



IEEE Standard for a Smart Transducer Interface for Sensors, Actuators, and Devices—Message Queue Telemetry Transport (MQTT) for Networked Device Communication

IEEE Industrial Electronics Society

Developed by the
Industrial Electronics Society Standards Committee

IEEE Std 1451.1.6™-2025

STANDARDS

IEEE Standard for a Smart Transducer Interface for Sensors, Actuators, and Devices—Message Queue Telemetry Transport (MQTT) for Networked Device Communication

Developed by the

Industrial Electronics Society Standards Committee
of the
IEEE Industrial Electronics Society

Approved 19 June 2025

IEEE SA Standards Board

Abstract: A method for transporting IEEE 1451™ messages over a network using Message Queue Telemetry Transport (MQTT) to establish a light-weight, simplified protocol structure to handle IEEE 1451 communications is defined by this standard. Also, the establishment of a light-weight, simplified protocol structure to handle IEEE 1451 communications over MQTT is enabled by this standard. Adding to the MQTT standards' recommended methods, three different formats of IEEE 1451.0™ message carriers are defined. Moreover, time-synchronization, security, and other useful extensions are also defined by this standard.

Keywords: IEEE 1451.1.6™, Internet of Things, IoT, LAN/WAN, MQTT, security, sensor network, smart transducer, time synchronization

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

Copyright © 2026 by The Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 6 February 2026. Printed in the United States of America.

IEEE is a registered trademark in the U.S. Patent & Trademark Office, owned by The Institute of Electrical and Electronics Engineers, Incorporated.

PDF: ISBN 979-8-8557-2797-5 STD28402
Print: ISBN 979-8-8557-2798-2 STDPD28402

IEEE prohibits discrimination, harassment, and bullying.

For more information, visit <https://www.ieee.org/about/corporate/governance/p9-26.html>.

No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.

Important Notices and Disclaimers Concerning IEEE Standards Documents

IEEE Standards documents are made available for use subject to important notices and legal disclaimers. These notices and disclaimers, or a reference to this page (<https://standards.ieee.org/ipr/disclaimers.html>), appear in all IEEE standards and may be found under the heading “Important Notices and Disclaimers Concerning IEEE Standards Documents.”

Notice and Disclaimer of Liability Concerning the Use of IEEE Standards Documents

IEEE Standards documents are developed within IEEE Societies and subcommittees of IEEE Standards Association (IEEE SA) Board of Governors. IEEE develops its standards through an accredited consensus development process, which brings together volunteers representing varied viewpoints and interests to achieve the final product. IEEE standards are documents developed by volunteers with scientific, academic, and industry-based expertise in technical working groups. Volunteers involved in technical working groups are not necessarily members of IEEE or IEEE SA and participate without compensation from IEEE. While IEEE administers the process and establishes rules to promote fairness in the consensus development process, IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

IEEE makes no warranties or representations concerning its standards, and expressly disclaims all warranties, express or implied, concerning all standards, including but not limited to the warranties of merchantability, fitness for a particular purpose and non-infringement. IEEE Standards documents do not guarantee safety, security, health, or environmental protection, or compliance with law, or guarantee against interference with or from other devices or networks. In addition, IEEE does not warrant or represent that the use of the material contained in its standards is free from patent infringement. IEEE Standards documents are supplied “AS IS” and “WITH ALL FAULTS.”

Use of an IEEE standard is wholly voluntary. The existence of an IEEE standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard.

In publishing and making its standards available, IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity, nor is IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing any IEEE Standards document should rely upon their own independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

IN NO EVENT SHALL IEEE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO: THE NEED TO PROCURE SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE PUBLICATION, USE OF, OR RELIANCE UPON ANY STANDARD, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND REGARDLESS OF WHETHER SUCH DAMAGE WAS FORESEEABLE.

Translations

The IEEE consensus balloting process involves the review of documents in English only. In the event that an IEEE standard is translated, only the English language version published by IEEE is the approved IEEE standard.

Use by artificial intelligence systems

In no event shall material in any IEEE Standards documents be used for the purpose of creating, training, enhancing, developing, maintaining, or contributing to any artificial intelligence systems without the express, written consent of IEEE SA in advance. “Artificial intelligence” refers to any software, application, or other system that uses artificial intelligence, machine learning, or similar technologies, to analyze, train, process, or generate content. Requests for consent can be submitted using the Contact Us form.

Official statements

A statement, written or oral, that is not processed in accordance with the IEEE SA Standards Board Operations Manual is not, and shall not be considered or inferred to be, the official position of IEEE or any of its committees and shall not be considered to be, or be relied upon as, a formal position of IEEE or IEEE SA. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that the presenter’s views should be considered the personal views of that individual rather than the formal position of IEEE, IEEE SA, the Standards Committee, or the Working Group. Statements made by volunteers may not represent the formal position of their employer(s) or affiliation(s). News releases about IEEE standards issued by entities other than IEEE SA should be considered the view of the entity issuing the release rather than the formal position of IEEE or IEEE SA.

Comments on standards

Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE or IEEE SA. However, **IEEE does not provide interpretations, consulting information, or advice pertaining to IEEE Standards documents.**

Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Since IEEE standards represent a consensus of concerned interests, it is important that any responses to comments and questions also receive the concurrence of a balance of interests. For this reason, IEEE and the members of its Societies and subcommittees of the IEEE SA Board of Governors are not able to provide an instant response to comments or questions, except in those cases where the matter has previously been addressed. For the same reason, IEEE does not respond to interpretation requests. Any person who would like to participate in evaluating comments or revisions to an IEEE standard is welcome to join the relevant IEEE SA working group. You can indicate interest in a working group using the Interests tab in the Manage Profile & Interests area of the [IEEE SA myProject system](#).¹ An IEEE Account is needed to access the application.

Comments on standards should be submitted using the [Contact Us](#) form.²

¹ Available at: <https://development.standards.ieee.org/myproject-web/public/view.html#landing>.

² Available at: <https://standards.ieee.org/about/contact/>.

Laws and regulations

Users of IEEE Standards documents should consult all applicable laws and regulations. Compliance with the provisions of any IEEE Standards document does not constitute compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

Data privacy

Users of IEEE Standards documents should evaluate the standards for considerations of data privacy and data ownership in the context of assessing and using the standards in compliance with applicable laws and regulations.

Copyrights

IEEE draft and approved standards are copyrighted by IEEE under U.S. and international copyright laws. They are made available by IEEE and are adopted for a wide variety of both public and private uses. These include both use by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making these documents available for use and adoption by public authorities and private users, neither IEEE nor its licensors waive any rights in copyright to the documents.

Photocopies

Subject to payment of the appropriate licensing fees, IEEE will grant users a limited, non-exclusive license to photocopy portions of any individual standard for company or organizational internal use or individual, non-commercial use only. To arrange for payment of licensing fees, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400; <https://www.copyright.com/>. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Updating of IEEE Standards documents

Users of IEEE Standards documents should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect.

Every IEEE standard is subjected to review at least every 10 years. When a document is more than 10 years old and has not undergone a revision process, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE standard.

In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit [IEEE Xplore](#) or [contact IEEE](#).³ For more information about the IEEE SA or IEEE's standards development process, visit the IEEE SA Website.

Errata

Errata, if any, for all IEEE standards can be accessed on the [IEEE SA Website](#).⁴ Search for standard number and year of approval to access the web page of the published standard. Errata links are located under the Additional Resources Details section. Errata are also available in [IEEE Xplore](#). Users are encouraged to periodically check for errata.

Patents

IEEE standards are developed in compliance with the [IEEE SA Patent Policy](#).⁵

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken by the IEEE with respect to the existence or validity of any patent rights in connection therewith. If a patent holder or patent applicant has filed a statement of assurance via an Accepted Letter of Assurance, then the statement is listed on the IEEE SA Website at <https://standards.ieee.org/about/sasb/patcom/patents.html>. Letters of Assurance may indicate whether the Submitter is willing or unwilling to grant licenses under patent rights without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination to applicants desiring to obtain such licenses.

Essential Patent Claims may exist for which a Letter of Assurance has not been received. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims, or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

IMPORTANT NOTICE

Technologies, application of technologies, and recommended procedures in various industries evolve over time. The IEEE standards development process allows participants to review developments in industries, technologies, and practices, and to determine what, if any, updates should be made to the IEEE standard. During this evolution, the technologies and recommendations in IEEE standards may be implemented in ways not foreseen during the standard's development. IEEE standards development activities consider research and information presented to the standards development group in developing any safety recommendations. Other information about safety practices, changes in technology or technology implementation, or impact by peripheral systems also may be pertinent to safety considerations during implementation of the standard. Implementers and users of IEEE Standards documents are responsible for determining and complying with all appropriate safety, security, environmental, health, data privacy, and interference protection practices and all applicable laws and regulations.

³ Available at: <https://ieeexplore.ieee.org/browse/standards/collection/ieee>.

⁴ Available at: <https://standards.ieee.org/standard/index.html>.

⁵ Available at: <https://standards.ieee.org/about/sasb/patcom/materials.html>.

Participants

At the time this IEEE standard was completed, the P1451.1.6 Working Group had the following membership:

Hiroaki Nishi, *Chair*
John Schmalzel, *Vice Chair*

Reza Abrishambaf
Anthony Amaro, Jr.
Richard Bartel
Hans Peter Bernhard
Cheng-Jen Chen
Bob Cherek,
Dominique Dallet
Hans Dermot Doran
Koksai Erenturk
António Espirito Santo
Antoni Grzanka
Victor Huang

Ran Jin
Audalis Karnoney
Ryogo Kubo
Vanessa Lalitte
Kang Lee
Paulo Leitao
Yucheng Lin
Kanae Matsui
William Miller
Tobias Mitterer
Yuichi Nakamura
B. Banu Rekha
Helbert Rocha

Nirmala Sanikommu
Sangho Shin
Russell Trafford
Kim Fung Tsang
Joshua Velez
Peter Weher
Janaka Wijekoon
Darold Wobschall
Yingmeng Xiang
Hengzhao Yang
Hubert Zangl
Jing Zhang

The following members of the individual Standards Association balloting group voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

Mihaela Albu
Riccardo Brama
Sharmeela Chenniappan
Sunday Ekpo
Werner Hoelzl
Victor Huang

Piotr Karocki
John Kay
Greg Luri
Rajesh Murthy
Vicente Alonso Navarro
Valencia
Hiroaki Nishi

Jhony Sembiring
Yuyin Song
Stephen Webb
Ruqiang Yan
Yu Yuan
Janusz Zalewski

When the IEEE SA Standards Board approved this standard on 19 June 2025, it had the following membership:

Lei Wang, *Chair*
Jon Walter Rosdahl, *Vice Chair*
David J. Law, *Past Chair*
Alpesh Shah, *Secretary*

Edward Au
Ted Burse
Xiaofeng (Alfred) Chen
Doug Edwards
Nehad El-Sherif
J. Travis Griffith
Deborah R. Hagar

Guido R. Hiertz
Ronald W. Hotchkiss
Tyler L. Jaynes
Thomas Koshy
Howard Li
Xiaohui Liu
Kevin W. Lu
Hiroshi Mano

Daleep C. Mohla
Annette D. Reilly
Robby Robson
Daniel Sabin
F. Keith Waters
Sha Wei
Luyang (Eric) Zhang

Introduction

This introduction is not part of 1451.1.6-2025, IEEE Standard for a Smart Transducer Interface for Sensors, Actuators, and Devices—Message Queue Telemetry Transport (MQTT) for Networked Device Communication.

The IEEE Std 1451.1.6™ presents a robust framework for implementing a lightweight, scalable communication protocol for smart transducers, sensors, actuators, and devices over a network using the Message Queue Telemetry Transport (MQTT) protocol. This standard is an extension of the IEEE 1451™ family of standards, which aims to facilitate the efficient communication and interoperability of smart transducers in a networked environment. Specifically, IEEE Std 1451.1.6 leverages MQTT, a well-established, open standard that is recognized for its simplicity, low overhead, and suitability for devices with limited computational resources, such as those found in the Internet of Things (IoT).

MQTT is a publish-subscribe protocol that operates over TCP/IP and provides reliable, ordered, and lossless communication between devices through a centralized broker. The design of this standard is focused on enhancing the usability of IEEE 1451 messages for transducer data access, configuration, and management by incorporating the lightweight and flexible nature of MQTT.

The primary objective of this standard is to establish a method for transporting IEEE 1451 messages using MQTT, enabling the seamless integration of smart transducers within IoT environments. This includes defining mechanisms for transporting transducer data, helping to ensure time synchronization, and incorporating security measures to protect data integrity and confidentiality in a networked environment. Three distinct message formats for IEEE 1451.0-2024 communications are introduced to optimize the transmission of data over MQTT, each designed to handle specific communication needs.

Key features of this standard include provisions for time synchronization, security enhancements, and extensions to handle various network configurations. The use of MQTT allows devices with low bandwidth, memory, and power to communicate efficiently, which is particularly relevant for IoT applications that demand real-time data processing and low-power operation.

In conclusion, IEEE Std 1451.1.6 expands the IEEE 1451 family of standards by integrating the MQTT protocol, thereby enabling widespread adoption of smart transducer systems in IoT applications. It helps ensure compatibility with existing infrastructure while offering the flexibility and scalability required for future networked sensor and actuator systems. Through its adoption, this standard will support enhanced communication efficiency, interoperability, and security in smart sensor networks across a wide range of industries.

Contents

1. Overview.....	11
1.1 Scope.....	11
1.2 Purpose.....	11
1.3 Word usage.....	11
2. Normative references.....	12
3. Definitions, acronyms, and abbreviations	12
3.1 Definitions	12
3.2 Acronyms and abbreviations.....	15
4. Data types.....	17
5. IEEE 1451 family reference model as IEEE Std 1451.1.6	18
5.1 IEEE 1451 family reference model	18
5.2 IEEE 1451 family reference model as IEEE Std 1451.1.6.....	20
5.3 Merit of the use of IEEE 1451.1.6 and MQTT for IEEE 1451.....	21
6. Network/communication configuration for NCAP	22
6.1 Basic communication model in IEEE Std 1451.0-2024.....	22
6.2 MQTT structure for NCAP.....	29
6.3 Communication configuration for supporting MQTT	30
6.4 MQTT-NCAP from the perspective of IEEE Std 1451.0-2024.....	31
7. IEEE 1451.0 network services in IEEE 1451.1.6	45
7.1 General	45
7.2 Discovery services	45
7.3 Transducer access services	49
7.4 TEDS access services.....	52
7.5 Event notification services.....	53
7.6 Subscribe NCAP heartbeat (04 10)	54
8. Network service operations	55
8.1 Message broadcast and protocol examples for discovery services	55
8.2 Simple NCAP operation using MQTT	56
9. Time synchronization.....	58
9.1 General	58
9.2 BR-Sync	60
9.3 RR-Sync	61
9.4 TimeSync TEDS	64
10. Security	65
10.1 Security of IEEE Std 1451.0-2024 and IEEE 1451.1.6 WAN interface	65
10.2 MQTT security.....	66
10.3 IEEE 1451.1.6 security.....	67
10.4 Security TEDS	71
10.5 Malformed data.....	73
Annex A (informative) Bibliography.....	74

Annex B (normative) Specification of MQTT	76
B.1 General.....	76
B.2 MQTT v3 and MQTT v5	76
Annex C (informative) Request-reply communication of MQTT v5.....	78
Annex D (informative) Request-reply communication of MQTT v3 (Synchronous communication).....	80
D.1 General	80
D.2 Improvement in security	80
Annex E (informative) Request-reply communication on MQTT v3 by emulating MQTT v5 property	81
E.1 General.....	81
E.2 Emulating user property of MQTT v5 on MQTT v3	82
Annex F (informative) Implementation of Security TEDS	83
Annex G (informative) NCAP architecture and limitations for supporting MQTT	86
G.1 MQTT communication module.....	86
G.2 Use of MQTT communication API.....	86
G.3 MQTT message size limitations in the use of 1451.....	86
G.4 Method to confirm the version of MQTT v3 and MQTT v5.....	86
Annex H (informative) Identifier.....	87
H.1 Location field.....	87
H.2 Manufacturer field.....	87
H.3 Year field	87
H.4 Time field	87

IEEE Standard for a Smart Transducer Interface for Sensors, Actuators, and Devices—Message Queue Telemetry Transport (MQTT) for Networked Device Communication

1. Overview

This standard introduces a method for transporting IEEE 1451™ messages over a network using Message Queue Telemetry Transport (MQTT) to establish a light-weight, simplified protocol structure to handle IEEE 1451 communications. The standard uses the common network services defined in IEEE Std 1451.0™-2024.⁶

1.1 Scope

This standard defines a method for transporting IEEE 1451 messages over a network using Message Queue Telemetry Transport (MQTT) to establish a light-weight, simplified protocol structure to handle IEEE 1451 communications.

1.2 Purpose

The purpose of this standard is to provide a low-overhead, light-weight method to facilitate IEEE 1451 communications in systems with low bandwidth, power, or memory allocations.

1.3 Word usage

The word *shall* indicates mandatory requirements strictly to be followed in order to conform to the standard and from which no deviation is permitted (*shall* equals *is required to*).^{7,8}

⁶ Information on references can be found in Clause 2.

⁷ The use of the word *must* is deprecated and cannot be used when stating mandatory requirements; *must* is used only to describe unavoidable situations.

⁸ The use of *will* is deprecated and cannot be used when stating mandatory requirements; *will* is only used in statements of fact.

The word *should* indicates that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required (*should* equals *is recommended that*).

The word *may* is used to indicate a course of action permissible within the limits of the standard (*may* equals *is permitted to*).

The word *can* is used for statements of possibility and capability, whether material, physical, or causal (*can* equals *is able to*).

2. Normative references

The following referenced documents are indispensable for the application of this document (i.e., they must be understood and used, so each referenced document is cited in text and its relationship to this document is explained). For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.

IEEE Std 1451.0™-2024, IEEE Standard for a Smart Transducer Interface for Sensors and Actuators—Common Functions, Communication Protocols, and Transducer Electronic Data Sheet (TEDS) Formats.^{9,10}

MQTT v3.1.1, MQTT OASIS standard.¹¹

MQTT v5, MQTT OASIS standard.¹²

3. Definitions, acronyms, and abbreviations

3.1 Definitions

For the purposes of this document, the following terms and definitions apply. The *IEEE Standards Dictionary Online* should be consulted for terms not defined in this clause.¹³

actuator: A transducer that accepts a data sample or samples and converts them into an action. The action may be completely contained within the transducer interface module (TIM) or may change something outside of the TIM.

address: A character or group of characters that identifies a register, a particular part of storage, or some other data source or destination.

application (APP): A software module (IEEE 1451 client) running on a device that performs network communications with the network-capable application processor (NCAP; IEEE 1451 server) to access transducer data and the Transducer Electronic Data Sheet (TEDS) and processes transducer data for various Internet of Things (IoT) applications.

⁹ The IEEE standards or products referred to in Clause 2 are trademarks owned by The Institute of Electrical and Electronics Engineers, Incorporated.

¹⁰ IEEE publications are available from The Institute of Electrical and Electronics Engineers (<https://standards.ieee.org/>).

¹¹ MQTT specification available at: <https://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html>.

¹² MQTT specification available at: <https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html>.

¹³ *IEEE Standards Dictionary Online* is available at: <http://dictionary.ieee.org>. An IEEE account is required for access to the dictionary, and one can be created at no charge on the dictionary sign-in page.

buffer: An intermediate data storage location used to compensate for the difference in rate of flow of data or time of occurrence of events when transmitting information from one device to another.

calibration: The process used to determine the information that resides in the Calibration Transducer Electronic Data Sheet (TEDS) to support correction.

correction: The evaluation of a multinomial function using information from the Calibration Transducer Electronic Data Sheet (TEDS) together with data from one or more transducers.

data sheet: A set of information on a device that defines the parameters of operation and conditions of usage (usually produced by the device's manufacturer).

digital interface: A communications media and a protocol for transferring information by binary means only.

electronic data sheet: A data sheet stored in some form of electrically readable memory (as opposed to a piece of paper).

embedded transducer: A device that behaves as a transducer from the point of view of the network-capable application processor (NCAP) even though nothing outside of the transducer interface module (TIM) is sensed or changed. Embedded transducers are useful for setting or reading the operating parameters of other transducers.

enumeration: The assignment of a numeric value to a specific meaning within the context of a specific data field. Binary numbers are usually expressed in decimal terms for human convenience. Not all possible numeric values need to have a specific meaning. Values without meaning are declared to be unused or reserved for future use. Enumeration is the process of declaring the encoding of human interpretable information in a manner convenient for digital electronic machine storage and interchange. Any subclause that defines a Transducer Electronic Data Sheet (TEDS) data field to be enumerated is to contain a table that defines the meaning of the data field for each numeric value possible. The meanings encoded in each data field are to be specific and unique to that data field and only to that data field. The value becomes meaningless if not associated with the data field and its defining table.

interoperability: Ability of two or more systems or components to exchange information and to use the information that has been exchanged.

least significant bit (LSB): The bit in the binary notation of a number that is the coefficient of the lowest exponent possible.

message: Information that is to be passed between devices as a single logical entity. A message may occupy one or more packets.

meta-: A Greek prefix meaning that which pertains to the whole or overall entity or that which is in common or shared with all member entities comprising the whole.

meta-TEDS: The collection of those Transducer Electronic Data Sheet (TEDS) data fields that pertain to the whole or overall entity or those that are in common or shared with all member entities (TransducerChannels) comprising the whole product.

network-capable application processor (NCAP): A device between the transducer interface modules (TIMs) and the application. This processor performs network communications with the application via the network interface, network communications with TIM via the transducer interface, and data conversion or other processing functions. It is an IEEE 1451 server that provides network services for applications (IEEE 1451 clients), and a client that can access transducer data from TIMs via the transducer interface.

not-a-number (NaN): A bit pattern of a single-precision or double-precision real number data type that is a result of an invalid floating point operation. For single-precision real numbers, the bit pattern has an exponent of 255 (decimal) and a nonzero mantissa. The sign is not considered in determining whether a value is not a number.

NOTE—See IEEE Std 754™-1985 [B2].^{14,15}

octet: A group of 8 bits. In the United States, an octet is usually referred to as a *byte*.

packet: A block of information that is to be passed by the physical layer between devices in a single transmission.

sensor: A transducer that converts a physical, biological, or chemical parameter into an electrical signal.

signal conditioning: The transducer signal processing that involves operations such as amplification, compensation, filtering, and normalization.

smart transducer: A transducer that provides functions beyond those necessary for generating a correct representation of a sensed or controlled quantity and is as follows:

- Consists of transducers (sensors and/or actuators), signal conditioning and data conversion, an application processor or data processing unit, and a network communication interface and Transducer Electronic Data Sheet (TEDS).
- Provides functions, such as sensing and/or actuating, signal and data conversion, intelligent data processing, network communication with an application or outside world, self-identification, and self-description based on TEDS.
- From a sensor network perspective, can be constructed or deployed as a network-capable application processor (NCAP) and a number of transducer interface modules (TIMs).

transducer: A device that converts energy from one domain into another. The device may be either a sensor or an actuator.

transducer interface module (TIM): A module that contains the Transducer Electronic Data Sheet (TEDS), logic to implement the transducer interface, the transducer(s) or connection to the transducer(s), and any signal conversion or signal conditioning. It can contain sensors and/or actuators.

TransducerChannel: A single transducer and all of the signal conditioning and conversion components associated with that transducer.

TransducerChannel number: A 16-bit number assigned to an individual TransducerChannel within a transducer interface module (TIM) by the manufacturer.

TransducerChannel proxy: A device that is created to allow a collection of TransducerChannels to be treated as a single entity. A TransducerChannel proxy is similar to a normal TransducerChannel except that it does not require a TransducerChannel Transducer Electronic Data Sheet (TEDS), and it cannot have a Calibration TEDS, Transfer Function TEDS, or Frequency Response TEDS. It may support other TEDS. A TransducerChannel proxy may respond to commands.

¹⁴ Notes in text, tables, and figures of a standard are given for information only and do not contain requirements needed to implement this standard.

¹⁵ The numbers in brackets correspond to those of the bibliography in Annex A.

Transducer Electronic Data Sheet (TEDS): An electronic data sheet describing a transducer interface module (TIM) or a TransducerChannel or a network-capable application processor (NCAP).

NOTE 1—The structures of multiple TEDS are described in this standard.

NOTE 2—The term *TEDS* is used to refer to single or multiple instances in this standard.

transfer: The act or process of moving information from one digital device to another.

transfer function: The function that defines the response of a TransducerChannel in both amplitude and frequency.

virtual Transducer Electronic Data Sheet (TEDS): A Transducer Electronic Data Sheet (TEDS) that is stored permanently in a location other than the transducer interface module (TIM) or network-capable application processor (NCAP).

wireless transducer interface module (WTIM): A transducer interface module with a wireless communication interface. *See also:* **transducer interface module (TIM).**

3.2 Acronyms and abbreviations

API	application programming interface
APP	application
BR-Sync	broadcast-based synchronization (or BRS)
CMD	command
CSV	comma-separated values
C-OP	Comma-Separated Style Operation
D-OP	Direct Format Operation
D0	Dot0 (IEEE Std 1451.0-2024)
D0-OP	Dot0 Operation
FIFO	first-in-first-out
EOS	end of string
FQDN	Fully Qualified Domain Name
ID	identification/identifier
IoT	Internet of Things
IP	Internet Protocol
LAN	local area network
LNS	location name string

MQTT	Message Queuing Telemetry Transport
NaN	not-a-number
NCAP	network-capable application processor
NCAPA	network-capable application processor (NCAP) address
NTP	Network Time Protocol
OP	operation
OSI	Open Systems Interconnection
PTP	Precision Time Protocol
PAN	personal area network
PHY	physical layer
QoS	quality of service
RES	response
REQ	request
RR-Sync	request-reply-based synchronization (or RRS)
SPFX	IEEE 1451.1.6 standard prefix
SRQ	synchronization request
STYPE	service type
Sync	synchronization
SYN	synchronization message (i.e., SYN message)
TCP	Transmission Control Protocol
TEDS	Transducer Electronic Data Sheet
TEDSC	Transducer Electronic Data Sheet (TEDS) Command
TEDSR	Transducer Electronic Data Sheet (TEDS) Reply
TIM	transducer interface module
TNI	Transducer Name Identifier
TOM	Type of Message
UID	unique identifier
UUID	universally unique identifier

WAN wide area network

WTIM wireless transducer interface module (WTIM)

4. Data types

All data types used throughout the remainder of this standard are defined according to the definition given in IEEE Std 1451.0-2024. Refer to IEEE Std 1451.0-2024 for their details. Note that the definition is extended for ambiguity resolution, and all data types given in IEEE Std 1451.0-2024 are not explained here.

Data type name	Symbol	Octet size	Expression range or descriptions
Unsigned octet integer	UInt8	1	Positive integer value from 0 to 255.
Unsigned 16-bit integer	UInt16	2	Positive integer value from 0 to 65 535.
Signed 32-bit integer	Int32	4	Signed integer value from –2 147 483 648 to 2 147 483 647.
Unsigned 32-bit integer	UInt32	4	Positive integer value from 0 to 4 294 967 295.
Unsigned 48-bit integer	UInt48	6	Positive integer value from 0 to 281 474 976 710 655.
Signed 64-bit integer	Int64	8	Signed integer value from –9 223 372 036 854 775 808 to 9 223 372 036 854 775 807.
Single-precision real	Float32	4	Encoded real numbers using IEEE Std 754-1985 [B2].
Double-precision real	Float64	8	Encodes real numbers using IEEE Std 754-1985 [B2]. Floating-point not-a-number (NaN) expression in TEDS has a special meaning. According to IEEE Std 754-1985 [B2], a single-precision number with an exponent of 255 and a nonzero fractional portion (mantissa) is NaN regardless of the sign bit. The recommended value for use in a TEDS field for NaN is 0x7FFFFFFF (hexadecimal number).
_String	String	Var,	Originally, this is defined in IEEE Std 1451.0-2024. In this standard, this data type represents text strings and has a variable size. The basic type of text string is defined in IEEE Std 1451.2™-1997 [B4]. MQTT messages and topics have to use UTF-8 style text expression and can be coded on the standard. Moreover, if the size of String was not given, String does not contain a NULL character, and it represents EOS (end of string special character).
Boolean	Boolean	1	A bit or octet with a nonzero value is considered True. A zero value represents the False state of the variable.
TimeRepresentation	TimeRep	6+4	Struct of UInt48 sec and UInt 32 nsecs.
TimeInstance	TimeIns	6+4	Struct of UInt48 sec and UInt 32 nsecs.
TimeDuration	—	8	Time interval expressed in nanoseconds.

As for TimeInstance and TimeDuration, this abstract class defines the time representation. It is a subclass of TimeInstance and TimeDuration classes. The definition of the two arguments is as follows:

```
struct TimeRepresentation {
    UInt48 secs;
    UInt32 nsecs;
};
```

In this data structure, secs (UInt48) is seconds (s). This unsigned 32-bit number represents the number of seconds from the beginning of the epoch (normally 0 h on January 1, 1970); nsecs (UInt32) is nanoseconds. This unsigned 32-bit integer comprises two smaller fields. The most significant bit will be interpreted as the sign bit of the entire time value. The least significant 31 bits represent the number of nanoseconds to be added to the value identified by the Seconds field before the sign is applied. The value specified in the Nanoseconds field is constrained to the domain 0 to 999 999 999, inclusive. IEEE Std 1451.0™-2024 uses the time representation specified in IEEE Std 1588™-2002 [B6], and this standard observes IEEE Std 1451.0-2024 rules.

The TimeInstance can be either with or without leap seconds, and the designer decides. However, avoiding the use of leap seconds is recommended according to the content submitted to the General Conference of Weights and Measures (CGPM), which will vote on whether or not to abolish the system at its meeting on November 18, 2022. If the time can be obtained from the upper-level network-capable application processor (NCAP) using the time synchronization defined in Clause 8, and if this upper-level NCAP correctly expresses leap seconds using Network Time Protocol (NTP) or other methods, then leap seconds can be used. It is also possible to respond by having the administrator modify the internal register that has the timeinstance. In any case, the entire system should be unified whether the system uses the leap or not.

The following data types are not used in this standard and IEEE Std 1451.0-2024 but are provided to allow the applications using this standard to pass data in these formats. Var. means variable size.

8-bit signed integer	Int8	1	Integers value from –128 to 127.
16-bit signed integer	Int16	2	Integers value from –32 768 to 32 767.
Arbitrary octet array	OctetArray	Var.	An arbitrary number of octets as an array.
String array	StringArray	Var.	An arbitrary number of string data type, treated as an aggregate entity.
Boolean array	BooleanArray	Var.	An arbitrary number of _boolean data types, treated as an aggregate entity.
Array of 8-bit signed integers	Int8Array	Var.	An arbitrary number of octets, treated as an aggregate entity made up of 8-bit signed integers (Int8).
Array of 16-bit signed integers	Int16Array	Var.	An arbitrary number of 16-bit signed integers (Int16), treated as an aggregate entity.
Array of 32-bit signed integers	Int32Array	Var.	An arbitrary number of 32-bit signed integers (Int16), treated as an aggregate entity.
Array of 8-bit unsigned integers	UInt8Array	Var.	An arbitrary number of octets, treated as an aggregate entity made up of 8-bit unsigned integers (Int8).
Array of 16-bit unsigned integers	UInt16Array	Var.	An arbitrary number of 16-bit unsigned integers (Int16), treated as an aggregate entity.
Array of 32-bit unsigned integers	UInt32Array	Var.	An arbitrary number of 32-bit signed integers (Int16), treated as an aggregate entity.
Array of single-precision real numbers	Float32Array	Var.	An arbitrary number of single-precision real numbers, as specified in IEEE Std 754-1985 [B2], is treated as an aggregate entity.
Array of double-precision real numbers	Float64Array	Var.	An arbitrary number of double-precision real numbers, as specified in IEEE Std 754-1985 [B2], is treated as an aggregate entity.
Array of TimeDuration data types	TimeDurationArray	Var.	An arbitrary number of TimeDuration data type treated as an aggregate entity.
Array of TimeInstance data types	TimeInstanceArray	Var.	An arbitrary number of TimeInstance data types treated as an aggregate entity.

5. IEEE 1451 family reference model as IEEE Std 1451.1.6

5.1 IEEE 1451 family reference model

Several data networks are in general use throughout the industry section, as well as a myriad of industrial transducers, including sensors and actuators. A user must often turn to a single manufacturer for networks, transducers, and controls to help ensure compatible operation. It would make tremendous economic sense to enable any transducer or embedded transducer to communicate using any data network. The concept of smart sensors (transducers) strives to make this a reality.

Thus, luminaries of the transducer, measurement-and-control, and data-network industries discussed the concept of smart transducers during the September 1993 IEEE TC-9 meeting (IEEE Instrumentation and Measurement Society Technical Committee on Sensors). As a result, two draft standards were completed for review at the end of 1995. IEEE Std 1451.1™ [B3] dealt with the transducer electronic data sheet (TEDS), while IEEE Std 1451.2™ [B4] treated the common object model. The objectives of these standards are as follows:

- Provide a common communication interface between transducers and processors.
- Provide compatibility with multiple sensor actuator bus standards.
- Provide a common communication interface to interconnect analog transducers with digital networks.
- Do not develop a new network standard.

The advantages of the use of these standards were as follows:

- Allow interoperation among transducers and control systems of different manufacturers.
- Allow the use of existing control system installations.
- Allow transducers to share a common bus, reducing wiring cost and complexity.
- Increase the usage of existing networks.

IEEE 1451 defines a transducer interface module (TIM) as a bus between sensors and controllers and a sensor data network capable application processor (NCAP) to access TIM from the Internet. The IEEE 1451 reference model of APP, NCAP, and TIM is described in Figure 1 and Figure 2.

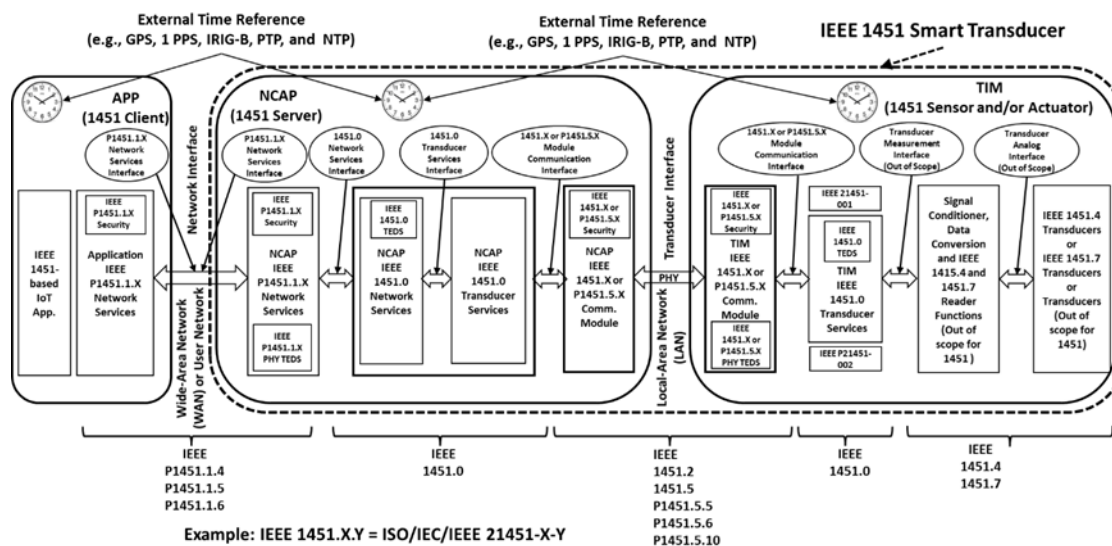


Figure 1—IEEE 1451 reference model

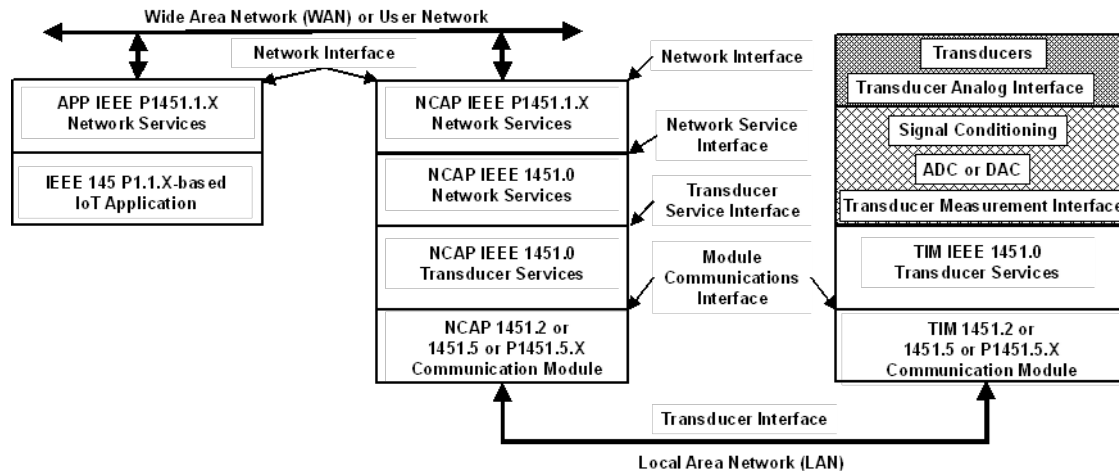


Figure 2—Alternative view of the reference model

This standard defines the connection between the NCAP applications and users' networks, such as the Internet, including user networks. The Internet environment has dynamically changed from the first publication of the IEEE 1451 family of standards. In the early 1990s, the discussion of sensor networks started, and the focus was on the structure. The ubiquitous computing and networks followed the sensor network, and it mainly discussed the operation of network-connected sensors and distributed embedded computing resources. The Internet of Things (IoT) has become popular, and it discusses the services and applications of those sensors and computing resources.

Accordingly, network communication protocols were also developed. One of the popular protocols for these environments is Message Queuing Telemetry Transport (MQTT), an open standard (ISO/IEC 20922 [B28]), a lightweight, publish-subscribe-based network protocol that transports messages between devices through a broker. MQTT usually runs over Transmission Control Protocol/Internet Protocol (TCP/IP); however, any network protocol that provides ordered, lossless, bi-directional connections can support MQTT. It is designed for connections with remote nodes with a small code footprint and limited computational resources or network bandwidth.

MQTT provides publish-and-subscribe messaging. In the specification opened as version 3.1, the formal name was given as "MQ Telemetry Protocol (MQTP)." In documents, the protocol is strictly referred to as MQTT, although the technical committee itself is named "OASIS Message Queuing Telemetry Transport Technical Committee." The communication style of MQTT is different from the general request-reply-based communication between a client and a server.

IEEE 1451.0-2024 network services require broadcast messages, which are not permitted in the Internet area. It was natural to use MQTT for IEEE 1451 as NCAP common network services. IEEE Std 1451.1.6 defines the communication rules for connecting with NCAPs from the Internet by using MQTT. In this standard, NCAP, which has a function of MQTT, is called MQTT-NCAP. IEEE 1451.1.6 communication protocol is based on IEEE Std 1451.0-2024, especially its network function definitions.

5.2 IEEE 1451 family reference model as IEEE Std 1451.1.6

5.2.1 General

This standard outlines the connection between applications (APPs) as IEEE 1451 clients and NCAPs through the utilization of MQTT. Within this standard, the IEEE 1451 client application uses IEEE 1451.1.6 network services, serving as a network interface, and IoT applications adhering to IEEE Std 1451. The primary

objective of the IEEE 1451 client application is to establish communication with the NCAP, utilizing the IEEE 1451.1.6 network services, using the user's preferred networks, such as wired or wireless IEEE 802.3™/Internet.

5.2.2 Wide area network

The wide area network (WAN) is defined in IEEE Std 1451.0-2024 as a pair of local area networks (LANs). The local area network in the IEEE 1451 family stands for a network between NCAP and TIMs, and the network shall be categorized in a sensor area network category. The wide area network is a network between NCAPs and their Applications. In this standard, the wide-area network in IEEE Std 1451.0-2024 means the IP network and shall be available to use MQTT. Note that LAN, WAN, personal area network (PAN), and other network categories using IP are applicable as a wide area network in IEEE Std 1451.0-2024. This standard regards the Internet as WAN. Local areas can use broadcast or multicast; however, the Internet does not provide broadcast or multicast methods from a security perspective.

5.2.3 NCAP (IEEE 1451 server)

This standard's NCAP (IEEE 1451 server) consists of IEEE 1451.1.6 and other IEEE 1451.1.X network services as network interfaces, IEEE 1451.0-2024 network services, IEEE 1451.0 transducer services as a transducer interface, and NCAP communication module. In addition, this standard defines the functionality and data representation that should be extended to support MQTT. For example, IEEE 1451.0 messages are translated to implement on MQTT-based systems and extended their original functionality to improve usability from the perspective of MQTT application targets.

5.2.4 APP as IEEE 1451 client

APP is a software module as IEEE 1451 client running on a device that performs network communications with the NCAP (1451 server) to access transducer data and the TEDS, and processes transducer data.

APP is an IEEE 1451 standard-based IoT application including the IEEE 1451.1.6 network interface and IEEE 1451.0 network services. The APP should communicate with the NCAP based on the IEEE 1451.1.6 interface and IEEE 1451.0-2024 network services via a user network or WAN that the user desires; it could be wired or wireless IEEE 802.3/intranet/Internet/VPN.

5.3 Merit of the use of IEEE 1451.1.6 and MQTT for IEEE 1451

MQTT is a popular communication protocol for small IoT devices. The reason for this affinity to such devices can be summarized as follows:

- MQTT is based on a simple protocol design. For example, MQTT header size can be 2 B at minimum. IoT devices are configured with a tiny processor with a small memory size and limited processing power to reduce their power consumption. Moreover, when using wireless communications for these devices, the power consumption of wireless communication modules becomes dominant. Reducing the power consumption for wireless communication is indispensable to extending the lifetime of the devices. The simplicity of MQTT contributes to the reduction of power consumption from several perspectives.
- MQTT uses a broker server to distribute messages from MQTT clients. The need for a broker server can be a demerit that needs another network service to distribute messages. However, when multicasting or broadcasting messages, it is required to manage a protocol to identify all IP addresses for the network nodes. It is also required to send the messages one by one to all the network nodes

on the Internet. There are many problems with multicasting or broadcasting, such as using a new protocol to know the IP address set to nodes in a group, increasing power consumption to maintain the protocol and multicasting or broadcasting messages, increasing the possibility of security issues when using these complex protocols. MQTT provides broadcasting and multicasting method controlled by its special addressing rule of topic names and its publish-subscribe communication style.

- IEEE Std 1451.0-2024 requires broadcast messages to announce and discover NCAPs, which are used to construct the group of NCAPs to communicate. However, the broadcast messages are not available on the Internet because they cause security issues or communication problems, such as node searches to find attack targets, denial of service attacks, and network congestion. Therefore, all broadcast addresses were closed on the Internet. Furthermore, full IEEE 1451.0 functions can be used only in a Layer-2 network domain for these reasons. Therefore, the functionality of IEEE 1451 is deteriorated in its use on the Internet. For using the full function of IEEE 1451 over the Internet, the broadcast and multicast functions of MQTT are indispensable. IEEE Std 1451.1.6 provides the rule to use IEEE 1451 over MQTT. IEEE 1451.1.6 enables the use of the IEEE 1451.0 full functions over the Internet.

In general communication models, the power cost for processing and network communication to distribute messages becomes larger according to the number of network nodes. When using MQTT, MQTT broker receives only one published message, and the broker distributes the message to all subscribers. Therefore, MQTT can drastically reduce the power consumption of MQTT clients when the number of destination nodes is increased. In this delivery, the topic name plays an important role. MQTT message has the topic name, and the topic name is the keyword that indicates a group of nodes receiving the message. Therefore, the message distribution is completely controlled only by using the topic name.

The public-subscribe style communication of MQTT is the essential contribution of IEEE 1451.1.6 to using IEEE 1451 over the Internet and for IoT devices. This standard defines the topic name rules for emulating IEEE 1451.0 message transactions to enable the use of full functions defined in IEEE 1451.0-2024 network functions.

6. Network/communication configuration for NCAP

6.1 Basic communication model in IEEE Std 1451.0-2024

6.1.1 General

The basic communication models shown in Figure 3 are command-reply communication models. The client shall send the command to the server. The server shall receive and process the commands from the client and then send the corresponding replies back to the client. The synchronous and asynchronous communication category in IEEE Std 1451.0-2024 can be understood as a communication style and is not based on the general synchronous (blocking) and asynchronous (non-blocking) communication protocol design. For example, synchronous communication consists of command and reply messages. After the client sends the command message, the client can do other processes and even send another command message before receiving a reply message. This kind of asynchronous communication design is commonly used, and this asynchronous communication design and alternative synchronous communication, which block the process of waiting for the reply message, are different from synchronous and asynchronous communication in IEEE Std 1451.0-2024. For historical reasons, IEEE Std 1451.0-2024 prepares asynchronous communication; however, asynchronous communication design can be used for synchronous and asynchronous communication of IEEE Std 1451.0-2024, which is a typical communication design style. Moreover, the asynchronous communication design in IEEE 1451.0 can be replaced with synchronous communication of IEEE 1451.0. Regarding the handling of synchronous and asynchronous operations, IEEE 1451.0 provides its own interpretation, while IEEE 1451.1.6 explains them using the more general terms “blocking” and “non-

blocking.” MQTT generally provides callback functions, making it asynchronous in the general sense. By leveraging this feature, both synchronous and asynchronous behavior as defined in IEEE Std 1451.0-2024 can be described.

Figure 3(a) shows the command-reply communication model. The client shall send the command to the server. The server shall receive and process the commands from the client and then send the corresponding replies to the client. The client-server communication model supports both synchronous communication (blocking, namely, waiting for the connecting message) with and without the callback service and an asynchronous communication model (non-blocking, namely, continuing other processes, including other communications) with the callback service.

Figure 3(b) shows synchronous and asynchronous client-server communications between the APP (IEEE 1451 client) and the NCAP (IEEE 1451 server) based on network services. APPs (IEEE 1451 clients) shall connect to and communicate with one NCAP (IEEE 1451 server) in a synchronous or asynchronous (optional) way. The NCAP (IEEE 1451 server) provides network services for the APP (IEEE 1451 client) to access transducer data and TEDS. The synchronous client-server models shall be applied to the communication between the APP (IEEE 1451 client) and the NCAP (IEEE 1451 server). The asynchronous client-server model with the callback service is optional for client-server communications between APP and NCAP.

Figure 3(c) shows synchronous and asynchronous client-server communication between the NCAP (client) and TIM (server) based on transducer services. Normally, one TIM or wireless transducer interface module (WTIM) shall only register to one NCAP (client). One NCAP (client) shall connect and communicate with multiple TIMs/WTIMs that are registered to it in a synchronous or asynchronous (optional) way. The asynchronous client-server model with the callback service is optional for client-server communications between NCAP and TIM.

Figure 3(d) shows synchronous and asynchronous client-server communications between the APP (IEEE 1451 client) and the NCAP (IEEE 1451 server) based on network services and the client-server communication between the NCAP (client) and TIM (server) based on transducer services. The asynchronous client-server communication model with the callback service is optional for communications between APP and NCAP and between NCAP and TIM.

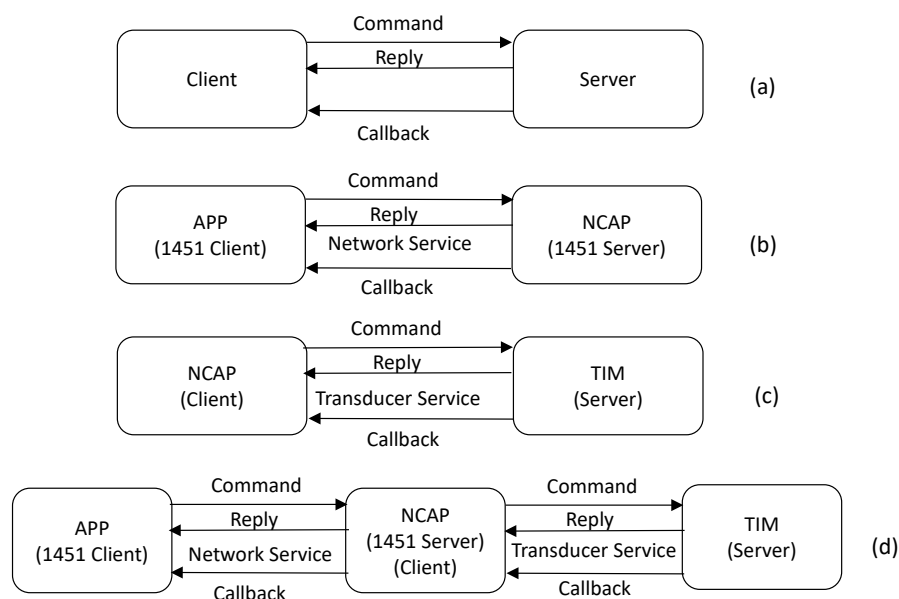


Figure 3—Basic communication models

6.1.2 Publisher-subscriber communication model in IEEE Std 1451.0-2024

The publish-subscribe communication model is described in Figure 4. This model consists of a publisher, two or three subscribers, and a broker. The subscribers shall subscribe to the topic that the publisher published. To publish a message means sending a message to an MQTT broker with a topic specifying the message's distribution. The broker distributes the message to the subscribers who subscribe to the same topic. Namely, the topic and message can be regarded as destination identifier (ID) and transferred data contents. For MQTT, a subscriber can subscribe to multiple topics.

Publish-subscribe could be unicast, multicast, or broadcast. For unicast, each subscriber has a unique topic name, and the publisher can send a message by using the topic name. For multicast, the multicast group of subscribers subscribes to the same group topic name. The published message on the topic is shared with the subscriber group. For broadcast, all subscribers subscribe to only one topic. As for the MQTT used in this standard, the middle node as the message distributor is an MQTT broker. Additionally, the publish-subscribe communication is based on an asynchronous communication design; however, it can be used for both synchronous and asynchronous communication of IEEE Std 1451.0-2024.

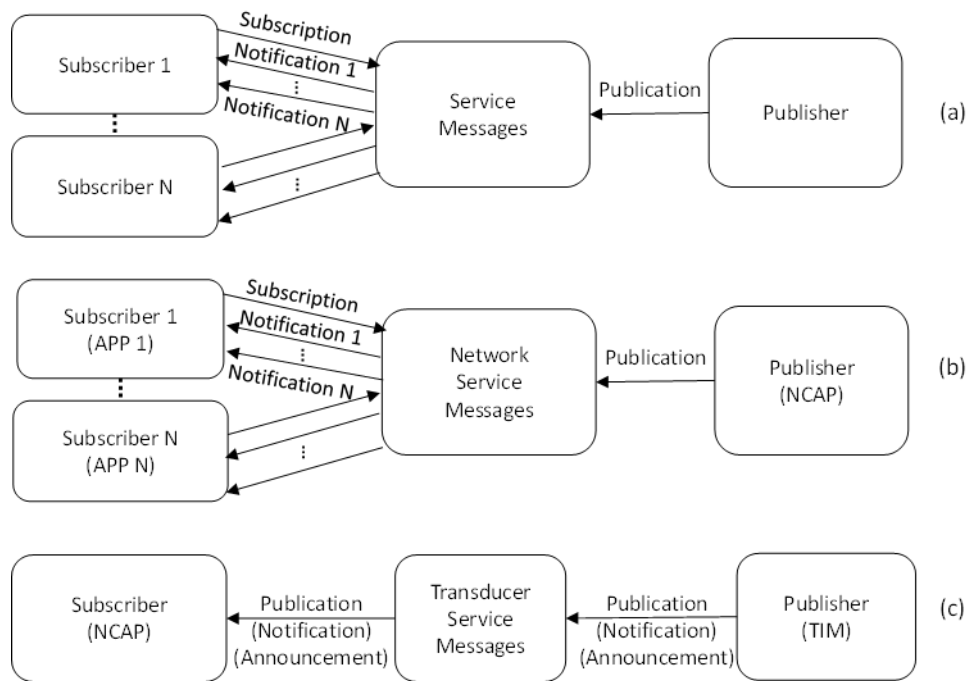


Figure 4—Publish-subscribe model

A publish-subscribe communication model is shown in Figure 4. Figure 4(a) is one type of publish-subscribe model in which the service messages are events. An event publisher, which is a generator of events, publishes event notifications to those event subscribers who have subscribed to the events. Figure 4(b) shows a publish-subscribe event between the NCAP and APPs. Figure 4(c) shows a publish-subscribe event between the TIM and NCAP.

6.1.3 Network services in IEEE Std 1451.0-2024

IEEE 1451.0-2024 network services support a set of services (command and reply messages) for IEEE 1451 smart transducers to communicate transducer data and TEDS information to and from a WAN or user network via various network communication protocols, including MQTT. Network services aim to achieve smart transducer data interoperability in the network interface. In addition, this standard provides network services over MQTT.

The network services can be classified into four service types: discovery services, transducer access services, TEDS access services, and event notification services. Each network service type consists of a set of services that consist of two or three messages. The network service type names and their enumeration expressions are given in Table 1 and as in IEEE Std 1451.0-2024.

Table 1—Network service type

Network service type name (meaning)	Network service type (enumeration)
Reserved	0
Discovery services	1
Transducer access services	2
TEDS access services	3
Event notification services	4
Reserved	5–127
User-defined	128–255

The network service consists of a command message, reply message, announcement message, notification message, and callback message. The message names and their enumeration expressions are given in Table 2 and as in IEEE Std 1451.0-2024.

In this standard, the word command is used as a message type, which causes the following reply message according to IEEE Std 1451.0-2024. The word request is used separately and is not used with reply messages and a communication style of request-response style communication. The typical communication patterns of network services are given in Figure 5.

Table 2—Network service message type

Message name	Message type (enumeration)
Reserved	0
Command	1
Reply	2
Announcement	3
Notification	4
Callback	5
Reserved	6–127
User-defined	128–255

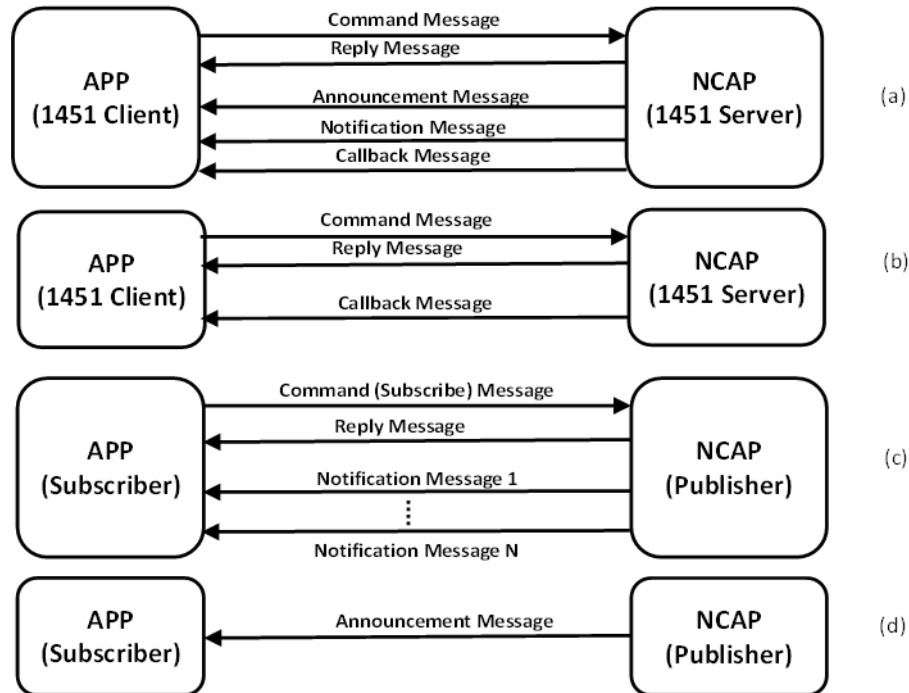


Figure 5—Network service messages on publish-subscribe communication model

Table 7 shows the network service message structure given in IEEE Std 1451.0-2024, which consists of Network Service Type, Network Service Id, length, and message body.

`NetSvcType` is a variable type of Network service type—the values can be 1 to 4 integer values—which is defined in Table 1.

`NetSvcID` is a network service ID, which is used to distinguish services. The `NetSvcID` is defined for each service type. Discovery service, transducer access service, TEDS access service, and event notification service have each network service as given in Table 3, Table 4, Table 5, and Table 6.

Table 3—Discovery service

Service type	NetSvcId (enumeration)	Service name	Service Code
1	1	NCAPAnnouncement	01 01
	2	NCAPDeparture	01 02
	3	NCAPAbandonment	01 03
	4	NCAPTIMAnnouncement	01 04
	5	NCAPTIMDeparture	01 05
	6	NCAPTIMTransducerAnnouncement	01 06
	7	NCAPTIMTransducerDeparture	01 07
	8	NCAPDiscovery	01 08
	9	NCAPTIMDiscovery	01 09
	10	NCAPTIMTransducerDiscovery	01 10