INTERNATIONAL STANDARD

ISO 21806-5

First edition
2020-09

Road vehicles — Media Oriented Systems Transport (MOST) —

Part 5:
Transport layer and network layer conformance test plan

Véhicules routiers — Système de transport axé sur les médias —

Partie 5: Plan d'essais de conformité de la couche transport et de la couche réseau

Reference number
ISO 21806-5:2020(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

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 procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/TC 22, *Road vehicles*, Subcommittee SC 31, *Data communication*.

A list of all parts in the ISO 21806 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Introduction

The Media Oriented Systems Transport (MOST) communication technology was initially developed at the end of the 1990s in order to support complex audio applications in cars. The MOST Cooperation was founded in 1998 with the goal to develop and enable the technology for the automotive industry. Today, MOST[1)] enables the transport of high quality of service (QoS) audio and video together with packet data and real-time control to support modern automotive multimedia and similar applications. MOST is a function-oriented communication technology to network a variety of multimedia devices comprising one or more MOST nodes.

Figure 1 shows a MOST network example.

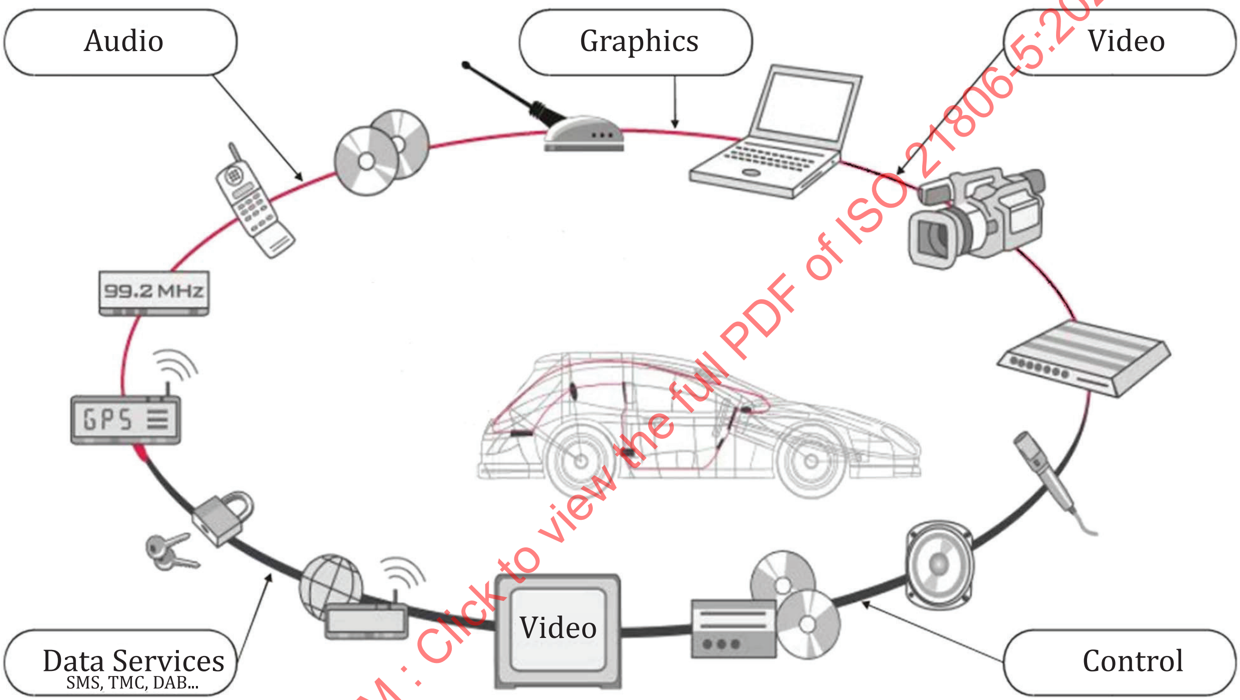


Figure 1 — MOST network example

The MOST communication technology provides:

— synchronous and isochronous streaming,

— small overhead for administrative communication control,

— a functional and hierarchical system model,

— API standardization through a function block (FBlock) framework,

— free partitioning of functionality to real devices,

— service discovery and notification, and

— flexibly scalable automotive-ready Ethernet communication according to ISO/IEC/IEEE 8802-3[2].

MOST is a synchronous time-division-multiplexing (TDM) network that transports different data types on separate channels at low latency. MOST supports different bit rates and physical layers. The network clock is provided with a continuous data signal.

1) MOST® is the registered trademark of Microchip Technology Inc. This information is given for the convenience of users of this document and does not constitute an endorsement by ISO.

Within the synchronous base data signal, the content of multiple streaming connections and control data is transported. For streaming data connections, bandwidth is reserved to avoid interruptions, collisions, or delays in the transport of the data stream.

MOST specifies mechanisms for sending anisochronous, packet-based data in addition to control data and streaming data. The transmission of packet-based data is separated from the transmission of control data and streaming data. None of them interfere with each other.

A MOST network consists of devices that are connected to one common control channel and packet channel.

In summary, MOST is a network that has mechanisms to transport the various signals and data streams that occur in multimedia and infotainment systems.

The ISO standards maintenance portal (https://standards.iso.org/iso/) provides references to MOST specifications implemented in today's road vehicles because easy access via hyperlinks to these specifications is necessary. It references documents that are normative or informative for the MOST versions 4V0, 3V1, 3V0, and 2V5.

The ISO 21806 series has been established in order to specify requirements and recommendations for implementing the MOST communication technology into multimedia devices and to provide conformance test plans for implementing related test tools and test procedures.

To achieve this, the ISO 21806 series is based on the open systems interconnection (OSI) basic reference model in accordance with ISO/IEC 7498-1[1] and ISO/IEC 10731[3], which structures communication systems into seven layers as shown in Figure 2. Stream transmission applications use a direct stream data interface (transparent) to the data link layer.

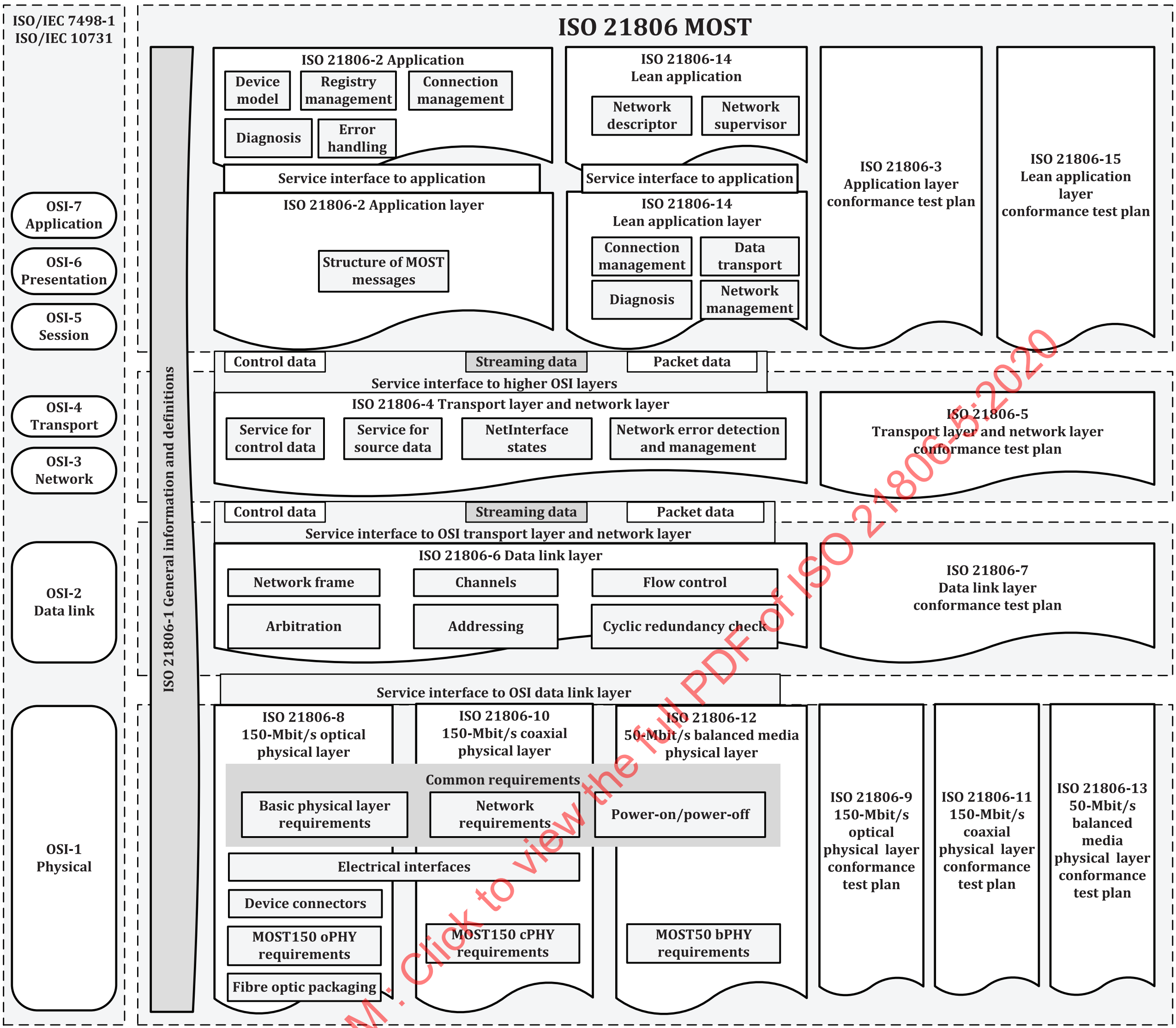


Figure 2 — The ISO 21806 series reference according to the OSI model

The International Organization for Standardization (ISO) draws attention to the fact that it is claimed that compliance with this document may involve the use of a patent.

ISO takes no position concerning the evidence, validity and scope of this patent right.

The holder of this patent right has assured ISO that he/she is willing to negotiate licences under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holder of this patent right is registered with ISO. Information may be obtained from the patent database available at www.iso.org/patents.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights other than those in the patent database. ISO shall not be held responsible for identifying any or all such patent rights.

Road vehicles — Media Oriented Systems Transport (MOST) —

Part 5: Transport layer and network layer conformance test plan

1 Scope

This document specifies the conformance test plan (CTP) for the transport layer and network layer for MOST, a synchronous time-division-multiplexing network, as specified in ISO 21806-4.

This document specifies conformance test cases (CTCs) in the following categories:

— network layer services;

— data transport mechanism;

— dynamic behaviour of a node.

Interoperability testing is not in the scope of this document.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 9646-1:1994, *Information technology — Open Systems Interconnection — Conformance testing methodology and framework — Part 1: General concepts*

ISO 21806-1, *Road vehicles — Media Oriented Systems Transport (MOST) — Part 1: General information and definitions*

ISO 21806-2, *Road vehicles — Media Oriented Systems Transport (MOST) — Part 2: Application layer*

ISO 21806-4, *Road vehicles — Media Oriented Systems Transport (MOST) — Part 4: Transport layer and network layer*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO 21806-1, ISO 21806-2, ISO 21806-4, ISO/IEC 9646-1, and the following apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at http://www.electropedia.org/

3.1
REPEAT
pseudo code command for an iteration

3.2
REPEAT END
pseudo code command for ending an iteration

4 Symbols and abbreviated terms

4.1 Symbols

--- empty cell/undefined

4.2 Abbreviated terms

CTC conformance test case

CTP conformance test plan

ET EnhancedTestability

FBlock function block

IUT implementation under test

LT lower tester

MPI maximum position information

NCE network change event

OSI Open Systems Interconnection

UT upper tester

5 Conventions

This document is based on OSI service conventions as specified in ISO/IEC 10731[3] and ISO/IEC 9646-1 for conformance test system setup.

6 CTP overview

6.1 Test set-up

All CTCs are based on the same test set-up with an upper tester (UT) and a lower tester (LT). The LT contains the lower tester pre-IUT (LT pre-IUT) and the lower tester post-IUT (LT post-IUT).

Figure 3 specifies the test set-up.

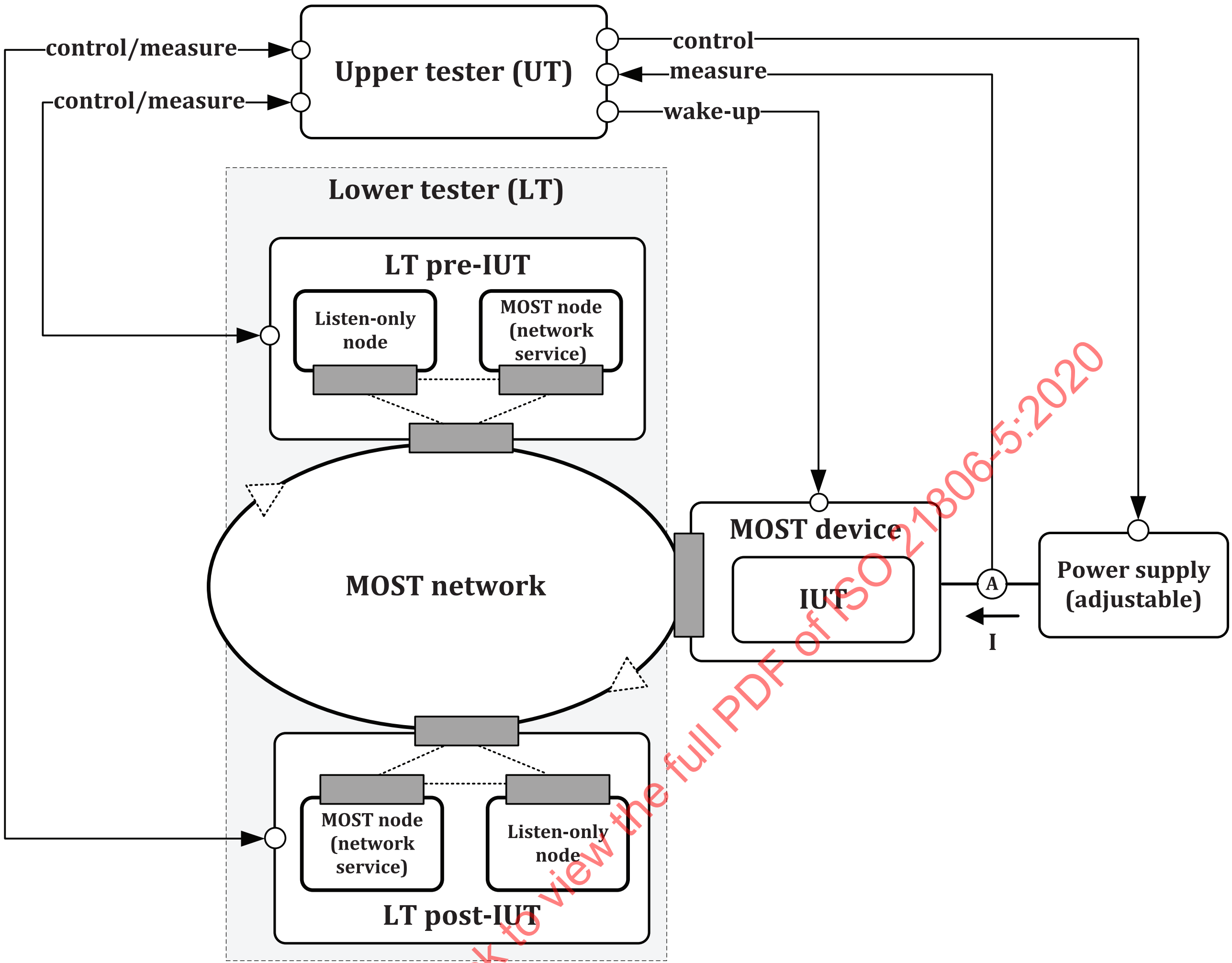


Figure 3 — Test set-up

The LT pre-IUT and the LT post-IUT implement the application layer services and the lower layer services of a MOST node in accordance with the ISO 21806 series. They also contain a listen-only node in front of the MOST node to log the whole communication. The MOST node is able to operate as TimingMaster or TimingSlave; alternatively, it can be physically disconnected from the MOST network. If it is disconnected, the associated LT pre-IUT or LT post-IUT serves as listen-only node.

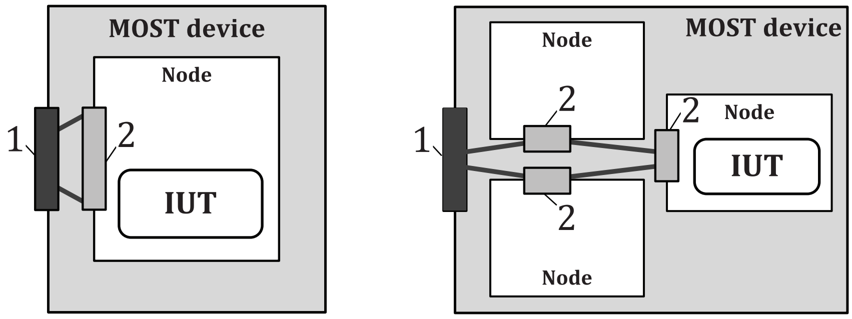
Every CTC specifies the roles of the LT pre-IUT and the LT post-IUT.

During testing of the MOST device that implements the IUT, avoid over-temperature by following the manufacturer recommendations regarding cooling.

The power supply of the MOST device that contains the IUT is adjustable and the power consumption can be monitored by the UT. This is necessary to determine whether a node has entered `s_NetInterface_Sleep`.

A MOST device contains one or more nodes, which are connected to an external MOST physical interface. One of the nodes contains the implementation under test (IUT). All tests and timings, specified by the CTP, are related to the external MOST physical interface.

Figure 4 shows a MOST device with one node and a MOST device with three nodes.



1 external MOST physical interface
2 internal MOST physical interface

Figure 4 — MOST device with one node and MOST device with three nodes

6.2 CTP organisation

CTCs are independent of one another. Each CTC checks the behaviour of the IUT for requirements stated in ISO 21806-4. Within CTCs, which require variations of individual parameters, each specified value of the parameter is iterated.

The measurement uncertainty for each CTC shall be in accordance with Annex A.

7 CTP general information

7.1 CTC remarks

7.1.1 Timer naming

For conformance testing of the IUT, the UT and LT need minimum and maximum timers. The names of the timers used by this document are based on ISO 21806-2 and ISO 21806-4. To obtain the timer name, for minimum and maximum, "$_{\_min}$" and "$_{\_max}$" are appended, respectively. Table 1 shows a timer naming definition example for t_{Config}.

Table 1 — Timer naming example

Name	Minimum value name	Typical value name	Maximum value name	Unit	Purpose
t_{Config}	t_{Config_min}	t_{Config}	t_{Config_max}	ms	Time before ev_Init_Error_Shutdown or delay for RBD result.

7.1.2 Deadlock prevention

This document specifies the timeouts $t_{DeadLockShort}$, $t_{DeadLockMid}$, and $t_{DeadLockLong}$ to prevent deadlock situations during conformance testing. These are the default values:

— $t_{DeadLockShort}$: 1 s;

— $t_{DeadLockMid}$: 20 s;

— $t_{DeadLockLong}$: 5 min.

These timeouts are only relevant for conformance testing and may be extended.

7.1.3 Un-initialised logical node address

The variable `uninitialised_node_address` is defined as the address of an un-initialised node, which is specified in ISO 21806-2.

7.1.4 Addresses of MOST nodes in the LT

The address of a MOST node in the LT is the default logical node address corresponding to the node position.

If this address is in conflict with the address of a node that contains the IUT (e.g. if a supplier uses static addresses in the dynamic address range), the affected MOST node in the LT shall use a valid free address.

7.1.5 Device manufacturer information list

This list contains all information that is provided by the device manufacturer for conformance testing. It also includes remarks and references to corresponding CTCs.

Table 2 shows the device manufacturer information list, which does not include information stored in FBlock EnhancedTestability.

Table 2 — Device manufacturer information list

Category	Item/property	Description	Reference to CTC
MOST network configuration	IUT in the TimingMaster	Determines whether the IUT is part of the TimingMaster.	All CTCs
	IUT in the NetworkMaster	Determines whether the IUT is part of the NetworkMaster.	All CTCs
	IUT in the PowerMaster	Determines whether the IUT is part of the PowerMaster.	All CTCs
	Multi-node device	If the IUT is part of a MOST device that contains more than one node, the following information is provided: — number of nodes in the MOST device; — topology of the MOST device (position of PowerMaster and TimingMaster/NetworkMaster); — position of the node that contains the IUT.	All CTCs
	IUT sample frequency	If the IUT is not part of the TimingMaster, the LT provides the correct network frame rate (44,1 kHz or 48,0 kHz).	All CTCs
	Required value of boundary descriptor (if the TimingMaster is in the LT)	Value of the boundary descriptor Unless otherwise stated, all CTCs are performed with this value of the boundary descriptor.	All CTCs
Power management	Node that contains the IUT is capable of waking via network startup (i.e. switching on its MOST output)	---	CTC_2.1.1-6a, CTC_2.1.3-1
	Delay between connection to power (of the MOST device that contains the IUT) and the ability of the node that contains the IUT to detect wake-up events	Potentially, the UT (see Figure 3) waits for a short period of time between connecting the MOST device that contains the IUT to power and switching on the MOST output to wake up the node that contains the IUT. Otherwise, the node that contains the IUT does not detect a wake-up event.	All CTCs

Table 2 *(continued)*

Category	Item/property	Description	Reference to CTC
General communication	$t_{Property}$	Limit for responding to a command that reads a property.	CTC_2.8.4-3
Physical parameter (voltage levels)	$U_{IUT\_Operating}$	At this voltage level, the MOST device that contains the IUT operates normally. Unless otherwise stated, all CTCs are performed at this voltage level.	All CTCs
Messaging	Node that contains the IUT supports segmented transfers	The IUT can send and receive segmented transfers.	CTC_2.8.4-2, CTC_2.8.4-3, CTC_2.8.4-7, CTC_2.8.4-8

7.1.6 States of the node that contains the IUT

Table 3 specifies how the NetInterface state `s_NetInterface_Normal_Operation` is effectuated and detected in the node that contains the IUT.

Table 3 — Effectuate and detect s_NetInterface_Normal_Operation

Effectuate state	Detect state
a) The IUT is contained in a NetworkSlave: The UT shall: — start the network, — wait for the node that contains the IUT to open its bypass (MPI is nominal[a]), — send `NetworkMaster.Configuration.Status(NotOK)`, — perform an FBlock scan (including retries if the address of the node that contains the IUT is invalid), — send `NetworkMaster.Configuration.Status(OK)`, and — wait for $t_{WaitForApplication}$.	a) The IUT is contained in a NetworkSlave: the node that contains the IUT responds to `NetBlock.FBlockIDs.Get`
b) The IUT is contained in the NetworkMaster: — the UT shall behave like a NetworkSlave. It shall process and respond to all requests from the node that contains the IUT so that the node can enter central registry state OK; — the UT shall respond to an FBlock scan by the node that contains the IUT. Additionally, the UT shall wait for the node to open its bypass (MPI is nominal[a]) if the node is part of a multi-node device; — finally, the UT shall wait for $t_{WaitForApplication}$.	b) The IUT is contained in the NetworkMaster: the node that contains the IUT responds to `NetBlock.FBlockIDs.Get`
[a] The nominal MPI is the total number of nodes in the test set-up, based on the device manufacturer information and the test equipment.	

Table 4 specifies how the NetInterface state `s_NetInterface_Off` is effectuated and detected in the node that contains the IUT.

Table 4 — Effectuate and detect s_NetInterface_Off

Effectuate state	Detect state
The LT shall switch off the MOST output.	No network activity

7.1.7 Procedures

Table 5 specifies the procedures of the LT.

Table 5 — Procedures of the LT

Purpose	Description
Perform wake-up	a) If the LT contains the TimingMaster, it shall execute this sequence to perform a wake-up: 1. switch on the MOST output; 2. wait for network activity (timeout $t_{DeadLockShort}$); 3. wait for stable lock (timeout $t_{DeadLockMid}$). b) If the LT does not contain the TimingMaster, it shall execute this sequence to perform a wake-up: 1. generate a wake-up event; 2. wait for network activity; 3. switch on the MOST output; 4. wait for stable lock; 5. wait for the lock flag to evaluate to true. If the LT does not generate a wake-up event and detects network activity, it shall switch on the MOST output. In some cases, the node that contains the IUT needs some preconditions for wake-up. These preconditions are established before testing is started. The preconditions depend on the device manufacturer. If `EnhancedTestability.AutoWakeup` is triggered, the node that contains the IUT does not enter `s_NetInterface_Sleep` before creating the corresponding wake-up event. The node enters `s_NetInterface_Off`. This state is not detectable by monitoring the power consumption of the MOST device that contains IUT. With entering `s_NetInterface_Off`, the node that contains the IUT switches off the MOST output.
Perform shutdown	a) If the IUT is part of the PowerMaster, the LT shall execute this sequence to perform shutdown: — trigger shutdown by means of FBlock EnhancedTestability; — if no network activity is detected, the node that contains the IUT has performed shutdown; — if $t_{DeadLockMid}$ expires after triggering shutdown and the LT still detects network activity, it shall switch off the MOST output. b) If the IUT is part of a PowerSlave, the LT shall switch off the MOST output to perform shutdown. If the IUT is part of the PowerMaster, no preconditions are established that prevent the node that contains the IUT from performing shutdown.
Generate unlock	To generate an unlock event of predictable duration, the LT — shall invalidate or delay the preamble at the beginning of at least every third network frame during the period of unlock (see ISO 21806-6), and — shall avoid a PLL unlock.
MOST output on	The LT shall switch on the MOST output.
MOST output off	The LT shall switch off the MOST output.

Table 5 *(continued)*

Purpose	Description
Network change event with unlock	The NCE shall be generated between the TimingMaster and the node that contains the IUT.
Network change event without unlock	The NCE shall be generated between the node that contains the IUT and the TimingMaster.

7.1.8 Violation of prerequisites of the CTC

If the node that contains the IUT does not meet the prerequisites of the CTC (such as network activity, lock, FBlock scan performed, central registry state OK), the CTC results in "IUT not ok: the IUT does not meet the prerequisites".

7.2 CTC items

7.2.1 FBlock EnhancedTestability (ET)

The FBlock EnhancedTestability is used to trigger sequences that are specified in the CTC. These are normally triggered by a project specific, sometimes complicated, mechanism. FBlock EnhancedTestability implements neither notification nor processing messages. The UT shall initialise FBlock EnhancedTestability every time the `s_NetInterface_Normal_Operation` state is reached. The FBlock is only available during `s_NetInterface_Normal_Operation`. All properties are reset to their default state when entering `s_NetInterface_Normal_Operation`, unless otherwise stated. The functions in this FBlock describe a general interface for starting functionality partly implemented in the application, partly in the MOST network service. If an application callback returns wrong or unexpected values, the FBlock sends an error message with `ErrorCode` $0B_{16}$ (device malfunction).

If the FBlock EnhancedTestability returns errors, the corresponding CTC result is indicated as “IUT not ok”.

7.2.2 Multi-node devices

A multi-node device is a MOST device with an external MOST physical interface, which is connected to more than one internal node.

When dealing with MOST devices with several external MOST physical interfaces, each MOST physical interface shall be treated separately. A MOST physical interface may be connected to more than one node; each node shall be treated individually.

NOTE A MOST device passes conformance testing successfully if all contained IUTs pass all relevant CTCs.

7.2.3 Node kinds excluded from conformance testing

CTCs for remote controlled nodes according to ISO 21806-2 are not in the scope of this document.

CTCs for remote nodes according to ISO 21806-14[4] are not in the scope of this document.

CTCs for root nodes according to ISO 21806-14[4] are not in the scope of this document.

8 CTC specification

8.1 Wake-up

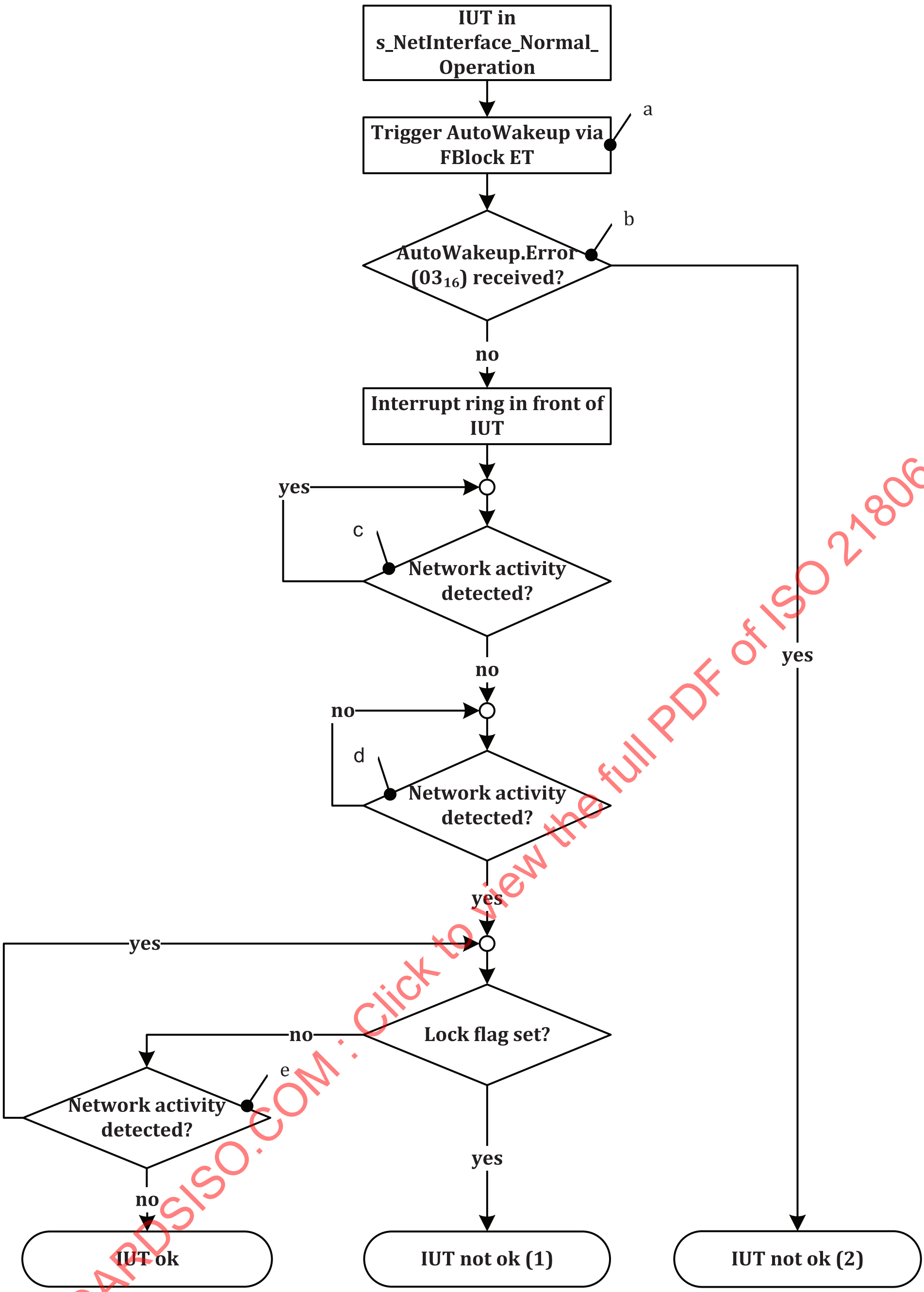
8.1.1 TimingMaster wake-up

8.1.1.1 CTC_2.1.1-6a – Lock flag test (a)

Table 6 specifies the lock flag test (a).

Table 6 — CTC_2.1.1-6a – Lock flag test (a)

Item	Content
CTC # – Title	CTC_2.1.1-6a – Lock flag test (a)
Purpose	This CTC verifies that the TimingMaster is capable of waking the network and setting the lock flag correctly. This CTC applies to all MOST devices that contain the TimingMaster and are capable of waking via network startup.
Reference	ISO 21806-4:2020, 8.2.1.3.4 NL – NetInterface state `s_NetInterface_Init`
Prerequisite	The UT shall effectuate `s_NetInterface_Normal_Operation` in the IUT.
Set-up	— The LT pre-IUT shall be a TimingSlave. — The LT post-IUT shall be a listen-only node.
Step	1. The UT shall execute the sequence as specified in Figure 5.
Iteration	Not applicable
Expected response	Step 1: IUT ok: the IUT supports the wake-up correctly. Step 1: IUT not ok (1): the IUT sets the lock flag erroneously without stable lock. Step 1: IUT not ok (2): the IUT does not support wake-up.
Remark	---



a The UT shall trigger a wake-up event and interrupt the ring in front of the IUT.
Use `EnhancedTestability.AutoWakeup` with the following parameters:

— DelayTime = 1 s;

— Duration = 0 s.

b The IUT does not support wake-up.

c To avoid deadlocks, the test may be aborted when $t_{DeadLockMid}$ expires.
The IUT is expected to switch off the MOST output within $t_{Shutdown\_max}$ after $t_{SSO\_Shutdown\_max}$ expires.

d To avoid deadlocks, the test is aborted when $t_{DeadLockMid}$ expires.

e To avoid deadlocks, the test is aborted when t_{Config_max} expires.
As long as the IUT keeps the MOST output switched on, the lock flag is not set.

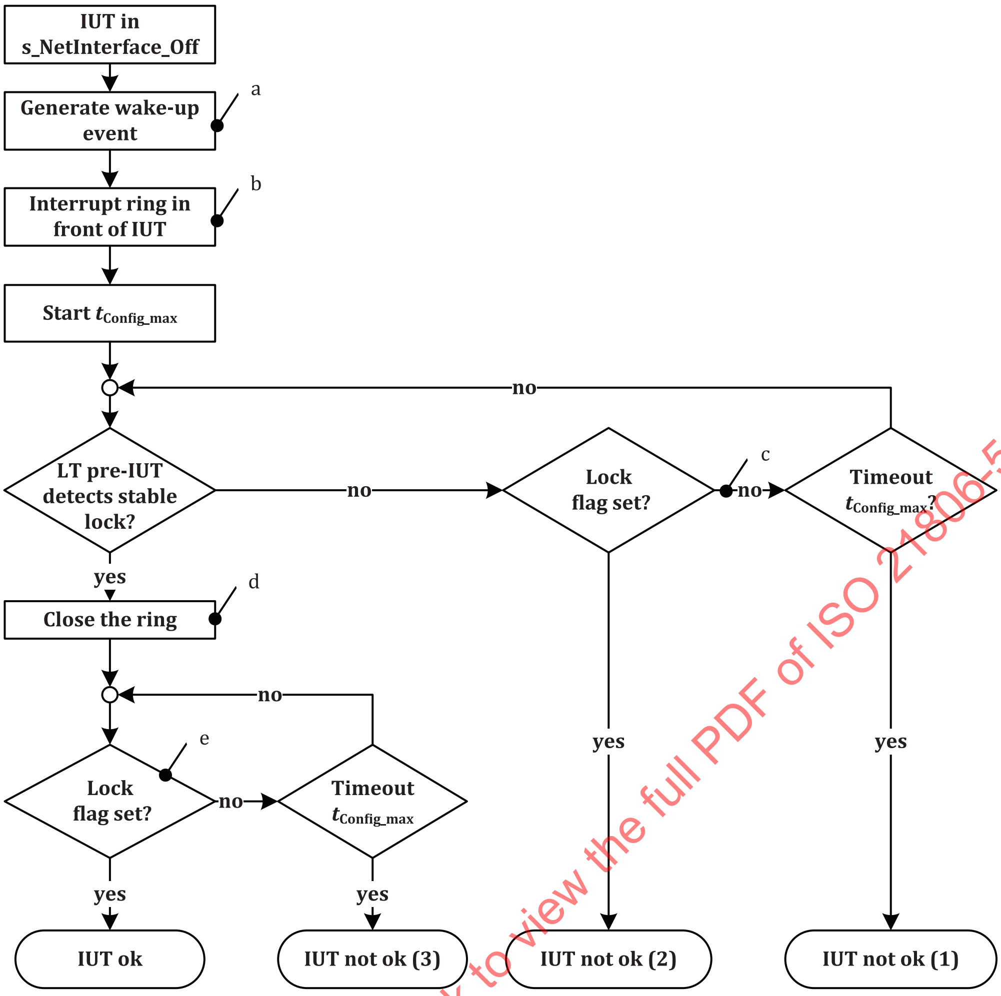
Figure 5 — CTC_2.1.1-6a – Lock flag test (a)

8.1.1.2 CTC_2.1.1-6b – Lock flag test (b)

Table 7 specifies the lock flag test (b).

Table 7 — CTC_2.1.1-6b – Lock flag test (b)

Item	Content
CTC # – Title	CTC_2.1.1-6b – Lock flag test (b)
Purpose	This CTC verifies that the TimingMaster detects a closed ring and sets the lock flag within t_{Config_max}. This CTC applies to all MOST devices that contain the TimingMaster.
Reference	ISO 21806-4:2020, 8.2.1.3.4 NL – NetInterface state `s_NetInterface_Init`
Prerequisite	— The UT shall effectuate `s_NetInterface_Off` in the IUT. — The ring shall be closed.
Set-up	— The LT pre-IUT shall be a TimingSlave. — The LT post-IUT shall be a listen-only node.
Step	1. The UT shall execute the sequence as specified in Figure 6.
Iteration	Not applicable
Expected response	Step 1: IUT ok: the IUT detects a closed ring and sets the lock flag accordingly. Step 1: IUT not ok (1): timeout t_{Config} without stable lock. Step 1: IUT not ok (2): lock flag is erroneously set. Step 1: IUT not ok (3): lock flag is not set.
Remark	---



a The UT shall trigger a wake-up event.

b The UT shall disable the MOST output of the LT pre-IUT (ring interrupted in front of the IUT).

c The IUT does not set the lock flag.

d If the UT detects a stable lock, it shall close the ring, using the LT pre-IUT, by switching on the MOST output.

e The IUT sets the lock flag within t_{Config_max}.

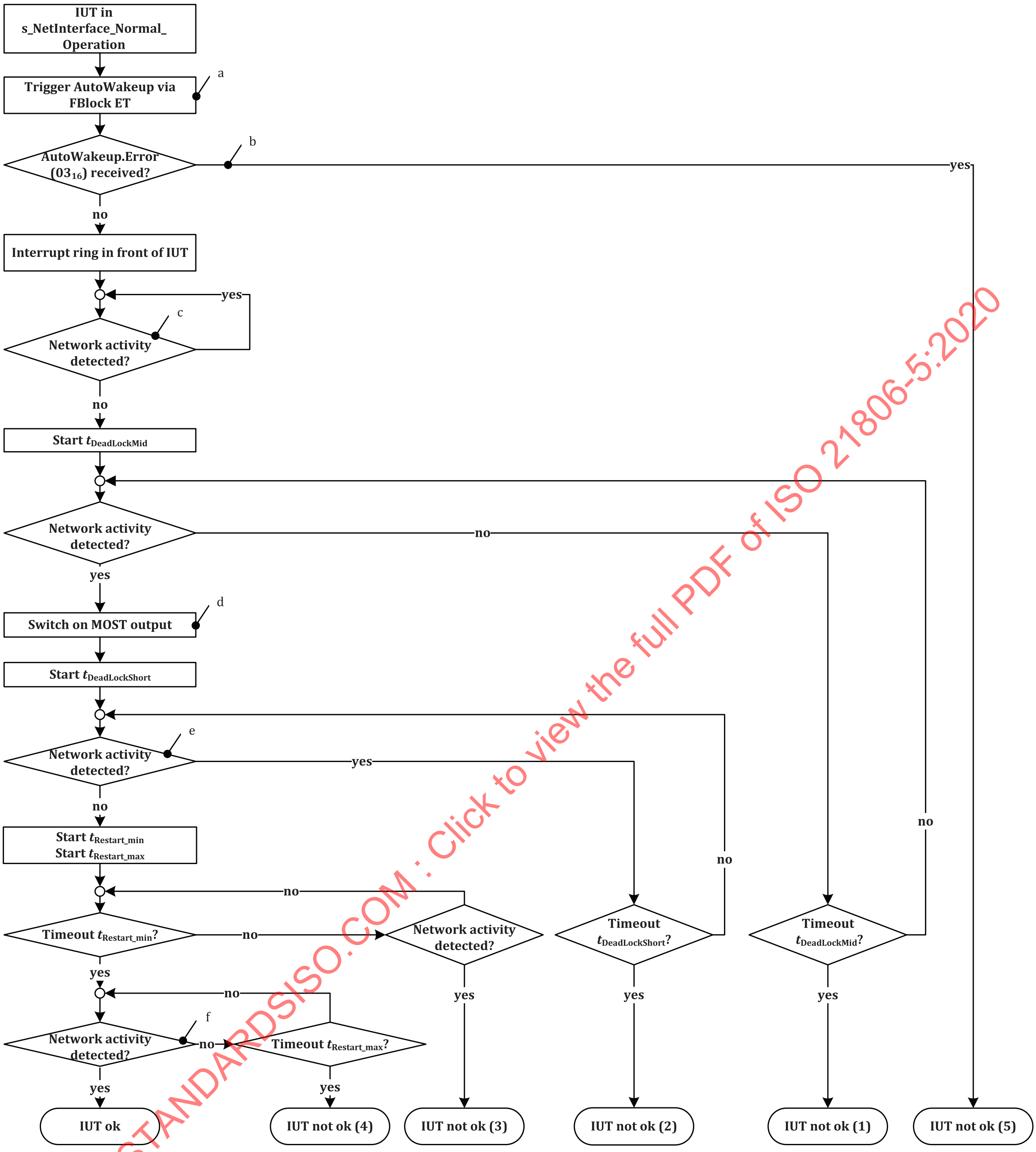
Figure 6 — CTC_2.1.1-6b – Lock flag test (b)

8.1.2 CTC_2.1.3-1 – TimingSlave wake-up test

Table 8 specifies the TimingSlave wake-up test.

Table 8 — CTC_2.1.3-1 – TimingSlave wake-up test

Item	Content
CTC # – Title	CTC_2.1.3-1 – TimingSlave wake-up test
Purpose	This CTC verifies that a TimingSlave is capable of performing a network startup attempt correctly and under consideration of $t_{Restart}$. This CTC applies to all MOST devices that do not contain the TimingMaster and are capable of waking by switching on the MOST output.
Reference	ISO 21806-4:2020:
	— 8.2.1.3.4 NL – NetInterface state `s_NetInterface_Init`; — 8.2.1.4 NL – Network wake-up and startup.
Prerequisite	The UT shall effectuate `s_NetInterface_Normal_Operation` in the IUT.
Set-up	— The LT pre-IUT shall be the TimingMaster. — The LT post-IUT shall be a listen-only node.
Step	1. The UT shall execute the sequence as specified in Figure 7.
Iteration	Not applicable
Expected response	Step 1: IUT ok: the IUT performs network startup attempts and retries correctly. Step 1: IUT not ok (1): the IUT does not switch on the MOST output after wake-up (or external wake-up not supported). Step 1: IUT not ok (2): the IUT does not switch off the MOST output within $t_{DeadLockShort}$. Step 1: IUT not ok (3): the IUT does not wait $t_{Restart}$ before switching on the MOST output again. Step 1: IUT not ok (4): the IUT does not switch on the MOST output before $t_{Restart}$ expires. Step 1: IUT not ok (5): the IUT does not support wake-up.
Remark	---



a The UT shall trigger `AutoWakeup`, using the LT pre-IUT, via FBlock EnhancedTestability.
Use `EnhancedTestability.AutoWakeup` with the following parameters:

— DelayTime = 1 s;

— Duration = 0 s.

b The IUT does not support wake-up.

c The IUT switches on the MOST output according to Table 2. To avoid deadlocks, test can be aborted when $t_{DeadLockMid}$ expires. The IUT is expected to switch off the MOST output within $t_{Shutdown\_max}$ after $t_{SSO\_Shutdown\_max}$ expires.

d If the LT pre-IUT detects network activity, it shall switch on the MOST output.

e The IUT switches off the MOST output within $t_{DeadLockShort}$.

f The IUT waits for $t_{Restart\_min}$ before switching on the MOST output again.

Figure 7 — CTC_2.1.3-1 – TimingSlave wake-up test

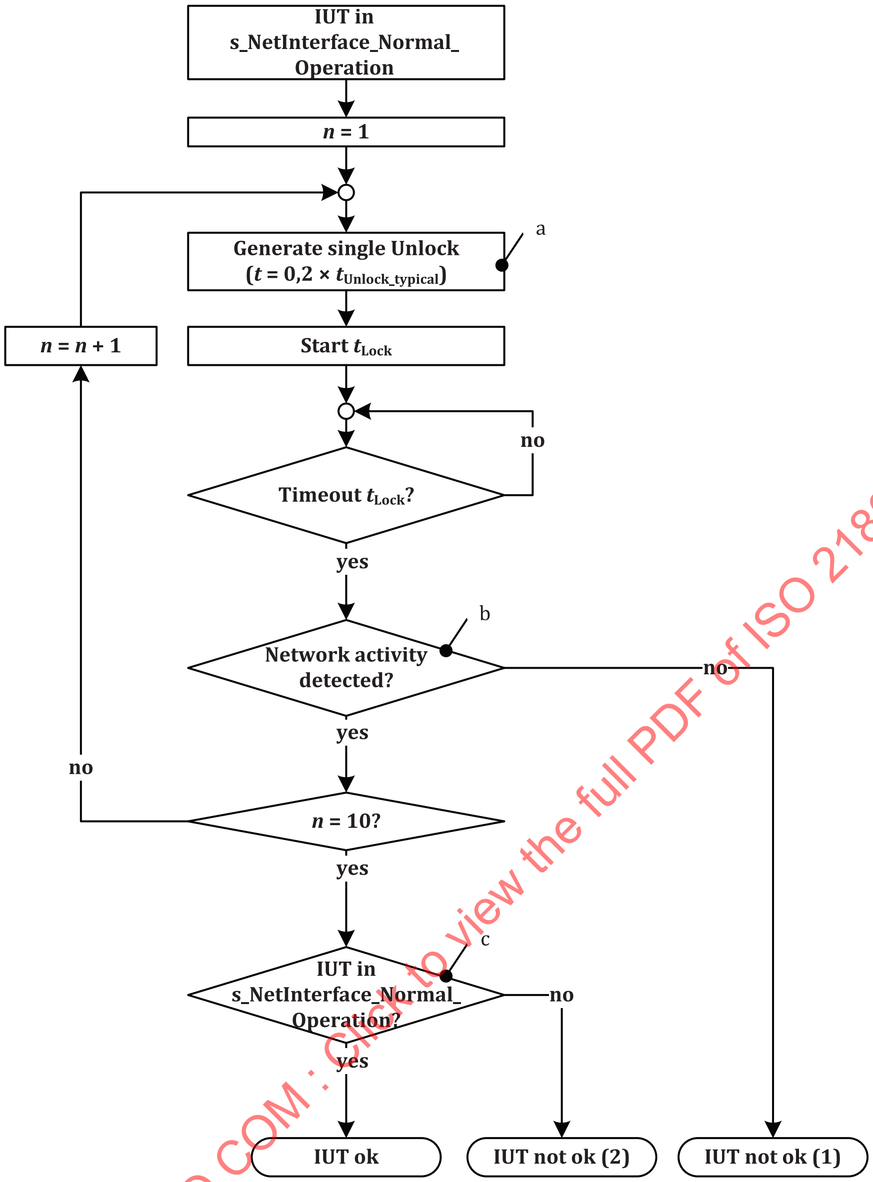
8.2 s_NetInterface_Normal_Operation – Unlock

8.2.1 CTC_2.2.1-3 – Short unlock test

Table 9 specifies the CTC_2.2.1-3 – Short unlock test.

Table 9 — CTC_2.2.1-3 – Short unlock test

Item	Content
CTC # – Title	CTC_2.2.1-3 – Short unlock test
Purpose	This CTC verifies that a node remains in `s_NetInterface_Normal_Operation` when a sequence of short unlocks that are interspersed with locks occurs. This CTC applies to all MOST devices.
Reference	ISO 21806-4:2020, 8.2.1.3.5 NL – NetInterface state `s_NetInterface_Normal_Operation`
Prerequisite	The UT shall effectuate `s_NetInterface_Normal_Operation` in the IUT.
Set-up	— The IUT is part of a MOST device that contains the TimingMaster: — the LT pre-IUT shall be a TimingSlave; — the LT post-IUT shall be a listen-only node. — The IUT is part of a MOST device that does not contain the TimingMaster: — the LT pre-IUT shall be a TimingSlave; — the LT post-IUT shall be the TimingMaster.
Step	1. The UT shall execute the sequence as specified in Figure 8.
Iteration	Not applicable
Expected response	Step 1: IUT ok: the IUT remains in `s_NetInterface_Normal_Operation`. Step 1: IUT not ok (1): the IUT has switched off the MOST output. Step 1: IUT not ok (2): the IUT does not reach `s_NetInterface_Normal_Operation` at the end of the test.
Remark	1. The reaction of the application (mute) is not tested by this CTC. 2. Procedure for unlock generation is described in Clause 7.



a The UT, using the LT pre-IUT, shall generate 10 short unlocks (each 0,2 × $t_{Unlock\_typical}$) with a pause of t_{Lock} between each unlock.

b The IUT does not switch off the MOST output.

c At the end of the test, the IUT remains in `s_NetInterface_Normal_Operation`.

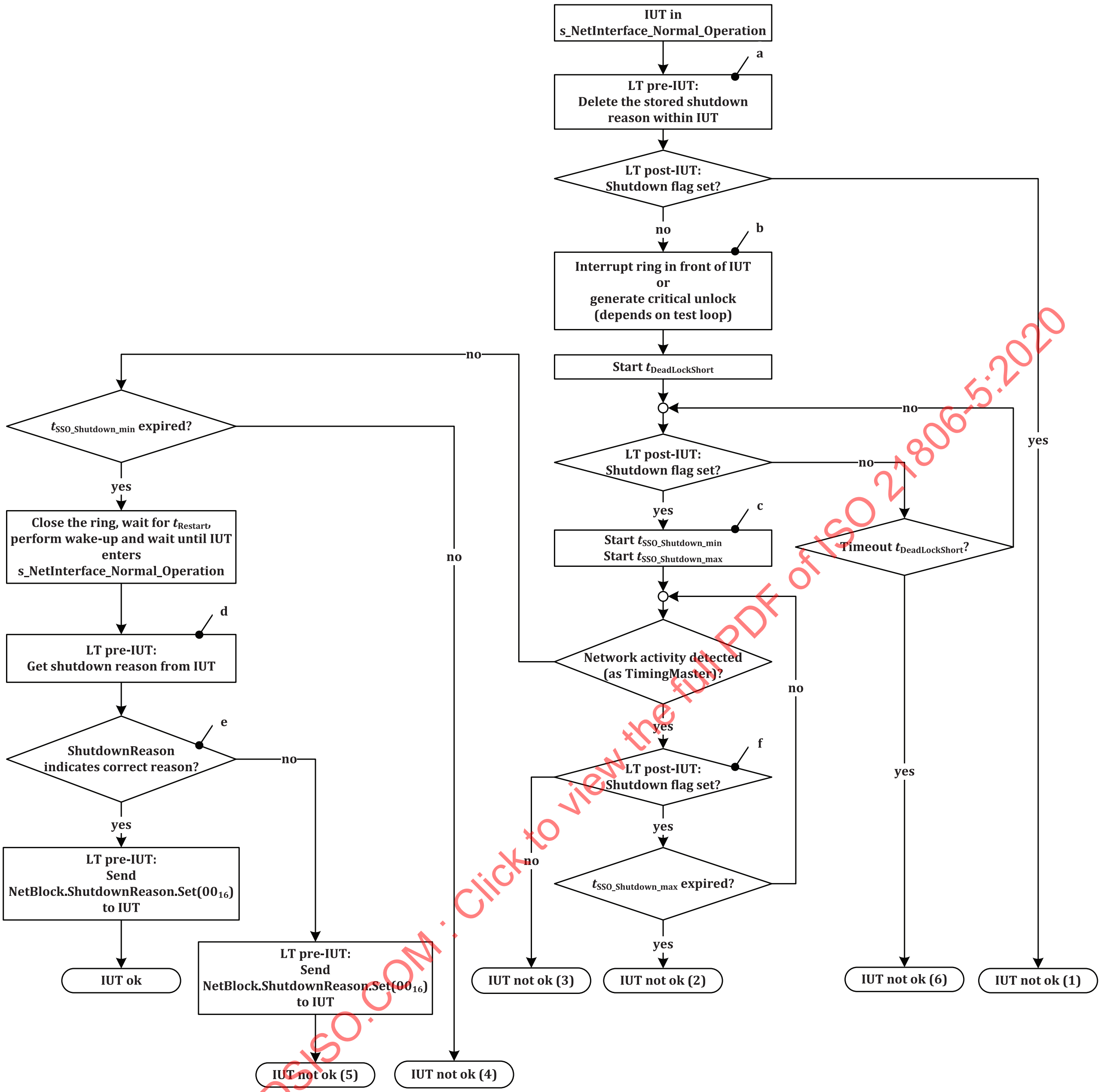
Figure 8 — CTC_2.2.1-3 – Short unlock test

8.2.2 CTC_2.2.1-6 – Sudden signal off test

Table 10 specifies the sudden signal off test.

Table 10 — CTC_2.2.1-6 – Sudden signal off test

Item	Content
CTC # – Title	CTC_2.2.1-6 – Sudden signal off test
Purpose	This CTC verifies that a node correctly detects a sudden signal off or an unlock, behaves accordingly, and stores the respective shutdown reason. This CTC applies to all MOST devices. In multi-node devices only the first node is tested.
Reference	ISO 21806-4:2020, 8.2.2.3 NL – Handling sudden signal off and critical unlock
Prerequisite	The UT shall effectuate `s_NetInterface_Normal_Operation` in the IUT.
Set-up	— If the IUT is part of a MOST device that contains the TimingMaster, the LT pre-IUT shall be a TimingSlave. — If the IUT is part of a MOST device that does not contain the TimingMaster, the LT pre-IUT shall be the TimingMaster. — The LT post-IUT shall be a TimingSlave.
Step	1. The UT shall execute the sequence as specified in Figure 9.
Iteration	REPEAT Step 1; n = 2 times 1. generate sudden signal off (interrupt ring); 2. critical unlock. REPEAT END.
Expected response	Step 1: IUT ok: the IUT correctly reacts to sudden signal off. Step 1: IUT not ok (1): the IUT does not set shutdown flag. Step 1: IUT not ok (2): the IUT does not switch off the MOST output. Step 1: IUT not ok (3): the IUT does not set shutdown flag. Step 1: IUT not ok (4): the IUT switches off the MOST output too early. Step 1: IUT not ok (5): the IUT reports the wrong shutdown reason after restart. Step 1: IUT not ok (6): the IUT does not set the shutdown flag correctly.
Remark	---



a To delete the shutdown reason, send `NetBlock.ShutdownReason.Set`(00_{16}) to the node that contains the IUT. If it responds with “FktID not available”, retry later until the result is available.

b The test shall be performed twice:

1. generate sudden signal off (The UT shall interrupt the ring in front of the IUT.);
2. critical unlock (generated by the LT pre-IUT).

c Start the timer as soon as the shutdown flag is set.

[d] To get the shutdown reason, send `NetBlock.ShutdownReason.Get` to the node that contains the IUT, using node position addressing.

If the node that contains the IUT responds with "`FktID` not available", retry later until the result is available.

In the case of a multi-node device: shutdown reason only valid for first node of the MOST device.

[e] After restart, the node that contains the IUT reports the correct shutdown reason.

Correct reason depends on test loop:

— In the case of sudden signal off, the shutdown reason is "Sudden Signal Off";

— In the case of critical unlock, the shutdown reason is "Critical Unlock".

[f] The IUT sets the shutdown flag and keeps the MOST output switched on as TimingMaster until $t_{SSO\_Shutdown}$ expires.

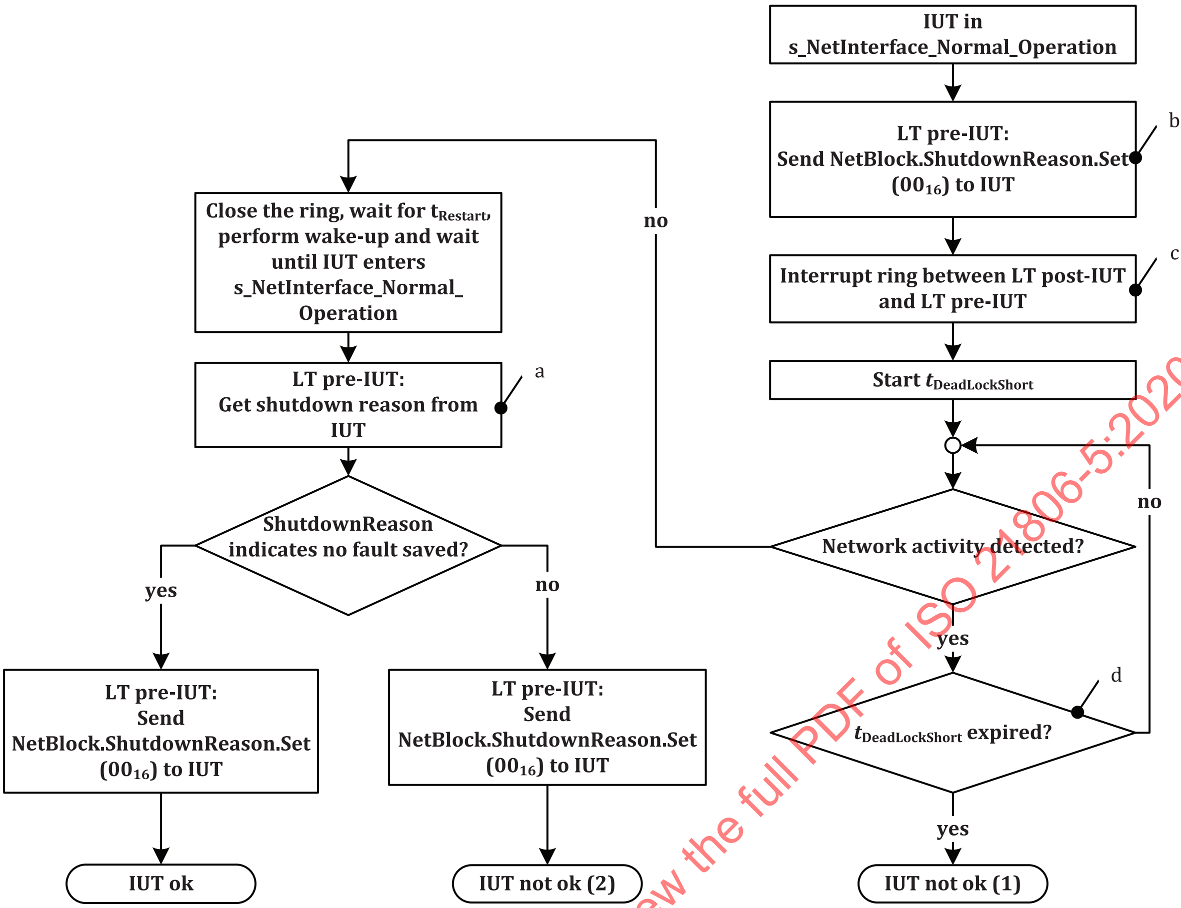
Figure 9 — CTC_2.2.1-6 – Sudden signal off test

8.2.3 CTC_2.2.1-7 – Shutdown flag present test

Table 11 specifies the shutdown flag present test.

Table 11 — CTC_2.2.1-7 – Shutdown flag present test

Item	Content
CTC # – Title	CTC_2.2.1-7 – Shutdown flag present test
Purpose	This CTC verifies that a TimingSlave stores the correct shutdown reason. This CTC applies to all MOST devices that do not contain the TimingMaster.
Reference	ISO 21806-4:2020, 8.2.2.3 NL – Handling sudden signal off and critical unlock
Prerequisite	The UT shall effectuate `s_NetInterface_Normal_Operation` in the IUT.
Set-up	— The LT pre-IUT shall be the TimingMaster. — The LT post-IUT shall be a listen-only node.
Step	1. The UT shall execute the sequence as specified in Figure 10.
Iteration	Not applicable
Expected response	Step 1: IUT ok: the IUT switches off the MOST output and stores the correct shutdown reason. Step 1: IUT not ok (1): the IUT does not switch off the MOST output. Step 1: IUT not ok (2): the IUT reports wrong shutdown reason after restart.
Remark	---



a After restart, the node that contains the IUT reports the correct shutdown reason.

To get the shutdown reason, send `NetBlock.ShutdownReason.Get` to the node that contains the IUT, using node position addressing.

If the node that contains the IUT responds with "`FktID` not available", retry later until the result is available.

b Set the shutdown reason of the node that contains the IUT to "No result available".

c The UT shall interrupt the ring between the LT post-IUT and the LT pre-IUT.

This causes the LT pre-IUT to detect an SSO and set the shutdown flag automatically.

d The IUT switches off the MOST output within $t_{DeadLockShort}$.

Figure 10 — CTC_2.2.1-7 – Shutdown flag present test

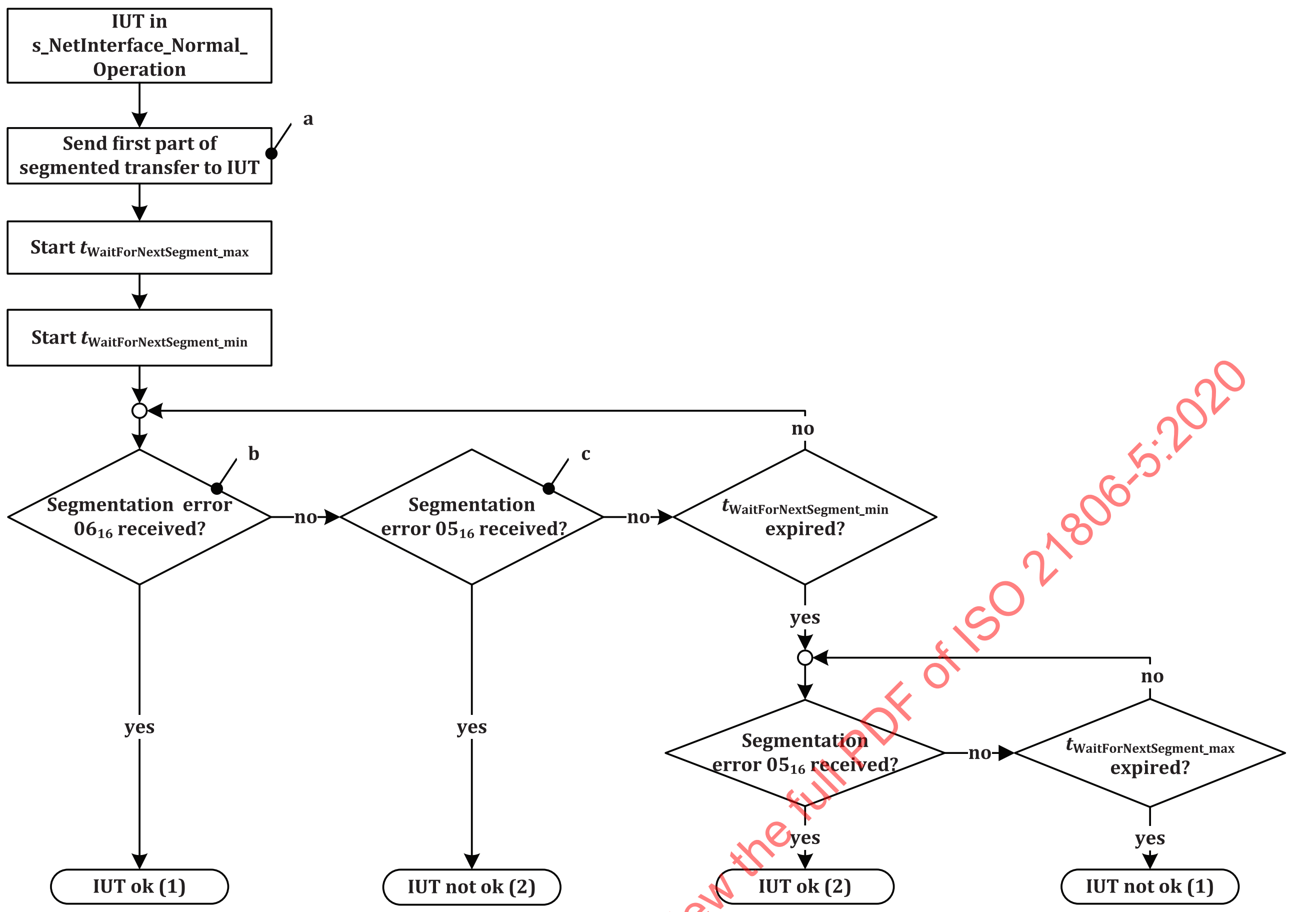
8.3 Segmented transfer

8.3.1 CTC_2.8.4-1 – Segmented transfer rejection test

Table 12 specifies the CTC_2.8.4-1 – Segmented transfer rejection test.

Table 12 — CTC_2.8.4-1 – Segmented transfer rejection test

Item	Content
CTC # – Title	CTC_2.8.4-1 – Segmented transfer rejection test
Purpose	This CTC verifies that a node that — supports segmented transfer correctly detects interruption of segmented transfer; — does not support segmented transfer reports the corresponding error. This CTC applies to all MOST devices.
Reference	ISO 21806-4:2020, 7.2.2.1 TL – Application message service (AMS)
Prerequisite	— The UT shall effectuate `s_NetInterface_Normal_Operation` in the IUT. — The ring shall be closed.
Set-up	— If the IUT is part of a MOST device that contains the TimingMaster, the LT pre-IUT shall be a TimingSlave. — If the IUT is part of a MOST device that does not contain the TimingMaster, the LT pre-IUT shall be the TimingMaster. — The LT post-IUT shall be a listen-only node.
Step	1. The UT shall execute the sequence as specified in Figure 11.
Iteration	Not applicable
Expected response	Step 1: IUT ok (1): the IUT does not support segmented transfers. Step 1: IUT ok (2): the IUT supports segmented transfers. Step 1: IUT not ok (1): the IUT does not report a segmentation error (`ErrorCode` $0C_{16}$, `ErrorInfo` 05_{16}) in time. Step 1: IUT not ok (2): the IUT reports a segmentation error (`ErrorCode` $0C_{16}$, `ErrorInfo` 05_{16}) too early.
Remark	Timer values for $t_{WaitForNextSegment}$ are: — minimum 4 950 ms; — typical 5 000 ms; — maximum 10 150 ms.



a The UT, using the LT pre-IUT, shall send the first part of a segmented transfer (EnhancedTestability.EchoMessage.StartResult) to the node that contains the IUT.
Use the message EnhancedTestability.EchoMessage.StartResult with arbitrary data.

b If the IUT does not support segmented transfers, it responds with a segmentation error (ErrorCode $0C_{16}$, ErrorInfo 06_{16}).

c If the IUT supports segmented transfers, it responds with a segmentation error (ErrorCode $0C_{16}$, ErrorInfo 05_{16}).

Figure 11 — CTC_2.8.4-1 – Segmented transfer rejection test

8.3.2 CTC_2.8.4-2 – Segmented transfer sending test

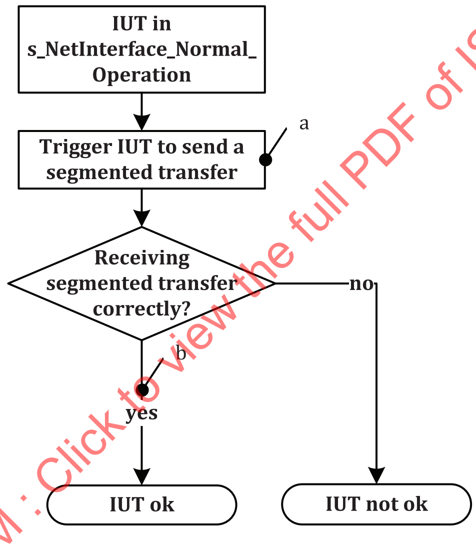
Table 13 specifies segmented transfer sending test.

Table 13 — CTC_2.8.4-2 – Segmented transfer sending test

Item	Content
CTC # – Title	CTC_2.8.4-2 – Segmented transfer sending test
Purpose	This CTC verifies that a node is capable of sending a complete segmented transfer. This CTC applies to all MOST devices that contain at least one node that supports segmented transfer.
Reference	ISO 21806-4:2020, 7.2.2.8 TL – Segmented transfer
Prerequisite	— The UT shall effectuate s_NetInterface_Normal_Operation in the IUT. — The ring shall be closed.

Table 13 *(continued)*

Item	Content
Set-up	— If the IUT is part of a MOST device that contains the TimingMaster, the LT pre-IUT shall be a TimingSlave. — If the IUT is part of a MOST device that does not contain the TimingMaster, the LT pre-IUT shall be the TimingMaster. — The LT post-IUT shall be a listen-only node.
Step	1. The UT shall execute the sequence as specified in Figure 12.
Iteration	Not applicable
Expected response	Step 1: IUT ok: the IUT sends a segmented transfer correctly. Step 1: IUT not ok: the IUT does not send a segmented transfer correctly.
Remark	Possibly, some nodes can send segmented transfers but cannot receive them.



a The UT, using the LT pre-IUT, shall trigger the node that contains the IUT (via `EnhancedTestability.SendMessage`) to send a segmented transfer to the LT pre-IUT:

— depending on the buffer size given by FBlock EnhancedTestability, the message consists of the size-prefix and two, three, or four segments;

— in the case that the MessageBufSize is dynamic (indicated by value 0) the IUT sends the size-prefix and at least L_{AMSmax} + 1 bytes and not more than four segments.

b The IUT sends the message correctly. Correct reception means:

— transfer is started with transmission of the size-prefix (`TelID 4`);

— the sequence of messages is correct;

— all segments are received before the garbage collection gets activated (maximum $t_{WaitForNextSegment\_max}$ between two segments);

— the whole message is sent with correct timing ($t_{WaitForNextSegment\_max}$ multiplied by the number of segments).

— The `TelID` sequence adheres to ISO 21806-4 transfer of payload, for example, 1, 2, 2, 2, 3.

Figure 12 — CTC_2.8.4-2 – Segmented transfer sending test

8.3.3 CTC_2.8.4-3 – Segmentation error test

Table 14 specifies the CTC_2.8.4-3 – Segmentation error test.

Table 14 — CTC_2.8.4-3 – Segmentation error test

Item	Content
CTC # – Title	CTC_2.8.4-3 – Segmentation error test
Purpose	This CTC verifies that a node correctly detects segmentation errors when receiving segmented transfers. This CTC applies to all MOST devices that contain at least one node that supports segmented transfer.
Reference	ISO 21806-4:2020, TL – Segmented transfer
Prerequisite	— The UT shall effectuate s_NetInterface_Normal_Operation in the IUT. — The ring shall be closed.
Set-up	— If the IUT is part of a MOST device that contains the TimingMaster, the LT pre-IUT shall be a TimingSlave. — If the IUT is part of a MOST device that does not contain the TimingMaster, the LT pre-IUT shall be the TimingMaster. — The LT post-IUT shall be a listen-only node.
Step	1. The UT shall execute the sequence as specified in Figure 13.
Iteration	Not applicable
Expected response	Step 1: IUT ok: the IUT detects segmentation errors when receiving segmented transfers. Step 1: IUT not ok (1): the IUT does not detect/indicate missing first part of segmented transfer. Step 1: IUT not ok (2): the IUT does not detect/indicate wrong order of parts of segmented transfer. Step 1: IUT not ok (3): the IUT does not detect/indicate parts of segmented transfer that are sent twice.
Remark	---