

TECHNICAL
REPORT

TC 184
ISO
TR 12178

First edition
1994-04-01

**Industrial automation — Time-critical
communications architectures — User
requirements**

*Automatisation industrielle — Architectures des communications en
temps réel — Prescriptions des utilisateurs*



Reference number
ISO/TR 12178:1994(E)

Contents

	Page
Foreword	iii
Introduction	iv
Section 1: General	1
1.1 Scope	1
1.2 Normative references	1
1.3 Definitions	1
Section 2: User requirements for time-critical communications in discrete parts manufacturing	5
2.1 Introduction	5
2.2 Application models	7
2.3 Integration issues	9
2.4 Message differentiation and PDU attributes	12
2.5 Timing related issues	14
2.6 Resilience	20
2.7 Topology aspects	22
2.8 Management and sovereignty	22
Section 3: User requirements for metrics and benchmarks	24
3.1 Introduction	24
3.2 Standard metrics	25
3.3 Benchmarks for TCCA	29
3.4 Quality of Service Parameters	36

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The main task of technical committees is to prepare International Standards. In exceptional circumstances a technical committee may propose the publication of a Technical Report of one of the following types:

- type 1, when the required support cannot be obtained for the publication of an International Standard, despite repeated efforts;
- type 2, when the subject is still under technical development or where for any other reason there is the future but not immediate possibility of an agreement on an International Standard;
- type 3, when a technical committee has collected data of a different kind from that which is normally published as an International Standard ("state of the art", for example).

Technical Reports of types 1 and 2 are subject to review within three years of publication, to decide whether they can be transformed into International Standards. Technical Reports of type 3 do not necessarily have to be reviewed until the data they provide are considered to be no longer valid or useful.

ISO/TR 12178, which is a Technical Report of type 3, was prepared by Technical Committee ISO/TC 184, *Industrial automation systems and integration* Sub-Committee SC 5, *Architecture and communications*.

For an explanation of the relationship of this Technical Report to normative aspects of time-critical communications architectures that are dealt with in International Standards, see the Introduction.

Introduction

0.1 Summary

It is becoming apparent that in their present form the OSI standards cannot readily accommodate time-critical applications. The network architectures which have been standardized so far were primarily intended for general traffic and are not always capable of providing adequate performance and resilience for time-critical communications, especially where time-critical and non-time-critical traffic coexists. In particular in many CIM and control installations there appears to be a requirement for an intermediate network, between general enterprise-wide networks and the fieldbus-type networks. This intermediate network should carry both bulk data transfers and time-critical messages, and be able to operate over considerable distances and in hostile environments.

A task force was therefore set up under ISO/TC 184/SC 5/WG 2 to look at the requirements for a time-critical communications architecture, concentrating particularly on the requirements in intermediate networks and to prepare a technical report on that topic. This work could be extended to take into account requirements for time-critical communications in fieldbus.

The focus of this report is to identify the communication needs of tightly coupled control systems. This effort is intended to complement efforts which have focused on factory information networks (e.g. MAP) which are characterized by high throughput, large packet service between information processors (e.g. mini computers) and control processors (e.g. special purpose controllers). The effort is also intended to complement efforts to identify sensor networks (e.g. fieldbus) which concentrate on providing a low complexity interface to simple factory floor sensors, actuators and other devices.

There is no clear consensus on whether the time-critical communications architecture should be connection-oriented or connectionless. Either solution is acceptable providing that the requirements are met; however, some requirements may be easier to meet with one of these solutions than with the other. Since MMS uses a connection-oriented session layer then the overall service may appear connection-oriented even if the network and transport layers are connectionless.

This report summarizes the user requirements for time-critical communications systems that have been identified by the task force and provides a short explanation of each of the requirements. Consideration is also given to the requirement for standard metrics and benchmarks to allow users to assess the suitability of particular architectures for their applications.

There is still no all-embracing definitive quantification of the meaning of time-critical communication; however, communication is time-critical if the application process sending a message requires it to be received (or received and acted upon, or

received and acted upon and confirmed) within a certain time after it submits its 'send' request to the system. The network time constraints are determined by the requirements of the application, and there is no intention of defining absolute time constraints for any particular architecture. Indeed, many issues such as predictability are definable only in statistical terms. In all the following discussions it is assumed that the system under consideration is primarily concerned with dynamic sequencing, i.e. there is variable loading on the network because the applications are event driven. It is also assumed that practical applications will require the coexistence of time-critical and non-time-critical traffic on a single network.

By concentrating primarily on the requirements for event driven communications scenarios, the requirements for state driven scenarios and those where there is a mixture of state driven and event driven communications are also satisfied, since the state driven communications can be considered to be special, more predictable, cases of event driven communications.

A network which provides time-critical communications system functionality may, in certain applications, or in smaller systems, provide functionality of information and sensor networks as well. However, in no way should this possibility compromise its capability to serve as a control network.

The time-critical communications architecture shall provide services which allow for efficient interworking with systems which use MMS (ISO/IEC 9506). The time-critical communications architecture should be applicable in environments where the full 7 layer OSI stack is implemented but in certain applications reduced stack architectures may be more appropriate.

In certain applications requiring time-critical communications it may not be essential for the TCC architecture to meet all the identified requirements. Nevertheless, it is clear that all the requirements are essential for some applications.

0.2 Time windows

The term 'time-critical' is used to represent the presence of a time window, within which one or more specified actions is required to be completed with some defined level of certainty. Failure to complete specified actions within the time window risks failure of the applications requesting the actions, with attendant risk to equipment, plant and possibly human life.

While much agreement exists with respect to the presence of the window, the duration of the window continues to be an application-dependent matter. Time constants associated with batch and continuous manufacturing range from milliseconds to seconds, and sequencing times in discrete manufacturing range from microseconds to hours. It is not possible to define, in absolute terms, a universally acceptable time window for all manufacturing applications. Moreover, actual network performance is substantially perturbed in some implementations by dynamic

factors such as loading or collisions or token loss, and by static variables such as the number of participating nodes and message frame size. The user has to take all such factors into account when establishing the time window required for a particular implementation. Network performance changes resulting from such factors should not compromise the ability of the network implementation to meet the required time window.

However, it remains essential to provide a uniform means for determining whether a given configuration of networks and equipment will meet the time window requirements of a specific application. The metrics section of this document addresses such issues. Furthermore it is also not possible to define a networking protocol suite that is properly optimized for all conditions encountered in manufacturing applications. In general, there are three overlapping requirements that arise in varying degrees in manufacturing, and these requirements may lead to the definition of conflicting protocols.

- a) A network is required that is optimized for robust interworking and guaranteed message delivery under harsh conditions.
- b) A network is required that is optimized for cyclical or repetitive polling or data transmission. Possible bases for the repetition cycle include time and externally induced bases such as rotational position of associated mechanical equipment.
- c) A network is required that is optimized for dynamic, unpredictable communications sequencing, both on the network and within the attached network time-critical communications entities.

This Technical Report focuses primarily on c).

It is recognized that all time-critical communications systems differ from other networks in their requirement for resilience. The tighter the time constraints are, the more important strategies for resilience become, since networks capable of meeting tight time constraints need to exhibit great resilience if the dataflow is not to be perturbed by minor faults.

Industrial automation — Time-critical communications architectures — User requirements

Section 1: General

1.1 Scope

This Technical Report identifies user requirements for systems supporting time-critical communications. Such systems need networks which allow both time-critical and non-time-critical communications. It focuses on the requirements identified in the field of discrete parts manufacturing, but the requirements may also be applicable in other fields, including process control. Other work has concentrated on the requirements of state-driven systems, in which traffic flow patterns and network configuration are static, but this report concentrates primarily on the requirements of event-driven systems, in which traffic flow and configuration change dynamically.

It also discusses user requirements for metrics and benchmarks which can be used to compare and manage the performance of networks to ensure that the time constraints of applications can be met and proposes some simple benchmarks.

1.2 References

ISO 7498:1987, *Information processing systems - Open Systems Interconnection - Basic Reference Model*.

ISO/IEC 9506:1990, *Industrial automation systems - Manufacturing Message Specification (MMS) - Part 1: Service definition, Part 2: Protocol specification*.

ISO/IEC 9545:1989, *Information technology - Open Systems Interconnection - Application Layer Structure*.

1.3 Definitions

For the purposes of this Technical Report, the following definitions apply.

1.3.1 application dependent requirement: A requirement related to the usage of the TCC architecture which is independent of the implementation.

1.3.2 application entity: Part of the application process that deals with the communications system.

1.3.3 implementation-dependent requirement: A requirement which depends on the specific aspects of the implementation of the time-critical communications architecture being used.

1.3.4 interaction: Any causal relation induced by message transfer.

1.3.5 multipeer group: A group of peer entities which are mutually willing and able to be senders or receivers of multi-peer data transmissions with other members of the group.

1.3.6 multipeer data transmission: The transmission of a PDU to one or more destinations.

1.3.7 performance: The behaviour of the system with respect to time.

1.3.8 QoS attribute: An attribute of a managed object relating to QoS.

1.3.9 QoS category: A general class of QoS system elements relating to the user view of the system.

1.3.10 QoS characteristic: A quantifiable aspect of QoS system elements which is defined independently of the means by which it is represented or controlled.

1.3.11 QoS context: The set of information retained by one or more system elements, used for the purposes of QoS management.

1.3.12 QoS measure: A set of one or more observed values relating to a QoS characteristic.

1.3.13 QoS metrics: A set of measurement methods for quantifying QoS characteristics.

1.3.14 QoS parameter: A variable relating to a QoS characteristic conveyed between system elements as part of a mechanism for the management and provision of QoS.

1.3.15 spatial coherence: A property of duplicated lists of variables. It indicates whether or not all the copies are identical at a given time or within a given time window.

1.3.16 temporal coherence: A property of a list of variables. It indicates whether or not the value of each variable in the list has been produced and transmitted and/or received within a given time window.

1.3.17 time-critical communications: When one or more application process which sends a message requires it to be received (or received and acted upon, or received and acted upon

and confirmed) within a certain bounded time period "window" after its send request to the system. (This can be a time-critical communication transaction or a time-critical data transmission.)

1.3.18 time-critical communications architecture: An architecture, in the sense of the OSI reference model, which supports the identified requirements for time-critical communications. It is a structure which has layers, entities, service access points, protocols, connections etc. in accordance with the concepts introduced in ISO 7498.

1.3.19 time-critical data transmission: The transmission of a PDU within a given time window.

1.3.20 time-critical communications entity: An entity participating in time-critical communications (a time-critical communications entity may or may not map directly to an application entity since a single application entity could have time-critical and non-time-critical components).

1.3.21 time-critical communications group: A group of time-critical communications entities in a time-critical communications system.

1.3.22 time-critical communications system: A system in which there is time-critical communication; it is an implementation of a time-critical communications architecture.

1.3.23 time-critical multipeer data transmission: The transmission of a protocol data unit to more than one destination within a defined time-window or time windows.

1.3.24 time-critical communications transaction: An ordered set of message transfers (or exchanges) coming from different application entities which all have to be completed within a given time window. This may be a time-critical multipeer communications transaction or a time-critical peer to peer communications transaction. Such transactions are complex information processing operations involving multiple data transmissions for an application with time constraints.

1.3.25 time window: A bounded time interval which is characterized by starting time and delay or starting time and end time which are application dependent; the degree of resolution of times and delays is implementation dependent. Specific time windows are defined in 3.2.3.4.

1.4 Abbreviations

The following symbols and abbreviations are used in this Technical Report with the meanings indicated.

AE Application entity
AP Application process
ASE Application service element

CIM	Computer Integrated Manufacturing
LAN	Local Area Network
MAC	Media Access Control
MAP	Manufacturing Automation Protocol
MMS	Manufacturing Message Specification
MTBF	Mean time between failures
MTTR	Mean time to repair
OSI	Open Systems Interconnection
PDU	Protocol Data Unit
QoS	Quality of Service
SAP	Service Access Point
TCC	Time-critical communications
TCCA	Time-critical communications architecture
TCCE	Time-critical communications entity
TCCG	Time-critical communications group
TCCS	Time-critical communications system
TCCT	Time-critical communications transaction
TCDT	Time-critical data transmission
TW	Time window

STANDARDSISO.COM : Click to view the full PDF of ISO/TR 12178:1994

Section 2: User requirements for time-critical communications in discrete parts manufacturing

2.1 Introduction

Automated factories use networks for distributed control applications. These applications require communications systems which are designed to accommodate worst-case performance as opposed to average performance; in such systems timeliness is of paramount importance. The network should be able to cope with both bulk data transfers and time-critical messages and should support methods for achieving resilience.

If necessary, interfacing the TCC system with the main factory data communication network should be a simple procedure, using a linking device. Because of the requirement to operate in very harsh environments, highly reliable media and signalling methods are necessary so that a very low bit error rate results and a minimum number of retransmissions is necessary. The user needs to be able to define communications priorities and control error recovery mechanisms. Network management functions are necessary to allocate resources, control access to TCC groups, detect latent failures and actual failures etc.

The user requirements that are identified allow the concept of time windows to be realized in OSI. The interrelationship between TCC architecture, temporal coherence, time windows and applications requiring time-critical communications are set out in figure 1.

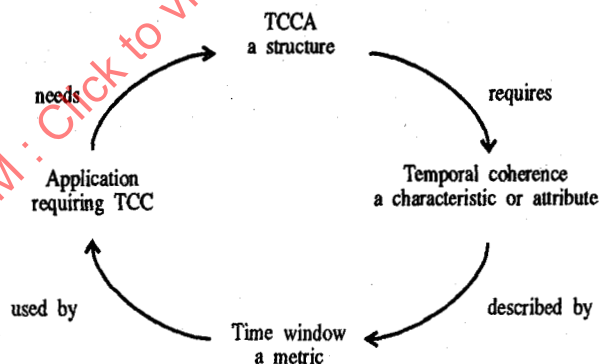


Figure 1

The following user requirements for time-critical communications systems (TCCSs) are grouped into six major categories: integration issues; message differentiation and PDU attributes; timing-related issues; resilience; topology aspects; and management and sovereignty issues. These requirements are discussed separately.

a) Integration issues

- 1) TCCSs supporting MMS functionalities in a timely fashion
- 2) TCCA requiring standardized user interfaces

- 3) TCCA defined within the context of OSI
 - 4) Time-critical messages short enough to be sent without segmentation
 - 5) Membership of a TCCS controlled
 - 6) Applications requiring time-critical communications not committing time-critical PDUs to the transmitting stack faster than it or the network or the receiving stack can handle the traffic
 - 7) Classes of conformance to individual requirement level
- b) Message differentiation and PDU attributes
- 1) Differentiation of messages
 - 2) Non-time-critical PDUs not prejudicing the delivery of time-critical PDUs
 - 3) Distinguishing the relative urgency of messages
 - 4) A TCCS coping with "dynamic sequencing"
- c) Timing-related issues
- 1) Necessity of Predictability of message transfer times and transaction times
 - 2) Necessity of agreed probability of interaction/transaction completion time
 - 3) Requirement for temporal coherence and possible requirement for spatial coherence in TCCA
 - 4) Reconfigurability of the application and the use of the TCCS
- d) Resilience
- 1) User control of error recovery
 - 2) Mechanisms to assist latent failure detection in TCCSs
 - 3) Availability
- e) Topology aspects
- 1) Off segment communication
- f) Management and sovereignty
- 1) Management issues
 - 2) Sovereignty issues

NOTE 1 The ordering of the above categories and points does not reflect their relative importance.

Some other sets of combinations of the identified user requirements may be made, for example: architectural requirements covering message differentiation, management and resilience issues; timeliness requirements; relationship to OSI standards and architectures. In practice users will probably order the requirements to reflect the importance of each in their particular application.

2.2 Application models

A TCC architecture needs to provide for a number of different communication scenarios. In general the user's overview is perceived from the application; an AE may map directly to the TCC entity or partly to the TCC entity because it also has non-TCC-entity elements in it (see figure 2).

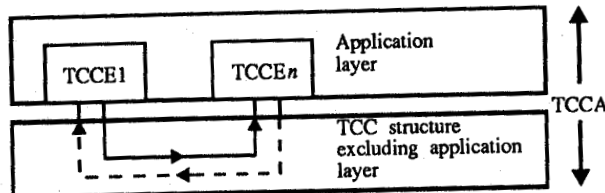


Figure 2 - Relationship of TCCEs and TCCA

A variety of application models have been identified and are referred to as client-server, initiator-sink, producer-consumer, producer-distributor-consumer, etc. The following types of transfers have been identified as essential for TCC architecture but the technical implications for TCC systems are not discussed in detail as these will depend on the particular TCC system chosen and its implementation.

- point to point, i.e. an exchange from one sender to one receiver.
- multicast, i.e. an exchange from one sender to many (n) receivers which form a known group of TCC entities.

NOTE 2 Where an exchange is from one sender to all receivers without negotiation of the communications context it may be referred to as broadcast.

In applications there will be the following types of interactions:

- Peer to peer interaction, where the application which initiates the interaction or requests the service (the initiator) is the sender and there may or may not be a requirement for a response [see figures 3 a) and 3 b)].
- Multipeer interaction, where the initiator may make one multipoint transfer or n point to point transfers and again there may, or may not, be a requirement for responses, which may be a confirmation by the group as a whole or n individual confirmations from each member of the group [see figures 4 a), 4 b), and 4 c)].
- Peer to distributing peer is sometimes recognized as a third category, but this can be equated to figure 3a or 3b plus figures 4 a) or 4 b) or 4 c) depending on what level of confirmation is required (see figure 5).

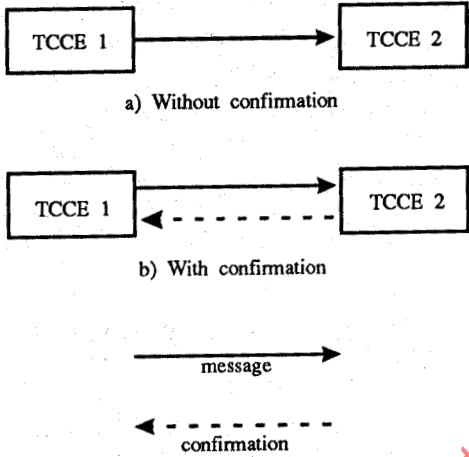
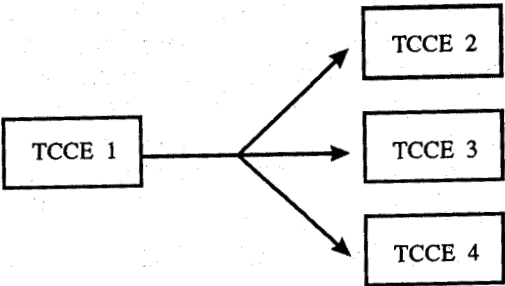
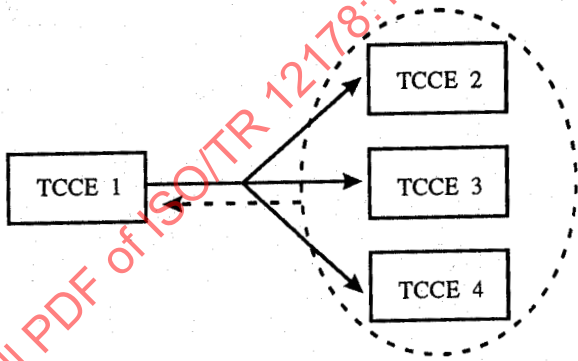


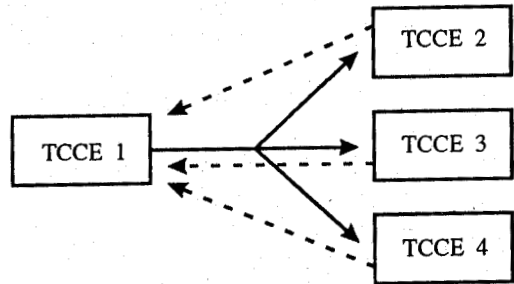
Figure 3 - Peer to peer



a) Without confirmation



b) With one confirmation from group of receipt of that message transfer



c) With individual confirmations to that message transfer from each group member

Figure 4 - Multicast to members of a group

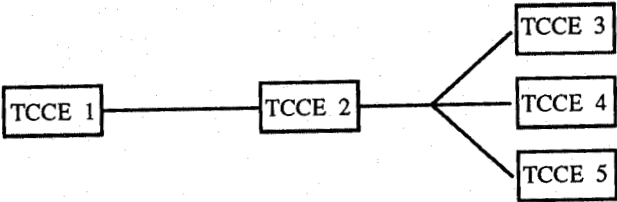


Figure 5 - Peer to distributing peer

It is clear that a TCC system requires point to point communication. There should be a facility which allows an application to invoke procedures or request services, or information, or both, on an ad-hoc basis. In many situations a TCC system needs the capability to conduct communications with several applications at the same instant, such as the synchronization of real-time clocks at multiple locations. It is apparent that TCC systems require multicast capability for spatial and temporal coherence and for efficient use of limited resources, e.g. bandwidth. Finally, there is a requirement to minimize the communications requirements on simple devices, while allowing open access to the most current data.

Distributed real-time applications are expected to have a varied mix of messages including periodic traffic (e.g. remote data samples), aperiodic regular traffic (e.g. file transfers, user requests) and sporadic traffic (e.g. alarms, reconfiguration requests). Therefore a TCC system has to support both periodic and aperiodic traffic, i.e. both state-driven and event-driven traffic. Since many time-critical applications will be event driven (i.e. there will be variable loading on the network in terms of traffic, TCC entities etc), time-critical traffic has to be able to be sent within prescribed time windows with an acceptable probability of success in all circumstances.

2.3 Integration issues

NOTE 3 Six requirements are considered under the general topic of integration issues. The discussion of user requirements in the following sections assumes that these requirements will have been met in the TCC architecture.

2.3.1 TCCs supporting MMS functionalities in a timely fashion

2.3.1.1 Summary

A requirement of TCC systems in discrete parts manufacturing shall be a set of services aligned with ISO/IEC 9506 (Manufacturing Message Specification).

2.3.1.2 Explanation

MMS services provide peer to peer application interactions in accordance with standardized syntax and semantics. Standardized semantics are necessary to reduce the programming burden for application developers, and to promote interoperability between different implementations of the TCC architecture. It is clear that not all of the services provided by MMS will be required by all applications using the architecture. Where the functionality provided by MMS services is implemented in a real end system, the corresponding MMS services shall also be implemented. In applications which do not use the client-server model the services should 'look and feel' like MMS.

Modifications or extensions to the content of MMS may be required to support multicast and broadcast message transfers.

2.3.2 TCCA requiring standardized user interfaces

2.3.2.1 Summary

Standardized interfaces to the application communication services shall be required.

2.3.2.2 Explanation

These standardized interfaces shall be in the form of function and/or procedure calls providing for synchronous operation, or asynchronous operation, or both. Interfaces shall be required for service invocation at the client side. Where the service functionality is implemented at a server by a 'user' application, there shall also be standardized interfaces from the service provider to the user at the server side. Additional service interfaces, which are not standardized and add value to an implementation, are not precluded by this architecture.

Standardized interfaces are necessary to ensure application program transportability across different implementations of the architecture. Standard service specifications are also necessary; these functional definitions would include the time constraints relating to the use of the TCC system.

A standardized environment, such as an operating system, that supports real-time requirements is a prerequisite for defining a standardized user interface.

NOTE 4 Specific standardized application program interfaces for TCC systems will be discussed in a future technical report on the management of TCC systems.

2.3.3 TCCA defined within the context of OSI

2.3.3.1 Summary

This architecture shall conform to the OSI layer standards wherever possible while also meeting the other requirements described in this paper.

2.3.3.2 Explanation

An open systems orientation is necessary to allow integration with other real open end systems. TCC architecture should enable communications in the manufacturing environment to benefit from the same flexibility as other implementations developed in accordance with ISO 7498.

Technological improvements may allow OSI to be used in an increasing number of time-critical applications in which time windows are of relatively short duration. The introduction of yet another technology to meet time critical requirements should be avoided. However, changes to the layer standards may be necessary in order to meet the requirements described in this Technical Report.

2.3.4 Time-critical messages short enough to be sent without segmentation

2.3.4.1 Summary

A time-critical message shall be conveyed across the network in a single datalink frame.

2.3.4.2 Explanation

Most existing MAC methods will only guarantee that a TCC entity that gains control of the medium can send one single datalink frame before the TCC entity has to relinquish control of the medium (or has control taken away by another TCC entity). When something has gone wrong it is no good sending half of a 'panic' message with the rest to follow when the TCC entity next gains control of the medium.

In effect, the requirement that a time-critical message shall not be segmented limits the size of the message, because the message, plus all protocol octets added by each layer in the stack, has to go into the 'user data' part of the maximum size datalink frame allowed. For some architectures it may be necessary to further limit the maximum message size in some implementations to ensure satisfactory message transfer times. Therefore the application will require knowledge of such constraints.

2.3.5 Membership of a TCCS controlled

2.3.5.1 Summary

It may be necessary to limit the number of associations set up by AEs and/or to limit the load offered on them so that the network can carry time-critical traffic within prescribed time-windows with the defined acceptable probability of success.

2.3.5.2 Explanation

Time-critical associations may be prearranged so that they can be available when required. This avoids the overhead of setting up an application association at the time a time-critical message has to be sent, and also allows resource to be reserved for such messages. Because resource should always be available for time-critical communications it is essential to control the number of TCC entities generating such traffic. Whenever there are no time-critical communications it is desirable for all the available resource to be used for non-time-critical transfers.

2.3.6 Applications requiring time-critical traffic not committing time-critical PDUs to the transmitting stack faster than it or the network or the receiving stack being able to handle the traffic

2.3.6.1 Summary

It is apparent that communication on a TCC system ceases to be determinate if its virtual channel capacity is overloaded.

2.3.6.2 Explanation

The problem of ensuring there is an upper bound on the demands made of a finite resource (communications) is not the responsibility of the communications system. In any TCC system it is a requirement that resources should always be available for time-critical communications. The systems designer should provide broad rules for a network that will reflect the network constraints, e.g. bandwidth or throughput.

2.3.7 Classes of conformance to individual requirement level

2.3.7.1 Summary

Classes of conformance allow TCC architectures to match individual application requirements.

2.3.7.2 Explanation

In some time-critical applications it is not essential for the TCC architecture to meet all the requirements. Therefore there will be identifiable classes of TCC architecture which meet specific subsets of the requirements. The identification of these classes is not within the scope of the document. Specific benchmarks may need to be developed to test TCC systems with respect to conformance classes.

2.4 Message differentiation and PDU attributes

2.4.1 Differentiation of messages

2.4.1.1 Summary

The application should differentiate between time-critical and non-time-critical messages. In addition, it is necessary for the communications system to distinguish between time-critical and non-time-critical messages and provide appropriate service.

2.4.1.2 Explanation

The differentiation of messages should be dynamically allocated when the application commits the message to the TCC entity. Once the messages have been differentiated the communication system should carry the messages from source to destination without altering their designation.

Differentiation could be done by using attributes attached to the PDUs, and/or multiple (dual) stacks. The former is the most elegant solution but involves changing many existing standards whereas the latter could be introduced more immediately. The coexistence of time-critical and non-time-critical PDUs in a single stack architecture may lead to processing difficulties so multiple stacks may be required in addition to PDU tagging. Other alternative mechanisms may be devised to allow the differentiation of messages in TCC systems. It may be a requirement that time-critical PDUs can pre-empt non-time-critical PDUs.

2.4.2 Non-time-critical PDUs not prejudicing the delivery of time-critical PDUs

2.4.2.1 Summary

Clearly in practical applications, for economic reasons, the two types of traffic will be required to share a single network. Development of tuning parameters, prioritization etc. may allow this to work.

2.4.2.2 Explanation

Applications' timing requirements for time-critical traffic should still be met if the offered load is at its maximum. However a single non-time-critical transmission, such as downloading a file, on an otherwise idle network should be able to use a large proportion of the network's bandwidth, which implies that non-time-critical traffic will be slowed down when the network is busy. At the moment OSI stacks cannot take appropriate action if there is a conflict, i.e. where the delivery requirements of the time-critical PDU are incompatible with the baseload, PDU size, etc., of the non-time-critical PDU.

2.4.3 Distinguishing the relative urgency of messages

2.4.3.1 Summary

In order to reflect the relative urgency of messages it may be appropriate to assign attributes such as priority to PDUs. This will facilitate the use of mechanisms to ensure that specified message transfer times are achievable.

2.4.3.2 Explanation

There are many possibilities for the most appropriate number of priority levels; for example, two levels of priority might correspond directly with time-critical and non-time-critical PDUs, while three priority levels could distinguish between periodic time-critical PDUs, aperiodic time-critical PDUs and all other PDUs. Once a priority mechanism has been established it could be argued that more levels are desirable but this would probably greatly complicate the architecture and its management. Clearly, the priority of the PDU needs to be distinguished throughout its traverse of both stacks, since the time-critical

PDU should be propagated by all layers in preference to less time-critical PDUs. Such pre-emption should occur:

- at the datalink layer, where preference is given to the highest priority PDU;
- at layers concerned with segmenting; time-critical PDUs should be propagated in preference to discrete frames of segmented large non-time-critical PDUs;
- in other layers, where any opportunity for the pre-emption should be considered if this leads to better performance.

It should be obvious that this strategy offers the virtual channel, in turn, to the hierarchy of priorities. Escalating priorities with time is undesirable since this will rapidly remove the differentiation of the traffic, e.g. under fault conditions. When there are a number of priorities only the highest level will get the best determinism and this will be especially critical under fault conditions. If more than two priorities are supported by a TCC system, a mechanism should exist to ensure that all implementations of the architecture are able to treat identical priority levels in identical ways.

Mechanisms such as deadline scheduling may assist the management of priorities in TCC systems or may be used as other ways of distinguishing the relative urgency of messages. If such mechanisms can only be used at the application layer it is essential that the lower layers can be relied upon to transmit the message within a given time window.

2.4.4 A TCCS coping with 'dynamic sequencing'

2.4.4.1 Summary

The sequence of the PDUs in a TCC system is determined by the application and circumstances, and is not prescribed.

2.4.4.2 Explanation

In a TCC system there will be periodic time-critical transfers, aperiodic time-critical transfers and non-time-critical transfers. The system shall operate with any combination of such transfers. The different applications using the TCC system will dictate the sequence of the PDUs.

2.5 Timing related issues

2.5.1 Necessity of predictability of message transfer times and transaction times

2.5.1.1 Summary

With shared media and asynchronous PDUs the delivery time is invariably a statistical quantity.

2.5.1.2 Explanation

With a reservation MAC strategy, an upper bound of delivery time can be determined on the assumption that no faults occur. With most contention MAC strategies there is no upper bound, although it may be possible to determine the probability that a message will be delivered within a certain time.

Even with a reservation strategy, there is a non-zero probability of a fault (e.g. a broken cable or a lost token) causing the delivery time to exceed the calculated upper bound. Therefore, whatever the MAC strategy, the performance shall be measured in terms of the probability of a message failing to be delivered within a specified time (see figures 6 and 7).

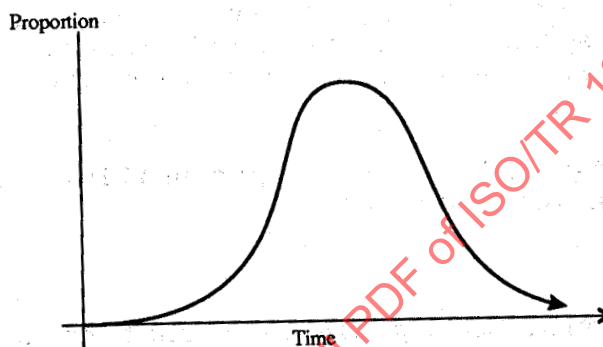


Figure 6 - Proportion of messages received in a given time

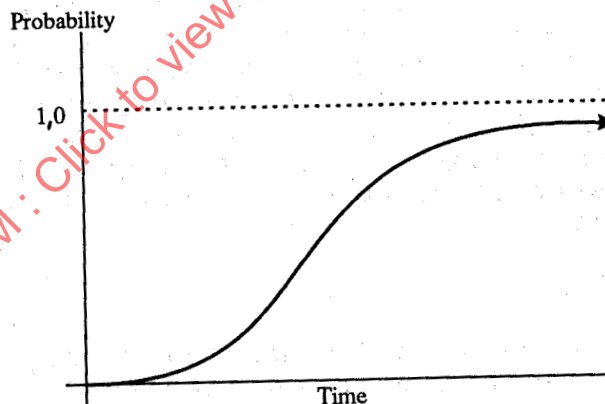


Figure 7 - Probability of frame being received as a function of time

Various factors will affect the message transfer time (depending on the architecture of the TCC system) such as the following

- The incoming application layer PDUs might well add latency to an outgoing PDU (since most stacks share one processor). While the TCC entity is always in control of the amount of outgoing traffic, in general it is not in control of the amount of incoming traffic and thus the stack can still be overwhelmed. Therefore, incoming time-critical PDUs should be removed such that flow control is not invoked.

- Changing the frame size will modify protocol efficiency. In a stack large PDUs impede the flow of time-critical PDUs. If application PDU size exceeds frame size then segmenting etc is invoked with consequent time delays.
- Number of nodes (application entities) in the TCC system.
- Dynamic allocation of bandwidth is desirable since then users will get a better service.

Message transfer times can be measured experimentally (but it will often be impossible to find a representative system to measure or to vary the traffic enough to ensure that the 'worst case' transfer has been found), heuristically by gathering statistics while operating, and by simulating temporal and procedural aspects to calculate performance. Standard metrics will be required for this, otherwise meaningful comparisons are impossible.

2.5.2 Necessity of agreed probability of interaction/transaction completion time

2.5.2.1 Summary

The TCC system should be capable of transferring messages within a predetermined time interval with a specified probability. In order to assess the suitability of a particular network for a specific application, it is essential to be able to establish that the network will be able to meet the application's timing constraints. In practice this will be an assessment of the probability that the timing constraints will be met. The application should include 'fail-safe' procedures for use in the event of the network failing to deliver (or the receiving AE failing to act on) a time-critical message within the required time. It is necessary to be able to calculate the probability of failure per message.

2.5.2.2 Explanation

QoS characteristics need to be handled coherently through the OSI stack so that applications can specify parameters such as the following.

- Required message transfer time. This allows the network to distinguish between urgent and less urgent traffic.
- Maximum tolerable probability of failing to meet this requirement. This allows the network to identify whether there is a requirement for retransmission, renegotiation of QoS, etc.
- Maximum throughput across a connection. This allows the lower layers to reserve a sufficient share of the network's resource, e.g. bandwidth for the traffic that the connection may be required to carry; the network may allow any of this "reserved" resource that is not required for

time-critical messages to be used for other data flows such as file transfer and the downloading of programs.

Provision needs to be made for service providers to renegotiate the QoS after the connection has been made (e.g. because more nodes have been added to the LAN segment or because the original route is no longer available). The values of these QoS parameters need to be related to the metrics derived from standard benchmarks or it will not be possible to assess the performance of a network with benchmark tests and then translate this into QoS parameter values.

It may be appropriate that the differentiation of PDUs or the priority assigned to a PDU is ultimately handled as a QoS matter.

2.5.3 Requirement for temporal coherence and possible requirement for spatial coherence in TCCA

2.5.3.1 Summary

A TCC system shall support timeliness and synchronization of information. The concept of temporal coherence is required in all TCC applications; the concept of spatial coherence is required in many TCC applications. These important characteristics set TCC systems apart from other communication systems.

2.5.3.2 Temporal coherence and timeliness

Temporal coherence is a property of a list of variables; it indicates whether or not the value of each variable of the list has been produced and transmitted and/or received in a given time window.

The TCC system shall support explicit timeliness attributes of information, for example indications of temporal properties of values of different objects. This will then provide the receiving TCC entity with an indication of the time elapsed since the information was originally sampled from a process, product or network, or since the information was requested by a TCC entity. In many applications it is important to know whether or not values have been:

- produced at the same time; or
- transmitted and/or received at the same time.

Thus there are two distinct types of temporal coherence; a temporal coherence with regard to the production of information and a temporal coherence with regard to the transmission of information.

Temporal coherence is not only important in information transfers but may also be important when there is a delivery deadline for a message. Timeliness attributes would assist users in determining whether the message transfer time of the system met their requirements. Solutions to the requirement for

temporal coherence could be absolute timestamping or relative timestamping (e.g. using refreshment and punctuality parameters).

2.5.3.3 Spatial coherence

Spatial coherence is a property of a duplicated list or multiple copies of a list of variables. It indicates whether or not all the copies are identical at a given time or within a given time window.

Spatial coherence is useful when broadcast or multicast exchanges are necessary. It is necessary to provide such a facility if the user needs to know that copies are the same on different AEs. Where TCC systems support redundancy, spatial coherence is necessary so that multiple controllers can have an identical view of the distributed control system and take appropriate action. Broadcasting or multicasting may allow TCC systems to offer better performance in some implementations than discrete point to point connections which require acknowledgements but in such circumstances spatial coherence will be essential.

2.5.3.4 Time-critical communications transaction

A time-critical communications transaction is a set of message transfers (or exchanges) coming from different AEs which all have to be completed within a given time window.

In many applications it is important to be able to complete a number of message transfers within a given time window. For example the MMS service "Read a list of variables" takes no account of time; ideally it should be possible to provide a time-critical service which could specify some parameters so that the user could be certain that the variables are temporally coherent, i.e. that all the values of the variables were taken in a given time window. In addition the variables may be located in different VMDs and with MMS it will be necessary to use several different associations, and it will be important to be able to complete all the transfers within a certain time window.

A time-critical communications transaction is coherent with respect to production time when all the application's messages or information units have been produced within a given time window.

A time-critical communications transaction is coherent with respect to transmission time when all the application's messages have been received within a given time window.

A time-critical communications transaction is globally coherent when it is coherent with respect to both production and transmission time.

A time-critical communications transaction is spatially coherent when copies of messages, or lists of variables, are associated with a given time or time window.

2.5.3.5 Synchronization

The TCC system may frequently be required to support the synchronization of actions such as event triggering or information retrieval. It may be necessary for a TCC system to have a mechanism by which all remote TCC entities have an agreement of absolute or relative time, within some maximum jitter time under normal (non-fault) conditions. When this is required, maximum jitter time is generally much less than maximum transport time.

With this synchronization mechanism, a system designer can be confident that information from multiple sources can be sampled at the same instant in time (within the jitter specified). Given this mechanism, it is the responsibility of other elements in the system to synchronize the sampling of information subsequently presented to the communication system.

The feasibility of this approach is demonstrated by existing networks; some networks use a broadcast synchronization mechanism while other networks use timestamping. Accurate time stamping requires remote clocks to be periodically adjusted. This can be done by periodic broadcast messages, or by a reliable reference (e.g. a satellite clock) on each remote node. Synchronization is also important in TCC systems which have inactive devices.

2.5.3.6 Reference clock

Applications using the TCC system may require access to a common reference clock and a TCC entity may need to be able to adjust its local clock to allow synchronization to the resolution required by the application.

2.5.4 Reconfigurability of the application and the use of the TCCS

2.5.4.1 Summary

A TCC system may require a preset time for applications to be activated or deactivated or modified.

2.5.4.2 Explanation

Many factory automation users require that an application be modified or initiated and/or terminated within a preset time window. Initiation may include, but not be limited to: network access, connection establishment, association negotiation and application program download. Initiation is complete when the application program is fully supported by the TCC entity. Termination is complete when the application program is no longer supported by, and no longer affects, the TCC entity. Modification is complete when the application is operating in the changed state in accordance with the modification request. It may be important that reconfigurability occurs in such a way that the performance of any stable regime is not degraded or that any process is not perturbed. There may be a requirement

for dynamic reconfiguration without disruption to the network. Metrics should enable the user to decide how quickly the application can be modified.

2.6 Resilience

2.6.1 User control of error recovery

2.6.1.1 Summary

The detection and reporting of failure to complete the delivery of a message or to receive an acknowledgement within a nominated time frame is necessary.

2.6.1.2 Explanation

After notification of failure, retransmissions will typically cease unless the user requests a new attempt to send the message. A message can be nominated to belong to a particular category or nominated to use a given time frame, and perhaps be allowed to be retransmitted a specific number of times (possibly zero).

TCC architecture shall be able to handle any type and combination of types of data traffic such as periodic and aperiodic time-critical traffic and periodic and aperiodic non-time-critical traffic.

Nomination of message could be regarded as a dynamic (QoS) parameter. Dynamic QoS parameters are those that can be set for a particular invocation of the service and should in a TCC system include - at the application layer service interface - an item such as nomination.

2.6.2 Mechanisms to assist latent failure detection in TCCs

2.6.2.1 Summary

Wherever possible latent communication failures, i.e. failures not yet apparent to participants, should be detectable to allow the highest probability of successful time-critical communication to be offered to the TCC entity.

2.6.2.2 Explanation

In a TCC system it is desirable to have monitoring mechanisms to allow the detection of failures or partial failures in components of the TCC system before these failures have affected the time-critical traffic on the TCC system. AEs in a TCC system may perform this role themselves or there may be special monitoring entities. Management functions will be necessary to administer such error detection processes and coordinate reporting of errors etc.

2.6.3 Availability

2.6.3.1 Summary

Availability is defined as the probability of the communication system being in the functional state. This can be expressed by the equation

$$\text{availability} = \text{MTBF} / (\text{MTBF} + \text{MTTR})$$

2.6.3.2 Factors affecting availability

It is essential that a TCC system offer a high degree of availability because the execution time of a service is lengthened by the recovery time (time to repair) if a failure occurs during execution.

In a communication system two different failures can occur:

- permanent failure due to line or component failure;
- temporary failures due to high noise on the line or other temporary causes such as lost or duplicated tokens.

It is desirable that the impact of a single fault does not take away the system's availability. There are three methods to improve availability:

- The 'perfectionist' approach. With this concept the mean time between failure will be lengthened by selecting components with lower failure rates or by choosing environments with lower noise level, or by choosing media with less susceptibility to environmental noise.
- The 'fault tolerant' approach. With this concept the mean time to repair (to recover) will be reduced. The most promising way is to use redundancy. One example is code redundancy, which is used to recover from transmission errors. Another example is the use of duplicated transmission lines, where a failure of one line can be tolerated.
- The 'fault containment' approach. With this concept the system continues to operate in the presence of any single fault. The system becomes unavailable only if a second fault occurs before the first is repaired.

The overall optimal solution may combine all three approaches.

2.7 Topology aspects

2.7.1 Off segment communication

2.7.1.1 Summary

Off segment communication can be allowed only if the application time constraints are not broken. There is clearly a requirement for off segment communication; e.g., configuration, maintenance, diagnostics, setup.

2.7.1.2 Explanation

Determinism on any time-critical segment ends when the propagation of the frame passes to another segment. In a bridge there is an implied application (the cloud) sitting above the two data link interfaces. The latency within the cloud may be indeterminate; in a router and a gateway the indeterminate latency is simply higher up the stack. A relay (bridge, router or gateway) in a TCC system segment will have a stack that is a normal conforming instance of the protocol in that segment. The relay can receive or dispatch time-critical traffic in exactly the same manner as any other stack in that segment. In an interconnecting segment the same situation will be true. Thus the latency of a time-critical PDU despatched to the next segment will be the sum of the guarantees for each segment latency plus the latency (which may be indeterminate) in the relay application. Such a situation will result in latency that is more closely bounded than for non-time-critical traffic.

Inter-segment traffic cannot be allowed to enter a time-critical segment if this will impact on already promised performance or reserved resource. Control of such traffic would have to be a local issue.

2.8 Management and sovereignty

2.8.1 General

Detailed consideration of the management required for time-critical communications and discussion of sovereignty issues will form a separate technical report. However the following general observations are relevant to this review of user requirements for time-critical communications.

2.8.2 Management issues

Many of the identified requirements for time-critical communication will require network management to allow their implementation in a real system. Management of TCC entities in TCC groups and TCC systems will be necessary and management of resources will be required. It is probable that automatic management functions will be required in TCC systems.

2.8.3 Sovereignty issues

TCC systems are groups of co-operating AEs that interact to form a regulated and stable system. Thus there is an overriding requirement for control in TCC systems to ensure that individual applications do not perturb the system with their requirements for time-critical communications. The TCC architecture has to manage dynamic, event-driven requirements for time-critical communications and ensure that all members of a TCC group get the network performance and resilience that they need. It may be necessary to provide messages with temporal attributes and differentiation at all layers, and have a controlling AE that identifies and qualifies the TCC entities in a TCC system and monitors and manages the network.

STANDARDSISO.COM : Click to view the full PDF of ISO/TR 12178:1994

Section 3: User requirements for metrics and benchmarks

3.1 Introduction

It is now generally acknowledged that issues of performance have become somewhat neglected as IT standards have evolved; in particular it has become apparent that time-critical applications do not have appropriate support within OSI standards. The network architectures which have been standardized to date were primarily intended for general data traffic and performance considerations were not given a high priority. There is now an emerging desire to redress this balance and re-examine the ways in which network performance can be characterized, measured and managed.

In current OSI standards the principal performance considerations are throughput and the average time delay/throughput trade-off rather than performance measures related to the transfer of specific messages. In TCC architecture it is the ability to characterize the transfer times and delays of specific time-critical messages and express these as time windows which is important. Unfortunately, existing standards do not allow applications to specify performance requirements directly, so that in practice the only way to ensure that such requirements can be met is to decide on a particular network architecture and configuration when the application programs are implemented. This is particularly important in industrial systems where it is often necessary to be able to perform a certain action within a bounded time window.

There is an increasing desire to decouple the application layer and the network to give flexibility both in the functions of the application and in the configuration of the network so that new application systems can easily be incorporated onto a network. Users will then have a much greater choice of systems and this means that they will need some means of comparing different solutions.

Moreover, implementers may wish to run an application on a variety of networks and will not want to have to reassess quality of service parameters afresh for each implementation.

The only comparable metrics that exist today define the physical characteristics of the medium, including bandwidth, cable range, and design limitations of hardware and software (e.g. maximum number of simultaneous application connections, maximum theoretical number of connected TCC entities). The queue handling characteristics of the system, its behavior in overload and error situations, and details of the actual effect of tuning parameters are often only understood when the system is actually operating.

There is thus a positive need to provide a uniform set of criteria related to performance to permit potential users to make accurate and objective comparisons between different

solutions and choose the most appropriate for their particular application. Standard metrics and benchmarks are required which can easily be applied to different candidate network architectures to compare their relative performance. It is also essential to be able to determine whether a given configuration of network and equipment will meet the time window requirements of a specific application under both typical and extreme loading conditions. In addition the performance of network hardware cannot be studied in isolation: the user is concerned with system performance and this includes application software, network hardware and end-equipment performance and the resilience of the whole installation.

Performance modelling will allow users to investigate whether a given network configuration will provide adequate performance to fulfil the needs of a given application in the presence of loads generated by other applications. There is a growing desire to have truly open systems and use 7-layer OSI stacks in CIM networks wherever possible, and many of these networks will have a requirement to support some time-critical communications. Advances in technology mean that 7-layer stacks may now be able to meet users' performance requirements for TCC and therefore benchmarks are required to adequately evaluate technology alternatives. Users need standard criteria so that simulation models can be developed and used to assess the suitability of different implementations for their applications. It is desirable to verify that the network design will allow applications to have communication services that will meet the required time windows under all possible combinations of circumstances, and in some implementations the number of possible combinations will be very large indeed.

3.2 Standard metrics

3.2.1 Measurement of performance

3.2.1.1 Summary

It is essential to measure performance of a time-critical system in terms of standard metrics which are independent of network architecture.

3.2.1.2 Explanation

In order to ascertain the performance qualities of the network it is first necessary to define some basic parameters that can be used to measure the effectiveness of the network. Standard metrics that can be applied to these parameters, irrespective of the particular TCC system architecture and technology, are required. Ideally the metrics should be simple, relatively easy to use, and meaningful to the network user, and they should align with time window definitions and tuning parameters which may be used to manage the TCC system.

There is a different perspective on the relative importance of different metrics in TCC systems: for example, a crude

comparison of bandwidth may allow an assessment of the network potential to be made but it will not ascertain whether a particular network can meet the time windows demanded by an application. In control scenarios the suitability of a particular architecture for a specific application will concentrate on assessing the worst case timing and the identification of boundary conditions, i.e. when the timings no longer meet the specification.

With appropriate metrics, performance measurements will provide a means of determining what time windows a specific implementation of a particular technology will support. TCC architecture will not lay down any prescriptive values although as conformance classes are defined there may be some specific performance limits imposed on certain classes.

3.2.2 Time parameters in message transfers

3.2.2.1 Summary

There are three basic time parameters associated with the communications procedure. These are send time, transfer time, and receive time.

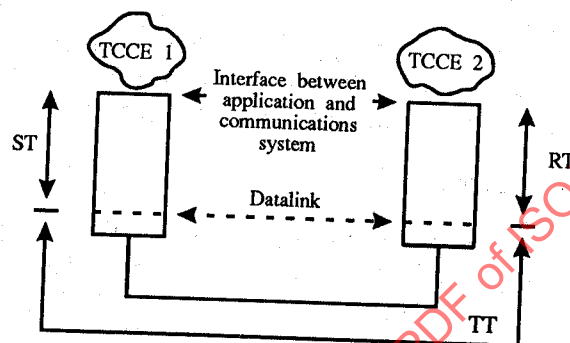
3.2.2.2 Explanation

- Send time is the time taken from the TCC entity requesting a transfer to the first item of information being placed on the transmission media. It is the message handling time at the sending TCC service element.
- Transfer time is the time taken from the first item of information in the message being given to the datalink layer to the last item of information being received by the receiver's datalink layer (as opposed to first item to first item which is propagation time). It is the time taken to transmit a message across the network and includes MAC and transmission delays.
- Receive time is the time taken from the last item of information being received by the receiver's datalink layer to the last item of information being received by the receiver. It is the message handling time at the destination TCC service element.

Both send and receive times can be broken down with respect to the layers of communications protocol utilized within the implementation. The actual times are given by the summation of the processing time taken by each layer along with an overhead figure to represent any latent delays between layers. Given an appropriate sample, average figures can be established which will be associated with a corresponding probability factor. The times may be displayed in the manner of figure 6 or figure 7 or can be expressed simply as a scalar figure being one point of the graph of figure 7 with an associated probability.

It is also important to note that the time taken by a layer on transmission may be different to that taken by the same layer on reception, especially in multi-vendor environments. Also in many cases the latency of a layer on the transmitting stack may be affected by the number of messages being received by the in-bound stack at the same TCC entity and similarly the latency of a layer on the receiving stack may be affected by the number of messages on the out-bound stack at the same TCC entity.

Figure 8 shows the basic time parameters associated with message transfers between two TCC entities.



Key:

ST - Send time

RT - Receive time

TT - Transfer time

Figure 8 - Time parameters in message transfers

The sum of these times is the end to end delay, the transit time. This message transit time can be used to establish whether the time window required by an application for a one-way message transfer can be realized by the proposed network implementation.

3.2.3 Proposed metrics

3.2.3.1 M: Message transit time

This is the time in which the message is not under control of the application programs; that is, from the instant at which the message is released for transmission by the sending application program until the time the message arrives at the receiving application program. It can be subdivided into three components:

- MSO: Stack outbound message transit time (Send time)
- MDL: Datalink message transit time (Transfer time)
- MSI: Stack inbound message transit time (Receive time)

The message transit time is the sum of these three components.

NOTE 5 Due to implementation differences, the definition of the 'point of release for transmission' of a message from an application is arbitrary.

3.2.3.2 Q: Message throughput

This can be subdivided into three components:

- QSO: Stack outbound message throughput
- QDL: Datalink to datalink message throughput
- QSI: Stack inbound message throughput

The achievable message throughput cannot be greater than the smallest of these three components for time-critical communications.

3.2.3.3 R: Recovery time

This is the time to recover from a specific type of failure and restore service, and will be further characterized by an indication of failure type.

3.2.3.4 TW: Time window

There are three major types of time window which are defined by applications requiring time-critical communications:

- TWW: Time window in which a message must be sent from one TCC entity to another (i.e. one way time window); the maximum message transit time may be shorter than this time window if significant delays can occur within the AP
- TWR: Time window in which a message must be sent from one TCC entity to another and a response received by the originating TCC entity (i.e. round trip time window)
- TWT: Time window in which a time-critical communications transaction (ordered set of message transfers from different TCC entities) must be completed.

3.2.3.5 Additional metrics

A standard metric might be the time to recognize and enrol a new TCC entity in the network. This has no meaning where the configuration is essentially fixed.

Another metric might be the time taken to initiate a remote batch job, which is a feature of a specific application layer standard, and thus has meaning only in certain profiles.

3.2.3.6 Standard test message

Comparisons of all the above metrics depend on there being a recognized test message of predefined length. Since the size of the "message" varies as it passes down the stack it is proposed

to define the test message in terms of data octets passed from the application program to the communications system. A 64 data octet length is proposed for abstract comparisons. It may be that it is necessary to define a standard time-critical message length and also a standard length for non-time-critical traffic as the performance of such transfers can also be important in some applications and it may be appropriate to define benchmarks for both.

3.2.4 Testing methodology

Comparisons of a family of curves obtained by plotting message transit time, message throughput and recovery time against load will allow comparisons to be made between networks of different architectures [figures 9 a), 9 b), and 9 c)].

In TCC architecture there are two distinct types of load, time-critical and non-time-critical, and results will have to be obtained for loads with different proportions of time-critical and non-time-critical traffic. In addition, in order to relate more closely to a particular user's application requirement it will be necessary to investigate the effect of varying other factors, e.g. the number of participating TCC entities, on these results. Such an exercise should lead to a closer definition of the vital boundary conditions and allow this information to be incorporated into network management procedures.

Because it is frequently difficult to actually determine all these figures from an actual implementation it is important that the tests can be simulated as well as performed on hardware implementations. This, however, will also lead to a requirement to assess the accuracy of any simulation tests by comparison with data from real installations.

There is a need to develop a benchmark or series of benchmarks that can be used to compare network performance and from which extrapolations may be made for different conditions.

3.3 Benchmarks for TCCA

3.3.1 General

A benchmark is a standard measure of performance that enables one architecture to be compared meaningfully with another. For any application it is the performance with that particular application which is important to the user, and benchmarks are only relevant if they reflect this. Benchmarks may be useful tools for users allowing them to select networks and monitor and manage networks.

The benchmarks discussed in later sections of this report should be considered as examples of performance test definitions. A further report will provide a comprehensive exploration of testing methodologies for time-critical communications and further detail on benchmarks.