refer to a data element or a data frame.
Referenced Object	An object instance that is supported by a device and that may be referenced from a dynamic object via the use of the dynObjVariables field.
Response Time	The time between the receipt of the last byte of the request and the start of the transmission of the first byte of the response.
Variable Binding	A structure used within SNMP to couple the identifier for a piece of data with its value.
Variable Binding List	A sequence of variable bindings.

1.4 ACRONYMS

AID	Application Identifier
SFMP	Simple Fixed Message Protocol
SMI	Structure of Management Information
SNMP	Simple Network Management Protocol
STMP	Simple Transportation Management Protocol
T2	Transportation Transport Profile (see NTCIP 2201)
TMIB	Transportation Management Information Base
TMP	Transportation Management Protocol

1.5 LAYOUT

NTCIP 1103 v01 is divided into the following sections:

Section 2 describes the overall structure of the Transportation Management Protocols and how the three component protocols merge together to form a single identifiable protocol.

Section 3 provides an overview of how the Internet standard Simple Network Management Protocol (SNMP) works.

Section 4 describes how Simple Fixed Message Protocol (SFMP) enhances the concepts of SNMP provide for more compact encoding of data while still providing a truly simple design with some loss of flexibility.

Section 5 describes how Simple Transportation Management Protocol (STMP) provides for flexible, compact encoding of data, with some loss of simplicity.

Section 6 is reserved for future definition of rules for how TMP is expected to handle exception reporting (i.e., to allow for an agent to send unsolicited information to the management station within a semi-controlled environment).

Section 7 discusses how logical names are defined.

Section 8 describes security issues related to the TMP.

Section 9 defines conformance to NTCIP 1103 v01.

Annex A is normative and provides the definition of the Transportation Management Protocols Management Information Base, which replaces the Transportation Management Information Base (TMIB) previously defined in NEMA TS 3.2/NTCIP 1101:1996.

Annex B is normative and defines deprecated objects.

Annex C is informative and provides information about the new ASN.1 Syntax of Relative OID.

Annex D is informative describes the EntryStatus type that was defined in previous NTCIP standards, but has now been deprecated.

SECTION 2 TRANSPORTATION MANAGEMENT PROTOCOL

Transportation Management Protocol (TMP) is a composition of three distinct protocols all providing nearly identical services, but designed to meet different data exchange and processing requirements. The three component protocols are:

- a) Simple Network Management Protocol (SNMP), version 1, as defined in RFC 1157 and according to the rules defined in Section 3.
- b) Simple Fixed Message Protocol (SFMP), as defined in Section 4.
- c) Simple Transportation Management Protocol (STMP), as defined in Section 5.

TMP was carefully designed to provide 100% interoperability with the Internet-standard SNMP, but extends this protocol structure to provide for additional requirements of the transportation environment. While SNMP meets the flexibility and simplicity requirements of the transportation industry, it produces a very verbose encoding that does not meet the functional requirements of existing communication infrastructures. As a result, the NTCIP effort produced two additional protocols that yield much more compact encoding with some loss of either simplicity or flexibility. The relationships among these three protocols are described in Figure 1.

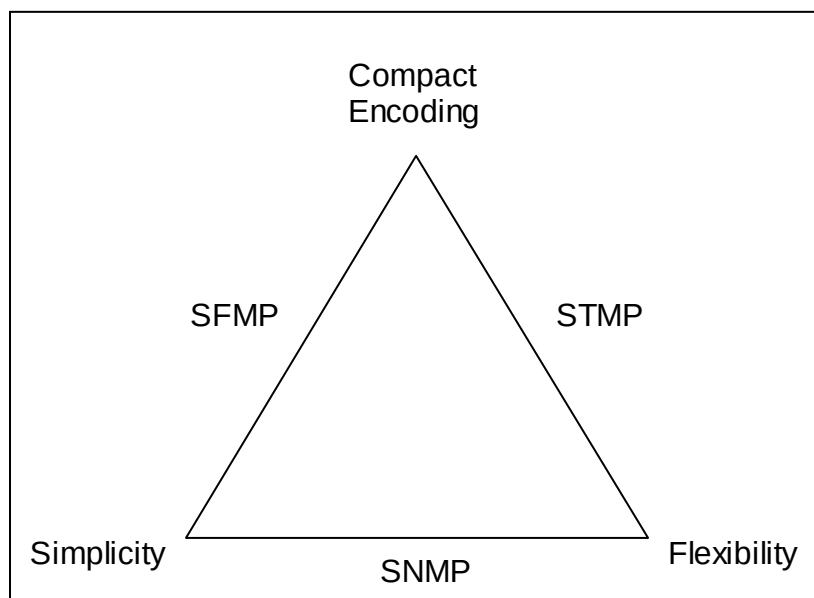


Figure 1 Requirements Relationship

The information exchanged by all three protocols shall be defined according to the rules set forth in NTCIP 8004, which is fully compatible with RFC 1155 as used by SNMP.

2.1 COMPOSITION OF TMP

TMP refers to the data construct that allows all three protocols to coexist while using the same protocol identifier. This was achieved due to the fact that all SNMP messages start with an initial byte of 0x30 (i.e., SNMP uses Basic Encoding Rules and all SNMP messages are defined as a SEQUENCE of data). Thus, the TMP construct has been designed to use the value of this first byte to identify which protocol is being referenced. The value of 0x30 identifies an SNMP message. Both SFMP and STMP messages use the high-order nibble (i.e., highest order four bits) of the first byte to determine the type of message (e.g., get request, set request, etc.) The low-order nibble is then used to identify whether the message is a fixed message (i.e., SFMP, indicated by the value of zero), or one of the 13 dynamic objects (i.e., STMP, indicated by a value of 1 through 13). The specific mapping of the first byte value is defined in Table 1.

Table 1 TMP Protocol Mapping

Protocol	Value
SNMP—All SNMP Messages (including traps)	0x30
SFMP—Get Request	0x80
SFMP—Set Request	0x90
SFMP—Set No Reply	0xA0
SFMP—Get Response	0xC0
SFMP—Set Response	0xD0
SFMP—Error Response	0xE0
SFMP—Trap (Reserved for future definition)	0xF0
STMP—Get Request (for 13 dynamic objects)	0x81 - 0x8D
STMP—Set Request (for 13 dynamic objects)	0x91 - 0x9D
STMP—Set No Reply (for 13 dynamic objects)	0xA1-0xAD
STMP—Get Next (for 13 dynamic objects)	0xB1-0xBD
STMP—Get Response (for 13 dynamic objects)	0xC1-0xCD
STMP—Set Response (for 13 dynamic objects)	0xD1-0xDD
STMP—Error Response (for 13 dynamic objects)	0xE1-0xED
Reserved for compatibility with TP-T2	0x31
Reserved for compatibility with TP-T2	0x41
Reserved	0x00 – 0x29
Reserved	0x32 – 0x40
Reserved	0x42-0x7F
Reserved	0x8E - 0x8F
Reserved	0x9E - 0x9F
Reserved	0xAE - 0xB0
Reserved	0xBE - 0xBF
Reserved	0xCE - 0xCF
Reserved	0xDE - 0xDF
Reserved	0xEE - 0xEF
Reserved	0xF1-0xFF

When decoding, TMP transmits the entire data stream, including the first byte, to the correct component protocol. When encoding, TMP simply transmits the entire data stream to the lower layer without changing the encoding from the component protocol.

2.2 SIMULTANEOUS PROCESSING

A management station takes into account the variable binding list processing nature of TMP. In TMP, all objects contained in a single set-request data packet appear to be set to their new values simultaneously. Therefore, a management station shall not combine a state change request with a request to set an

instance value associated with that state change. If such an operation is attempted, the operation may not be correctly processed. For example, using a single set command to change both the status and contents of a dynamic object can have unpredictable results.

2.3 PROTOCOL IDENTIFICATION LOGIC

The structure of the mappings defined in Table 1 is based on the following general principles as depicted in Figure 2:

- a) If the first byte is 0x30, the message is an SNMP message.
- b) If the high order bit of the first byte is set, i.e., the value of the first byte is greater than or equal to 0x80, the message is either STMP or SFMP as defined by the following;
 - 1) If the low order nibble (i.e., the four low order bits) of the first byte is equal to 0, the message is SFMP.
 - 2) If the low order nibble of the first byte is between 1 and 13 (0xD) inclusive and the high order nibble of the first byte is not 15 (0xF), the message is STMP.
 - 3) Otherwise the message is undefined.
- c) Otherwise the message is undefined.

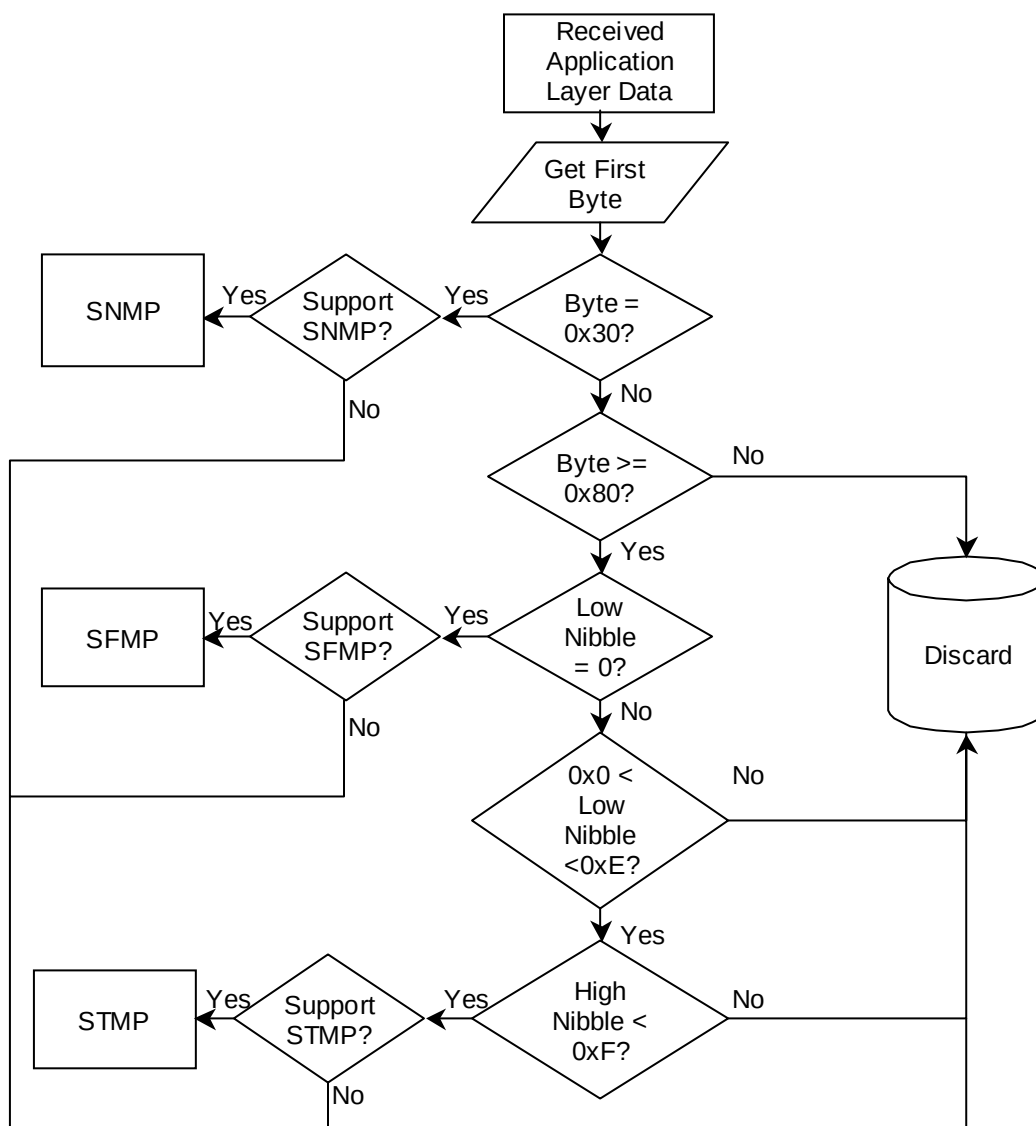


Figure 2 Process to Determine Component Protocol

NOTE—This value should not be confused with the AID of T2. Specifically, the values of 0x31 and 0x41 should never be received at the Application Layer of TMP; these values may be used by the T2 Transport Profile to identify an SNMP Trap or a header using port numbers, but this information is stripped by T2 prior to delivering the Application Layer Data to this process.

SECTION 3 SIMPLE NETWORK MANAGEMENT PROTOCOL

SNMP is a major standard developed by the Internet Engineering Task Force (IETF). The NTCIP effort selected this protocol for use in the ITS industry due to its wide use within the Internet community, the flexibility it provides management stations to define their own message content, and the simplicity of the protocol. While there were concerns about the encoding overhead that SNMP imposed on data communications, it was decided that the protocol provided a core set of functionality and that companion protocols could be developed to circumvent the overhead issues. While SNMP is only one of the three protocols employed by TMP, all three protocols follow the same basic get-set paradigm employed by SNMP. This paradigm is described briefly below and numerous textbooks describing SNMP in greater detail.

3.1 OVERVIEW

SNMP uses a get-set paradigm to exchange individual pieces of data. Each piece of data stored within a device and that is accessible via the SNMP protocol is called an *object*. Each object consists of two parts: the *object type* and the *instance*. Some object types may only occur once within a device, these are called *scalar* objects and are assigned the instance of zero (0). Other objects, i.e., those that exist in a table, may have multiple instances; these objects are called *columnar* objects and their instance is determined based on which index (i.e., row of the table) they are associated with. The first row of a table typically has an index of 1.

Each object type stored within a device is defined in a computer readable file called a Management Information Base (MIB). The MIB associates each object type with a precise syntax, a definition, and an Object Identifier, which is generally about 15 bytes long. An object instance is identified by appending the instance number to this base Object Identifier. Thus, each piece of data within the device has a unique number associated with it.

An SNMP management station exchanges data by sending each subject object identifier along with the get or set request. A single SNMP message may, and typically does, include the request for multiple objects simultaneously. Thus, any one SNMP data packet is likely to contain several of these large object identifiers. Likewise, the response also returns the object identifiers along with the data, even for responses to set operations.

SNMP also allows an agent to transmit unsolicited information called a *trap*. Data transmitted along with the trap notification also includes Object Identifiers. Section 6 has been reserved to define future additional rules for traps within NTCIP environments.

This is a reasonable approach to exchanging data when the data exchanges are infrequent and may change content from one request to another, which is typical of the Internet. However, within the ITS environment, the majority of the communications volume between a management application and an agent consists of a small number of messages that are repeatedly exchanged. In many cases, these exchanges occur frequently over dedicated channels. Thus, a significant reduction in the size of these frequently repeated messages could significantly reduce the size of the communications channel required for a link.

3.2 DEFINITION

SNMP shall be in accordance with SNMP version 1 as defined in RFC 1157, the requirements defined in Section 3.2, and the requirements defined in Section 6 and Section 7.

3.2.1 General Rules

A management station must take into account the variable binding list processing nature of TMP. In TMP, all objects contained in a single set-request data packet appear to be set to their new values simultaneously. Therefore, a management station shall not combine a state change request with a request to set an instance value associated with that state change. If such an operation is attempted, the operation may not be correctly processed.

3.2.2 Set Operations on a Read-Only Variable

Upon an agent receiving a SetRequest-PDU for a read-only object, the first condition listed under Clause 4.1.5 of RFC 1157 shall apply.

NOTE—The definition of ErrorStatus in RFC 1157 Clause 4.1.1 and the wording of RFC 1157 Clause 4.1.5 have caused some confusion as to how a device should handle a SetRequest-PDU for a read-only object. For the purposes of conformance with NTCIP 1103 v01, the most literal meaning of the RFC 1157 text applies. Thus, upon receipt of a SetRequest-PDU for a read-only object, an agent will respond with a noSuchName error. This interpretation is consistent with that of the broader Internet community as indicated in the clarifications provided by RFC 1213, which deprecates the snmpOutReadOnlys object and indicates that transmitting readOnly error is in fact a protocol error (within the definition of snmpInReadOnlys). The other TMP protocols have been designed to return the read-only error value.

A management station shall accept all of the following error codes as valid responses to an attempt to set a read-only object: noSuchName, readOnly, and genErr.

NOTE—This requirement ensures that a management station will be able to work properly with any NTCIP version 1 device that may have interpreted RFC 1157 differently.

3.2.3 Extra Data Prohibition

An agent receiving a get-request or get-next-request containing anything other than a NULL in a variable binding value field shall silently drop the data packet.

NOTE—RFC 1157 Clause 4.1.1 requires that any value in this field be ignored. The above NTCIP prohibition further clarifies the intent of this clause by completely prohibiting the insertion of any data; thereby minimizing the size of the data packet.

3.2.4 Response Time

The SNMP agent shall process all requests in accordance with Section 3, including processing the request sufficiently to generate the transmission of the appropriate response (assuming that the SNMP agent has permission to transmit) within the maximum Response Time. If the specification does not indicate the maximum Response Time, the maximum Response Time shall be 100 milliseconds plus one millisecond for each byte in the response variable-bindings field.

SECTION 4 SIMPLE FIXED MESSAGE PROTOCOL

4.1 OVERVIEW

SFMP can be viewed as a simplified, more compact version of SNMP. A careful analysis of SNMP reveals that the size and complexity of the data packets can be reduced by:

- a) Identifying the data contents within a data packet by using a single identifier that references a group of data elements rather than using a separate identifier in association with each data element in the data packet,
- b) Defining a data packet structure that only includes the information that is required for a given message type (e.g., a set response does not need to echo the values), and
- c) Using a set of encoding rules that are more efficient than the Basic Encoding Rules as used by SNMP.

4.1.1 Data Identification

SFMP decreases the size of the overhead consumed by data identification in two ways. First, it is designed with the assumption that it will exchange a single composite object, i.e., an object that consists of a defined sequence of other objects. This approach decreases the overhead by allowing the use of a single object identifier rather than a separate identifier for each component object. Second, the design of SFMP incorporates the concept that all of the composite objects are located under the NEMA node of the ISO tree. As such, it includes an encoding mechanism to shorten the object identifiers for objects under this node. Further, the complexity of the protocol is reduced due to the fact that an agent is not required to handle a get or set command with any combination of data in any order, it only is required to support one object at a time and if the object is a block object, the ordering of data is always fixed. This allows less powerful and less complex devices to support NTCIP.

4.1.2 Packet Structure

In SNMP, all data packets use a very similar data structure. While this provides for some advantages in code reusability, it also results in extra information being sent in many SNMP data packets. Due to the need to minimize the overhead for the most frequently exchanged messages and to minimize the processing requirements to decode these extra bytes; the NTCIP effort developed the SFMP Data Packet Structure to more efficiently exchange the fixed messages while still providing necessary security functions. This packet structure is defined in detail in Section 4.2.

4.1.3 Encoding

SNMP encodes all of its information according to the ASN.1 Basic Encoding Rules (BER). BER uses a three *tuple* to encode data for transmission. The first element of the tuple, the type, specifies what type of data follows. The second element of the tuple, the length, specifies how many octets the data occupies. The third and final element of the tuple, the value, is the actual data being transmitted. This encoding is sometimes referred to as *TLV* encoding, which stands for 'type', 'length', 'value'. This provides a very flexible method of encoding information for transmission; however, if both sides have already agreed on a specific data structure, it includes unnecessary overhead by including the 'type' field and the 'length' field for fixed length data.

As a result, the NTCIP effort has defined a separate set of ASN.1 encoding rules, known as Octet Encoding Rules (OER), as defined in NTCIP 1102:2004. OER eliminates the type field completely, and it

eliminates the length field under those conditions where the length is known. Given that most of the object definitions (data elements) defined by NTCIP consist of INTEGERS in the range of 0 to 255, OER is able to significantly reduce the size of many NTCIP data packets.

4.2 DEFINITION

All SFMP implementations shall be declared to be a management station, agent, or both.

Management stations may transmit SFMP-GetRequest-PDU's, SFMP-SetRequest-PDU's, and SFMP-SetRequest-NoReply-PDU's as necessary in order to manage agents. Management stations shall be able to receive and process SFMP-GetResponse-PDU's, SFMP-SetResponse-PDU's, SFMP-ErrorResponse-PDU's per the rules defined within this clause.

Agents shall be able to receive and process SFMP-GetRequest-PDU's, SFMP-SetRequest-PDU's, and SFMP-SetRequest-NoReply-PDU's per the rules defined within Section 4.2.1, including the ability to transmit SFMP-GetResponse-PDU's, SFMP-SetResponse-PDU's, and SFMP-ErrorResponse-PDU's as required.

The transmission of SFMP data packets shall be governed by the rules and procedures defined in Section 4.2.1 and Section 4.2.2, and shall conform to the structures defined in Section 4.2.3.

4.2.1 Rules

Similar to SNMP, SFMP models all device functions as alterations (i.e., SETs) or inspections (i.e., GETs) of variables (i.e., objects). This strategy avoids the complexities of defining a different message type as a part of the protocol definition for each command desired.

Communication among protocol entities is accomplished by the exchange of protocol messages, each of which may be entirely and independently represented within a single datagram. An implementation of this protocol shall accept valid messages whose length does not exceed 484 octets. However, it is recommended that implementations support larger messages whenever feasible.

A management station necessarily takes into account the variable binding list processing nature of TMP. In TMP, all objects contained in a single set-request data packet appear to be set to their new values simultaneously. Therefore, a management station shall not combine a state change request with a request to set an instance value associated with that state change. If such an operation is attempted, the operation may not be correctly processed.

4.2.2 Elements of Procedure

Section 4.2.2 describes the actions of a protocol entity implementing the SFMP.

NOTE—It is not intended to constrain the internal architecture of any conformant implementation.

An SFMP protocol entity transmits an SFMP message by passing the serialized message to a transport service capable of sending the message to the peer protocol entity.

An SFMP protocol entity receives an SFMP message as follows:

- a) It performs a rudimentary parse of the incoming datagram to build an ASN.1 object corresponding to the referenced Objects. If the parse fails, it discards the datagram and performs no further actions.
- b) It then verifies the version number of the SFMP message. If there is a mismatch, it discards the datagram and performs no further actions.
- c) It then authenticates the community name. If the authentication fails, the protocol records an authentication failure event by incrementing the *sfmp-inBadCommunityNames* object (see Annex A.4)

and the *agentHealth-authenticationFailures* object, discards the datagram, and performs no further actions.

- d) It then processes the message according to the rules defined in Section 4.2.2.2 and Section 4.2.2.3 using the identified community.

4.2.2.1 Initiating a Request

An SFMP management station may, at any time, initiate a get or set request by generating an SFMP-Data-Packet containing an SFMP-GetRequest-PDU, an SFMP-SetRequest-PDU, or an SFMP-SetRequest-NoReply-PDU. The conditions that may result in the management station initiating such a request are the subject of the end-application functionality and are beyond the scope of this standard.

An SFMP agent shall not issue an SFMP-Data-Packet containing an SFMP-GetRequest-PDU, an SFMP-SetRequest-PDU, or an SFMP-SetRequest-NoReply-PDU. This restriction shall not preclude a single device from acting as both a management station and an agent.

4.2.2.2 Processing a Request

An SFMP management station shall silently drop (discard and perform no further action on the request) any SFMP-GetRequest-PDU, SFMP-SetRequest-PDU, or SFMP-SetRequest-NoReply-PDU.

The SFMP agent shall process all requests in accordance with Sections 4.2.2.2.1 through 4.2.2.2.3, including processing the request sufficiently to generate the transmission of the appropriate response (assuming that the SFMP agent has permission to transmit) within the maximum Response Time. If the specification does not indicate the maximum Response Time, the maximum Response Time shall be 100 milliseconds plus one millisecond for each byte in the response SFMP-PDU data field.

4.2.2.2.1 Processing an SFMP Get Request

Upon receipt of an SFMP-GetRequest-PDU, an SFMP agent shall respond according to the following rules, in order:

- a) If the SFMP-GetRequest-PDU contains an information field, the agent shall silently drop the data packet.
- b) If the value of the message-oid field does not exactly correspond to an object available for get operations in the relevant MIB view, the agent shall transmit to the originator of the request an SFMP-Data-Packet containing an SFMP-ErrorResponse-PDU. The error-status field shall indicate noSuchName, and the error-index field shall indicate zero (0).
- c) If the value of the message-oid field references an object that is of a non-accessible aggregate type (as defined by RFC 1155, a normative reference of NTCIP 8004 v01), the agent shall transmit to the originator of the request an SFMP-Data-Packet containing an SFMP-ErrorResponse-PDU. The error-status field shall indicate noSuchName and the error-index field shall indicate zero (0).
- d) If the size of the SFMP-GetResponse-PDU would exceed a local limitation, the agent shall transmit to the originator of the request an SFMP-Data-Packet containing an SFMP-ErrorResponse-PDU. The error-status fields shall indicate tooBig and the error-index field shall indicate zero (0).
- e) If the value of the object referenced by the message-oid field cannot be retrieved for reasons not covered by any of the foregoing rules, the agent shall transmit to the originator of the request an SFMP-Data-Packet containing an SFMP-ErrorResponse-PDU. The error-status field shall indicate genErr and the error-index field shall either indicate zero (0) or shall indicate the element within the structure that is preventing the operation.
- f) If none of the foregoing rules apply, the agent shall transmit to the originator of the request an SFMP-Data-Packet containing an SFMP-GetResponse-PDU such that the request number field shall be that used in the SFMP-GetRequest-PDU and the data field shall contain the requested information.

4.2.2.2.2 Processing an SFMP Set Request

Upon receipt of an SFMP-SetRequest-PDU, an SFMP agent shall respond according to the following rules, in order:

- a) If the SFMP-SetRequest-PDU does not contain a data field, the agent shall silently drop the data packet.
- b) If the object referenced by the value of the message-oid field is only available for get operations within the relevant MIB view, the agent shall transmit to the originator of the request an SFMP-Data-Packet containing an SFMP-ErrorResponse-PDU. The error-status field shall indicate readOnly and the error-index field shall indicate zero (0).
- c) If the value of the message-oid field does not exactly correspond to an object available for set operations in the relevant MIB view, the agent shall transmit to the originator of the request an SFMP-Data-Packet containing an SFMP-ErrorResponse-PDU. The error-status field shall indicate noSuchName and the error-index field shall indicate zero (0).
- d) If the contents of the data field cannot be parsed to fit the SYNTAX defined for the referenced object, the agent shall transmit to the originator of the request an SFMP-Data-Packet containing an SFMP-ErrorResponse-PDU. The error-status field shall indicate badValue and the error-index field shall indicate the field number at which the parsing first failed.
- e) If the value of the object referenced by the message-oid field cannot be altered for reasons not covered by any of the foregoing rules, the agent shall transmit to the originator of the request an SFMP-Data-Packet containing an SFMP-ErrorResponse-PDU. The error-status field shall indicate genErr and the error-index field shall either indicate zero (0) or shall indicate the element within the structure that is preventing the operation.
- f) If none of the foregoing rules apply, the agent shall assign the requested value to the subject object. If the object is a block object, each sub-variable assignment specified by the request shall be effected as if simultaneously set with respect to all other assignments specified in the same message. The agent shall then transmit to the originator of the request an SFMP-Data-Packet containing an SFMP-SetResponse-PDU such that the request number field shall be that used in the SFMP-SetRequest-PDU.

4.2.2.2.3 Processing an SFMP Set Request-No Reply

Upon receipt of an SFMP-SetRequest-NoReply-PDU, an SFMP agent shall respond according to the following rules, in order:

- a) If the SFMP-SetRequest-NoReply-PDU does not contain a data field, the agent shall silently discard the data packet.
- b) If the object referenced by the value of the message-oid field is only available for get operations within the relevant MIB view, the agent shall silently discard the datagram and perform no further actions.
- c) If the value of the message-oid field does not exactly correspond to an object available for get or set operations in the relevant MIB view, the agent shall silently discard the datagram and perform no further actions.
- d) If the contents of the data field cannot be parsed to fit the SYNTAX defined for the referenced object, the agent shall silently discard the datagram and perform no further actions.
- e) If the value of the object referenced by the message-oid field cannot be altered for reasons not covered by any of the foregoing rules, the agent shall silently discard the datagram and perform no further actions.
- f) If none of the foregoing rules apply, the agent shall assign the requested value to the subject object.