INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

X.743
(06/98)

SERIES X: DATA NETWORKS AND OPEN SYSTEM COMMUNICATIONS

OSI management – Management functions and ODMA functions

Information technology – Open Systems Interconnection – Systems Management: Time management function

ITU-T Recommendation X.743

(Previously CCITT Recommendation)

ITU-T X-SERIES RECOMMENDATIONS

DATA NETWORKS AND OPEN SYSTEM COMMUNICATIONS

PUBLIC DATA NETWORKS	
Services and facilities	X.1–X.19
Interfaces	X.20–X.49
Transmission, signalling and switching	X.50–X.89
Network aspects	X.90–X.149
Maintenance	X.150–X.179
Administrative arrangements	X.180–X.199
OPEN SYSTEMS INTERCONNECTION	
Model and notation	X.200–X.209
Service definitions	X.210–X.219
Connection-mode protocol specifications	X.220–X.229
Connectionless-mode protocol specifications	X.230–X.239
PICS proformas	X.240–X.259
Protocol Identification	X.260–X.269
Security Protocols	X.270–X.279
Layer Managed Objects	X.280–X.289
Conformance testing	X.290–X.299
INTERWORKING BETWEEN NETWORKS	
General	X.300–X.349
Satellite data transmission systems	X.350–X.399
MESSAGE HANDLING SYSTEMS	X.400–X.499
DIRECTORY	X.500–X.599
OSI NETWORKING AND SYSTEM ASPECTS	
Networking	X.600–X.629
Efficiency	X.630–X.639
Quality of service	X.640–X.649
Naming, Addressing and Registration	X.650–X.679
Abstract Syntax Notation One (ASN.1)	X.680–X.699
OSI MANAGEMENT	
Systems Management framework and architecture	X.700–X.709
Management Communication Service and Protocol	X.710–X.719
Structure of Management Information	X.720–X.729
Management functions and ODMA functions	**X.730–X.799**
SECURITY	X.800–X.849
OSI APPLICATIONS	
Commitment, Concurrency and Recovery	X.850–X.859
Transaction processing	X.860–X.879
Remote operations	X.880–X.899
OPEN DISTRIBUTED PROCESSING	X.900–X.999

For further details, please refer to ITU-T List of Recommendations.

INTERNATIONAL STANDARD 10164-20

ITU-T RECOMMENDATION X.743

INFORMATION TECHNOLOGY – OPEN SYSTEMS INTERCONNECTION – SYSTEMS MANAGEMENT: TIME MANAGEMENT FUNCTION

Source

The ITU-T Recommendation X.743 was approved on the 26th of June 1998. The identical text is also published as ISO/IEC International Standard 10164-20.

FOREWORD

ITU (International Telecommunication Union) is the United Nations Specialized Agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of the ITU. The ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Conference (WTSC), which meets every four years, establishes the topics for study by the ITU-T Study Groups which, in their turn, produce Recommendations on these topics.

The approval of Recommendations by the Members of the ITU-T is covered by the procedure laid down in WTSC Resolution No. 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

INTELLECTUAL PROPERTY RIGHTS

The ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. The ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, the ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementors are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database.

CONTENTS

Page

Page

Introduction

ITU-T Rec. X.700 series | ISO/IEC 10164 is a Series of Recommendations | International Standards developed according to ITU-T Rec X.200 | ISO/IEC 7498-1 and ITU-T Rec. X.700 | ISO/IEC 7498-4. ITU-T Rec. X.700 series | ISO/IEC 10164 is related to the following International Standards:

- CCITT X.710 | ISO/IEC 9595:1990, Information technology – Open System Interconnection – Common management information service definition;
- CCITT X.711 and CCITT X.712 | ISO/IEC 9596:1990, Information technology – Open System Interconnection – Common management information protocol specification;
- CCITT X.701 | ISO/IEC 10040:1992, Information technology – Open Systems Interconnection – Systems management overview;
- CCITT Recs. X.730, X.740 series and ITU-T Rec. X.750 series | ISO/IEC 10064:1992, Information technology – Open Systems Interconnection – Systems Management.

OSI management standardization inevitably involves coordinated work by a number of standards bodies. ITU-T SG 7 and ISO/IEC JTC 1 SC 21/WG 4 are jointly responsible for the development of Recommendations | International Standards that describe the architecture for OSI managemnt, the services, protocols, and functions that are used for systems management, and the structure of management information. Other working groups, in ITU-T, ISO/IEC JTC 1 SC 21, ISO/IEC JTC 1 SC 6 and elsewhere, are responsible for the development of Recommendations | International Standards that describe the management aspects of particular layers of the OSI Basic Reference Model; these may describe (N)-layer management protocols, management aspects of (N)-layer operation, and managed objects that provide a "management view" of aspects of the layer operation and are visible to systems management.

INTERNATIONAL STANDARD

ITU-T RECOMMENDATION

INFORMATION TECHNOLOGY – OPEN SYSTEMS INTERCONNECTION – SYSTEMS MANAGEMENT: TIME MANAGEMENT FUNCTION

1 Scope

This Recommendation | International Standard defines a Systems Management Function that may be used by an application process in a centralized or decentralized management environment to interact for the purpose of systems management, as defined by ITU-T Rec. X.200 | ISO/IEC 7498-1. This Recommendation | International Standard defines a function which consists of generic definitions, services, and functional units. This function is positioned in the application layer of ITU-T Rec.X.200 | ISO/IEC 7498-1 and is defined according to the model provided by ISO 9545. The role of systems management functions is described by CCITT Rec. X.701 | ISO /IEC 10040.

This Recommendation | International Standard:

- defines a service for the management of clocks for use by OSI management and available for use by OSI applications and others;
- establishes user requirements for this Recommendation | International Standard;
- establishes a time management function model, addressing the components of a generic time service involving communication between systems, that relates the service and generic definitions provided by this function to the user requirements;
- defines generic object classes, attribute types, operation types, notification types, and parameters documented in accordance with CCITT Rec. X.722 | ISO/IEC 10165-4;
- specifies compliance requirements placed on other standards that make use of these generic definitions;
- defines the services provided by the function;
- specifies the management protocol that is necessary in order to provide the services;
- defines the relationship between these services and systems management operations and notifications;
- specifies the abstract syntax necessary to identify and negotiate the function unit in the protocol;
- defines relationships with other systems management functions;
- specifies conformance requirements to be met by implementation of this Recommendation | International Standard;
- identifies time synchronization protocols.

This Recommendation | International Standard does not:

- address the provision of time information within a local system;
- define the nature of any implementation intended to provide the Time Management Function;
- specify the manner in which management is accomplished by the user of the Time Management Function;
- define the nature of any interaction which results in the use of the Time Management Function;
- specify the services necessary for the establishment, use, and normal or abnormal release of a management association.

2 Normative references

The following Recommendations and International Standards contain provisions which, through reference in this text, constitute provisions of this Recommendation | International Standard. At the time of publication, the editions indicated were valid. All Recommendations and Standards are subject to revision, and parties to agreements based on this Recommendation | International Standard are encouraged to investigate the possibility of applying the most recent edition of the Recommendations and International Standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards. The Telecommunications Standardizations Bureau of the ITU maintains a list of currently valid ITU-T Recommendations.

2.1 Identical Recommendations | International Standards

- ITU-T Recommendation X.210 (1993) | ISO/IEC 10731:1994, *Information technology – Open Systems Interconnection – Basic Reference Model – Conventions for the definition of OSI services.*
- CCITT Recommendation X.701 (1992) | ISO/IEC 10040:1992, *Information technology – Open Systems Interconnection – Systems management overview*[1].
- CCITT Recommendation X.720 (1992) | ISO/IEC 10165-1:1993, *Information technology – Open Systems Interconnection – Structure of management information: Management Information Model.*
- CCITT Recommendation X.721 (1992) | ISO/IEC 10165-2:1992, *Information technology – Open Systems Interconnection – Structure of management information: Definition of management information.*
- CCITT Recommendation X.722 (1992) | ISO/IEC 10165-4:1992, *Information technology – Open Systems Interconnection – Structure of management information: Guidelines for the definition of managed objects.*
- ITU-T Recommendation X.723 (1993) | ISO/IEC 10165-5:1994, *Information technology – Open Systems Interconnection – Structure of management information: Generic management information.*
- ITU-T Recommendation X.724 (1996) | ISO/IEC 10165-6:1997, *Information technology – Open Systems Interconnection – Structure of management information: Requirements and guidelines for implementation conformance statement proformas associated with OSI management.*
- CCITT Recommendation X.730 (1992) | ISO/IEC 10164-1:1993, *Information technology – Open Systems Interconnection – Systems Management: Object management function.*
- CCITT Recommendation X.731 (1992) | ISO/IEC 10164-2:1993, *Information technology – Open Systems Interconnection – Systems Management: State management function.*
- CCITT Recommendation X.732 (1992) | ISO/IEC 10164-3:1993, *Information technology – Open Systems Interconnection – Systems Management: Attributes for representing relationships.*
- ITU-T Recommendation X.738 (1993) | ISO/IEC 10164-13:1995, *Information technology – Open Systems Interconnection – Systems Management: Summarization function.*
- ITU-T Recommendation X.739 (1993) | ISO/IEC 10164-11:1994, *Information technology – Open Systems Interconnection – Systems Management: Metric objects and attributes.*
- CCITT Recommendation X.740 (1992) | ISO/IEC 10164-8:1993, *Information technology – Open Systems Interconnection – Systems Management: Security audit trail function.*
- ITU-T Recommendation X.741 (1995) | ISO/IEC 10164-9:1995, *Information technology – Open Systems Interconnection – Systems Management: Objects and attributes for access control.*
- ITU-T Recommendation X.742 (1995) | ISO/IEC 10164-10:1995, *Information technology – Open Systems Interconnection – Systems Management: Usage metering function for accounting purposes.*
- ITU-T Recommendation X.745 (1993) | ISO/IEC 10164-12:1994, *Information technology – Open Systems Interconnection – Systems Management: Test management function.*
- ITU-T Recommendation X.746 (1995) | ISO/IEC 10164-15:1995, *Information technology – Open Systems Interconnection – Systems Management: Scheduling function.*

---

[1] As amended by ITU-T Rec. X.701/Cor.2 | ISO/IEC 10040/Cor.2.

2.2 Paired Recommendations | International Standards equivalent in technical content

- CCITT Recommendation X.208 (1988), *Specification of Abstract Syntax Notation One (ASN.1).*

  ISO/IEC 8824:1990, *Information technology – Open Systems Interconnection – Specification of Abstract Syntax Notation One (ASN.1).*
- CCITT Recommendation X.209 (1988), *Specification of basic encoding rules for Abstract Syntax Notation One (ASN.1).*

  ISO/IEC 8825:1990, *Information technology – Open Systems Interconnection – Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1)*
- ITU-T Recommendation X.291 (1995), *OSI conformance testing methodology and framework for protocol Recommendations for ITU-T applications – Abstract test suite specification.*

  ISO/IEC 9646-2:1994, *Information technology – Open Systems Interconnection – Conformance testing methodology and framework – Part 2: Abstract Test Suite specification.*
- ITU-T Recommendation X.296 (1995), *OSI conformance testing methodology and framework for protocol Recommendations for ITU-T applications – Implementation conformance statements.*

  ISO/IEC 9646-7:1995, *Information technology – Open Systems Interconnection – Conformance testing methodology and framework – Part 7: Implementation Conformance Statements.*
- CCITT Recommendation X.700 (1992), *Management framework for Open Systems Interconnection (OSI) for CCITT applications.*

  ISO/IEC 7498-4:1989, *Information processing systems – Open Systems Interconnection – Basic Reference Model – Part 4: Management framework.*
- CCITT Recommendation X.710 (1991), *Common management information service definition for CCITT applications.*

  ISO/IEC 9595:1991, *Information technology – Open Systems Interconnection – Common management information service definition.*
- CCITT Recommendation X.711 (1991), *Common management information protocol specification for CCITT applications.*

  ISO/IEC 9596-1:1991, *Information technology – Open Systems Interconnection – Common management information protocol – Part 1: Specification.*

2.3 Additional references

- ITU-T Recommendation M.3100 (1995), *Generic Network Information Model.*
- ITU-T Recommendation M.3101 (1995), *Managed object conformance statements for the Generic Network Information Model.*
- ISO/TR 8509:1987, *Information processing systems – Open Systems Interconnection – Service conventions.*

3 Definitions

For the purposes of this Recommendation | International Standard, the following definitions apply.

3.1 Management framework definitions

This Recommendation | International Standard uses the following term defined in CCITT Rec. X.700 | ISO/IEC 7498-4:

- managed object;

3.2 Systems management overview definitions

This Recommendation | International Standard uses the following terms defined in CCITT Rec. X.701 | ISO/IEC 10040:

a) managed object class;

b) Management Information Conformance Statement (MICS);

c) Managed Object Conformance Statement (MOCS);

d) MICS proforma;

e) MOCS proforma;

f) notification.

3.3 CMIS definitions

This Recommendation | International Standard uses the following term defined in CCITT Rec. X.710 | ISO/IEC 9595:

– attribute.

3.4 Management information model definitions

This Recommendation | International Standard uses the following terms defined in CCITT Rec. X.720 | ISO/IEC 10165-1:

a) action;

b) behaviour;

c) name binding;

d) package;

e) superclass.

3.5 Guidelines for the definition of managed objects definitions

This Recommendation | International Standard uses the following term defined in CCITT Rec. X.722 | ISO/IEC 10165-4:

– template.

3.6 Implementation conformance statement proforma definitions

This Recommendation | International Standard uses the following terms defined in ITU-T Rec.X.724 | ISO/IEC 10165-6:

a) Managed Relationship Conformance Statement (MRCS);

b) Management Conformance Summary (MCS);

c) MCS proforma;

d) MRCS proforma.

3.7 Additional definitions

For the purposes of this Recommendation | International Standard, the following definitions apply.

3.7.1 actual clock rate: The actual clock rate is the frequency or rate at which a clock increments, including any modifications resulting from frequency adjustment or clock training. The actual clock rate is equivalent to the basic clock rate in the absence of or prior to any frequency adjustment modifications.

3.7.2 accuracy: Accuracy is a measure of how well a local clock's time value and frequency compare to UTC.

3.7.3 adjustment rate: Adjustment rate is the frequency or rate at which a single time adjustment is applied to the local clock.

3.7.4 basic clock rate: The basic clock rate is the frequency or rate at which a clock increments in the absence of any modifications resulting from frequency adjustment.

3.7.5 Coordinated Universal Time (UTC): The time reference that is assumed to be universally correct. UTC was adopted by CCIR Recommendation 470 and described in CCIR Report 517. This is not the ASN.1 representation of generalized time.

3.7.6 correct clock: A clock where the absolute value of the error is less than its maximum error.

3.7.7 frequency offset: The first derivative of the clock's error. That is, the frequency offset is the actual rate of change of error of the clock.

3.7.8 **error of a clock**: The time offset between the clock's reading and UTC at a given instant.

3.7.9 **functioning clock**: A clock in which either the frequency offset is within the maximum frequency error of the clock or the clock is undergoing an adjustment. A functioning clock may be correct or incorrect.

3.7.10 **granularity**: The maximum precision permitted by a representation of time.

3.7.11 **local clock**: The collection of hardware and software that comprises a local source of time for a system.

3.7.12 **maximum drift of a clock**: The manufacturer's specified maximum value of frequency offset.

3.7.13 **maximum error of a clock**: The maximum error bound of the absolute value of the error of a clock.

3.7.14 **precision**: The smallest value by which a clock changes.

3.7.15 **rapport**: The state when the local clock is correct and the maximum error of the clock is within the user specified maximum error.

3.7.16 **synchronization domain**: The set of local clocks involved in the exchange of time information for the purposes of coordination. This includes local clock and clock coordination resources. The members of this set are defined by administrative, platform, or environmental considerations.

3.7.17 **synchronization source**: The source chosen by an algorithm of policy for time synchronization.

3.7.18 **time offset**: The algebraic difference between the readings of two clocks at a given instant in time.

4 Abbreviations

For the purposes of this Recommendation | International Standard, the following abbreviations apply:

- DTS Distributed Time Service
- GPS Global Positioning System
- LAN Local Area Network
- NTP Network Time Protocol
- PCS Probabilistic Clock Synchronization
- RPC Remote Procedure Call
- UTC Coordinated Universal Time

5 Conventions

This Recommendation | International Standard defines services for the Time Management function following the descriptive conventions defined in ISO/TR 8509.

The following notation is used in the service parameter tables:

- M The parameter is mandatory
- C The parameter is conditional
- (=) The value of the parameter is identical to the corresponding parameter in the interaction described by the preceding related service primitive
- U The use of the parameter is a service-user option
- – The parameter is not present in the interaction described by the primitive concerned

6 Requirements

Systems management functions have requirements to record accurately the time of occurrence of alarm notifications, fault event notifications, summarization notifications, and accesses of attribute values of managed objects. Observations of attribute values of managed objects can be observation time of attribute value, time that attribute value was changed, and time interval calculations. Also, system management includes the scheduling of managed objects. Scheduling includes the control of object attributes such as start-time, stop-time, begin-time, and end-time, and involves the tracking of seconds, hours, weeks, months, and years. In addition, applications beyond the scope of systems management require a stable robust time service.

The service objective of the Time Management function is to provide correct, accurate, and stable time among systems. The implementation of the Time Management function shall be consistent with the user's communication system application.

The derived requirements are summarized below and detailed in the following subclauses:

The time management function shall:

- define a representation of time that incorporates both a time value and an accuracy, has a granularity of at least 1 nanosecond, has a range of at least AD 1 to AD 3000, and represents time instants that occur with leap days;
- provide accurate and correct time;
- minimize the time and frequency error of each system;
- accommodate the distribution of time-related information to other systems;
- preserve the correctness of clocks;
- be robust against single failures;
- provide mechanisms to set or adjust the time value of the local clock;
- provide mechanisms to automatically configure the synchronization subnet; and/or
- provide mechanisms to adjust the frequency of the local clock.

6.1 Time representation requirements

The time management function shall define a representation of time that incorporates both a time value and an accuracy. The time representation shall have a granularity equal to or smaller than 1 nanosecond. The time representation range shall cover the period AD 1 to AD 3000.

NOTE – The following information is provided to illustrate the period of time and granularity that can be represented in 64 bits. A time representation of 64 bits with a granularity of 100 nanoseconds will cover approximately 59,973 years. Reducing the granularity to 1 nanosecond will reduce the time range represented to approximately 600 years.

The time representation shall represent time instants that occur within leap days.

The time representation need not permit the direct representation of time instants that occur within leap seconds.

6.2 Time accuracy and precision requirements

Each time value shall have an accuracy and precision associated with it. The precision is reflected in the accuracy as well as being a separate parameter. The accuracy can be represented in terms of an estimated error.

NOTE – For specialized environments, it may be necessary for accuracy and precision requirements to be specified. This is discussed under the context of the user time service in Annex I.

The accuracy of any system's clock shall not be constrained by parameters in the time management function or the underlying time synchronization protocol. A time synchronization protocol will minimize the error and maximum error of a system's clock, subject to the limitations of the underlying hardware and networks.

The bound on the deviation of any two system's clocks shall not be constrained by parameters in the time management protocol. The time management protocol will minimize the deviation between any two system's clocks, subject to the limitations of the underlying hardware and networks.

The time management protocol shall provide an indication of the maximum error at each system. (This implicitly bounds the maximum possible deviation, since it is the sum of the two maximum errors.)

Optionally, the time management function will provide the user with a management parameter that allows the required accuracy of the local clock to be specified.

6.3 Time distribution requirements

The time management function shall allow for the distribution of time management information between systems. The time management function shall be capable of operating over a wide-area network that may have large stochastic delays on the transmission paths.

The time management function shall have a mechanism for accommodating leap seconds.

6.4 Time service reliability requirements

The time management function shall preserve the correctness of clocks. If all the clocks in a synchronization domain are functioning and are correct at some time, they will remain correct at future times.

The time management function shall be configurable such that it will be robust against single failures, including intentionally induced failures. More precisely, it should be possible to configure the time management function such that in a managed network if the clock on a single system fails or is compromised, this will not affect the correctness of the clocks on any other system. This should include the failure of an external reference clock.

Each local time system shall maintain information about the state of its own time service as well as that of time services with which it is exchanging time information. Upon detection of a fault within itself or at a remote system, notifications shall be raised for potential transmission to a managing system.

The time management function shall be self-correcting in the presence of single failures. Specifically, if a single system in a managed network has an incorrect but functioning clock, it will converge to become a correct clock.

The local clocks shall be able to provide accurate and correct time even with relatively large stochastic delays on the transmission paths.

6.5 Local clock requirements

The implementation of a local clock is outside the scope of this Recommendation | International Standard. However, in order to support the time management function, a local clock has the following requirements.

A local clock shall provide mechanisms to set its time in the event of initialization or fault and to adjust its time periodically during normal operation.

As part of the procedures for periodically adjusting the time during normal operation, a local clock shall provide mechanisms to prevent itself from running backward. A local clock which has a positive error shall temporarily slow down, and a local clock which has a negative error shall temporarily speed up. The adjustment rate of the local clock must be greater than the maximum drift rate of the clock. A local clock shall converge to correct time.

NOTE 1 – Care must be taken to correctly tune the adjustment rate. An adjustment rate that is too fast will result in an unstable clock, and an adjustment rate that is too small will never converge.

Optionally, a local clock shall provide mechanisms to effect permanent adjustment of the basic clock rate of the local clock. This adjustment is reflected in a new actual clock rate for the local clock. This may be accomplished using mechanisms either within or outside of the time management function protocol.

In order to minimize the human configuration management of local clocks, the time management function shall have an automatic mechanism to configure their local clocks to the most accurate and stable clock (reference source) within its synchronization domain. A mechanism to allow a method for a new local clock to request information on available references shall be provided. Frequent changing among reference sources shall be minimized.

NOTE 2 – A directory service mechanism may be used here.

The time management function shall compensate for the expected frequency offset of the local clock used in the local system.

The local clock shall be able to switch reference clocks in the event that:

a) the current reference fails to respond for a period of time that threatens the accuracy of the local clock; or

b) the synchronizing dispersion indicates that statistical tolerance limits have been exceeded at the current reference.

7 Model

The purpose of the time management function is to manage the resources related to the provision of quality time information in a system. In this clause, the generic functionality involved in the provision of time information is defined and the components of that functionality that are within the scope of the time management function are identified. The time-related resources in a system are identified. A model for the time management function is provided, and the clock coordination function is defined.

7.1 Generic time functionality

All of the components necessary to provide and manage time information in a system make up a set of generic time functions. The foundation of all these functions is a clock which includes a local clock and optionally external time references. These generic time functions can be organized as three basic components that interact with these clocks. These components are a clock coordination function, time management function, and time user function. Figure 1 illustrates generic time functionality.

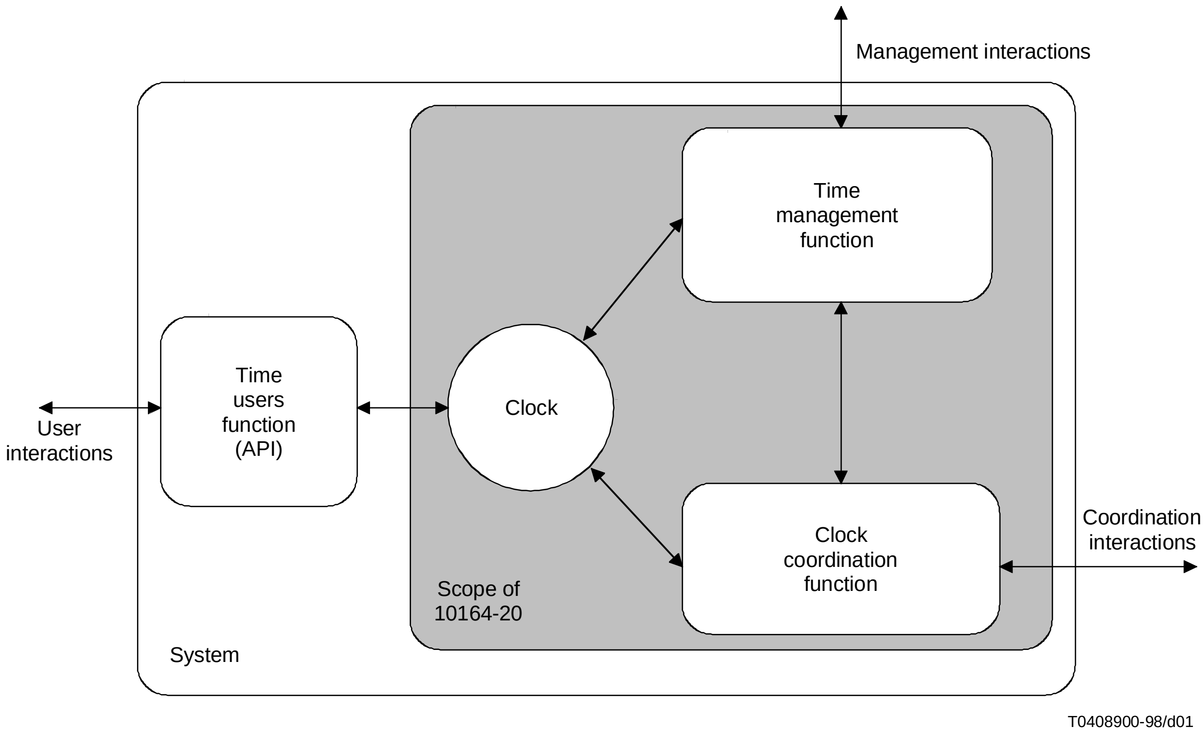


Figure 1 – Generic time functionality

The clock coordination function synchronizes individual clocks in different systems to each other and to national and international time standards. It includes the mechanisms necessary to exchange time information between individual local clocks and the algorithms required to process this information to arrive at meaningful conclusions. There may be multiple clock coordination solutions being utilized in a system.

The time management function includes the functionality necessary to monitor and control both the clocks and the clock coordination process.

Finally, the time user function of generic time functionality provides users with access to time information including the current time value and the accuracy of that value. A time user is any consumer of time values including application processes, operating systems, and OSI communication and management processes.

This Recommendation | International Standard addresses those components of generic time functionality that involve communication between systems. This includes the clock coordination and time management components defined above. Clock coordination is by its very nature a distributed algorithm; however, time management conforms to the standard manager/agent model present in other system management functions. The time user component, while very important to a local system, is considered a local issue and outside the scope of this effort.

NOTE – There are a number of issues related to the provision of a time user service in a local system. These are discussed in Annex I.

7.2 Time Management Function

This subclause identifies the resources managed by the Time Management Function and presents the model for the management of these resources.

7.2.1 Time-related resources

There are two resources related to the provision of time information to the users or consumers of that information. These resources are clocks and clock coordination tools.

Clocks can be either local clocks or external time references. A local clock is the collection of hardware and software components that comprises a single source of time information within a system. An external time reference is an interface located within a system that provides access to a specialized external clock with specified parameters and a relationship to national or international time standards.

Clock coordination is the collection of protocol mechanisms, procedures, and algorithms that are used to exchange time information between individual clocks and to process that information to provide for the coordination of the same clocks. Generally, this coordination takes place between local clocks in different systems using a clock coordination protocol. Clock coordination also takes place between local clocks and external references within a system. This coordination may be through either a clock coordination protocol or local means and is outside the scope of this Recommendation | International Standard.

The time management function is primarily concerned with managing two types of time-related resources, clocks and synchronization protocols. To this end, the time management function defines two classes and two subclasses of managed objects and the functions that pertain to the management of those objects. These objects include those that model time sources or clocks and those that model the clock coordination process. This relationship is shown in Figure 2.

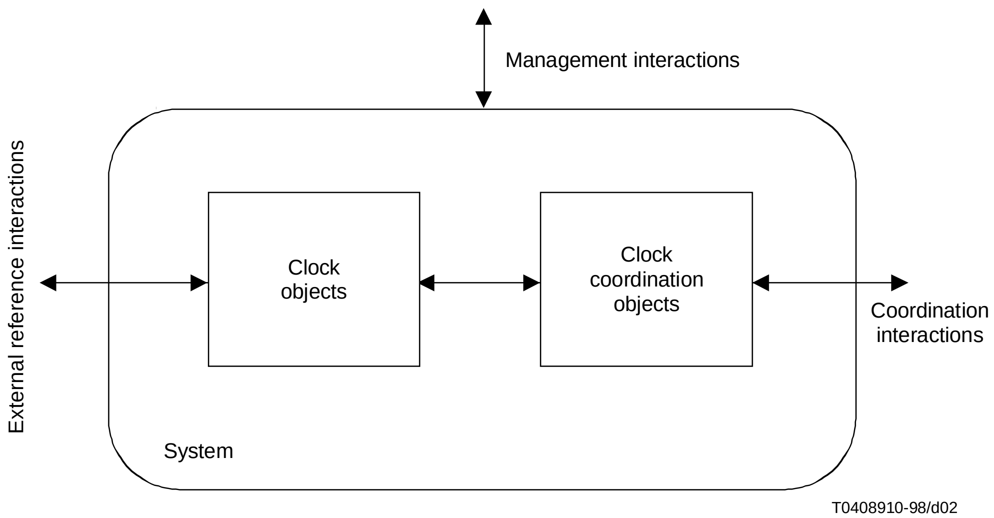


Figure 2 – Time-related resources

7.2.2 Time management functions

The following functions related to time management are identified. Of these, the first three are related to clock objects and the last four are related to clock coordination objects:

- Get Clock Status;
- Modify Clock parameters;
- Reset Clock;
- Distribute Leap Second Warning;
- Get Protocol Status (overall and per association);
- Modify Protocol Machine (add/remove peer, change polling interval, change required accuracy);
- Start/Stop Protocol Machine.

7.2.3 Time management function managed objects

The time management function defines four managed objects:

1) the clockSource object;

2) the localClock object;

3) the referenceClock object; and

4) the synchronizationProtocol object.

The first three objects are used to model time sources or clocks while the latter models the clock coordination process. The synchronizationProtocol object class can be specialized (i.e. subclassed) to represent specific time synchronization protocols. For example, Annex B specifies a subclass for the Network Time Protocol (NTP). The specification of further subclasses for additional time synchronization protocols is for further study. Figure 3 illustrates the inheritance hierarchy of the TMF, and Figure 4 shows the name bindings.

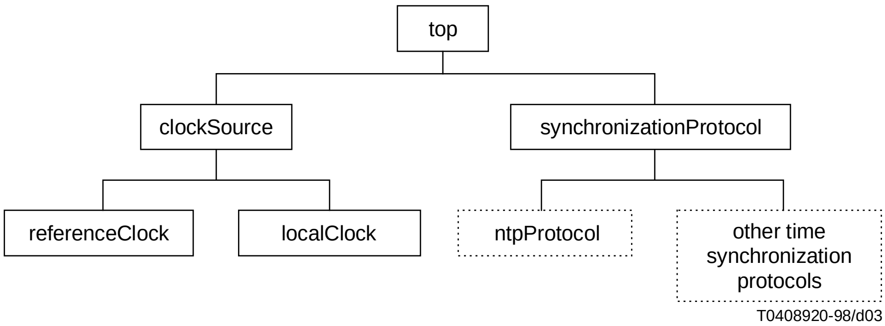


Figure 3 – Time management function inheritance hierarchy

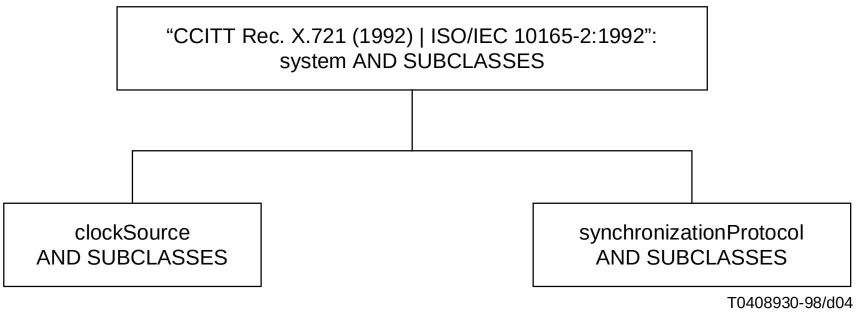


Figure 4 – TMF name bindings

7.2.4 The clockSource managed object

The clockSource object models the dynamic state of a clock. Two subclasses are defined to further distinguish between internal system clocks and external reference clocks. The clockSource object must contain the following attributes:

– the identity of the clock;

– the operational state;

– the clock status;

– the clock value;

– the clock event counter;

– the clock event code; and

– clock event time.

The clockSource object may contain the following attributes:

- precision;
- clock drift;
- the maximum error;
- the estimated error;
- the time, date and form (insert/delete) of the next leap second; and
- the time, type and total count of leap second clock events.

The clockSource object provides an action to:

- reset the clock.

The clockSource object provides a notification for:

- state change.

7.2.4.1 The localClock managed object

Each source of time information in a real system is considered to be a local clock. A local clock is conceptually the hardware and software that constitutes the source of time information in the system. A functioning local clock is one in which the maximum frequency error of the clock does not exceed the manufacturer's tolerance specified for that clock.

The above statement requires very little of a clock. It asserts only that a clock must run at about one second per second. This statement makes no assertion about the current time or the clock's correctness. It also makes no assertion about predicting the future rate of the clock based on its previous behaviour. As a statement about the properties of the local clock, this statement is manifestly independent of the network and the network's properties.

The local clock object is the model of the local clock resource used in the time management function. This object is derived from the clockSource object and contains the following attributes in addition to those specified in the clockSource object:

- the network address of the clock;
- the network addresses of its peer clocks;
- the adjustment interval of the clock;
- the current synchronization source for this clock;
- the maximum error acceptable for this clock;
- the clock stratum.

7.2.4.2 The referenceClock managed object

The reference clock object provides a mechanism for modeling the interfaces to unique sources of external time. This include interfaces to such time sources as Global Positioning System (GPS), radio sources (WWV), and atomic oscillators (cesium clock standards). It is expected that there would only be a few of these in any particular synchronization domain. The referenceClock object is derived from the clockSource object and contains the following attribute in addition to those specified in the clockSource object:

- the source/type of the external time.

7.2.5 The synchronizationProtocol managed object

A synchronizationProtocol object represents an individual instantiation of a protocol used to exchange time information between various local clocks. The synchronizationProtocol object can be used (e.g. as in subclasses) to represent different time synchronization protocols. The synchronizationProtocol object includes attributes to indicate:

- the identify of the time synchronization protocol;
- the type of the time synchronization protocol;
- the local clock(s) currently being coordinated;
- the list of other clocks with which time information has been exchanged.

The synchronizationProtocol object provides actions to:

- distribute leap second indications;
- reset the coordination protocol.

The synchronizationProtocol object class can be specialized (i.e. subclassed) to represent specific time synchronization protocols. For example, Annex B specifies a subclass for the Network Time Protocol (NTP). The specification of further subclasses for additional time synchronization protocols is for further study.

7.3 Clock coordination function

The clock coordination function provides for the coordination of clocks for the purposes of time synchronization. The clock coordination function represents one of the resources being managed by the time management function. Because of the lack of an appropriate Recommendation | International Standard defined elsewhere, it has been decided to define one solution for this function within the scope of this Recommendation | International Standard.

Different clock coordination functions exist and may be used. Multiple clock coordination functions can exist in a single system. The interaction between these various clock coordination functions is outside the scope of this Recommendation | International Standard. Additionally, local clock coordination functions between local clocks and external references within a single system are outside the scope of this Recommendation | International Standard. For completeness, one clock coordination function will be defined as part of this Recommendation | International Standard.

For the purposes of modeling, clock coordination is divided into two components, the time synchronization protocol and the time synchronization procedures. The time synchronization protocol includes the mechanism used to exchange time information between clocks in a synchronization domain. The time synchronization procedures component incorporates the procedures and algorithms required to process and act on this information locally for the purposes of clock coordination. An overview of these procedures is given in the following subclauses.

7.3.1 Time synchronization protocol

The time synchronization protocol is used to exchange time information between systems for the purposes of synchronization. There are a number of current time synchronization protocols identified and discussed in Annex H. The time management function will enable the management of these protocols. In addition, this function will define a time synchronization protocol that addresses the requirements identified in clause 6. The following procedures are identified for the time synchronization protocol.

7.3.1.1 Time inquiry procedure

The **time inquiry procedure** provides a mechanism to obtain a time value. The time inquiry procedure is abstractly presented as a remote procedure call. A time inquiry procedure is the mechanism by which the local clock obtains time information from other entities. Abstractly, it consists of a remote procedure call to remote local clocks on other real systems. The remote local clock returns the following information regarding itself: the time and maximum error, a warning concerning the time of occurrence of the next leap second (as available from a national means of dissemination), and a minimum bound on the time delay associated with processing the request at that system.

7.3.1.2 Time transmit procedure

The **time transmit procedure** provides an optional mechanism to periodically broadcast the current time and maximum error. This permits light-weight local clock implementations which obtain the time by listening for this broadcast and adjusting their local clocks accordingly.

7.3.2 Procedures for time synchronization

Time synchronization procedures are used to process and make decisions with the information collected by the time synchronization protocol. The protocols described in Annex H incorporate aspects of both the time synchronization protocol and the time synchronization procedures components of clock coordination.

7.3.2.1 Time supply procedure

The **time supply procedure** provides a mechanism to provide a time value upon request. The local clock presents this time value. The local clock may maintain its maximum error as well as its time. For instance, if the local clock is implemented as a counter in memory which is incremented by the clock precision at each clock tick, a second counter (the maximum error) would be incremented by the product of the clock precision and the maximum drift (as specified by the clock manufacturer) of the clock at each tick. This maintenance of the maximum error must imply the statement that a clock which is functioning and initially correct remains correct. The two values (time value and accuracy) are consistent with each other in reference to the same point in time (i.e. they are atomic at the interface to the time service).

7.3.2.2 Time synchronization procedure

The **time synchronization procedure** is periodically invoked to compute a new time offset for the local clock. This is done by invoking the time inquiry procedure for each of the local clocks currently sharing time information with this particular local clock. After the information is gathered, a network time and maximum error are computed based on the responses. This is compared to the local clock's time, and a time offset is calculated. Based on the value of the time offset and the management policies for a particular local clock, either a time adjustment or a time update is performed.

7.3.2.3 Time adjustment procedure

The **time adjustment procedure** provides a mechanism to advance or retard the frequency of the local clock for a specified period of time. This results in an adjustment of the local clock's value by small amounts, gradually reducing the clock's error. This adjustment should use the clock adjustment rate. In this case, since the purpose of the adjustment is to reduce the error of the local clock, the maximum error should be reduced during the course of the adjustment by the magnitude of the adjustment so far completed.

7.3.2.4 Time update procedure

The **time update procedure** provides a mechanism to abruptly change the value of a clock when gradual adjustments will not suffice. This is most commonly used during initialization and when it has been determined that the local clock is faulty. It could also be used instead of the time adjustment procedure to make the standard adjustments necessary to maintain synchronization. Changing the time of the local clock requires specifying a maximum error. The update of the time and maximum error must be consistent with each error (atomic at the interface to the time service).

7.3.2.5 Next leap second procedure

The **next leap second procedure** provides a mechanism to specify the time of the next leap second and whether it is to be inserted or deleted. When the time of the next leap second is reached, the clock adjusts its time to compensate for the leap second. Whether this adjustment is a step adjustment or a gradual adjustment is currently a local matter. Systems requiring a more stable timeframe around the occurrence of a leap second will need to address this issue in a more rigorous manner.

7.3.2.6 Frequency adjustment computation procedure

The **frequency adjustment computation procedure** is periodically invoked to compute a new frequency adjustment for the local clock to be used by the frequency adjustment procedure. This is an optional procedure that examines the time offset adjustments required by previous time synchronization procedures. If the local clock exhibits an explicit systematic pattern of time offsets required, the local clock's frequency (actual clock rate) may be adjusted within the limits of the hardware and clock software.

7.3.2.7 Frequency adjustment procedure

The **frequency adjustment procedure** provides a mechanism to adjust the apparent frequency of the local clock (sometimes referred to as clock training). This is an optional procedure that changes the apparent frequency (actual clock rate) of the local clock. In this case, the mechanism must also permit modifying the specified maximum drift of the clock. The two attributes should be atomically updated.

7.4 Time user function

The time user function component provides users with access to time information including time values in various formats, the quality (accuracy, precision, etc.) of the time values, time interval counters, etc. However, this portion of generic time functionality is considered a local issue and outside the scope of the time management function.

8 Generic definitions

8.1 The representation of time

A representation of time for use by the time management function and any OSI management functions or user applications requiring this service shall include a time value, a maximum error, and an epoch. There are several additional representations of time specified by various international standards bodies.

This representation of time is two values representing the number of seconds plus the number of nanoseconds since an epoch, with a base date of 0 hours, 0 minutes, 0 seconds GMT on January 1, 1970. This representation has a precision of 1 nanosecond and a range of approximately 600 years per epoch. The occurrence of leap seconds will be noted for conversion to other time formats; however, it will not cause a discontinuity in this time representation. Additionally, the local time zone is noted for conversion to other time formats.

The representation of the maximum error of the time stamp is an integer representing a number of nanoseconds. The maximum error has a range of zero nanoseconds to approximately 3 days (281 474 976 710 654 nanoseconds). The maximum value represents the condition where there is no estimate of the error available.

The representation of the local time zone is an integer representing the number of minutes east of GMT. Values outside of the range from 780 to –780 minutes are undefined or unknown. The local time zone indicates the timezone in which the timestamp was created.

The representation of an epoch is an integer representing the approximately 600 year period (4 294 967 296 seconds) being represented (with an epoch of 0 indicating the period beginning in 1970). The epoch effectively increases the range of the representation to between approximately 74 800 BCE and AD 79 400, representing a range of around 154 000 years.

8.2 Managed object classes

8.2.1 Clock source

This object class provides information concerning the dynamic state of a clock within a system. Two subclasses are also defined to further distinguish between internal system clocks and external reference clock interfaces. An instantiation of this object is required for each manageable clock.

The clockSource object provides access to and information about a source of time within a system. The clockStatus attribute is identified as state attribute. A change in the value of the operationalState attribute or in the clockStatus attribute causes a stateChange notification to be emitted. This managed object class is a subclass of "CCITT Rec. X.721 | ISO/IEC 10165-2":top and adds the following attributes:

- clockID;
- "CCITT Rec. X.721 | ISO/IEC 10165-2": operationalState;
- clockStatus;
- clockValue;
- clockEventCounter;
- clockEventCode;
- clockEventTime.

If an instance supports more detailed clock source information, the following attributes are present:

- clockPrecision;
- clockDrift;
- clockMaximumError;
- clockEstimatedError.

If an instance supports leap second information, the following attributes are present:

- leapSecondIndication;
- leapSecondCount.

This managed object class adds the following notification:

- "CCITT Rec. X.721 | ISO/IEC 10165-2": stateChange.

This managed object class provides the following action:

- clockReset.

8.2.2 Local clock

This object class provides information concerning the dynamic state of a local clock internal to a system. It is a subclass of the managed object class clock source. The localClock object provides access to and information about an internal source of time within a system. This managed object class adds the following attributes:

- localClockAddress;
- peerClockAddresses;
- synchronizationSourceAddress;
- clockStratum;
- clockAdjustmentInterval.

8.2.3 Reference clock

This object class provides information concerning the dynamic state of a clock interface residing in a system and providing that system access to an external time reference. This is a subclass of the managed object class clock source. It adds the following attribute:

– referenceClockType.

8.2.4 Synchronization protocol

This object provides general information about clock coordination function present in a system and provides access to the basic parameters of the time synchronization protocol. It is a subclass of "CCITT Rec. X.721 | ISO/IEC 10165-2":top. It adds the following attributes:

– synchronizationProtocolID;
– synchronizationProtocolType;
– synchronizedClock;
– synchronizingClocks;

It adds the following actions:

– leapSecondAction;
– protocolResetAction.

8.3 Attribute definitions

8.3.1 Clock Adjustment Interval

This attribute specifies the interval over which gradual phase adjustments to the local clock are to be applied.

8.3.2 Clock Drift

This attribute indicates the clock manufacturer's specified value of drift.

8.3.3 Clock Estimated Error

This attribute indicates the estimated error of the clock.

8.3.4 Clock Event Code

This attribute identifies the latest clock system exception event.

8.3.5 Clock Event Counter

This attribute specifies a counter indicating the number of system exception events that have occurred since the last time the counter was checked and cleared.

8.3.6 Clock Event Time

This attribute indicates the time at which the latest system exception event occurred.

8.3.7 Clock ID

This attribute identifies the clock being modeled by the managed object.

8.3.8 Clock Maximum Error

This attribute indicates the maximum error of the clock.

8.3.9 Clock Precision

This attribute indicates the precision of the clock.

8.3.10 Clock Status

This attribute indicates the current status of the clock.

8.3.11 Clock Stratum

This attribute indicates the current stratum value for this local clock in this node.

8.3.12 Clock Value

This attribute indicates the current time of the clock.

8.3.13 Leap Second Count

This attribute specifies the cumulative number of leap seconds that have occurred since January 1, 1972.

8.3.14 Leap Second Indication

This attribute indicates that a leap second is going to occur at the end of the current day.

8.3.15 Local Clock Address

This attribute indicates the network address of this node.

8.3.16 Peer Clock Addresses

This attribute lists the network addresses of the peers currently being maintained by this node.

8.3.17 Reference Clock Type

This attribute specifies the type of reference clock or external source that this object represents.

8.3.18 Synchronization Protocol ID

This attribute identifies the synchronization protocol being modeled by the managed object.

8.3.19 Synchronization Protocol Type

This attribute identifies the synchronization protocol type being modeled.

8.3.20 Synchronization Source Address

This attribute specifies the network address or the reference clock type of the current synchronization source for this node.

8.3.21 Synchronized Clock

This attribute specifies the clock being synchronized by this instance of the time synchronization protocol.

8.3.22 Synchronizing Clocks

This attribute specifies the set of clocks exchanging information with this clock for the purposes of synchronization.

8.4 Action definitions

The set of generic action parameters and semantics defined by this Recommendation | International Standard provide the detail for the following general parameters of the M-ACTION service defined by CCITT Rec. X.710 | ISO/IEC 9595:

- action type;
- action information;
- action reply.

8.4.1 Clock reset

The clock reset action provides the capability to reset an instance of a clock source to a given value. This service uses the M-ACTION service and procedures defined in CCITT Rec. X.710 | ISO/IEC 9595.

8.4.2 Leap second

The leap second action provides the capability to distribute an indication that a leap second is about to occur. It includes a mechanism to set the appropriate parameters in the protocol. This service uses the M-ACTION service and procedures defined in CCITT Rec. X.710 | ISO/IEC 9595.

8.4.3 Protocol reset

The protocol reset action provides the capability to restart the time synchronization protocol. This service uses the M-ACTION service and procedures defined in CCITT Rec. X.710 | ISO/IEC 9595.

8.5 Name binding definitions

8.5.1 Clock Source – System

This name binding is used for naming a clock source object with respect to a system object.

8.5.2 Synchronization Protocol – System

This name binding is used for naming a synchronization protocol object with respect to a system object.

9 Service definitions

This Recommendation | International Standard defines three services; clock reset, leap second distribution, and protocol reset. These services are defined below. In addition, the use of services defined in other functions is described below.

Clock functions include:

- creation of a clock managed object;
- deletion of a clock managed object;
- modification of clock parameters;
- accessing clock status;
- reset of clock.

Clock coordination functions include:

- creation of clock coordination managed object;
- deletion of clock coordination managed object;
- modification of clock parameters;
- accessing clock coordination protocol status;
- reset of clock coordination protocol machine;
- leap second notification distribution.

9.1 PT-CREATE service

The PT-CREATE service defined in CCITT Rec. X.730 | ISO/IEC 10164-1 is used to allow one open system to request that another open system create a managed object to model either the clock or the clock coordination resources available in that system for the purposes of management. This does not create the underlying resource.

9.2 PT-DELETE service

The PT-DELETE service defined in CCITT Rec. X.730 | ISO/IEC 10164-1 is used to allow one system to request that another open system delete a managed object modeling either the clock or the clock coordination resources available in that system for the purposes of management. This does not delete the underlying resource.

9.3 PT-SET service

The PT-SET service defined in CCITT Rec X.730 | ISO/IEC 10164-1 is used to allow one open system to request that another open system change the value of settable attributes in either clock or clock coordination managed objects.

9.4 PT-GET service

The PT-GET service defined in CCITT Rec. X.730 | ISO/IEC 10164-1 may be used to retrieve any of the readable attributes of the clock or clock coordination managed objects.

9.5 State Change service

The State Change notification service defined in CCITT Rec. X.731 | ISO/IEC 10164-2 may be used to monitor the state status of the clock or clock coordination managed objects.

9.6 Clock Reset service

The Clock Reset service allows a manager to request that another open system (the managed system) reset the clock. Table 1 lists the parameters for this service.

The Clock Reset service uses the parameters defined in clause 8 in addition to the general M-ACTION service parameters defined in CCITT Rec. X.710 | ISO/IEC 9595.

Table 1 – Clock reset parameters

Parameter name	Req/Ind	Rsp/Conf
Invoke Identifier	P	P
Linked Identifier	–	P
Mode	P	–
Base object class	P	–
Base object instance	P	–
Scope	P	–
Filter	P	–
Managed object class	–	P
Managed object instance	–	P
Access Control	P	–
Synchronization	P	–
Clock reset type	M	C(=)
Clock reset info	M	–
Clock Value	M	–
Current time	–	P
Errors	–	C

9.7 Leap Second service

The Leap Second service allows a manager to request that another open system (the managed system) initiate distribution of a leap second indication. Table 2 lists the parameters for this service.

The Leap Second service uses the parameters defined in clause 8 in addition to the general M-ACTION service parameters defined in CCITT Rec. X.710 | ISO/IEC 9595.

9.8 Protocol Reset service

The Protocol Reset service allows a manager to request that another open system (the managed system) reset the time synchronization protocol. Table 3 lists the parameters for this service.

The Protocol Reset service uses the parameters defined in clause 8 in addition to the general M-ACTION service parameters defined in CCITT Rec. X.710 | ISO/IEC 9595.

Table 2 – Leap second parameters

Parameter name	Req/Ind	Rsp/Conf
Invoke Identifier	P	P
Linked Identifier	–	P
Mode	P	–
Base object class	P	–
Base object instance	P	–
Scope	P	–
Filter	P	–
Managed object class	–	P
Managed object instance	–	P
Access Control	P	–
Synchronization	P	–
Leap second type	M	C(=)
Leap second info	M	–
Leap Indication	M	–
Date of Leap	M	
Current time	–	P
Errors	–	C

Table 3 – Protocol reset parameters

Parameter name	Req/Ind	Rsp/Conf
Invoke Identifier	P	P
Linked Identifier	–	P
Mode	P	–
Base object class	P	–
Base object instance	P	–
Scope	P	–
Filter	P	–
Managed object class	–	P
Managed object instance	–	P
Access Control	P	–
Synchronization	P	–
Protocol reset type	M	C(=)
Protocol reset info	M	–
Current time	–	P
Errors	–	C

10 Functional units

Two functional units are defined in this Recommendation | International Standard for the management of time:

a) clock control functional unit;

b) clock coordination functional unit.

The clock control functional unit requires the support of the PT-CREATE, PT-DELETE, PT-SET, PT-GET, State Change, and Clock Reset services. The clock coordination control functional unit requires the support of the PT-CREATE, PT-DELETE, PT-SET, PT-GET, State Change, Leap Second and Protocol Reset services.

11 Protocol

11.1 Elements of procedure

11.1.1 Clock reset procedure

11.1.1.1 Manager role

11.1.1.1.1 Invocation

The clock reset procedure is initiated by the clock reset primitive. On receipt of a clock reset primitive, the SMAPM shall construct an MAPDU and issue a CMIS M-ACTION request service primitive with parameters derived from the clock reset primitive. The confirmed mode shall be used.

11.1.1.1.2 Receipt of response

On receipt of a CMIS M-ACTION confirm service primitive containing an MAPDU responding to a clock reset operation, the SMAPM shall issue a deliver confirmation primitive to the Clock Reset service user with parameters derived from the CMIS M-ACTION confirm service primitive, thus completing the clock reset procedure.

NOTE – The SMAPM shall ignore all errors in the received MAPDU. The Clock Reset service user may ignore such errors, or abort the association as a consequence of such errors.

11.1.1.2 Agent role

11.1.1.2.1 Receipt of request

On receipt of a CMIS M-ACTION indication service primitive containing an MAPDU requesting the Clock Reset service, the SMAPM shall, if the MAPDU is well formed, issue a clock reset indication primitive to the Clock Reset service user with parameters derived from the CMIS M-ACTION indication service primitive. Otherwise, the SMAPM shall construct an appropriate MAPDU indicating the error, and shall issue a CMIS M-ACTION response service primitive with an error parameter present.

11.1.1.2.2 Response

The SMAPM shall accept a clock reset response primitive and shall construct an MAPDU confirming the operation and issue a CMIS M-ACTION response service primitive with parameters derived from the clock reset response primitive.

11.1.2 Leap second procedure

11.1.2.1 Manager role

11.1.2.1.1 Invocation

The leap second procedures are initiated by the leap second primitive. On receipt of a leap second primitive, the SMAPM shall construct an MAPDU and issue a CMIS M-ACTION request service primitive with parameters derived from the leap second primitive. The confirmed mode shall be used.

11.1.2.1.2 Receipt of response

On receipt of a CMIS M-ACTION confirm service primitive containing an MAPDU responding to a leap second operation, the SMAPM shall issue a deliver confirmation primitive to the Leap Second service user with parameters derived from the CMIS M-ACTION confirm service primitive, thus completing the leap second procedure.

NOTE – The SMAPM shall ignore all errors in the received MAPDU. The Leap Second service user may ignore such errors, or abort the association as a consequence of such errors.

11.1.2.2 Agent role

11.1.2.2.1 Receipt of request

On receipt of a CMIS M-ACTION indication service primitive containing an MAPDU requesting the Leap Second service, the SMAPM shall, if the MAPDU is well formed, issue a leap second indication primitive to the Leap Second service user with parameters derived from the CMIS M-ACTION indication service primitive. Otherwise, the SMAPM shall construct an appropriate MAPDU indicating the error, and shall issue a CMIS M-ACTION response service primitive with an error parameter present.

11.1.2.2.2 Response

The SMAPM shall accept a leap second response primitive and shall construct an MAPDU confirming the operation and issue a CMIS M-ACTION response service primitive with parameters derived from the leap second response primitive.

11.1.3 Protocol reset procedure

11.1.3.1 Manager role

11.1.3.1.1 Invocation

The protocol reset procedures are initiated by the protocol reset primitive. On receipt of a protocol reset primitive, the SMAPM shall construct an MAPDU and issue a CMIS M-ACTION request service primitive with parameters derived from the protocol reset primitive. The confirmed mode shall be used.

11.1.3.1.2 Receipt of response

On receipt of a CMIS M-ACTION confirm service primitive containing an MAPDU responding to a protocol reset operation, the SMAPM shall issue a deliver confirmation primitive to the Protocol Reset service user with parameters derived from the CMIS M-ACTION confirm service primitive, thus completing the protocol reset procedure.

NOTE – The SMAPM shall ignore all errors in the received MAPDU. The Protocol Reset service user may ignore such errors, or abort the association as a consequence of such errors.

11.1.3.2 Agent role

11.1.3.2.1 Receipt of request

On receipt of a CMIS M-ACTION indication service primitive containing an MAPDU requesting the Protocol Reset service, the SMAPM shall, if the MAPDU is well formed, issue a protocol reset indication primitive to the Protocol Reset service user with parameters derived from the CMIS M-ACTION indication service primitive. Otherwise, the SMAPM shall construct an appropriate MAPDU indicating the error, and shall issue a CMIS M-ACTION response service primitive with an error parameter present.

11.1.3.2.2 Response

The SMAPM shall accept a protocol reset response primitive and shall construct an MAPDU confirming the operation and issue a CMIS M-ACTION response service primitive with parameters derived from the protocol reset response primitive.

11.2 Abstract syntax

11.2.1 Objects

This Recommendation | International Standard references the following support objects, the abstract syntax for which is specified in Annex A.

a) clockSource;

b) localClock;

c) referenceClock;

d) synchronizationProtocol.

11.2.2 Attributes

This Recommendation | International Standard references the following specific management attributes, the abstract syntax for which is specified in Annex A.

a) clockAdjustmentInterval;

b) clockDrift;

c) clockEstimatedError;

d) clockEventCode;

e) clockEventCounter;

f) clockEventTime;

g) clockID;

h) clockMaximumError;

i) clockPrecision;

j) clockStatus;

k) clockStratum;

l) clockValue;

m) leapSecondCount;

n) leapSecondIndication;

o) localClockAddress;

p) peerClockAddresses;

q) referenceClockType;

r) synchronizationProtocolID;

s) synchronizationProtocolType;

t) synchronizationSourceAddress;

u) synchronizedClock;

v) synchronizingClocks.

11.2.3 Actions

This Recommendation | International Standard references the following specific action types, the abstract syntax for which is specified in Annex A.

a) clockReset;

b) leapSecond;

c) protocolReset.

11.2.4 Name bindings

This Recommendation | International Standard references the following specific name bindings, the abstract syntax for which is specified in Annex A.

a) clockSource-system;

b) synchronizationProtocol-system.

11.3 Negotiation of functional units

This Recommendation | International Standard assigns the following object identifier:

{joint-iso-ccitt ms(9) function(2)part20(20) functionalUnitPackage(1)}

as a value of the ASN.1 type FunctionalUnitPackageId defined in CCITT Rec. X.701 | ISO/IEC 10040 for negotiating the following functional units:

0 clock control functional unit

1 clock coordination functional unit

where the number identifies the bit position assigned to the functional unit, and the name references the functional unit as defined in clause 10.

12 Relationships with other functions

The following function is provided by other Systems Management Functions:

– Support for security, covered by Objects and Attributes for Access Control (ISO/IEC 10164-9).

13 Conformance

Implementations claiming to conform to this Recommendation | International Standard shall comply with the conformance requirements as defined in the following subclauses.

13.1 Static conformance

The implementation shall conform to the requirements of this Recommendation | International Standard in the manager role, the agent role, or both roles. A claim of conformance to at least one role shall be made in Table D.1.

If a claim of conformance is made for support in the manager role, the implementation shall support at least one management operation or notification of the managed objects specified by this Recommendation | International Standard. The conformance requirements in the manager role for those management operations, notifications and actions are identified in Table D.3 and further tables referenced by Annex D.

If a claim of conformance is made for support in the agent role, the implementation shall support one or more instances of the managed object class and at least one clock coordination protocol specified in Table D.4 and further tables referenced by Annex D.

The implementation shall support the transfer syntax derived from the encoding rules specified in CCITT Rec. X.209 | ISO/IEC 8825 named {joint-iso-ccitt asn1(1) basicEncoding(1)} for the abstract data types referenced by the definitions for which support is claimed.

13.2 Dynamic conformance

Implementations claiming to conform to this Recommendation | International Standard shall support the elements of procedure and definitions of semantics corresponding to the definitions for which support is claimed.

13.3 Management implementation conformance statement requirements

Any MCS proforma, MICS proforma, MOCS proforma, and MRCS proforma which conforms to this Recommendation | International Standard shall be technically identical to the proformas specified in Annexes D, E, F, and G, preserving table numbering and the index numbers of items, and differing only in pagination and page headers.

The supplier of an implementation which is claimed to conform to this Recommendation | International Standard shall complete a copy of the Management Conformance Summary (MCS) provided in Annex D as part of the conformance requirements together with any other ICS proformas referenced as applicable from that MCS. A MCS, MICS, MOCS and MRCS which conforms to this Recommendation | International Standard shall:

– describe an implementation which conforms to this Recommendation | International Standard;

– have been completed in accordance with the instructions for completion given in ITU-T Rec.X.724 | ISO/IEC 10165-6;

– include the information necessary to uniquely identify both the supplier and the implementation.

Annex A

Definition of Time Management Information

(This annex forms an integral part of this Recommendation | International Standard)

-- <GDMO.Document "ITU-T Rec. X.743 | ISO/IEC 10164-20:1998" --
-- {joint-iso-ccitt ms(9) function(2) part20(20)}> --
-- <GDMO.Version 1.3 "ITU-T Rec. X.743 | ISO/IEC 10164-20:1998"> --

A.1 Managed object classes

A.1.1 clockSource

This object class provides information concerning the dynamic state of a clock within a system. Two subclasses are also defined to further distinguish between internal system clocks and external reference clock interfaces. An instantiation of this object is required for each manageable clock.

```
clockSource MANAGED OBJECT CLASS
        DERIVED FROM "CCITT Rec. X.721 | ISO/IEC 10165-2":top;
        CHARACTERIZED BY
            clockSourcePkg PACKAGE
            BEHAVIOUR clockSourceBeh BEHAVIOUR
                DEFINED AS
                "The clockSource object provides access to and information about a source of time within a system. Clock
                status attribute is identified as state attribute. A change in the value of the operationalState attribute
                causes a stateChange notification to be emitted. ";;
            ATTRIBUTES
                clockID GET SET-BY-CREATE NO-MODIFY,
                "Rec. CCITT X.721 | ISO/IEC 10165-2": operationalState      GET NO-MODIFY,
                clockStatus GET,
                clockValue GET,
                clockEventCounter GET,
                clockEventCode GET,
                clockEventTime GET;
            ACTIONS
                clockReset ;
            NOTIFICATIONS
                "CCITT Rec. X.721 | ISO/IEC 10165-2": stateChange;;;
        CONDITIONAL PACKAGES
            clockSourceDetailPkg PACKAGE
            BEHAVIOUR clockSourceDetailBeh BEHAVIOUR
                DEFINED AS
                "The clockSourceDetailPkg package provides detailed information about a source of time within a system.
                ";;
            ATTRIBUTES
                clockPrecision GET,
                clockDrift GET,
                clockMaximumError GET,
                clockEstimatedError GET;
            REGISTERED AS {TimeMF.clockSourceDetailPkgOID};
            PRESENT IF !an instance supports it.!,
            leapSecondPkg PACKAGE
            BEHAVIOUR leapSecondBeh BEHAVIOUR
                DEFINED AS
                "The leapSecondPkg package provides access to and information about the leap seconds of a source of
                time within a system. ";;
            ATTRIBUTES
                leapSecondIndication GET-REPLACE SET-BY-CREATE,
                leapSecondCount GET-REPLACE SET-BY-CREATE;
            REGISTERED AS {TimeMF.leapSecondPkgOID};
            PRESENT IF !an instance supports it.!;

REGISTERED AS {TimeMF.clockSourceOID};
```

A.1.2 localClock

This object class provides information concerning the dynamic state of a local clock internal to a system.

```
localClock MANAGED OBJECT CLASS
        DERIVED FROM clockSource;
        CHARACTERIZED BY
            localClockPkg PACKAGE
            BEHAVIOUR localClockBeh BEHAVIOUR
                DEFINED AS
                "The localClock object provides access to and information about an internal source of time within a
                system.";;
            ATTRIBUTES
                localClockAddress GET,
                peerClockAddresses GET-REPLACE ADD-REMOVE SET-BY-CREATE,
                synchronizationSourceAddress GET,
                clockStratum GET,
                clockAdjustmentInterval GET-REPLACE SET-BY-CREATE;
            ;;
REGISTERED AS {TimeMF.localClockOID};
```

A.1.3 referenceClock

This object class provides information concerning the dynamic state of a clock interface residing in a system and providing that system access to an external time reference.

```
referenceClock MANAGED OBJECT CLASS
        DERIVED FROM clockSource;
        CHARACTERIZED BY
            referenceClockPkg PACKAGE
            BEHAVIOUR referenceClockBeh BEHAVIOUR
                DEFINED AS
                "The referenceClock object provides access to and information about a source of external time
                information within a system.";;
            ATTRIBUTES
                referenceClockType GET;;;
REGISTERED AS {TimeMF.referenceClockOID};
```

A.1.4 synchronizationProtocol

This object provides access to the basic parameters of the time synchronization protocol.

```
synchronizationProtocol MANAGED OBJECT CLASS
        DERIVED FROM "CCITT Rec. X.721 | ISO/IEC 10165-2":top;
        CHARACTERIZED BY
            synchronizationProtocolPkg PACKAGE
            BEHAVIOUR synchronizationProtocolBeh BEHAVIOUR
                DEFINED AS
                "The synchronizationProtocol object provides general information about clock coordination service
                present in a system.";;
            ATTRIBUTES
                synchronizationProtocolID GET SET-BY-CREATE NO-MODIFY,
                synchronizationProtocolType GET,
                synchronizedClock GET,
                synchronizingClocks GET;
            ACTIONS
                leapSecond,
                protocolReset ;;;
REGISTERED AS {TimeMF.synchronizationProtocolOID};
```

A.2 Attribute definitions

A.2.1 clockAdjustmentInterval

```
clockAdjustmentInterval ATTRIBUTE
        WITH ATTRIBUTE SYNTAX       TimeMF.AdjustmentInterval;
        MATCHES FOR EQUALITY;
        BEHAVIOUR clockAdjustmentIntervalBeh BEHAVIOUR
            DEFINED AS
            "This attribute specifies the interval over which gradual phase adjustments to the local clock are to be
            applied.";;
REGISTERED AS {TimeMF.clockAdjustmentIntervalOID};
```

A.2.2 clockDrift

```
clockDrift ATTRIBUTE
        WITH ATTRIBUTE SYNTAX       TimeMF.ClockDrift;
        MATCHES FOR EQUALITY;
        BEHAVIOUR clockDriftBeh BEHAVIOUR
            DEFINED AS
            "This attribute indicates the clock manufacturer's specified value of drift.";;
REGISTERED AS {TimeMF.clockDriftOID};
```

A.2.3 clockEstimatedError

```
clockEstimatedError ATTRIBUTE
        WITH ATTRIBUTE SYNTAX       TimeMF.ClockEstimatedError;
        MATCHES FOR EQUALITY;
        BEHAVIOUR clockEstimatedErrorBeh BEHAVIOUR
            DEFINED AS
            "This attribute indicates the estimated error of the clock.";;
REGISTERED AS {TimeMF.clockEstimatedErrorOID};
```

A.2.4 clockEventCode

```
clockEventCode ATTRIBUTE
        WITH ATTRIBUTE SYNTAX       TimeMF.ClockEventCode;
        MATCHES FOR EQUALITY;
        BEHAVIOUR clockEventCodeBeh BEHAVIOUR
            DEFINED AS
            "This attribute identifies the latest system exception event.";;
REGISTERED AS {TimeMF.clockEventCodeOID};
```

A.2.5 clockEventCounter

```
clockEventCounter ATTRIBUTE
        WITH ATTRIBUTE SYNTAX       TimeMF.ClockEventCounter;
        MATCHES FOR EQUALITY, ORDERING;
        BEHAVIOUR clockEventCounterBeh BEHAVIOUR
            DEFINED AS
            "This attribute specifies a counter indicating the number of system exception events that have occurred since
            the last time the counter was checked and cleared.";;
REGISTERED AS {TimeMF.clockEventCounterOID};
```

A.2.6 clockEventTime

```
clockEventTime ATTRIBUTE
        WITH ATTRIBUTE SYNTAX       TimeMF.ClockEventTime;
        MATCHES FOR EQUALITY;
        BEHAVIOUR clockEventTimeBeh BEHAVIOUR
            DEFINED AS
            "This attribute indicates the time at which the latest system exception event occurred.";;
REGISTERED AS {TimeMF.clockEventTimeOID};
```

A.2.7 clockID

```
clockID ATTRIBUTE
        WITH ATTRIBUTE SYNTAX     TimeMF.ClockID;
        MATCHES FOR EQUALITY;
        BEHAVIOUR clockIDBeh BEHAVIOUR
            DEFINED AS
            "This attribute identifies the clock being modeled by the managed object.";;
REGISTERED AS {TimeMF.clockIDOID};
```

A.2.8 clockMaximumError

```
clockMaximumError ATTRIBUTE
        WITH ATTRIBUTE SYNTAX     TimeMF.ClockMaximumError;
        MATCHES FOR EQUALITY;
        BEHAVIOUR clockMaximumErrorBeh BEHAVIOUR
            DEFINED AS
            "This attribute indicates the maximum error of the clock.";;
REGISTERED AS {TimeMF.clockMaximumErrorOID};
```

A.2.9 clockPrecision

```
clockPrecision ATTRIBUTE
        WITH ATTRIBUTE SYNTAX     TimeMF.Precision;
        MATCHES FOR EQUALITY;
        BEHAVIOUR clockPrecisionBeh BEHAVIOUR
            DEFINED AS
            "This attribute indicates the precision of the clock.";;
REGISTERED AS {TimeMF.clockPrecisionOID};
```

A.2.10 clockStatus

```
clockStatus ATTRIBUTE
        WITH ATTRIBUTE SYNTAX     TimeMF.Status;
        MATCHES FOR EQUALITY;
        BEHAVIOUR clockStatusBeh BEHAVIOUR
            DEFINED AS
            "This attribute indicates the current status of the clock";;
REGISTERED AS {TimeMF.clockStatusOID};
```

A.2.11 clockStratum

```
clockStratum ATTRIBUTE
        WITH ATTRIBUTE SYNTAX     TimeMF.Stratum;
        MATCHES FOR EQUALITY, ORDERING;
        BEHAVIOUR clockStratumBeh BEHAVIOUR
            DEFINED AS
            "This attribute indicates the current stratum value for this local clock in this node.";;
REGISTERED AS {TimeMF.clockStratumOID};
```

A.2.12 clockValue

```
clockValue ATTRIBUTE
        WITH ATTRIBUTE SYNTAX     TimeMF.ClockValue;
        MATCHES FOR EQUALITY;
        BEHAVIOUR clockValueBeh BEHAVIOUR
            DEFINED AS
            "This attribute indicates the current time of the clock.";;
REGISTERED AS {TimeMF.clockValueOID};
```

A.2.13 leapSecondCount

```
leapSecondCount ATTRIBUTE
        WITH ATTRIBUTE SYNTAX       TimeMF.CumLeapSeconds;
        MATCHES FOR EQUALITY, ORDERING;
        BEHAVIOUR leapSecondCountBeh BEHAVIOUR
            DEFINED AS
            "This attribute specifies the cumulative number of leap seconds that have occurred since January 1, 1972.";;

REGISTERED AS {TimeMF.leapSecondCountOID};
```

A.2.14 leapSecondIndication

```
leapSecondIndication ATTRIBUTE
        WITH ATTRIBUTE SYNTAX       TimeMF.LeapIndication;
        MATCHES FOR EQUALITY;
        BEHAVIOUR leapSecondIndicationBeh BEHAVIOUR
            DEFINED AS
            "This attribute indicates that a leap second is going to occur at the end of the current day.";;

REGISTERED AS {TimeMF.leapSecondIndicationOID};
```

A.2.15 localClockAddress

```
localClockAddress ATTRIBUTE
        WITH ATTRIBUTE SYNTAX       TimeMF.ClockAddress;
        MATCHES FOR EQUALITY;
        BEHAVIOUR localClockAddressBeh BEHAVIOUR
            DEFINED AS
            "This attribute indicates the network address of this node.";;

REGISTERED AS {TimeMF.localClockAddressOID};
```

A.2.16 peerClockAddresses

```
peerClockAddresses ATTRIBUTE
        WITH ATTRIBUTE SYNTAX       TimeMF.PeerClockAddresses;
        MATCHES FOR EQUALITY;
        BEHAVIOUR peerClockAddressesBeh BEHAVIOUR
            DEFINED AS
            "This attribute lists the network addresses of the peers currently being maintained by this node.";;

REGISTERED AS {TimeMF.peerClockAddressesOID};
```

A.2.17 referenceClockType

```
referenceClockType ATTRIBUTE
        WITH ATTRIBUTE SYNTAX       TimeMF.ReferenceClockType;
        MATCHES FOR EQUALITY;
        BEHAVIOUR referenceClockTypeBeh BEHAVIOUR
            DEFINED AS
            "This attribute specifies the type of reference clock or external source that this object represents.";;

REGISTERED AS {TimeMF.referenceClockTypeOID};
```

A.2.18 synchronizationProtocolID

```
synchronizationProtocolID ATTRIBUTE
        WITH ATTRIBUTE SYNTAX       TimeMF.SynchronizationProtocolID;
        MATCHES FOR EQUALITY;
            BEHAVIOUR syncronizationProtocolIDBeh BEHAVIOUR
            DEFINED AS
            "This attribute identifies the synchronization protocol being modeled by the managed object. This attribute is
            used for naming";;

REGISTERED AS {TimeMF.synchronizationProtocolIDOID};
```

A.2.19 synchronizationProtocolType

```
synchronizationProtocolType ATTRIBUTE
        WITH ATTRIBUTE SYNTAX       TimeMF.SynchronizationProtocolType;
        MATCHES FOR EQUALITY;
            BEHAVIOUR syncronizationProtocolTypeBeh BEHAVIOUR
            DEFINED AS
            "This attribute identifies the synchronization protocol type being modeled by the managed object.";;
REGISTERED AS {TimeMF.synchronizationProtocolTypeOID};
```

A.2.20 synchronizationSourceAddress

```
synchronizationSourceAddress ATTRIBUTE
        WITH ATTRIBUTE SYNTAX       TimeMF.CurrSynchSourceAddress;
        MATCHES FOR EQUALITY;
        BEHAVIOUR synchronizationSourceAddressBeh BEHAVIOUR
            DEFINED AS
            "This attribute specifies the network address or the reference clock type of the current synchronization source
            for this node.";;
REGISTERED AS {synchronizationSourceAddressOID};
```

A.2.21 synchronizedClock

```
synchronizedClock  ATTRIBUTE
        WITH ATTRIBUTE SYNTAX       TimeMF.SynchronizedClock;
        MATCHES FOR EQUALITY;
        BEHAVIOUR syncronizedClockBeh BEHAVIOUR
            DEFINED AS
            "The clock being synchronized by this instance of the time synchronization protocol.";;
REGISTERED AS {TimeMF.synchronizedClockOID};
```

A.2.22 synchronizingClocks

```
synchronizingClocks ATTRIBUTE
        WITH ATTRIBUTE SYNTAX       TimeMF.SynchronizingClocks;
        MATCHES FOR EQUALITY;
        BEHAVIOUR syncronizingClocksBeh  BEHAVIOUR
            DEFINED AS
            "The set of clocks exchanging information with this clock for the purposes of synchronization.";;
REGISTERED AS {TimeMF.synchronizingClocksOID};
```

A.3 Action definitions

A.3.1 clockReset

```
clockReset ACTION
        BEHAVIOUR clockResetBeh BEHAVIOUR
            DEFINED AS
                "The BEHAVIOUR of this action is undefined in this Recommendation | International Standard. It
                provides the capability to distribute an indication to all instances of the time service to restart the time
                synchronization protocol.";;
        MODE CONFIRMED;
        WITH INFORMATION SYNTAX       TimeMF.ClockResetInfo;
REGISTERED AS {TimeMF.clockResetActionOID};
```

A.3.2 leapSecond

```
leapSecond ACTION
        BEHAVIOUR leapSecondActionBeh BEHAVIOUR
            DEFINED AS
                "The BEHAVIOUR of this action is undefined in this Recommendation | International Standard. It
                provides the capability to distribute an indication that a leap second is about to occur. It includes a
                mechanism to set the appropriate parameters in the protocol.";;
        MODE CONFIRMED;
        WITH INFORMATION SYNTAX       TimeMF.LeapSecondInfo;
REGISTERED AS {TimeMF.leapSecondActionOID};
```

A.3.3 protocolReset

```
protocolReset ACTION
        BEHAVIOUR protocolResetBeh BEHAVIOUR
            DEFINED AS
                "The BEHAVIOUR of this action is undefined in this Recommendation | International Standard. It
                provides the capability to distribute an indication to all instances of the time service to restart the time
                synchronization protocol.";;
        MODE CONFIRMED;
        WITH INFORMATION SYNTAX      TimeMF.ProtocolResetInfo;

REGISTERED AS {TimeMF.protocolResetActionOID};
```

A.4 Name binding definitions

A.4.1 clockSource-system

```
clockSource-system NAME BINDING
        SUBORDINATE OBJECT CLASS clockSource AND SUBCLASSES;
        NAMED BY SUPERIOR OBJECT CLASS "CCITT Rec. X.721 | ISO/IEC 10165-2:1992": system AND
        SUBCLASSES;
        WITH ATTRIBUTE clockID;
        CREATE WITH-AUTOMATIC-INSTANCE-NAMING;
        DELETE DELETES-CONTAINED-OBJECTS;

REGISTERED AS{TimeMF.clockSource-systemOID};
```

A.4.2 synchronizationProtocol-system

```
synchronizationProtocol-system NAME BINDING
        SUBORDINATE OBJECT CLASS synchronizationProtocol AND SUBCLASSES;
        NAMED BY SUPERIOR OBJECT CLASS "CCITT Rec. X.721 | ISO/IEC 10165-2:1992": system AND
        SUBCLASSES;
        WITH ATTRIBUTE synchronizationProtocolID;
        CREATE WITH-AUTOMATIC-INSTANCE-NAMING;
        DELETE DELETES-CONTAINED-OBJECTS;

REGISTERED AS{TimeMF.synchronizationProtocol-systemOID};
```

A.5 ASN.1 definition module for management information

```
-- <ASN1.Version 1990,1994 TimeMF --
-- {joint-iso-ccitt ms(9) function(2) part20(20) asn1Module(2) timeMF(1)}> --

TimeMF {joint-iso-ccitt ms(9) function(2) part20(20) asn1Module(2) timeMF(1)}

DEFINITIONS IMPLICIT TAGS ::=   BEGIN

-- EXPORTS everything --

IMPORTS

Attribute, ObjectInstance

FROM

CMIP-1 {joint-iso-ccitt ms(9) cmip(1) modules(0) protocol(3)}

SimpleNameType

FROM

Attribute-ASN1Module {joint-iso-ccitt ms(9) smi(3) part2(2) asn1Module(2) 1};

-- object identifier values --

timeManagement OBJECT IDENTIFIER ::= { joint-iso-ccitt ms(9) function(2) part20(20)}

clockSourceOID OBJECT IDENTIFIER ::= {timeManagement managedObjectClass(3) clockSource(0)}

localClockOID OBJECT IDENTIFIER ::= {timeManagement managedObjectClass(3) localClock(1)}

referenceClockOID OBJECT IDENTIFIER ::= {timeManagement managedObjectClass(3) referenceClock(2)}
```

synchronizationProtocolOID OBJECT IDENTIFIER ::= {timeManagement managedObjectClass(3) synchronizationProtocol(3)}

ntpProtocolOID OBJECT IDENTIFIER ::= {timeManagement managedObjectClass(3) ntpProtocol(4)}

clockSourceDetailPkgOID OBJECT IDENTIFIER ::= {timeManagement package(4) clockSourceDetailPkg(0)}

leapSecondPkgOID OBJECT IDENTIFIER ::= {timeManagement package(4) leapSecondPkg(1)}

clockAdjustmentIntervalOID OBJECT IDENTIFIER ::= {timeManagement attribute(7) clockAdjustmentInterval(0)}

clockDriftOID OBJECT IDENTIFIER ::= {timeManagement attribute(7) clockDrift(1)}

clockEstimatedErrorOID OBJECT IDENTIFIER ::= {timeManagement attribute(7) clockEstimatedError(2)}

clockEventCodeOID OBJECT IDENTIFIER ::= {timeManagement attribute(7) clockEventCode(3)}

clockEventCounterOID OBJECT IDENTIFIER ::= {timeManagement attribute(7) clockEventCounter(4)}

clockEventTimeOID OBJECT IDENTIFIER ::= {timeManagement attribute(7) clockEventTime(5)}

clockIDOID OBJECT IDENTIFIER ::= {timeManagement attribute(7) clockID(6)}

clockMaximumErrorOID OBJECT IDENTIFIER ::= {timeManagement attribute(7) clockMaximumError(7)}

clockPrecisionOID OBJECT IDENTIFIER ::= {timeManagement attribute(7) clockPrecision(8)}

clockStatusOID OBJECT IDENTIFIER ::= {timeManagement attribute(7) clockStatus(9)}

clockStratumOID OBJECT IDENTIFIER ::= {timeManagement attribute(7) clockStratum(10)}

clockValueOID OBJECT IDENTIFIER ::= {timeManagement attribute(7) clockValue(11)}

filterSizeOID OBJECT IDENTIFIER ::= {timeManagement attribute(7) filterSize(12)}

filterWeightOID OBJECT IDENTIFIER ::= {timeManagement attribute(7) filterWeight(13)}

leapSecondCountOID OBJECT IDENTIFIER ::= {timeManagement attribute(7) leapSecondCount(14)}

leapSecondIndicationOID OBJECT IDENTIFIER ::= {timeManagement attribute(7) leapSecondIndication(15)}

localClockAddressOID OBJECT IDENTIFIER ::= {timeManagement attribute(7) localClockAddress(16)}

maximumClockAgeOID OBJECT IDENTIFIER ::= {timeManagement attribute(7) maximumClockAge(17)}

maximumDispersionOID OBJECT IDENTIFIER ::= {timeManagement attribute(7) maximumDispersion(18)}

maximumDistanceOID OBJECT IDENTIFIER ::= {timeManagement attribute(7) maximumDistance(19)}

maximumPollIntervalOID OBJECT IDENTIFIER ::= {timeManagement attribute(7) maximumPollInterval(20)}

maximumSelectClockOID OBJECT IDENTIFIER ::= {timeManagement attribute(7) maximumSelectClock(21)}

maximumSkewOID OBJECT IDENTIFIER ::= {timeManagement attribute(7) maximumSkew(22)}

maximumStratumOID OBJECT IDENTIFIER ::= {timeManagement attribute(7) maximumStratum(23)}

minimumDispersionOID OBJECT IDENTIFIER ::= {timeManagement attribute(7) minimumDispersion(24)}

minimumPollIntervalOID OBJECT IDENTIFIER ::= {timeManagement attribute(7) minimumPollInterval(25)}

minimumSelectClockOID OBJECT IDENTIFIER ::= {timeManagement attribute(7) minimumSelectClock(26)}

peerClockAddressesOID OBJECT IDENTIFIER ::= {timeManagement attribute(7) peerClockAddresses(27)}

reachabilityRegisterSizeOID OBJECT IDENTIFIER ::= {timeManagement attribute(7) reachabilityRegisterSize(28)}

referenceClockTypeOID OBJECT IDENTIFIER ::= {timeManagement attribute(7) referenceClockType(29)}

selectWeightOID OBJECT IDENTIFIER ::= {timeManagement attribute(7) selectWeight(30)}

synchronizationProtocolIDOID OBJECT IDENTIFIER ::= {timeManagement attribute(7) synchronizationProtocolID(31)}

synchronizationProtocolTypeOID OBJECT IDENTIFIER ::= {timeManagement attribute(7) synchronizationProtocolType(32)}

synchronizationSourceAddressOID OBJECT IDENTIFIER ::= {timeManagement attribute(7) synchronizationSourceAddress(33)}

synchronizedClockOID OBJECT IDENTIFIER ::= {timeManagement attribute(7) synchronizedClock(34)}

synchronizingClocksOID OBJECT IDENTIFIER ::= {timeManagement attribute(7) synchronizingClocks(35)}

```
clockResetActionOID OBJECT IDENTIFIER ::= {timeManagement action(9) clockResetAction(1)}

leapSecondActionOID OBJECT IDENTIFIER ::= {timeManagement action(9) leapSecondAction(2)}

protocolResetActionOID OBJECT IDENTIFIER ::= {timeManagement action(9) protocolResetAction(3)}

synchronizationProtocol-systemOID OBJECT IDENTIFIER ::= {timeManagement nameBinding(6) synchronizationProtocol-
system(1)}

clockSource-systemOID OBJECT IDENTIFIER ::= {timeManagement nameBinding(6) clockSource-system(2)}

ntp SynchronizationProtocolType ::= { joint-iso-ccitt ms(9) function(2) part20(20) synchProtocolType(20) ntp(1) }

-- type references --

AdjustmentInterval ::= TimeInterval
ClockAddress ::= CHOICE {
                isoNsap [1] OCTET STRING (SIZE (0 | 3..20)),
                ip [2] SEQUENCE {
                    host OCTET STRING (SIZE(4)),
                    port INTEGER (0..65536)
                    }
                }

ClockDrift ::=  REAL

ClockEstimatedError ::=  TimeInterval

ClockEventCode ::= INTEGER {
                unspecified                 (0),
                restart                     (1),
                systemOrHardwareFault       (2),
                newStatusWord               (3),
                newSynchSourceOrStratum     (4),
                systemClockReset            (5),
                systemInvalidTimeOrDate     (6),
                systemClockException        (7),
                reserved8                   (8),
                reserved9                   (9),
                reserved10                  (10),
                reserved11                  (11),
                reserved12                  (12),
                reserved13                  (13),
                reserved14                  (14),
                reserved15                  (15)
                }

ClockEventCounter ::= INTEGER (0 .. 255)

ClockEventTime ::= GlobalTime

ClockID ::= SimpleNameType

ClockMaximumError ::=  TimeInterval

ClockValue ::= GlobalTime

CumLeapSeconds ::= INTEGER (0 .. 255)

ClockResetInfo ::= ClockValue

CurrSynchSourceAddress ::= CHOICE {
                refPeerAssoc [0] ClockAddress,
                refClockID [1] ReferenceClockType
                }

DateOfLeap ::=  GeneralizedTime

Dispersion ::= TimeInterval

                -- This field represents the dispersion (positive values only). --

FilterSize ::= INTEGER (0 .. 32)

FilterWeight ::= REAL (0 .. {mantissa 1, base 10, exponent 0})

GlobalTime ::= OCTET STRING (SIZE (8))   -- See 8.1.--
```

```
LeapIndication ::= ENUMERATED {
                    noWarning               (0),
                    minuteHas61Seconds      (1),
                    minuteHas59Seconds      (2),
                    alarmCondition          (3) }

LeapSecondInfo ::= SEQUENCE {
                    leapIndication   LeapIndication,
                    dateOfLeap    DateOfLeap
                    }

MaxAperature ::= TimeInterval

MaxClockAge ::= TimeInterval

MaxDistance ::= TimeInterval

MaxSkew ::= TimeInterval

PeerClockAddresses ::= SET OF SinglePeerClock

PollInterval ::= INTEGER (0..MAX)

                    -- This field represents the polling interval in seconds and can only contain positive values. --

Precision ::= TimeInterval

                    -- This field represents precision and can only contain positive values. --

ProtocolResetInfo ::=  SET OF Attribute

ReachRegSize ::= INTEGER (0 .. 32)

ReferenceClockType ::= INTEGER {
                    unspecifiedOrUnknown   (0),
                    calibratedAtomicClock  (1),
                    radioVLForLF           (2),
                    radioHF                (3),
                    radioUHF               (4),
                    localNet               (5),
                    synch                  (6),
                    wallclock              (7),
                    telephoneModem         (8),
                    gps                    (9),
                    loranC                 (10),
                    other                  (11)
                    }

SelectClock ::= INTEGER (0 .. 255)

SelectWeight ::= REAL (0 .. {mantissa 1, base 10, exponent 0})

SinglePeerClock ::= SEQUENCE {
                    assocNum [0] INTEGER,
                    assocClock [1] ClockAddress
                    }

Stratum ::= INTEGER (0..255)

-- A value of zero means that the stratum is not specified. --
-- A value of one indicates a primary reference. --
-- Values from 2 to 255 indicate secondary references of increasing --
-- distance from the root of the synchronization subnet .--

Status ::= INTEGER {
                    operatingWithinNominals    (0),
                    replyTimeout               (1),
                    badReplyFormat             (2),
                    hardwareSoftwareFault      (3),
                    propagationFailure         (4),
                    badDateFormatOrValue       (5),
                    badTimeFormatOrValue       (6)
                    }
SynchronizationProtocolID ::= SimpleNameType
SynchronizationProtocolType ::= OBJECT IDENTIFIER
```

```
SynchronizedClock ::= ObjectInstance

SynchronizingClocks ::= SET OF ObjectInstance

TimeInterval ::= OCTET STRING (SIZE (8))  -- See 8.1. --

TSelect ::= OCTET STRING (SIZE (4))

END -- End of syntax definitions --
```

A.6 ASN.1 definition module for time representation

```
-- <ASN1.Version 1990,1994 TimeRepresentation --
-- {joint-iso-ccitt ms(9) function(2) part20(20) asn1Module(2) --
-- timeRepresentation(2) }> --

TimeRepresentation {joint-iso-ccitt ms(9) function(2) part20(20) asn1Module(2) timeRepresentation(2)}

DEFINITIONS::= BEGIN
        Epochs::= INTEGER (-128 .. 127)
        Seconds::= INTEGER (0 .. 4294967295)
        Nanoseconds::= INTEGER (0 .. 999999999)
        MaximumErrorInNanoseconds::= INTEGER {noEstimate (281474976710655)}
            (0 .. 281474976710654)
        CumLeapSeconds::= INTEGER (0 .. 65536)
        TimeZone::= INTEGER {unknown (781)} (-780 .. 781)
            -- Represents minutes east of GMT.--
        TimeStamp::= SEQUENCE {
            epoch           Epochs,
            second          Seconds,
            nanosecond      Nanoseconds,
            maximumError    MaximumErrorInNanoseconds
            }

        ClockTime::= SEQUENCE {
            time            TimeStamp,
            leapSeconds     CumLeapSeconds,
            localTimeZone   TimeZone
            }

        TimeInterval::= SEQUENCE {
            epochs          Epochs,
            seconds         Seconds,
            nanoseconds     Nanoseconds
            }

        TimeDifference::= SEQUENCE {
            sign  ENUMERATED {positive (0), negative (1)},
            epochs          Epochs,
            seconds         Seconds,
            nanoseconds     Nanoseconds,
            maximumError    MaximumErrorInNanoseconds
            }

END
```

Annex B

The Network Time Protocol and Time Management Information

(This annex forms an integral part of this Recommendation | International Standard)

B.1 The Network Time Protocol

The Network Time Protocol may be implemented as the time synchronization service underlying this Time Management Function. If so, it is implemented in accordance with RFC 1305 (see [5] in H.4). The NTP managed object class is intended to be used with the Network Time Protocol time synchronization service.

B.2 The ntpProtocol managed object class definition

The ntpProtocol object provides access to the basic parameters of the Network Time Protocol (NTP) time synchronization protocol. It is a subclass of the synchronization protocol managed object class. The basic parameters of NTP include both the protocol to exchange time information and the procedures and algorithms used to process and act on the time information gathered. The ntpProtocol object includes attributes to indicate:

- the current state of the time synchronization protocol (polling intervals, modes of service, etc.) (specific to particular subclass);
- the offset, delay, maximum error (and other relevant data) associated with each clock with which time information has been exchanged.

```
ntpProtocol MANAGED OBJECT CLASS
        DERIVED FROM synchronizationProtocol;
        CHARACTERIZED BY ntpProtocolPkg PACKAGE
            BEHAVIOUR ntpProtocolBeh BEHAVIOUR
                DEFINED AS
                "This object provides general information about the Network Time Protocol (ntp) time synchronization
                protocol.";;
            ATTRIBUTES
                maximumStratum GET,
                maximumClockAge GET,
                maximumSkew GET,
                maximumDistance GET,
                minimumPollInterval GET,
                maximumPollInterval GET,
                minimumSelectClock GET,
                maximumSelectClock GET,
                minimumDispersion GET,
                maximumDispersion GET,
                reachabilityRegisterSize GET,
                filterSize GET,
                filterWeight GET,
                selectWeight GET;;;
REGISTERED AS {TimeMF.ntpProtocolOID};
```

B.3 Attribute definitions

B.3.1 filterSize

```
filterSize ATTRIBUTE
        WITH ATTRIBUTE SYNTAX       TimeMF.FilterSize;
        MATCHES FOR EQUALITY, ORDERING;
        BEHAVIOUR filterSizeBeh BEHAVIOUR
            DEFINED AS
            "This attribute specifies the size of the clock filter shift register.";;
REGISTERED AS {TimeMF.filterSizeOID};
```

B.3.2 filterWeight

```
filterWeight ATTRIBUTE
        WITH ATTRIBUTE SYNTAX       TimeMF.FilterWeight;
        MATCHES FOR EQUALITY;
        BEHAVIOUR filterWeightBeh BEHAVIOUR
            DEFINED AS
            "This attribute specifies the weight used to compute the filter dispersion.";;
REGISTERED AS {TimeMF.filterWeightOID};
```

B.3.3 maximumClockAge

```
maximumClockAge ATTRIBUTE
        WITH ATTRIBUTE SYNTAX       TimeMF.MaxClockAge;
        MATCHES FOR EQUALITY;
        BEHAVIOUR maximumClockAgeBeh BEHAVIOUR
            DEFINED AS
            "This attribute specifies the maximum interval without an update that a reference clock will be considered
valid.";;
REGISTERED AS {TimeMF.maximumClockAgeOID};
```

B.3.4 maximumDispersion

```
maximumDispersion ATTRIBUTE
        WITH ATTRIBUTE SYNTAX       TimeMF.Dispersion;
        MATCHES FOR EQUALITY;
        BEHAVIOUR maximumDispersionBeh BEHAVIOUR
            DEFINED AS
            "This attribute specifies the maximum dispersion increment allowable, also specifies the dispersion assumed for
missing data.";;
REGISTERED AS {TimeMF.maximumDispersionOID};
```

B.3.5 maximumDistance

```
maximumDistance ATTRIBUTE
        WITH ATTRIBUTE SYNTAX       TimeMF.MaxDistance;
        MATCHES FOR EQUALITY;
        BEHAVIOUR maximumDistanceBeh BEHAVIOUR
            DEFINED AS
            "This attribute specifies the maximum acceptable synchronization distance.";;
REGISTERED AS {TimeMF.maximumDistanceOID};
```

B.3.6 maximumPollInterval

```
maximumPollInterval ATTRIBUTE
        WITH ATTRIBUTE SYNTAX       TimeMF.PollInterval;
        MATCHES FOR EQUALITY, ORDERING;
        BEHAVIOUR maximumPollIntervalBeh BEHAVIOUR
            DEFINED AS
            "This attribute specifies the maximum polling interval allowable in the system.";;
REGISTERED AS {TimeMF.maximumPollIntervalOID};
```

B.3.7 maximumSelectClock

```
maximumSelectClock ATTRIBUTE
        WITH ATTRIBUTE SYNTAX       TimeMF.SelectClock;
        MATCHES FOR EQUALITY, ORDERING;
        BEHAVIOUR maximumSelectClockBeh BEHAVIOUR
            DEFINED AS
            "This attribute specifies the maximum number of peers considered for selection.";;
REGISTERED AS {TimeMF.maximumSelectClockOID};
```

B.3.8 maximumSkew

```
maximumSkew ATTRIBUTE
	WITH ATTRIBUTE SYNTAX	TimeMF.MaxSkew;
	MATCHES FOR EQUALITY;
	BEHAVIOUR maximumSkewBeh BEHAVIOUR
		DEFINED AS
		"This attribute specifies the maximum offset error caused by the skew of a local clock over the interval specified by maximumClockAge.";;
REGISTERED AS {TimeMF.maximumSkewOID};
```

B.3.9 maximumStratum

```
maximumStratum ATTRIBUTE
	WITH ATTRIBUTE SYNTAX	TimeMF.Stratum;
	MATCHES FOR EQUALITY, ORDERING;
	BEHAVIOUR maximumStratumBeh BEHAVIOUR
		DEFINED AS
		"This attribute specifies the maximum stratum value that can be encoded as a packet variable, also interpreted as network unreachable.";;
REGISTERED AS {TimeMF.maximumStratumOID};
```

B.3.10 minimumDispersion

```
minimumDispersion ATTRIBUTE
	WITH ATTRIBUTE SYNTAX	TimeMF.Dispersion;
	MATCHES FOR EQUALITY;
	BEHAVIOUR minimumDispersionBeh BEHAVIOUR
		DEFINED AS
		"This attribute specifies the minimum dispersion increment for each stratum level.";;
REGISTERED AS {TimeMF.minimumDispersionOID};
```

B.3.11 minimumPollInterval

```
minimumPollInterval ATTRIBUTE
	WITH ATTRIBUTE SYNTAX	TimeMF.PollInterval;
	MATCHES FOR EQUALITY, ORDERING;
	BEHAVIOUR minimumPollIntervalBeh BEHAVIOUR
		DEFINED AS
		"This attribute specifies the minimum polling interval allowable in the system.";;
REGISTERED AS {TimeMF.minimumPollIntervalOID};
```

B.3.12 minimumSelectClock

```
minimumSelectClock ATTRIBUTE
	WITH ATTRIBUTE SYNTAX	TimeMF.SelectClock;
	MATCHES FOR EQUALITY, ORDERING;
	BEHAVIOUR minimumSelectClockBeh BEHAVIOUR
		DEFINED AS
		"This attribute specifies the minimum number of peers acceptable for synchronization.";;
REGISTERED AS {TimeMF.minimumSelectClockOID};
```

B.3.13 reachabilityRegisterSize

```
reachabilityRegisterSize ATTRIBUTE
	WITH ATTRIBUTE SYNTAX	TimeMF.ReachRegSize;
	MATCHES FOR EQUALITY, ORDERING;
	BEHAVIOUR reachabilityRegisterSizeBeh BEHAVIOUR
		DEFINED AS
		"This attribute specifies the size of the reachability register.";;
REGISTERED AS {TimeMF.reachabilityRegisterSizeOID};
```

B.3.14 selectWeight

```
selectWeight ATTRIBUTE
        WITH ATTRIBUTE SYNTAX       TimeMF.SelectWeight;
        MATCHES FOR EQUALITY;
        BEHAVIOUR selectWeightBeh BEHAVIOUR
            DEFINED AS
            "This attribute specifies the weight used to compute the selection dispersion.";;
REGISTERED AS {TimeMF.selectWeightOID};
```

--<GDMO.EndDocument "ITU-T Rec. X.743 | ISO/IEC 10164-20:1997"
-- {joint-iso-ccitt ms(9) function(2) part20(20)}>--

Annex C

The Distributed Time service and Time Management Information

(This annex does not form an integral part of this Recommendation | International Standard)

C.1 The Distributed Time service

The Distributed Time service may be implemented as the time synchronization service underlying this Time Management Function. If so, it is implemented in accordance with the Distributed Time service as defined in OSF DCE 1.0 (see [5] in H.4).

C.2 The dtsProtocol managed object

A managed object class to support DTS is for further study.

Annex D[2]

MCS proforma

(This annex forms an integral part of this Recommendation | International Standard)

D.1 Introduction

D.1.1 Purpose and structure

The Management Conformance Summary (MCS) is a statement by a supplier that identifies an implementation and provides information on whether the implementation claims conformance to any of the listed set of documents that specify conformance requirements to OSI management.

The MCS proforma is a document in the form of a questionnaire that, when completed by the supplier of an implementation, becomes the MCS.

D.1.2 Instructions for completing the MCS proforma to produce an MCS[3]

The supplier of the implementation shall enter an explicit statement in each of the boxes provided. Specific instruction is provided in the text which precedes each table.

---

[2] **Copyright release for MCS proforma**

Users of this Recommendation | International Standard may freely reproduce the MCS proforma in this annex so that it can be used for its intended purpose, and may further publish the completed MCS.

[3] Instructions for completing the MCS proforma are specified in ITU-T Rec. X.724 | ISO/IEC 10165-6.

D.1.3 Symbols, abbreviations and terms

For all annexes of this Recommendation | International Standard, the following common notations, defined in ITU-T Rec. X.291 | ISO/IEC 9646-2 and ITU-T Rec. X.296 | ISO/IEC 9646-7, are used for the Status column:

- m Mandatory
- o Optional
- c Conditional
- x Prohibited
- – Not applicable or out of scope

NOTE 1 – "c", "m", and "o" are prefixed by a "c:" when nested under a conditional or optional item of the same table.

NOTE 2 – "o" may be suffixed by ".N" (where N is a unique number) for mutually exclusive or selectable options among a set of status values. Support of at least one of the choices (from the items with the same values of N) is required.

For all annexes of this Recommendation | International Standard, the following common notations, defined in ITU-T Rec. X.291 | ISO/IEC 9646-2 and ITU-T Rec. X.296 | ISO/IEC 9646-7 are used for the Support column:

- Y Implemented
- N Not implemented
- – No answer required
- Ig The item is ignored (i.e. processed syntactically but not semantically)

D.2 Identification of the implementation

D.2.1 Date of statement

The supplier of the implementation shall enter the date of this statement in the box below. Use the format DD-MM-YYYY.

Date of statement

D.2.2 Identification of the implementation

The supplier of the implementation shall enter information necessary to uniquely identify the implementation and the system(s) in which it may reside, in the box below.

D.2.3 Contact

The supplier of the implementation shall provide information on whom to contact if there are any queries concerning the content of the MCS, in the box below.

D.3 Identification of the Recommendation | International Standard in which the management information is defined

The supplier of the implementation shall enter the title, reference number and date of the publication of the Recommendation | International Standard which specifies the management information to which conformance is claimed, in the box below.

Recommendation | International Standard to which conformance is claimed

D.3.1 Technical corrigenda implemented

The supplier of the implementation shall enter the reference numbers of implemented technical corrigenda which modify the identified Recommendation | International Standard, in the box below.

D.3.2 Amendments implemented

The supplier of the implementation shall state the titles and reference numbers of implemented amendments to the identified Recommendation | International Standard, in the box below.

D.4 Management conformance summary

The supplier of implementation shall state the capabilities and features supported and provide summary of conformance claims to Recommendations | International Standards using the tables in this annex.

The supplier of the implementation shall specify the roles that are supported, in Table D.1

Table D.1 – Roles

Index	Roles supported	Status	Support	Additional information
1	Manager role support	o.1		
2	Agent role support	o.1		

The supplier of the implementation shall specify support for the systems management functional units, in Table D.2

Table D.2 – Systems management functional units

		Manager		Agent		
Index	Systems management functional unit name	Status	Support	Status	Support	Additional information
1	clock control functional unit	c1		c2		
2	clock coordination control functional unit	c1		c2		
c1: if D.1/1a then o else –						
c2: if D.1/2a then o else –						

The supplier of the implementation shall specify support for management information in the manager role, in Table D.3

Table D.3 – Manager role minimum conformance requirement

Index	Item	Status	Support	Additional information
1	Operations on managed objects	c3		
2	Clock reset action for local clock managed object	c4		
3	Clock reset action for reference clock managed object	c4		
4	Leap second action for synchronization protocol (or subclass) managed object	c5		
5	Protocol reset action for synchronization protocol (or subclass) managed object	c5		
6	State change notification for local clock managed object	c4		
7	State change notification for reference clock managed object	c4		
c3: if D.1/1a then o.2 else – c4: if D.2/1a then o.3 else (if D.1/1a then o.2 else –) c5: if D.2/2a then m else (if D.1/1a then o.2 else –)				

The supplier of the implementation shall specify support for management information in the agent role, in Table D.4

Table D.4 – Agent role minimum conformance requirement

Index	Item	Status	Support	Additional information
1	Local clock managed object	c6		
2	Reference clock managed object	c7		
3	Synchronization protocol managed object	c8		
4	NTP protocol managed object	c9		
c6: if D.1/2a then m else – c7: if D.1/2a then o else – c8: if support of a synchronization protocol for which there is no specialized managed object class (e.g. DTS) then m else – (Indicate synchronization protocol in Additional information column) c9: if support of NTP protocol then m else –				

The supplier of the implementation shall provide information on claims of conformance to any of the Recommendation | International Standards summarized in Tables D.5 to D.8. For each Recommendation | International Standard that the supplier of the implementation claims conformance to, the corresponding conformance statement(s) shall be completed, or referenced by, the MCS. The supplier of the implementation shall complete the Support, Table numbers and Additional information columns.

In Tables D.5 to D.8, the Status column is used to indicate whether the supplier of the implementation is required to complete the referenced tables or referenced items. Conformance requirements are as specified in the referenced tables or referenced items and are not changed by the value of the MCS Status column. Similarly, the Support column is used by the supplier of the implementation to indicate completion of the referenced tables or referenced items.

Table D.5 – PICS support summary

Index	Identification of the document that includes the PICS proforma	Table numbers of PICS proforma	Description	Constraints and values	Status	Support	Table numbers of PICS	Additional information
1	"CCITT Rec. X.730 (1992) | ISO/IEC 10164-1:1993"	Annex E all tables	SM application context	–	o			
2	"ITU-T Rec. X.743 (1998) | ISO/IEC 10164-20:1998"	– (PICS proforma do not exist, indicate support only)	NTP protocol	–	c10			
3	"ITU-T Rec. X.743 (1998) | ISO/IEC 10164-20:1998"	– (PICS proforma do not exist, indicate support only)	DTS protocol	–	c11			
c10: if D.4/4a then m else – c11: if support of DTS protocol then m else –								

Table D.6 – MOCS support summary

Index	Identification of the document that includes the MOCS proforma	Table numbers of MOCS proforma	Description	Constraints and values	Status	Support	Table numbers of MOCS	Additional information
1	"ITU-T Rec. X.743 (1998) | ISO/IEC 10164-20:1998"	Table F.1-F.6	localClock	–	c12			
2	"ITU-T Rec. X.743 (1998) | ISO/IEC 10164-20:1998"	Table F.7-F.12	referenceClock	–	c13			
3	"ITU-T Rec. X.743 (1998) | ISO/IEC 10164-20:1998"	Table F.13-F.17	synchronizationProtocol	–	c14			
4	"ITU-T Rec. X.743 (1998) | ISO/IEC 10164-20:1998"	Table F.18-F.22	ntpProtocol	–	c15			
c12: if D.4/1a then m else – c13: if D.4/2a then m else – c14: if D.4/3a then m else – c15: if D.4/4a then m else –								

Table D.7 – MRCS support summary

Index	Identification of the document that includes the MRCS proforma	Table numbers of MRCS proforma	Description	Constraints and values	Status	Support	Table numbers of MRCS	Additional information
1	"ITU-T Rec. X.743 (1998) | ISO/IEC 10164-20:1998"	Table G.1/1	clockSource-system	–	o			
2	"ITU-T Rec. X.743 (1998) | ISO/IEC 10164-20:1998"	Table G.1/2	synchronizationProtocol system	–	o			

Table D.8 – MICS support summary

Index	Identification of the document that includes the MICS proforma	Table numbers of MICS proforma	Description	Constraints and values	Status	Support	Table numbers of MICS	Additional information
1	"ITU-T Rec. X.743 (1998) | ISO/IEC 10164-20:1998"	Table E.1	management operations	–	c16			
2	"ITU-T Rec. X.743 (1998) | ISO/IEC 10164-20:1998"	Table E.2	actions	–	c17			
3	"ITU-T Rec. X.743 (1998) | ISO/IEC 10164-20:1998"	Table E.3	notification	–	c18			

c16: if D.3/1a then m else –

c17: if D.3/2a or D.3/3a or D.3/4a or D.3/5a then m else –

c18: if D.3/6a or D.3/7a then m else –

Annex E[4)]

MICS proforma

(This annex forms an integral part of this Recommendation | International Standard)

E.1 Introduction

The purpose of this MICS proforma is to provide a mechanism for a supplier of an implementation which claims conformance, in the manager role, to management information specified in this Recommendation | International Standard, to provide conformance information in a standard form.

E.2 Instructions for completing the MICS proforma to produce a MICS

The MICS proforma contained in this annex is comprised of information in tabular form, in accordance with ITU-T Rec. X.724 | ISO/IEC 10165-6. In addition to the general guidance given in ITU-T Rec. X.724 | ISO/IEC 10165-6. The supplier of the implementation shall state which items are supported in the tables below and if necessary, provide additional information.

E.3 Symbols, abbreviations and terms

The MICS proforma contained in this annex is comprised of information in tabular form, in accordance with CCITT Rec. X.291 | ISO/IEC 9646-2.

The notations used in the Status and Support columns are specified in D.1.3.

E.4 Statement of conformance to the management information

E.4.1 Attributes

The specifier of a manager role implementation that claims to support management operations on the attributes specified in this Recommendation | International Standard shall import a copy of Tables E.1 to E.7 and complete them.

[4)] **Copyright release for MICS proforma**

Users of this Recommendation | International Standard may freely reproduce the MICS proforma in this annex so that it can be used for its intended purpose, and may further publish the completed MICS.

Table E.1 – Attribute support

Index	Attribute template label	Value of object identifier for attribute	Constraints and values	Set by create Status	Set by create Support	Get Status	Get Support
1	"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2: 1992": allomorphs	{2 9 3 2 7 50}	SET OF Objectless	o		o	
2	"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992": nameBinding	{2 9 3 2 7 63}	OBJECT IDENTIFIER				
3	"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992": objectClass	{2 9 3 2 7 65}	ObjectClass	o.5		o.5	
4	"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992": packages	{2 9 3 2 7 66}	SET OF OBJECT IDENTIFIER	o		o	
5	clockId	{2 9 2 20 7 6}		–		o.5	
6	"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992": operationalState	{2 9 3 2 7 35}	ENUMERATED	–		o.5	
7	clockStatus	{2 9 2 20 7 9}		–		o.5	
8	clockValue	{2 9 2 20 7 11}		–		o.5	
9	clockPrecision	{2 9 2 20 7 8}		–		o.5	
10	clockDrift	{2 9 2 20 7 1}		–		o.5	
11	clockMaximumError	{2 9 2 20 7 7}		–		o.5	
12	clockEstimatedError	{2 9 2 20 7 2}		–		o.5	
13	leapSecondIndication	{2 9 2 20 7 15}		o.5		o.5	
14	leapSecondCount	{2 9 2 20 7 14}		o.5		o.5	
15	clockEventCounter	{2 9 2 20 7 4}		–		o.5	
16	clockEventCode	{2 9 2 20 7 3}		–		o.5	
17	clockEventTime	{2 9 2 20 7 5}		–		o.5	
18	localClockAddress	{2 9 2 20 7 16}		–		o.5	
19	peerClockAddresses	{2 9 2 20 7 27}		o.5		o.5	
20	synchronizationSourceAddress	{2 9 2 20 7 33}		–		o.5	
21	clockStratum	{2 9 2 20 7 10}		–		o.5	
22	clockAdjustmentInterval	{2 9 2 20 7 0}		o.5		o.5	
23	referenceClockType	{2 9 2 20 7 29}		–		o.5	
24	synchronizationProtocolID	{2 9 2 20 7 31}		–		o.5	
25	synchronizedClock	{2 9 2 20 7 34}		–		o.5	
26	synchronizingClocks	{2 9 2 20 7 35}		–		o.5	
27	maximumStratum	{2 9 2 20 7 23}		–		o.5	
28	maximumClockAge	{2 9 2 20 7 17}		–		o.5	
29	maximumSkew	{2 9 2 20 7 22}		–		o.5	
30	maximumDistance	{2 9 2 20 7 19}		–		o.5	
31	minimumPollInterval	{2 9 2 20 7 25}		–		o.5	
32	maximumPollInterval	{2 9 2 20 7 20}		–		o.5	
33	minimumSelectClock	{2 9 2 20 7 26}		–		o.5	
34	maximumSelectClock	{2 9 2 20 7 21}		–		o.5	
35	minimumDispersion	{2 9 2 20 7 24}		–		o.5	
36	maximumDispersion	{2 9 2 20 7 18}		–		o.5	
37	reachablityRegisterSize	{2 9 2 20 7 28}		–		o.5	
38	filterSize	{2 9 2 20 7 12}		–		o.5	
39	filterWeight	{2 9 2 20 7 13}		–		o.5	
40	selectWeight	{2 9 2 20 7 30}		–		o.5	
41	synchronizationProtocolType	{2 9 2 20 7 32}		–		o.5	

Table E.1 *(concluded)* **– Attribute support**

	Replace		Add		Remove		Set to default		
Index	Status	Support	Status	Support	Status	Support	Status	Support	Additional information
1	–		–		–		–		
2	–		–		–		–		
3	–		–		–		–		
4	–		–		–		–		
5	–		–		–		–		
6	–		–		–		–		
7	–		–		–		–		
8	–		–		–		–		
9	–		–		–		–		
10	–		–		–		–		
11	–		–		–		–		
12	–		–		–		–		
13	o.5		–		–		–		
14			–		–		–		
15			–		–		–		
16	–		–		–		–		
17	–		–		–		–		
18	–		–		–		–		
19	o.5		o.5		o.5		–		
20	–		–		–		–		
21	–		–		–		–		
22	o.5		–		–		–		
23	–		–		–		–		
24	–		–		–		–		
25	–		–		–		–		
26	–		–		–		–		
27	–		–		–		–		
28	–		–		–		–		
29	–		–		–		–		
30	–		–		–		–		
31	–		–		–		–		
32	–		–		–		–		
33	–		–		–		–		
34	–		–		–		–		
35	–		–		–		–		
36	–		–		–		–		
37	–		–		–		–		
38	–		–		–		–		
39	–		–		–		–		
40	–		–		–		–		
41	–		–		–		–		

E.4.2 Create and delete management operations

The specifier of a manager role implementation that claims to support the create or the delete management operations on the managed objects specified in this Recommendation | International Standard shall import a copy of Tables E.2 to E.5 and complete them.

E.4.2.1 Local clock managed object class

See Table E.2.

Table E.2 – Create and delete support

Index	Operation	Constraints and values	Status	Support	Additional information
1	Create support	localClock MO	o.5		
1.1	Create with reference object	–	–		
2	Delete support	localClock MO	o.5		

E.4.2.2 Reference clock managed object class

See Table E.3.

Table E.3 – Create and delete support

Index	Operation	Constraints and values	Status	Support	Additional information
1	Create support	referenceClock MO	o.5		
1.1	Create with reference object	–	–		
2	Delete support	referenceClock MO	o.5		

E.4.2.3 Synchronization protocol managed object class

See Table E.4.

Table E.4 – Create and delete support

Index	Operation	Constraints and values	Status	Support	Additional information
1	Create support	synchronizationProtocol MO	o.5		
1.1	Create with reference object	–	–		
2	Delete support	synchronizationProtocol MO	o.5		

E.4.2.4 NTP protocol managed object class

See Table E.5.

Table E.5 – Create and delete support

Index	Operation	Constraints and values	Status	Support	Additional information
1	Create support	ntpProtocol MO	o.5		
1.1	Create with reference object	–	–		
2	Delete support	ntpProtocol MO	o.5		

E.4.3 Actions

The specifier of a manager role implementation that claims to support the actions specified in this Recommendation | International Standard shall import a copy of Table E.6 and complete it.

Table E.6 – Action support

Index	Action type template label	Value of object identifier for action type	Constraints and values	Status	Support	Additional information
1	clockReset	{2 9 2 20 9 1}		c1		
2	leapSecond	{2 9 2 20 9 2}		c2		
3	protocolReset	{2 9 2 20 9 3}		c3		

Table E.6 *(concluded)* **– Action support**

Index	Subindex	Action field name label	Constraints and values	Status	Support	Additional information
1	1.1	ClockResetInfo	**Information Syntax** ClockValue	c1		
2	2.1	LeapSecondInfo	**Information Syntax** SEQUENCE	c2		
	2.1.1	LeapIndication	**ENUMERATED**	m		
	2.1.2	DayOfLeap	**GeneralizedTime**	m		
3	3.1	ProtocolResetInfo	**Information Syntax** SET OF SEQUENCE	c3		
c1: if D.3/2a or D.3/3a then m else – c2: if D.3/4a then m else – c3: if D.3/5a then m else –						

E.4.4 Notification

See Table E.7.

Table E.7 – Notification support

Index	Notification type template label	Value of object identifier for notification type	Constraints and values	Status	Support		Additional information
					Confirmed	Non-confirmed	
1	"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992": stateChange	{2 9 3 2 10 14}		c4			

Table E.7 *(concluded)* **– Notification support**

Index	Subindex	Notification field name label	Value of object identifier of attribute type associated with field	Constraints and values	Status	Support	Additional information
1	1.1	StateChangeInfo		**Information Syntax** SEQUENCE	c4		
	1.1.1	sourceIndicator	{2 9 3 2 7 26}	ENUMERATED	c:m		
	1.1.2	attributeIdentifierList	{2 9 3 2 7 8}	SET OF AttributeId	c:m		
	1.1.3	stateChangeDefinition	{2 9 3 2 7 28}	SET OF SEQUENCE	c:m		
	1.1.3.1	attributeID	–	AttributeId	c:m		
	1.1.3.2	oldAttributeValue	–	ANY DEFINED BY attributeID	c:m		
	1.1.3.3	newAttributeValue	–	ANY DEFINED BY attributeID	c:m		
	1.1.4	notificationIdentifier	{2 9 3 2 7 16}	INTEGER	c:m		
	1.1.5	correlatedNotifications	{2 9 3 2 7 12}	SET OF SEQUENCE	c:m		
	1.1.5.1	correlatedNotifications	{2 9 3 2 7 12}	SET OF INTEGER	c:m		
	1.1.5.2	sourceObjectInst	–	ObjectInstance	c:m		
	1.1.6	additionalText	{2 9 3 2 7 7}	GraphicString	c:m		
	1.1.7	additionalInformation	{2 9 3 2 7 6}	SET OF SEQUENCE	c:m		
	1.1.7.1	identifier	–	OBJECT IDENTIFIER	c:m		
	1.1.7.2	significance	–	BOOLEAN	c:m		
	1.1.7.3	information	–	ANY DEFINED BY identifier	c:m		
c4: if D.8/3a then m else –							

Annex F[5)]

MOCS proforma

(This annex forms an integral part of this Recommendation | International Standard)

F.1 Introduction

The purpose of this MOCS proforma is to provide a mechanism for a supplier of an implementation of a Recommendation | International Standard which claims conformance to a managed object class, to provide conformance information in a standard form.

F.1.1 Instructions for completing the MOCS proforma to produce a MOCS [6)]

The MOCS proforma contained in this annex is comprised of information in tabular form, in accordance with ITU-T Rec. X.724 | ISO/IEC 10165-6. The supplier of the implementation shall state which items are supported in the tables below and if necessary provide additional information.

F.1.2 Symbols, abbreviations and terms

The MOCS proforma contained in this annex is comprised of information in tabular form, in accordance with ITU-T Rec. X.291 | ISO/IEC 9646-2.

The notations used in the Status and Support columns are specified in D.1.3.

F.2 localClock

F.2.1 Statement of conformance to the managed object class

See Table F.1.

Table F.1 – localClock Managed object class support

Index	Managed object class template label	Value of object identifier for class	Support of all mandatory features? (Y/N)	Is the actual class the same as the managed object class to which conformance is claimed? (Y/N)
1	localClock	{2 9 2 20 3 1}		

If the answer to the actual class question in Table F.1 is No, the supplier of the implementation shall fill in the actual class support Table F.2.

Table F.2 – localClock Actual class support

Index	Managed object class template for actual class	Value of object identifier for managed object class definition of actual class	Additional information

6) Instructions for completing the MOCS proforma are specified in ITU-T Rec. X.724 | ISO/IEC 10165-6.

F.2.2 Packages

The supplier of the implementation shall state whether or not the packages specified by this managed object of this class are supported, in Table F.3.

Table F.3 – localClock Package support

Index	Package template label	Value of object identifier for package	Constraints and values	Status	Support	Additional information
1	"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992": allomorphicPackage	{2 9 3 2 4 17}	"if an object supports allomorphism"	c1		
2	"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992": packagesPackage	{2 9 3 2 4 16}	"any registered package, other than this package, has been instantiated"	c2		
3	"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992": topPackage	–	Mandatory	m		
4	clockSourcePkg	–	Mandatory	m		
5	clockSourceDetailPkg	{2 9 2 20 4 0}	"if an instance supports it"	o		
6	leapSecondPkg	{2 9 2 20 4 1}	"if an instance supports it"	o		
7	localClockPkg	–	Mandatory	m		
c1: if F.1/1b then – else m c2: if F.3/1a then m else –						

F.2.3 Attributes

The supplier of the implementation shall state whether or not the attributes specified by all of the packages instantiated in a managed object of this class are supported, in the Support and Additional information columns of Table F.4. The supplier of the implementation shall indicate support for each of the operations for each attribute supported.

Table F.4 – localClock Attribute support

Index	Attribute template label	Value of object identifier for attribute	Constraints and values	Set by create		Get	
				Status	Support	Status	Support
1	"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992": allomorphs	{2 9 3 2 7 50}	SET OF ObjectClass	c3		c4	
2	"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992": nameBinding	{2 9 3 2 7 63}	OBJECT IDENTIFIER	o		m	
3	"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992": objectClass	{2 9 3 2 7 65}	ObjectClass	m		m	
4	"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992": packages	{2 9 3 2 7 66}	SET OF OBJECT IDENTIFIER	c5		c6	
5	clockId	{2 9 2 20 7 6}		x		m	
6	"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992": operationalState	{2 9 3 2 7 35}	ENUMERATED	x		m	
7	clockStatus	{2 9 2 20 7 9}		c8		m	
8	clockValue	{2 9 2 20 7 11}		c8		m	
9	clockPrecision	{2 9 2 20 7 8}		c10		c9	
10	clockDrift	{2 9 2 20 7 1}		c10		c9	
11	clockMaximumError	{2 9 2 20 7 7}		c10		c9	
12	clockEstimatedError	{2 9 2 20 7 2}		c10		c9	
13	leapSecondIndication	{2 9 2 20 7 15}		c11		c11	
14	leapSecondCount	{2 9 2 20 7 14}		c11		c11	
15	clockEventCounter	{2 9 2 20 7 4}		c8		m	
16	clockEventCode	{2 9 2 20 7 3}		c8		m	
17	clockEventTime	{2 9 2 20 7 5}		c8		m	
18	localClockAddress	{2 9 2 20 7 16}		c8		m	
19	peerClockAddresses	{2 9 2 20 7 27}		m		m	
20	synchronizationSourceAddress	{2 9 2 20 7 33}		c8		m	
21	clockStratum	{2 9 2 20 7 10}		m		m	
22	clockAdjustmentInterval	{2 9 2 20 7 0}		m		m	

Table F.4 *(concluded)* **– localClock Attribute support**

	Replace		Add		Remove		Set to default		
Index	Status	Support	Status	Support	Status	Support	Status	Support	Additional information
1	–		–		–		–		
2	x		–		–		x		
3	x		–		–		x		
4	c7		c7		c7		c7		
5	x		–		–		x		
6	x		–		–		x		
7	c8		–		–		c8		
8	c8		–		–		c8		
9	c10		–		–		c10		
10	c10		–		–		c10		
11	c10		–		–		c10		
12	c10		–		–		c10		
13	c11		–		–		c12		
14	c11		–		–		c12		
15	c8		–		–		c8		
16	c8		–		–		c8		
17	c8		–		–		c8		
18	c8		–		–		c8		
19	m		m		m		c8		
20	c8		–		–		c8		
21	c8		–		–		c8		
22	m		–		–		c8		

c3: if F.3/1a then o else –
c4: if F.3/1a then m else –
c5: if F.3/2a then o else –
c6: if F.3/2a then m else –
c7: if F.3/2a then x else –
c8: if F.1/1b then x else –
c9: if F.3/5a then m else –
c10: if F.3/5a and F.1/1b then x else –
c11: if F.3/6a then m else –
c12: if F.3/6a and F.1/1b then x else –

F.2.4 Action

See Table F.5

Table F.5 – localClock Action support

Index	Action type template label	Value of object identifier for action type	Constraints and values	Status	Support	Additional information
1	clockReset	{2 9 2 20 9 1}		m		

Table F.5 *(concluded)* **– localClock Action support**

Index	Subindex	Action field name label	Constraints and values	Status	Support	Additional information
1	1.1	ClockResetInfo	**Information Syntax** ClockValue	m		

F.2.5 Notification

See Table F.6

Table F.6 – localClock Notification support

Index	Notification type template label	Value of object identifier for notification type	Constraints and values	Status	Support		Additional information
					Confirmed	Non-confirmed	
1	"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992": stateChange	{2 9 3 2 10 14}		m			

Table F.6 *(concluded)* **– localClock Notification support**

Index	Subindex	Notification field name label	Value of object identifier of attribute type associated with field	Constraints and values	Status	Support	Additional information
1	1.1	StateChangeInfo		**Information Syntax** SEQUENCE	m		
	1.1.1	sourceIndicator	{2 9 3 2 7 26}	ENUMERATED	o		
	1.1.2	attributeIdentifierList	{2 9 3 2 7 8}	SET OF AttributeId	o		
	1.1.3	stateChangeDefinition	{2 9 3 2 7 28}	SET OF SEQUENCE	m		
	1.1.3.1	attributeID	–	AttributeId	m		
	1.1.3.2	oldAttributeValue	–	ANY DEFINED BY attributeID	o		
	1.1.3.3	newAttributeValue	–	ANY DEFINED BY attributeID	m		
	1.1.4	notificationIdentifier	{2 9 3 2 7 16}	INTEGER	o		
	1.1.5	correlatedNotifications	{2 9 3 2 7 12}	SET OF SEQUENCE	o		
	1.1.5.1	correlatedNotifications	{2 9 3 2 7 12}	SET OF INTEGER	c:m		
	1.1.5.2	sourceObjectInst	–	ObjectInstance	c:o		
	1.1.6	additionalText	{2 9 3 2 7 7}	GraphicString	o		
	1.1.7	additionalInformation	{2 9 3 2 7 6}	SET OF SEQUENCE	o		
	1.1.7.1	identifier	–	OBJECT IDENTIFIER	c:m		
	1.1.7.2	significance	–	BOOLEAN	c:o		
	1.1.7.3	information	–	ANY DEFINED BY identifier	c:m		

F.3 referenceClock

F.3.1 Statement of conformance to the managed object class

See Table F.7.

Table F.7 – referenceClock Managed object class support

Index	Managed object class template label	Value of object identifier for class	Support of all mandatory features? (Y/N)	Is the actual class the same as the managed object class to which conformance is claimed? (Y/N)
1	referenceClock	{2 9 2 20 3 2}		

If the answer to the actual class question in Table F.7 is No, the supplier of the implementation shall fill in the actual class support Table F.8.

Table F.8 – referenceClock Actual class support

Index	Managed object class template for actual class	Value of object identifier for managed object class definition of actual class	Additional information

F.3.2 Packages

The supplier of the implementation shall state whether or not the packages specified by this managed object of this class are supported, in Table F.9.

Table F.9 – referenceClock Package support

Index	Package template label	Value of object identifier for package	Constraints and values	Status	Support	Additional information
1	"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992": allomorphicPackage	{2 9 3 2 4 17}	"if an object supports allomorphism"	c13		
2	"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992": packagesPackage	{2 9 3 2 4 16}	"any registered package, other than this package, has been instantiated"	c14		
3	"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992": topPackage	–	Mandatory	m		
4	clockSourcePkg	–	Mandatory	m		
5	clockSourceDetailPkg	{2 9 2 20 4 0}	"if an instance supports it"	o		
6	leapSecondPkg	{2 9 2 20 4 1}	"if an instance supports it"	o		
7	referenceClockPkg	–	Mandatory	m		
c13: if F.7/1b then – else m c14: if F.9/1a then m else –						

F.3.3 Attributes

The supplier of the implementation shall state whether or not the attributes specified by all of the packages instantiated in a managed object of this class are supported, in the Support and Additional information columns of Table F.10. The supplier of the implementation shall indicate support for each of the operations for each attribute supported.

Table F.10 – referenceClock Attribute support

				Set by create		Get	
Index	Attribute template label	Value of object identifier for attribute	Constraints and values	Status	Support	Status	Support
1	"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992": allomorphs	{2 9 3 2 7 50}	SET OF ObjectClass	c15		c16	
2	"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992": nameBinding	{2 9 3 2 7 63}	OBJECT IDENTIFIER	o		m	
3	"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992": objectClass	{2 9 3 2 7 65}	ObjectClass	m		m	
4	"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992": packages	{2 9 3 2 7 66}	SET OF OBJECT IDENTIFIER	c17		c18	
5	clockId	{2 9 2 20 7 6}		x		m	
6	"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992": operationalState	{2 9 3 2 7 35}	ENUMERATED	x		m	
7	clockStatus	{2 9 2 20 7 9}		c20		m	
8	clockValue	{2 9 2 20 7 11}		c20		m	
9	clockPrecision	{2 9 2 20 7 8}		c22		c21	
10	clockDrift	{2 9 2 20 7 1}		c22		c21	
11	clockMaximumError	{2 9 2 20 7 7}		c22		c21	
12	clockEstimatedError	{2 9 2 20 7 2}		c22		c21	
13	leapSecondIndication	{2 9 2 20 7 15}		c23		c23	
14	leapSecondCount	{2 9 2 20 7 14}		c23		c23	
15	clockEventCounter	{2 9 2 20 7 4}		c20		m	
16	clockEventCode	{2 9 2 20 7 3}		c20		m	
17	clockEventTime	{2 9 2 20 7 5}		c20		m	
18	referenceClockType	{2 9 2 20 7 29}		c20		m	

Table F.10 *(concluded)* **– referenceClock Attribute support**

	Replace		Add		Remove		Set to default		
Index	Status	Support	Status	Support	Status	Support	Status	Support	Additional information
1	–		–		–		–		
2	x		–		–		x		
3	x		–		–		x		
4	c19		c19		c19		c19		
5	x		–		–		x		
6	x		–		–		x		
7	c20		–		–		c20		
8	c20		–		–		c20		
9	c22		–		–		c22		
10	c22		–		–		c22		
11	c22		–		–		c22		
12	c22		–		–		c22		
13	c23		–		–		c24		
14	c23		–		–		c24		
15	c20		–		–		c20		
16	c20		–		–		c20		
17	c20		–		–		c20		
18	c20		–		–		c20		

c15: if F.9/1a then o else –
c16: if F.9/1a then m else –
c17: if F.9/2a then o else –
c18: if F.9/2a then m else –
c19: if F.9/2a then x else –
c20: if F.7/1b then x else –
c21: if F.9/5a then m else –
c22: if F.9/5a and F.7/1b then x else –
c23: if F.9/6a then m else –
c24: if F.9/6a and F.7/1b then x else –

F.3.4 Actions

See Table F.11.

Table F.11 – referenceClock Action support

Index	Action type template label	Value of object identifier for action type	Constraints and values	Status	Support	Additional information
1	clockReset	{2 9 2 20 9 1}		m		

Table F.11 *(concluded)* **– referenceClock Action support**

Index	Subindex	Action field name label	Constraints and values	Status	Support	Additional information
1	1.1	ClockResetInfo	**Information Syntax** ClockValue	m		

F.3.5 Notification

See Table F.12.

Table F.12 – referenceClock Notification support

Index	Notification type template label	Value of object identifier for notification type	Constraints and values	Status	Support		Additional information
					Confirmed	Non-confirmed	
1	"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992": stateChange	{2 9 3 2 10 14}		m			

Table F.12 *(concluded)* **– referenceClock Notification support**

Index	Subindex	Notification field name label	Value of object identifier of attribute type associated with field	Constraints and values	Status	Support	Additional information
1	1.1	StateChangeInfo		**Information Syntax** SEQUENCE	m		
	1.1.1	sourceIndicator	{2 9 3 2 7 26}	ENUMERATED	o		
	1.1.2	attributeIdentifierList	{2 9 3 2 7 8}	SET OF AttributeId	o		
	1.1.3	stateChangeDefinition	{2 9 3 2 7 28}	SET OF SEQUENCE	m		
	1.1.3.1	attributeID	–	AttributeId	m		
	1.1.3.2	oldAttributeValue	–	ANY DEFINED BY attributeID	o		
	1.1.3.3	newAttributeValue	–	ANY DEFINED BY attributeID	m		
	1.1.4	notificationIdentifier	{2 9 3 2 7 16}	INTEGER	o		
	1.1.5	correlatedNotifications	{2 9 3 2 7 12}	SET OF SEQUENCE	o		
	1.1.5.1	correlatedNotifications	{2 9 3 2 7 12}	SET OF INTEGER	c:m		
	1.1.5.2	sourceObjectInst	–	ObjectInstance	c:o		
	1.1.6	additionalText	{2 9 3 2 7 7}	GraphicString	o		
	1.1.7	additionalInformation	{2 9 3 2 7 6}	SET OF SEQUENCE	o		
	1.1.7.1	identifier	–	OBJECT IDENTIFIER	c:m		
	1.1.7.2	significance	–	BOOLEAN	c:o		
	1.1.7.3	information	–	ANY DEFINED BY identifier	c:m		

F.4 synchronizationProtocol

F.4.1 Statement of conformance to the managed object class

See Table F.13.

Table F.13 – synchronizationProtocol Managed object class support

Index	Managed object class template label	Value of object identifier for class	Support of all mandatory features? (Y/N)	Is the actual class the same as the managed object class to which conformance is claimed? (Y/N)
1	ntpProtocol	{2 9 2 20 3 3}		

If the answer to the actual class question in Table F.13 is No, the supplier of the implementation shall fill in the actual class support Table F.14.

Table F.14 – synchronizationProtocol Actual class support

Index	Managed object class template for actual class	Value of object identifier for managed object class definition of actual class	Additional information

F.4.2 Packages

The supplier of the implementation shall state whether or not the packages specified by this managed object of this class are supported, in Table F.15.

Table F.15 – synchronizationProtocol Package support

Index	Package template label	Value of object identifier for package	Constraints and values	Status	Support	Additional information
1	"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992": allomorphicPackage	{2 9 3 2 4 17}	"if an object supports allomorphism"	c24		
2	"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992": packagesPackage	{2 9 3 2 4 16}	"any registered package, other than this package, has been instantiated"	c26		
3	"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992": topPackage	–	Mandatory	m		
4	synchronizationProtocolPkg	–	Mandatory	m		
c25: if F.13/1b then – else m c26: if F.15/1a then m else –						

F.4.3 Attributes

The supplier of the implementation shall state whether or not the attributes specified by all of the packages instantiated in a managed object of this class are supported, in the Support and Additional information columns of Table F.16. The supplier of the implementation shall indicate support for each of the operations for each attribute supported.

Table F.16 – synchronizationProtocol Attribute support

				Set by create		Get	
Index	Attribute template label	Value of object identifier for attribute	Constraints and values	Status	Support	Status	Support
1	"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992": allomorphs	{2 9 3 2 7 50}	SET OF ObjectClass	c27		c28	
2	"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992": nameBinding	{2 9 3 2 7 63}	OBJECT IDENTIFIER	o		m	
3	"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992": objectClass	{2 9 3 2 7 65}	ObjectClass	m		m	
4	"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992": packages	{2 9 3 2 7 66}	SET OF OBJECT IDENTIFIER	c29		c30	
5	synchronizationProtocolID	{2 9 2 20 7 31}		m		m	
6	synchronizedClock	{2 9 2 20 7 34}		c32		m	
7	synchronizingClocks	{2 9 2 20 7 35}		c32		m	
8	synchronizationProtocolType	{2 9 2 20 7 32}		c32		m	

Table F.16 *(concluded)* **– synchronizationProtocol Attribute support**

	Replace		Add		Remove		Set to default		
Index	Status	Support	Status	Support	Status	Support	Status	Support	Additional information
1	–		–		–		–		
2	x		–		–		x		
3	x		–		–		x		
4	c31		c31		c31		c31		
5			–		–				
6	c32		–		–		c32		
7	c32		c32		c32		c32		
8	c32		–		–		c32		

c27: if F.15/1a then o else –
c28: if F.15/1a then m else –
c29: if F.15/2a then o else –
c30: if F.15/2a then m else –
c31: if F.15/2a then x else –
c32: if F.13/1b then x else –

F.4.4 Actions

See Table F.17.

Table F.17 – synchronizationProtocol Action support

Index	Action type template label	Value of object identifier for action type	Constraints and values	Status	Support	Additional information
1	clockReset	{2 9 2 20 9 1}		m		
2	leapSecond	{2 9 2 20 9 2}		m		
3	protocolReset	{2 9 2 20 9 3}		m		

Table F.17 *(concluded)* **– synchronizationProtocol Action support**

Index	Subindex	Action field name label	Constraints and values	Status	Support	Additional information
1	1.1	ClockResetInfo	**Information Syntax** ClockValue	m		
2	2.1	LeapSecondInfo	**Information Syntax** SEQUENCE	m		
	2.2	LeapIndication	**ENUMERATED**	m		
		DayOfLeap	**GeneralizedTime**	m		
3	3.1	ProtocolResetInfo	**Information Syntax** SET OF Attribute	m		

F.5 ntpProtocol

F.5.1 Statement of conformance to the managed object class

See Table F.18.

Table F.18 – ntpProtocol Managed object class support

Index	Managed object class template label	Value of object identifier for class	Support of all mandatory features? (Y/N)	Is the actual class the same as the managed object class to which conformance is claimed? (Y/N)
1	ntpProtocol	{2 9 2 20 3 4}		

If the answer to the actual class question in Table F.18 is No, the supplier of the implementation shall fill in the actual class support Table F.19.

Table F.19 – ntpProtocol Actual class support

Index	Managed object class template for actual class	Value of object identifier for managed object class definition of actual class	Additional information

F.5.2 Packages

The supplier of the implementation shall state whether or not the packages specified by this managed object of this class are supported, in Table F.20.

Table F.20 – ntpProtocol Package support

Index	Package template label	Value of object identifier for package	Constraints and values	Status	Support	Additional information
1	"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992": allomorphicPackage	{2 9 3 2 4 17}	"if an object supports allomorphism"	c33		
2	"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992": packagesPackage	{2 9 3 2 4 16}	"any registered package, other than this package, has been instantiated"	c34		
3	"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992": topPackage	–	Mandatory	m		
4	synchronizationProtocolPkg	–	Mandatory	m		
5	ntpProtocolPkg	–	Mandatory	m		
c33: if F.18/1b then – else m c34: if F.20/1a then m else –						

F.5.3 Attributes

The supplier of the implementation shall state whether or not the attributes specified by all of the packages instantiated in a managed object of this class are supported, in the Support and Additional information columns of Table F.21. The supplier of the implementation shall indicate support for each of the operations for each attribute supported.

Table F.21 – ntpProtocol Attribute support

				Set by create		Get	
Index	Attribute template label	Value of object identifier for attribute	Constraints and values	Status	Support	Status	Support
1	"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992": allomorphs	{2 9 3 2 7 50}	SET OF ObjectClass	c35		c36	
2	"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992": nameBinding	{2 9 3 2 7 63}	OBJECT IDENTIFIER	o		m	
3	"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992": objectClass	{2 9 3 2 7 65}	ObjectClass	m		m	
4	"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992": packages	{2 9 3 2 7 66}	SET OF OBJECT IDENTIFIER	c37		c38	
5	synchronizationProtocolID	{2 9 2 20 7 31}		m		m	
6	synchronizedClock	{2 9 2 20 7 34}		c40		m	
7	synchronizingClocks	{2 9 2 20 7 35}		c40		m	
8	maximumStratum	{2 9 2 20 7 23}		c40		m	
9	maximumClockAge	{2 9 2 20 7 17}		c40		m	
10	maximumSkew	{2 9 2 20 7 22}		c40		m	
11	maximumDistance	{2 9 2 20 7 19}		c40		m	
12	minimumPollInterval	{2 9 2 20 7 25}		c40		m	
13	maximumPollInterval	{2 9 2 20 7 20}		c40		m	
14	minimumSelectClock	{2 9 2 20 7 26}		c40		m	
15	maximumSelectClock	{2 9 2 20 7 21}		c40		m	
16	minimumDispersion	{2 9 2 20 7 24}		c40		m	
17	maximumDispersion	{2 9 2 20 7 18}		c40		m	
18	reachabilityRegisterSize	{2 9 2 20 7 28}		c40		m	
19	filterSize	{2 9 2 20 7 12}		c40		m	
20	filterWeight	{2 9 2 20 7 13}		c40		m	
21	selectWeight	{2 9 2 20 7 30}		c40		m	
22	synchronizationProtocolType	{2 9 2 20 7 32}		c40		m	

Table F.21 *(concluded)* **– ntpProtocol Attribute support**

	Replace		Add		Remove		Set to default		
Index	Status	Support	Status	Support	Status	Support	Status	Support	Additional information
1	–		–		–		–		
2	x		–		–		x		
3	x		–		–		x		
4	c39		c39		c39		c39		
5	x		–		–		x		
6	c40		–		–		c40		
7	c40		c40		c40		c40		
8	c40		–		–		c40		
9	c40		–		–		c40		
10	c40		–		–		c40		
11	c40		–		–		c40		
12	c40		–		–		c40		
13	c40		–		–		c40		
14	c40		–		–		c40		
15	c40		–		–		c40		
16	c40		–		–		c40		
17	c40		–		–		c40		
18	c40		–		–		c40		
19	c40		–		–		c40		
20	c40		–		–		c40		
21	c40		–		–		c40		
22	c40		–		–		c40		

c35: if F.20/1a then o else –
c36: if F.20/1a then c40 else –
c37: if F.20/2a then o else –
c38: if F.20/2a then c40 else –
c39: if F.20/2a then x else –
c40: if F.18/1b then x else –

F.5.4 Actions

See Table F.22

Table F.22 – ntpProtocol Action support

Index	Action type template label	Value of object identifier for action type	Constraints and values	Status	Support	Additional information
1	clockReset	{2 9 2 20 9 1}		m		
2	leapSecond	{2 9 2 20 9 2}		m		
3	protocolReset	{2 9 2 20 9 3}		m		

Table F.22 *(concluded)* **– ntpProtocol Action support**

Index	Subindex	Action field name label	Constraints and values	Status	Support	Additional information
1	1.1	ClockResetInfo	**Information Syntax** ClockValue	m		
2	2.1	LeapSecondInfo	**Information Syntax** SEQUENCE	m		
	2.2	LeapIndication	**ENUMERATED**	m		
		DayOfLeap	**GeneralizedTime**	m		
3	3.1	ProtocolResetInfo	**Information Syntax** SET OF Attribute	m		

Annex G[7]

MRCS proforma for name binding

(This annex forms an integral part of this Recommendation | International Standard)

G.1 Introduction

The purpose of this MRCS proforma for name bindings is to provide a mechanism for a supplier which claims conformance to a name binding to provide conformance information in a standard form.

G.2 Instructions for completing the MRCS proforma for name binding to produce a MRCS2[8]

The supplier of the implementation shall state which items are supported in the tables below and if necessary provide additional information.

G.3 Statement of conformance to the name binding

See Table G.1.

[7] **Copyright release for MRCS proforma**

Users of this Recommendation | International Standard may freely reproduce the MRCS proforma in this annex so that it can be used for its intended purpose, and may further publish the completed MRCS.

[8] Instructions for completing the MRCS proforma are specified in ITU-T Rec.X.724 | ISO/IEC 10165-6.

Table G.1 – Name Binding support

Index	Name binding template label	Value of object identifier for name binding	Constraints and values	Status	Support	Additional information
1	clockSource-system	{2 9 2 20 6 1}	Superior class: "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992": system AND SUBCLASSES	o		
2	synchronizationProtocol-system	{2 9 2 20 6 2}	Superior class: "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992": system AND SUBCLASSES	o		

Table G.1 *(concluded)* **– Name Binding support**

Index	Subindex	Operation	Constraints and values	Status	Support	Additional information
1	1.1	Create support		m		
	1.1.1	Create with reference object		–		
	1.1.2	Create with automatic instance naming		–		
	1.2	Delete support		m		
	1.2.1	Delete only if no contained objects		–		
	1.2.2	Delete contained objects		–		
2	2.1	Create support		m		
	2.1.2	Create with reference object		–		
	2.1.3	Create with automatic instance naming		–		
	2.2	Delete support		m		
	2.2.1	Delete only if no contained objects		–		
	2.2.2	Delete contained objects		–		

Annex H

Overview of common clock coordination protocols

(This annex does not form an integral part of this Recommendation | International Standard)

Three candidate clock coordination protocols have been identified during the course of this work. These protocols are the Network Time Protocol (NTP), the Distributed Time Service (DTS), and Probabilistic Clock Synchronization (PCS). A brief overview of each of the three is provided as information below.

H.1 The Network Time Protocol

The Network Time Protocol (NTP) is a protocol, service description, and algorithms for the distribution of time information in a large diverse internet system. NTP is fully documented in [5]. The principle features of NTP are listed below [7].

a) The synchronization subnet consists of a self-organizing, hierarchical network of time servers configured on the basis of estimated accuracy, precision and reliability of the participants.

b) The synchronization protocol operates in connectionless mode in order to minimize latencies, simplify implementations and provide ubiquitous internetworking.

c) The synchronization mechanism uses a symmetric design (which tolerates packet loss, duplication and misordering) together with filtering, selection, and combining algorithms that are based on maximum-likelihood principles.

d) The local-clock design is based on type II, adaptive-parameter, phase-lock loop with corrections computed using timestamps exchanged along the arcs of the synchronization subnet.

e) Multiple redundant time servers and multiple diverse transmission paths are used in the synchronization subnet. In addition, engineered algorithms select the most reliable synchronization source(s) and path(s) using weighted-voting procedures.

f) System overhead is reduced through the use of dynamic control of phase-lock loop bandwidth, poll intervals and association management.

H.1.1 Structure of the NTP synchronization subnet

An NTP system consists of a network of primary and secondary time servers, clients, and interconnecting transmission paths. A primary time server is directly synchronized to a reference source (such as a timecode receiver or calibrated atomic clock). A secondary time server derives synchronization, possibly via other secondary servers, from a primary server. These servers are organized in a hierarchical structure. This hierarchical structure is a logical tree reflecting the current state of the synchronization subnet. The dynamic location or depth of each server within the synchronization subnet is reflected by a number called its *stratum*, with higher stratum numbers reflecting a greater number of servers or a longer distance to a reference source. Figure H.1 illustrates this structure.

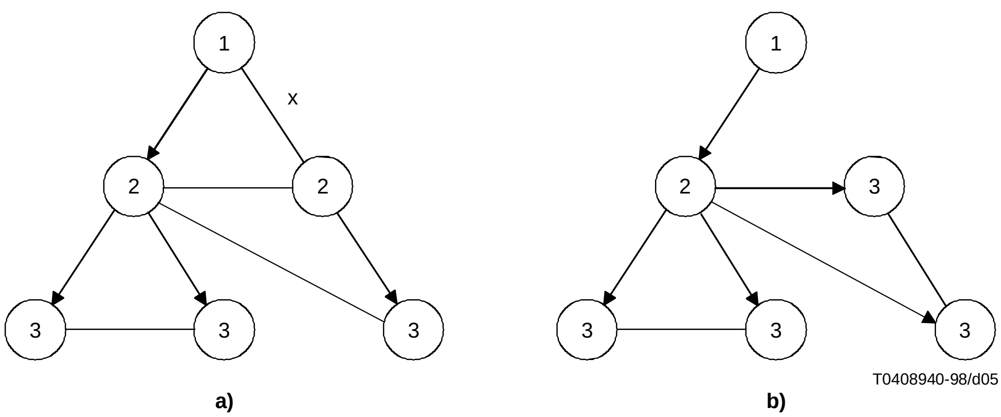


Figure H.1 – Synchronization subnet hierarchy

A number of factors impact the structure of the synchronization subnet. The first of these is the selection of peers or servers for each client. In this discussion, a server is an entity providing time information, and a client is an entity receiving time information. A peer is either a client or a server involved in a time information exchange. Using this generalization, secondary servers are servers when providing time information to others and clients when receiving time information for themselves. The selection of peers or servers for each client determines which entities will exchange time information for the purposes of synchronization. The selection of peers for a particular client bounds the possible configurations of the synchronization subnet.

NTP further bounds possible synchronization subnet configurations by providing three classes of service for time servers. These classes are multicast, procedure-call, and symmetric. These classes of service characterize the nature of the time information exchange between each client-server pair. The multicast class of service is intended primarily for use on high speed LANs where the highest accuracy is not required. In this class of service, the server transmits the time information and the client receives the information and uses an assumed delay to determine the time offset information. The procedure-call class of service is intended for operation in a file server and workstation environment. In this class, a server is willing to provide time information to the client, and the client is willing to be synchronized by the server. In this case, the server responds to requests from the client but maintains no state information reflecting the history of the exchange. The full generality of NTP is reflected in the symmetric class of service. In this class, a server exchanges time information with a peer and is willing to either provide synchronization to or receive synchronization from that peer.

The dynamic structure of the NTP synchronization subnet reflects the selection of a synchronization source made by each client at a particular moment in time. This selection is bounded by the particular peers the client exchanges information with (based on configuration setups) and the nature of the relationship between each individual pairing of client and server (as defined by the mode of service).

H.1.2 Determining clock offset

In the NTP model, timestamps are exchanged between a client and a time server. Using a single exchange of messages, individual round-trip delay and offset can be calculated for a particular peer association. Mechanisms are included that do not require that this exchange of messages be reliable. Figure H.2 represents a time-space diagram of the client's request and the servers response. The timestamps T_i, T_{i-1}, T_{i-2}, T_{i-3} represent the four most recent timestamps exchanged. Based on this, the round-trip delay δ_i and the clock offset O_i are defined in equations H.1 and H.2.

$$\delta_i = (T_i - T_{i-3}) - (T_{i-1} - T_{i-2}) \tag{H.1}$$

$$O_i = ((T_i - T_{i-3}) - (T_{i-1} - T_{i-2})) / 2 \tag{H.2}$$

It is also possible to calculate bounds on the network errors as a function of the measured delay. The true offset of the client relative to the server is O. The actual delay between the client and server (one-way) is x. Using this approach, $x + O = T_{i-2} - T_{i-3}$. Since x must be positive, $x = (T_{i-2} - T_{i-3}) - X \geq 0$, which requires $O \leq T_{i-2} - T_{i-3}$. It can also be shown that $T_{i-1} - T_i \leq O$. Therefore, $T_{i-1} - T_i \leq O \leq T_{i-2} - T_{i-3}$. It follows from this that

$$O_i - \delta_i / 2 \leq O \leq O_I + \delta_i / 2 \tag{H.3}$$

This means that the true clock offset must lie in the interval of size equal to the measured delay and centered about the measured offset.

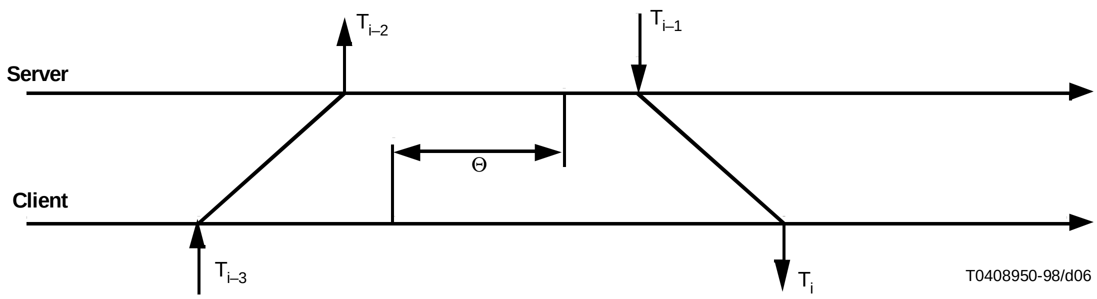


Figure H.2 – An exchange of time information

H.1.3 Model of NTP

The model for the Network Time Protocol is shown in Figure H.3. This model utilizes a phase-lock loop model for provision of a stable robust Time service. The basic components of this model are discussed further in the following subclauses.

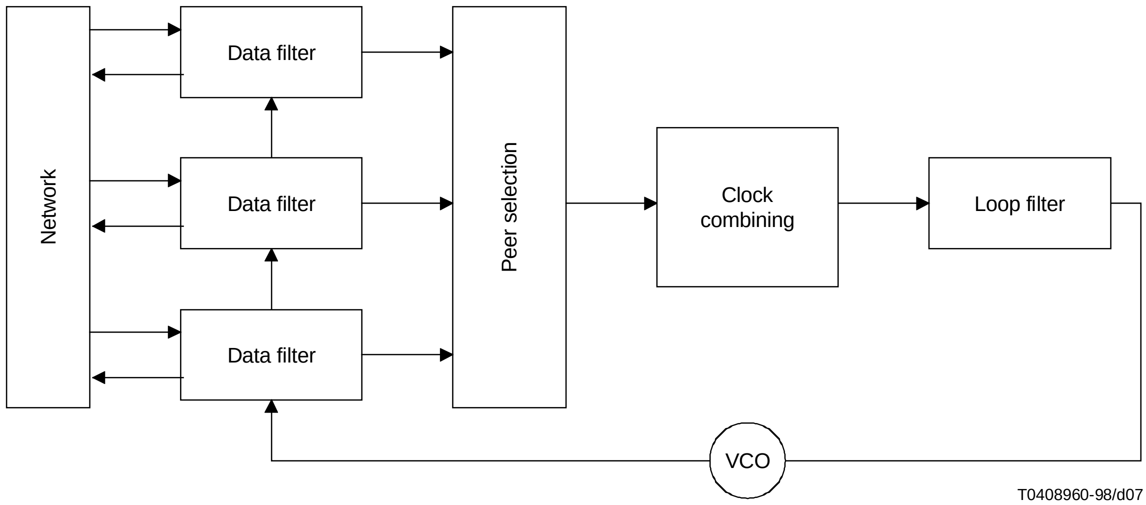


Figure H.3 – Network Time Protocol model

H.1.4 Data-filtering algorithm

For each peer association maintained by a client there is a set of the most recent delay and offset samples. The NTP data-filtering algorithm estimates the actual delay and offset between the client and the server participants in the peer association by selecting the sample with the lowest delay. To accomplish this, the most recent samples are ordered according to increasing delay. The sample with the lowest delay is selected as an estimate of the actual delay and offset.

In addition to the estimate of the actual delay and offset between a particular client and server, a filter dispersion is calculated. This filter dispersion is interpreted as a quality indicator for that peer association.

H.1.5 Peer-selection and combining algorithms

Using the estimates of delay and offset determined by the data-filtering algorithm for each client-server peer association, the peer-selection and combining algorithms are executed. These algorithms determine which peer should be selected as the synchronization source. In addition, the adjustments that need to be made to the local-clock and related protocol variables are determined.

The peer-selection algorithm is an adaptation of an agreement algorithm based on maximum likelihood statistical principles. This algorithm also makes use of the following two observations. First, the highest *reliability* is usually associated with the lowest stratum and synchronization dispersion. In addition, the highest accuracy is usually associated with the lowest stratum and synchronization distance. Using this information, this algorithm reduces the candidate list based on reliability constraints and then chooses a synchronization source based on accuracy. The peer-selection algorithm first checks each peer candidate using some basic sanity checks. Then, a list is constructed of candidate peers sorted first by stratum and then by synchronization dispersion within the same stratum level. This candidate list is pruned to reach a predetermined maximum size and maximum stratum. The candidate list then is resorted by stratum and synchronization distance. The procedure terminates when a single candidate remains or the maximum select dispersion over all the candidates remaining is less than the minimum filter dispersion of any one candidate. Now that a synchronization source has been chosen, the offset of the client relative to the source is known. The local stratum is set to one greater than the stratum of the selected peer. In addition, the synchronization distance (the sum of the total round-trip delays to the root of the synchronization subnet) and the synchronization dispersion (the sum of the total dispersions to the root of the synchronization subnet) are calculated.

H.1.6 Local clock model

NTP incorporates clock training by including a local clock model that provides corrections for both phase offset and frequency.

H.2 Distributed Time service

H.2.1 Obtaining a time value

This subclause describes how a DTS clerk or DTS server obtains time from some DTS server by performing a remote procedure call. Figure H.4 represents a time-space diagram of the messages that implement the clerk or server's remote procedure call to the server.

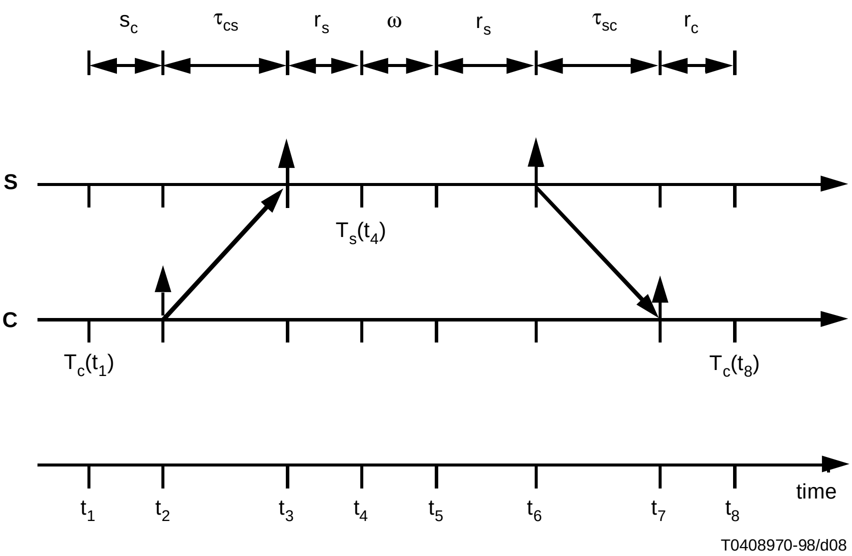


Figure H.4 – Components of Delay

The times t_k, where k = 1, 2, ...8, correspond to values of UTC, which can never be known exactly. A double arrow denotes a message transmission and a single arrow denotes a message reception.

The procedure begins with the client reading its clock in preparation for sending the request. The value of UTC at this instant is t_1 and the clock reads $T_C(t_1)$. Immediately thereafter, the clerk or server acting as client sends its request. In most systems, the request incurs some sending delay S_C as the operating system transfers it to the network adapter and as the adapter queues it for transmission. Although some component of S_C is deterministic (some minimum number of instructions must be executed in transferring the request from the client to the network), the remainder is random, depending on other system and network activities at that time. In Figure H.4, the request is actually transmitted at time t_2 and received at the server open system at time t_3 after a random propagation delay denoted by τ_{CS}.

The first action the server takes upon the arrival of a request is to note the time. As with sending, some random receiving delay r_S is incurred. Consequently, the server notes the arrival time of the request at time t_4, which is measured as $T_C(t_4)$ by the server's clock. The server then processes the request and sends a response datagram at time t_5. The client receives and time stamps this response at time t_8. The response incurs similar delays to those of the request, namely S_C, τ_{CS}, and r_C. We do not assume that the corresponding delays of the request and response messages are equal. That is, in general, $S_S \neq S_C$, $\tau_{SC}, \neq \tau_{CS}$ and $r_C \neq r_S$.

The processing delay at the server is explicitly accounted for in Figure H.4. Although you might expect the delay to be small, this may not be the case. For example, at a highly loaded server, a time request may be queued for service rather than processed immediately; in a secure system, a server may undertake the time-consuming task of signing the response message.

The processing delay is returned in the response message together with the value of the server's clock at t_4, $T_S(t_4)$, and the clock's *inaccuracy* at that instant, $I_S(t_4)$. This eliminates any need for servers to respond to requests with a high degree of timeliness. Instead, the client can compensate for processing and scheduling delays at the server as shown in the arithmetic below.

The value of the server's clock at t_4 is not useful to the client as it does not know the value of its own clock at that instant. So, the client must calculate the value of the server's clock at an instant for which it does know the value of its own clock. Let this instant be t_1 (however, we could equally have chosen t_8 or any other instant). The result of this calculation is effectively the reading of the server's clock at the instant when the client's clock read $T_C(t_1)$.

Assuming the server is non-faulty, the client knows that:

$$T_S(t_4) - I_S(t_4) \le t_4 \le T_S(t_4) + I_S(t_4)$$

Therefore, t_4 is in the range:

$$T_S(t_4) - I_S(t_4) - x \le t_1 \le T_S(t_4) + I_S(t_4) - x \qquad \text{(H.1)}$$

where, from Figure H.4 $x = S_C + \tau_{CS} + r_S$. Although x is unknown, it is in the range $0 \le x \le t_8 - t_1 - w$. Now, $t_8 - t_1$ is given by[9] :

$$t_8 - t_1 \le (T_C(t_8) + \rho - T_C(t_1))(1 + \delta_C) \qquad \text{(H.2)}$$

The clock resolution ρ, in the formula above, accounts for the discrete nature of the client's clock and the factor $(1 + \delta_C)$ accounts for its drift over the period $[t_1, t_8]$. Consequently:

$$0 \le x \le (T_C(t_8) + \rho - T_C(t_1))(1 + \delta_C) - w \qquad \text{(H.3)}$$

Combining the inequalities of equations (H.1) and (H.3), the client or server acting as client ascertains that, when its clock read $T_C(t_1)$, the server believed that UTC was in the range:

$$T_Z(t_4) - I_Z(t_4) - (T_C(t_8) + \rho - T_C(t_1))(1 + \delta_C) + w \le t_1 \le T_Z(t_4) + I_S(t_4) \qquad \text{(H.4)}$$

This estimate of the server's clock at time can be represented as a time with inaccuracy by[10] :

$$T_S^{(C)}(t_1) = T_S(t_4) - ((T_C(t_8) + \rho - T_C(t_1))(1 + \delta_C)) / 2 + w / 2$$
$$I_S^{(C)}(t_1) = I_S(t_4) + ((T_C(t_8) + \rho - T_C(t_1))(1 + \delta_C)) / 2 - w / 2 \qquad \text{(H.5)}$$

A sophisticated implementation of a server would include all known components of delay in $I_S^{(C)}(t_1)$ so as to reduce the estimated inaccuracy. For example, it could include any known components of S_S and r_S; but any component of r_S included in w requires that $T_S(t_4)$ be decremented by that amount.

Similarly, a sophisticated clerk or server acting as client could reduce $I_S^{(C)}(t_1)$ by compensating for known components of S_C, r_C, τ_{SC}, or τ_{CS}. To prevent double compensation, servers are prohibited from compensating for known components of τ_{SC} or τ_{CS}.

H.2.2 Computing a correct time

This subclause describes how a clerk or server acting as client computes a correct time from time values obtained from several servers or from the time provider, even if some are faulty. The description is presented as if the time values were obtained from other servers. However, the procedure is the same if the time values are obtained from the time provider interface.

Consider a clerk or server acting as client which has obtained M time values. For each server S_j with j = 1, 2 . . . M, the clerk or server acting as client has computed $T_j^{(C)}(t_j)$ and $I_j^{(C)}(t_j)$ where t_j corresponds to the instant t_1 in Figure H.4 but for the request sent to S_j.

9) This inequality is complicated further if one considers that a leap second might occur between t_8 and t_1. We do not discuss the modifications for this situation here, the full architecture incorporates this modification.

10) We use the notation $T_i^{(J)}(t)$ and $I_i^{(J)}(t)$ to indicate that the time and maximum error is an estimate of clock i by client J obtained as described in this subclause.

Calculating a correct time is feasible only if all the time values pertain to the same instant. So the first task is to translate these values to correspond to one synchronization instant, which is denoted by t_S. Any choice of t_S is adequate, the only requirement being that the value of the clerk or server acting as client's clock $T_C(t_S)$ is known. Least inaccuracy is achieved if t_S is close to the current time. All calculations that the clerk or server acting as client performs are in terms of $T_C(t_S)$ since t_S itself is never known. Translating the S_j time value to t_S is achieved by the following equations:

$$T_j^{(C)}(t_S) = T_j^{(C)}(t_j) + T_C(t_S) - T_C(t_j)$$
$$I_j^{(C)}(t_S) = I_j^{(C)}(t_j) + T_C(t_S) - T_C(t_j))\delta_C \quad \text{(H.6)}$$

Note that the inaccuracy of each time value increases to compensate for the maximum possible drift of the clerk's clock over the period $[t_j, t_S]$.

The basis of the calculation is now described. Assume for now that all servers are correct. Therefore, all the *M* time intervals contain UTC. The narrowest correct time that the clerk or server acting as client can compute is simply the intersection of these *M* time intervals. An example is shown in Figure H.5. We point out that this intersection is the smallest interval containing UTC that the clerk or server acting as client could possibly compute from the given information (the *M* time values).

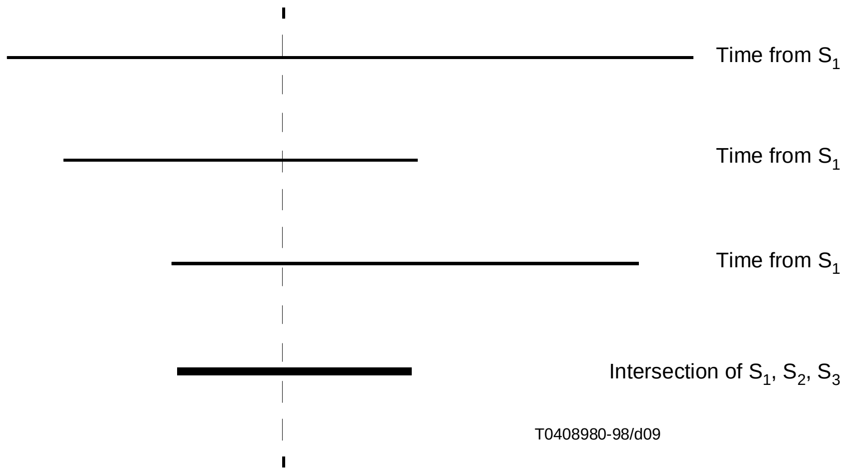


Figure H.5 – Computing the best correct time

But what if some servers are faulty? There may be no intersection or, even worse, the intersection may not contain UTC. The actual algorithm embellishes the idea of a simple intersection to accommodate the possibility of faulty servers. It is arrived at by the following reasoning.

The narrowest correct time that the clerk or server acting as client could possibly compute is the intersection of all the correct time intervals; any point on the real line contained in all the correct time values could potentially be the true value of UTC at the instant when the clerk's clock reads $T_C(t_S)$. However, the clerk or server acting as client does not know which servers are correct and which (if any) are faulty.

Let us assume that at most *f* servers are faulty. In this case, any point on the real line contained in at least $M - f$ of the time intervals could be a point in all the correct intervals, and hence could be UTC. While this is the smallest set of points guaranteed to contain UTC, it does not necessarily constitute a single interval as shown by the example in Figure H.5. Rather than considering the time to be multiple intervals, the clerk or server acting as client takes the correct time to be the smallest single interval containing all points in at least $M - f$ of the intervals.

The complexity of computing the narrowest correct time may at first appear to be high. However, it is simple and fast with a suitable data structure and algorithm. We now describe this computation algorithmically:

1) Arrange the endpoints of the *M* time values into a list. The list is of length 2*M* as each time value contributes two elements to the list: $T_j^{(C)}(t_S) - I_j^{(C)}(t_S)$ and $T_j^{(C)}(t_S) + I_j^{(C)}(t_S)$.
2) Mark each end point to indicate if it is a minimum or maximum endpoint.

3) Sort the list according to the values of the endpoints in ascending order. In the case where two or more endpoints take on the same value, those corresponding to lower bounds must precede those corresponding to upper bounds in the list; that is, if:

$$T_j^{(C)}(t_S) - I_j^{(C)}(t_S) = T_k^{(C)}(t_S) + I_k^{(C)}(t_S)$$

then $T_j^{(C)}(t_S) - I_j^{(C)}(t_S)$ must precede $T_k^{(C)}(t_S) + I_k^{(C)}(t_S)$ in the list.

4) Set the initial value of f.

5) Scan the list in ascending order to find the first endpoint that is contained in at least $M - f$ intervals. This point corresponds to the minimum value of the correct time interval.

6) If no such point is found, then there are more than f faulty servers. Increase f by one and go back to step 5). If a minimum value has been found, then continue with the current value of f.

7) Scan the list in descending order to find the first endpoint that is contained in at least $M - f$ intervals. This point corresponds to the maximum value of the correct time interval.

Computing the minimum and maximum endpoints of the narrowest correct time interval is equivalent to computing a time and inaccuracy. We denote this computed time and inaccuracy by $CT_C(t_S)$ and $CI_C(t_S)$, respectively.

H.2.3 Adjusting the clock

Once a clerk or server acting as client or server has computed a correct time $CT_C(t_S)$ and inaccuracy $CI_C(t_S)$, it adjusts its clock in accordance with the computed time. This subclause describes how that adjustment is made so that the clock never jumps forward or backward.

Recall that, at the synchronization point t_S, the clerk or server acting as client's clock reads $T_C(t_S)$. Consequently, the clerk or server acting as client must adjust the clock by $CT_C(t_S) - T_C(t_S)$.

Since time always advances, the clock too must always advance; that is, increase monotonically. A clerk or server acting as client should not set its clock backward to adjust a fast clock. When the clock is slow, it is desirable, but not strictly necessary, for the clerk or server acting as client to adjust the clock gradually so that users do not experience a sudden forward jump in the time. To satisfy these requirements, a clerk or server acting as client adjusts a fast clock by slowing it down so that UTC catches up, and a slow clock by speeding it up so that it catches up to UTC.

The procedure by which gradual monotonic adjustments are made depends on the underlying hardware and software constituting the clock. We describe this procedure for one particular realization of a clock, one common to many computer systems. Other realizations of clocks may use different procedures as long as they guarantee that a non-faulty clock always contains UTC in its interval.

A clock consists of memory containing its current measure of UTC and a hardware timer that periodically interrupts the processor. Normally, the routine servicing these tick interrupts increments the clock's memory by the resolution ρ, causing the contents of the memory to increase at (approximately) the same rate as UTC.

To adjust the clock, the clerk or server acting as client changes the amount by which the clock increments at each tick, increasing the amount if $CT_C(t_S) \geq T_C(t_S)$; otherwise, decreasing it. We denote the amount by which the nominal tick increment ρ is adjusted by ε. Suppose that the clerk or server acting as client increases the tick increment to ρ + ε for some number of ticks, say n. If ε is positive, the clock gains $n\varepsilon$ seconds, while if it is negative, the clock loses that amount. So to gain $CT_C(t_S) - T_C(t_S)$, the clerk or server acting as client modifies the tick increment by ε for:

$$N = (CT_C(t_S) - T_C(t_S)) / \varepsilon \qquad \text{(H.7)}$$

ticks. (If $CT_C(t_S) - T_C(t_S) < 0$, then ε is negative and the clock loses rather than gains.)

The value of ε is an implementation constant. However, the specification imposes two restrictions: to ensure that the clock is monotonic, ε must be greater than or equal to –ρ; and, because the adjustment compensates for drift, the adjustment rate must be greater than the drift, or $|\varepsilon| > \rho\delta$. If, in addition, $|\varepsilon|$ is small compared with ρ, users will barely perceive the adjustment, if at all.

H.2.4 Determining the maximum error

Our realization of the clock does not automatically measure inaccuracy as it does time. Instead, the clerk or server calculates the inaccuracy whenever the clock is read. This subclause presents the formula for making that calculation.

The inaccuracy of a clock consists of four components:

Base inaccuracy	The inaccuracy at the synchronization point t_S, which is given by $I_C(t_S) = CI_C(t_S) + \lvert CT_C(t_S) - T_C(t_S) \rvert$.
Drift increase	This is the most the base inaccuracy has increased due to drift. At time t, the drift increase is $(T_C(t) - T_C(t_S))\, \delta_C$.
Adjustment decrease	The amount by which the inaccuracy is decreased by adjusting the clock. For every tick for which the clock is adjusted by ε, inaccuracy is decreased by $\lvert \varepsilon \rvert$. Assuming adjustment begins at synchronization point t_S, the adjustment decrease is $\lvert ((T_C(t) - T_C(t_S))\, \varepsilon) / (\rho + \varepsilon) \rvert$ but no more than the total of $\lvert N\varepsilon \rvert$.
Clock resolution	The resolution ρ is included in the inaccuracy to reflect the amount by which UTC and the clock diverge between two consecutive ticks. It is scaled to account for drift.

Combining these four components gives the formula for calculating the inaccuracy at any time after t_S as:

$$I_C(t) = CI_C(t_S) + \lvert CT_C(t_S) - T_C(t_S) \rvert + (T_C(t) - T_C(t_S))\, \delta_C \\ - \min \{(T_C(t) - T_C(t_S)) / (\rho + \varepsilon), N\} / \varepsilon \rvert + (1 + \delta_C)\rho \qquad \text{(H.8)}$$

where N is given by equation H.7. Note that implementations can reduce the contribution of resolution to $0.5(1 = \delta_C)\rho$ if the time reported by the clock is increased by $0.5(1 = \delta_C)\rho$ over its actual value.

We note that it may not be possible to schedule the beginning of the adjustment at precisely t as required by the formula for adjustment decrease in the preceding list. If this is the case, the formula for calculating the inaccuracy must take into account the number of ticks between the synchronization instant and the instant at which adjustment actually begins. We now show one way to do this.

Call the instant at which adjustment begins the base time t_b. The clock reads $T(t_b)$ at this instant. The inaccuracy at t_b, which we call the base inaccuracy, is the inaccuracy at t, increased to account for the drift over the time t to t_b. So:

$$I_C(t_b) = CI_C(t_S) + \lvert CT_C(t_S) - T_C(t_S) \rvert + (T_C(t_b) - T_C(t_S))\, \delta_C$$

(Higher-order terms are ignored in this equation.) As the base inaccuracy now includes drift from t_S to t_b, the drift increase need only include drift thereafter. So at time t, it is given by $(T_C(t) - T_C(t_b))\, \delta_C$.

The adjustment decrease is calculated by measuring the number of ticks that have elapsed since adjustment began as long as adjustment began at t_b. So, it is given by $\lvert (T_C(t) - T_C(t_b))\, \varepsilon)/(\rho + \varepsilon) \rvert$ but still no more than the total of $\lvert N \varepsilon \rvert$. Thus, equation H.8 can be expressed in more general terms by:

$$I_C(t) = CI_C(t_S) + \lvert CT_C(t_S) - T_C(t_S) \rvert + (T_C(t_b) - T_C(t_S))\, \delta_C + \\ (T_C(t) - T_C(t_b))\, \delta_C - \min \{(T_C(t) - T_C(t_b)) / (\rho + \varepsilon), N\} / \varepsilon \rvert + (1 + \delta_C)\rho$$

H.2.5 Local faults

Despite periodic synchronization, an error (either transient or permanent) on a system can cause the clock on that system to become faulty. A faulty clock can be detected at synchronization by comparing the interval of the clock to the computed interval. If they do not intersect, the local clock is faulty.

The usual mechanism for adjusting the clock and computing the inaccuracy will correct a faulty clock. However, if the amount by which the clock is in error is large (say, several days), it may be undesirable to adjust the clock gradually to correct the error; instead, it may be preferable to set the clock to the computed time.

To facilitate this, there is a management attribute to control whether a faulty clock is set to the correct time or adjusted monotonically. This parameter is called *errorTolerance*. A faulty clock is set rather than monotonically adjusted if the separation between its interval and the computed interval is greater than *errorTolerance*. This is expressed quantitatively by the following:

$$| CT_C(t_S) - T_C(t_S) | - CI_C(t_S) - I_C(t_S)) \geq errorTolerance$$

By specifying the error tolerance of a faulty clock in this way, it is easy to accommodate the boundary conditions; namely, to never reset a faulty clock (*errorTolerance* = ∞) or, to immediately reset a faulty clock (*errorTolerance* = 0).

H.2.6 Configuration

The number of servers from which a clerk requests the time during each synchronization is determined by a management attribute called minServers. The Time Service must be configured so that there are enough servers to satisfy the needs of every clerk. In addition, it is desirable for servers to be located close to the clerks to which they provide the time. This minimizes communication delay that contributes to inaccuracy.

While small cells can satisfy these criteria with a single set of servers serving all clerks, large cells must be configured with many more servers than required for any one clerk. Consequently, the architecture partitions servers into sets with each set serving a subset of clerks.

To simplify management partitioning, we exploit the assumption that most systems are connected to LANs. Each LAN contains a (possibly empty) set of servers called the local set. Normally, there are sufficient servers in a local set to satisfy the needs of all clerks on the LAN of that set. If this is the case, clerks obtain the time from servers in their respective local sets. They discover these servers through RPC service profiles using the algorithm described in C.3.2. This algorithm makes it possible to autoconfigure the Time Service local sets.

While this arrangement is well suited for configuring local sets, it suffers two shortcomings:

1) If none of the servers in a local set have a time provider, an operator must periodically reset the time at *f* servers in this set.

2) Clerks whose local sets contain insufficient servers have no mechanism to discover additional servers.

To overcome these shortcomings, we provide an additional set of servers that are available throughout the cell. We call this set the global set and the servers of this set we call global servers. A global server is usually a member of some local set, but this is not required.

A clerk accesses global servers only if there are fewer servers in its local set than the number it requires for synchronization. A server accesses global servers if it does not have a time provider and either of the following two conditions holds:

1) There are fewer servers in the local set than the number it requires for synchronization.

2) The server is a courier.

Couriers are servers that import time from the global set into the local set. This is useful when none of the servers in the local set have time providers but some servers in the global set do. The mechanism by which a server becomes a courier is described in C.3.4.

Note that global servers do not synchronize with each other explicitly. A global server synchronizes with another global server only if the two are in different local sets and the first does not have a time provider.

H.2.7 Couriers

It is likely that some local sets will be configured without any servers that have time providers. With this configuration, an operator must periodically mimic a time provider to prevent inaccuracies in such a local set from becoming excessively large.

However, there may be some global servers in the cell with close proximity to time providers from which the servers in a local set could obtain accurate time. The architecture provides a mechanism for this with a design that limits the load placed on global servers and is easy to manage at the expense of reduced fault tolerance. With this mechanism, only those servers designated as couriers synchronize with global servers rather than all servers without time providers.

A server becomes a courier by a combination of an election mechanism and an attribute called courierRole which takes one of the three values: courier, non-courier, or backup-courier. A server with courierRole set to non-courier never becomes a courier; one with courierRole set to courier is always a courier; a server with courierRole set to backup-courier is, in general, not a courier but becomes one if:

1) There are no servers in the local set with courierRole set to courier .
2) It is the server whose security UUID (the UUID associated with the server's security principal) precedes all others with courierRole set to backup-courier.

Servers exchange their courierRole values in time request RPCs. Servers with courierRole set to backup-courier must redetermine whether or not they are couriers whenever they add an entry to, or remove one from, their lists of local servers.

H.2.8 Determining the next synchronization

A clerk determines when to synchronize its clock by attempting to bound its inaccuracy. The desired bound on the inaccuracy is specified by the management attribute maxInacc.

From the computed time and inaccuracy $CT_C(t_S)$ and $CI_C(t_S)$, the clerk can calculate the time at which its inaccuracy will reach the value of maxInacc. This is given by:

$$T = CT_C(t_S) + (\text{maxInacc} - CI_C(t_S)) / \delta_C$$

To keep $I_C(t) <$ maxInacc, the clerk must synchronize before its clock reads this time.

Note that the Time Service does not guarantee any bound on $CI_C(t_S)$. Consequently, it is possible that maxInacc $CI_C(t_S)$ is small or even negative. To prevent a clerk from synchronizing continuously, we require that the time between synchronizations be more than a minimum value which is specified by the management attribute syncHold.

Unfortunately there are likely conditions where, using the approach described in the previous paragraph, all clerks will choose the same instant at which to synchronize, causing bursty loads on the network and servers. To avoid these, some randomness is introduced into the times at which clerks synchronize.

The following steps describe how to schedule the next synchronization based on the computed time and the two parameters maxInacc and syncHold:

1) Compute the time for the inaccuracy to grow to maxInacc. This is given by the following:

$$D = (\text{maxInacc} - CI_C(t_S)) / \delta_C$$

2) If D < *syncHold*, go to step 4). Otherwise, continue with the next step.
3) Draw a random number, R, uniformly distributed over [*D*/2, *D*]. Go to step 5).
4) Draw a random number, R, uniformly distributed over [(3*syncHold*)/4, (5*syncHold*)]/4]
5) Schedule the next synchronization so that it can complete before the clock reads $CT_C(t_S) + R$.

H.2.9 Maintaining the server lists

To learn of new local servers, clerks must periodically perform an RPC import of the local set. To learn of new global servers, those clerks using global servers must periodically perform RPC imports of the global set.

The mechanism by which a clerk is forced to perform these imports is for it to periodically flush its entire lists of global and local servers. As can be seen from the synchronization procedure in C.4.2, this causes the clerk to import the local set and, if necessary, the global set the next time it synchronizes.

Flushing is done periodically. The period between flushes is specified by the management attribute cacheRefresh.

If the server synchronized with the time provider, it schedules the next synchronization for the time specified by the time provider. (The nextPoll field of the TPctlMsg returned by a call to the ContactProvider function of the Time Provider interface.)

If the server synchronized with other servers, or if it aborted the synchronization because there were too few servers, it determines the next synchronization in the same way that clerks do as specified in C.4.3.

H.2.10 Checking for faulty servers

To detect and report faulty servers in a timely fashion, servers must periodically obtain time from all other servers in the local set and check that their intervals intersect.

The procedure is the one used for synchronizing with other servers, as described in C.5.3.2, with the following changes:

- No global servers are queried.
- The procedure is not aborted if less than minServers –1 are queried.
- The step to adjust (or set) the clock is not carried out.
- The step to schedule the next synchronization is not carried out.

Checking for faulty servers is initiated by a periodic timer whose average period is specified by the management attribute checkInt. After initialization and after completing the checking procedure, the timer is set by drawing a random number in the range [(3checkInt)/4, (5checkInt)/4].

NOTE – Since checking is also done when a server synchronizes with other servers, the timer is restarted with a value that is a random number in the range [(3*checkInt*)/4, (5*checkInt*)/4].

H.3 Probabilistic clock synchronization

The Probabilistic Clock Synchronization (PCS) approach to achieving time synchronization between two processes addresses:

- an assumption about the PCS algorithm;
- a technique on how to read a remote clock with a specified accuracy;
- a method on how to keep clocks synchronized;
- a method on how to adjust a software clock.

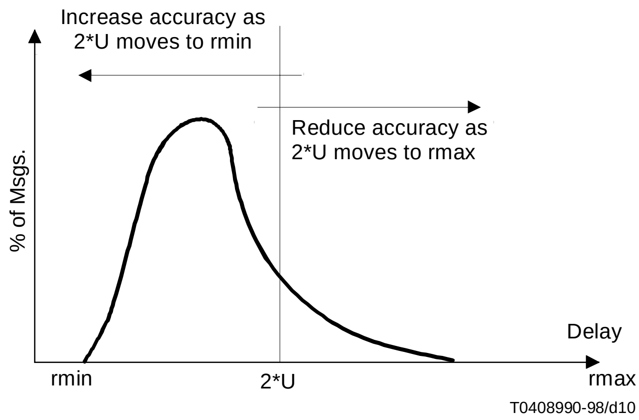
PCS does not address:

- a synchronization model;
- the achievement of fault tolerance;
- the solicitation of timestamps;
- a service interface.

All the equations quoted in this contribution can be found in [2] and [3].

H.3.1 Assumption

In distributed systems, process to process communication is affected by an unpredictable random real-time delay. This makes the task of synchronization difficult. From the study on message delay for process communication in distributed systems [2], the distribution of the message delay has a shape resembling that illustrated in Figure H.6. The distribution has a maximum density at a mode point between the minimum delay (rmin) and the median delay, usually close to rmin, with a long thin tail to the right. The rmin value can be computed by counting the time needed to prepare, transmit and receive an empty message in the absence of transmission errors and system load. This rmin value contributes to the algorithm of Probabilistic Clock Synchronization (PCS).



rmin Minimum round-trip delay
rmax Maximum round-trip delay
2*U Acceptable max. round-trip delay

Figure H.6 – Message delay

The synchronization algorithm is based on the following assumptions:

– clocks have a much finer resolution than the time intervals;

– min is half of the minimum round-trip message delay rmin;

– ρ is the maximum drift rate;

– a local clock implemented by software is adjustable.

When referencing a time of a process, a notation of CTx, where CT is the time value of the clock at process x, is used. For an estimated time of another process, CTx(y) is used to denote the time value of process y estimated at process x. For instance, a time value of a clock at process S is denoted as CTs, whereas a time value of process S estimated by C is CTc(s).

H.3.2 Reading a remote clock

Assuming two processes, S and C, C sends a request message for a time from S. When S receives the request, it replies with a time message which is timestamped with its clock's time. If process C does not receive any response within a predefined waiting period, the attempt at reading the clock S fails. The time at S, when C receives the replies message from process S, can be calibrated as:

$$CTs + min * (1 - \rho)$$

and

$$CTs + 2 * Dc * (1 + 2 * \rho) - min * (1 + \rho)$$

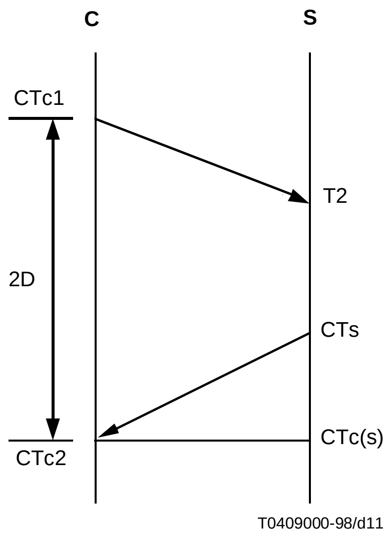
or it can be expressed as an interval of:

$$CTs + min * (1 - \rho) \leq CTc(s) \leq CTs + 2 * Dc * (1 + 2 * \rho) - min * (1 - \rho)$$

where:

CTs	The Clock value at S
Dc	The measured half round-trip delay at C
ρ	The maximum drift rate of the clock at C
min	Half of the minimum round-trip message delay rmin measured at C
CTc(s)	C's estimate of the clock value of S at the same instance on C

Figure H.7 illustrates the reading clock path.



CTc1 C requests time from S
T2 S receives request
CTs S replies with timestamp
CTc2 C receives response
CTc(s) Time at S when C received reply
2D Round-trip delay measured at C

Figure H.7 – Reading a remote clock

Also, it can be shown that the estimated clock value at S by C @CTc(s) is:

$$@CTc(s) = CTs + Dc * (1 + 2 * \rho) – min * \rho$$

and the maximum error of the estimated value (or inaccuracy) is:

$$\varepsilon = Dc * (1 + 2 * \rho) – min$$

In practice, the value Dc can be obtained by process C taking the time when it sends out the request message and the instance when it receives the replies message. The maximum drift rate is normally given by the clock manufacturer.

H.3.3 Reading a remote clock with a specified precision

For a process C to achieve a certain accuracy (i.e. the maximum reading error), say 'ε', process C must discard any reading attempts for which the measured actual round-trip delay '2 * Dc' is greater than the specified maximum round-trip delay '2 * Uc'. 'Uc' can be calculated based on the following equation:

$$Uc = (1 – 2 * \rho) * (\varepsilon + min)$$

When process C observes a successful round-trip, the process C is said to reach rapport with process S.

For completion of precision measurement, one has to consider the waiting period 'W' and maximum attempt 'k' retry. The former is to ensure that both processes stay correct, and any transient network traffic bursts do not affect their communication. It is evident that the higher the accuracy (i.e. the lower the value of 'ε'), the closer the acceptable maximum round-trip '2 * Uc' is to reaching the '2 * min' value. A higher probability of reaching rapport between processes can be achieved by attempting to read more messages. On the other hand, if the maximum round-trip delay '2 * Uc' is chosen to be closer to the maximum message delay, the algorithm becomes deterministic. This also results in lower accuracy and few attempts at sending messages.

Hence as '2 * U' is chosen closer to the 'rmin' as illustrated in Figure H.7, one can achieve higher precision on the clock reading, but it would also result in higher frequency of failure in reaching rapport. This may be accomplished by taking more 'k' attempts to reach rapport. As U tends to max, one would result in less accuracy and relatively fewer 'k' retry attempts in order to reach rapport. The average number of messages 'n' required for achieving rapport can be shown as a function of probability of not achieving precision 'ρ', that is:

$$n = 2 / (1 - \rho)$$

H.3.4 Keeping clocks synchronized

In order to keep a clock at C synchronized with S, C attempts periodically to reach rapport with S. Each attempt at rapport consists of at most 'k' attempts, where readings are separated by 'W' clock units. Intuitively, W > 2 * U. In the case where 'k' attempts have failed, S and C cannot be synchronized with the specific accuracy.

The maximum real-time delay, 'DNA', between a rapport and the next attempt at rapport can be expressed as:

$$DNA = ((1 - \rho) / \rho) * (sc - y) - k * W$$

where 'sc' is the deviation between S and C. It is also shown that 'sc' should be:

$$sc \geq Uc - min + \rho * k * (1 + \rho) * W$$

H.3.5 Clock adjustment

Given a local clock at process C with adjustable speed implemented in software, the value of clock at C can be represented as the sum of the local hardware clock H at C and a periodically computed adjustment function A; therefore:

$$C(t) = H(t) + A(t)$$

and

$$A(t) = m * H(t) + N$$

To avoid local clock discontinuities (i.e. jumps), 'A(t)' must be a continuous function of time, and for simplicity, A is a linear adjustment function. The value of 'm' and 'N' can be obtained as follows:

$$m = (CTc(s) - CTc) / \alpha$$

and

$$N = CTc - (1 + m) * H$$

'α' is a clock with a positive error which is defined as the time units following rapport for which a process C synchronized with S. After the adjustment period which corrects for 'α' has elapsed at C, the clock time CTc becomes:

$$CTc(at\ rapport) = CTc(s) + \alpha$$

Process C can be allowed to run again at the speed of local hardware clock until the next rapport by setting:

$$m = 0$$

and

N = local time of C at the end of the time adjustment period-hardware clock at the end of the same period

Notice that m and N are changed at the beginning and the end of each time adjustment period.

H.3.6 Conclusion

The Probabilistic Clock Synchronization (PCS) algorithm gives a controllable accuracy when reading a remote clock value. However, it does not guarantee that a processor can always read a remote clock with a specified accuracy. A process can read the clock of another process with a given accuracy with a probability as close to one as desired, by retrying a sufficient number of times while reaching rapport. When a rapport is achieved, a process knows the actual reading accuracy obtained. The algorithm also shows how to adjust a clock continuously within the time adjustment period and how to calculate the time for next rapport.

H.4 Additional references

Additional tutorial and background material can be found in the following sources:

[1] ISO/IEC JTC 1/SC 21/WG 7 N352 – Requirements on the time synchronization standard.

[2] CHRISTIAN (F.) Probabilistic Clock Synchronization, *Distributed Computing* (1989), 3:146-158.

[3] SCHMUCK (F.), CHRISTIAN (F.) Continuous clock amortization need not affect the precision of a clock synchronization algorithm, *Proceedings of the Ninth Annual ACM Symposium on Principles of Distributed Computing*, 1990.

[4] Digital Equipment Corporation, Digital Time Service Functional Specification – Version V1.0.

[5] MILLS (D.L.) Network Time Protocol (Version 3) Specification and Implementation. RFC 1305, March 1992.

[6] MARZULLO (K.A.) Maintaining time in a distributed system: An example of a loosely coupled distributed service, Ph. D. dissertation, Stanford University, Stanford, CA., February 1984.

[7] MILLS (D.L.) Internet Time Synchronization: The Network Time Protocol IEEE Transactions on Communications, Vol 39. No. 10, October 1991.

[8] CCIR, Standard-frequency and time-signal emissions (Recommendation 460), in XIIth Plenary Assembly CCIR, New Delhi, India, 1970, III, p.227, ITU, Geneva, Switzerland, 1970.

[9] CCIR, Detailed instructions by Study Group 7 for the implementation of Recommendation 460 concerning the improved Coordinated Universal Time (UTC) system, valid from 1 January 1972 in XIIth Plenary Assembly CCIR, New Delhi, India, 1970, III, p.228 a-d, ITU, Geneva, Switzerland, 1970.

[10] 1003.1b-1993: POSIX, Institute of Electrical and Electronics Engineers.

[11] OSF DCE 1.0 (Update 1.0.3), OFS DCE Application Environment Specification/Distributed Computing – Time Services.

Annex I

Time user service

(This annex does not form an integral part of this Recommendation | International Standard)

The time user service component of a generic time service provides users with access to time information. This portion of a generic time service is considered a local issue and outside the scope of the time management function. However, the provision of a time user service is important, and a few relevant issues are discussed in this annex.

I.1 Accuracy and precision

The time management function imposes no requirements for clock accuracy and precision on implementors of the time management function. However, in a practical sense, particular systems usually have precision and accuracy requirements. It is up to the system designer to identify the requirements of a particular system and specialize as needed.

I.2 User time formats

There are many existing formats for time information. This Recommendation | International Standard defines a single representation of time. However, in reality, a time user service may be required to convert between many different formats. This conversion includes such concepts as time zones and local time adjustments (Daylight Savings Time). This process of conversion, especially as related to leap seconds, must be accommodated by the time user service.

Also related to user time formats is the question of representation. The representation defined in the time management function is a good representation for coordinating time information effectively, for both time synchronization and general applications. However, many applications may only require a simple time value without the associated accuracy information. This is provided locally by the time user service.

I.3 Leap seconds

Leap seconds are adjustments to a standardized atomic time reference to account for slight variations in the rotation of the earth. These seconds are added to or subtracted from the UTC timescale as determined necessary. This is generally done at the end of June or December.

Two approaches are common for handling leap seconds when dealing with a representation of time that is a count of some unit of time (seconds) since an epoch. In the first, the time scale is a constant count of seconds that have occurred since the epoch. This represents a continuous timeline, in other words, the timeline generated by this representation is monotonic and regular. Leap seconds are noted for conversion to other time formats and to time of day. However, the occurrence of a leap second does not impact the basic representation. This approach simplifies the task of calculating intervals of time because there are no discontinuities. However, it somewhat complicates the task of calculating a precise calendar date. There must be knowledge of how many leap seconds have occurred prior to that particular date. One approach to address this issue is to include the count of leap seconds in the representation.

In the second approach, the occurrence of a leap second causes a discontinuity in the timeline. A one second unit step is applied to the count. This complicates the task of calculating time intervals because one needs to know if a leap second occurred sometime during the interval of interest. This approach simplifies the task of converting to other time formats and to time of day.

Another issue on this topic is what to do when a leap second occurs. Mechanisms need to be provided to give advance warning that a leap second is about to occur. In addition, decisions need to be made about how that event will be handled when it occurs. A decision needs to be made on how to apply the correction. It can be applied as a single step or it can be gradually phased in over time.

I.4 Time values for event ordering

Another time user service issue is that of unambiguous time stamps. Often, time values are used for the ordering of events. In this case, no two time stamps can be the same. There are a number of approaches for addressing this issue.

The first approach is to provide a local clock that increments faster than it can possibly be accessed. In many workstations today, the clock increments at 1 microsecond; however, it may take 40 microseconds to actually read the clock.

A second approach is to use the insignificant or meaningless bits in the time representation to provide ordering within a single increment of the clock. In this approach, if a clock increments every 10 milliseconds, the bits representing single microseconds can be used for ordering within the 10 millisecond interval without providing valid time information.

The final approach involves the use of an additional field attached to the time stamp as an ordinal. This additional field is then used for ordering within the clock increment.

These are all approaches that can be used locally within the time user service. The selection of one of these approaches is considered outside the scope of this Recommendation | International Standard.

ITU-T RECOMMENDATIONS SERIES

Series A	Organization of the work of the ITU-T
Series B	Means of expression: definitions, symbols, classification
Series C	General telecommunication statistics
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	TMN and network maintenance: international transmission systems, telephone circuits, telegraphy, facsimile and leased circuits
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	**Data networks and open system communications**
Series Y	Global information infrastructure
Series Z	Programming languages