MOST

Media **O**riented **S**ystems **T**ransport
Multimedia and Control
Networking Technology

MOST150 cPHY Duplex Network Diagnosis

Rev. 3.0
07/2017

MOSTCO CONFIDENTIAL

See page 3 for the terms of disclosure

Legal Notice

COPYRIGHT

LICENSE DISCLAIMER

Nothing on any MOST Cooperation Web Site, or in any MOST Cooperation document, shall be construed as conferring any license under any of the MOST Cooperation or its members or any third party's intellectual property rights, whether by estoppel, implication, or otherwise.

CONTENT AND LIABILITY DISCLAIMER

MOST Cooperation or its members shall not be responsible for any errors or omissions contained at any MOST Cooperation Web Site, or in any MOST Cooperation document, and reserves the right to make changes without notice. Accordingly, all MOST Cooperation and third party information is provided "AS IS". In addition, MOST Cooperation or its members are not responsible for the content of any other Web Site linked to any MOST Cooperation Web Site. Links are provided as Internet navigation tools only.

MOST COOPERATION AND ITS MEMBERS DISCLAIM ALL WARRANTIES WITH REGARD TO THE INFORMATION (INCLUDING ANY SOFTWARE) PROVIDED, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. Some jurisdictions do not allow the exclusion of implied warranties, so the above exclusion may not apply to you.

In no event shall MOST Cooperation or its members be liable for any damages whatsoever, and in particular MOST Cooperation or its members shall not be liable for special, indirect, consequential, or incidental damages, or damages for lost profits, loss of revenue, or loss of use, arising out of or related to any MOST Cooperation Web Site, any MOST Cooperation document, or the information contained in it, whether such damages arise in contract, negligence, tort, under statute, in equity, at law or otherwise.

FEEDBACK INFORMATION

Any information provided to MOST Cooperation in connection with any MOST Cooperation Web Site, or any MOST Cooperation document, shall be provided by the submitter and received by MOST Cooperation on a non-confidential basis. MOST Cooperation shall be free to use such information on an unrestricted basis.

TRADEMARKS

MOST Cooperation and its members prohibit the unauthorized use of any of their trademarks. MOST Cooperation specifically prohibits the use of the MOST Cooperation LOGO unless the use is approved by the Steering Committee of MOST Cooperation.

SUPPORT AND FURTHER INFORMATION

For more information on the MOST technology, please contact:

MOST Cooperation
Administration
Emmy-Noether-Str. 14
76131 Karlsruhe
Germany
Tel: (+49) (0) 721 966 50 00
E-mail: contact@mostcooperation.com
Web: www.mostcooperation.com

Contents

Document History

MOST150 cPHY Duplex Network Diagnosis Rev. 3.0

Change Ref.	Section	Changes
3V0_001	All	Initial document.

Bibliography

Information: Table Bibliography-1 lists all documents, which are referenced by this MOST document, along with their versions.

Document	Revision
MOST Specification	3.0
MOST150 cPHY Automotive Sub-Specification	1.1

Table Bibliography-1: Document references

Glossary

Information: Table Glossary-1 lists commonly used MOST terms and their definition.

Term	Definition
cPHY	MOST coaxial physical layer
Diagnosis evaluator	In the TimingMaster, the diagnosis evaluator starts the cPHY duplex network diagnosis process and evaluates the results that are provided by the diagnosis worker.
Diagnosis flag	The diagnosis flag in the administrative area of the network frame indicates whether diagnosis is active.
`Diagnosis_Initiate`	This request is used by the diagnosis worker to instruct the Network Interface Controller to switch on the MOST signal and set the diagnosis flag.
`Diagnosis_Initiated`	This report is sent from the Network Interface Controller to the diagnosis worker to confirm that the MOST signal is switched on and the diagnosis flag is set.
`Diagnosis_End`	This request is sent from the diagnosis worker to the Network Interface Controller to terminate the cPHY duplex network diagnosis process by switching off the MOST signal.
`Diagnosis_Ended`	This report is sent from the Network Interface Controller to the diagnosis worker to confirm that the MOST signal is switched off.
`Diagnosis_Result`	This report is sent from the diagnosis worker to the diagnosis evaluator to provide the information about a link on which cable link diagnosis was performed.
Diagnosis worker	In the TimingMaster, the diagnosis worker communicates with the nodes in the MOST network through the Network Interface Controller and provides results to the diagnosis evaluator.
`End_Diag`	This report is sent from the diagnosis worker to the diagnosis evaluator to indicate that the cPHY duplex network diagnosis process is completed.
MOST port	A MOST port is the MOST Network Interface Controller's connection point to the MOST Physical Interface.
`Node_Identified`	This report is sent from the diagnosis worker to the diagnosis evaluator to indicate that the next node in the network provided its signature.
Remote controlled node	For certain use cases that do not require FBlock communication outside the administrative range (`FBlockID 0x00 – 0x0F`), a node kind exists, which is called "remote controlled node".
Signature	The signature of a node contains the `DiagID` and other relevant information such as version information.
`Start_Diag`	This request is used by the diagnosis evaluator to instruct the diagnosis worker to begin the cPHY duplex network diagnosis process.
Weak link	A weak link between two ports is a link with reduced signal quality. A weak link exists if network activity is present but Stable Lock cannot be reached.

Table Glossary-1: Glossary entries

1 Introduction

1.1 Purpose

This document specifies the diagnosis concept for a MOST network that is based on a full-duplex version of the MOST coaxial physical layer (cPHY).

1.2 Scope

This document contains the specification of the diagnosis method including the cable link diagnostics.

1.3 Motivation

The motivation and advantage of using the cPHY duplex network diagnosis are:

- Evaluation is independent of the physical layout of a MOST network.
- A MOST port that requires cable link diagnostics can be detected during the cPHY duplex network diagnosis process.
- A unique scheme of reserved addresses is established, independent of the node's node position address.
- The physical position of a node in the network is determined.

2 Overview

2.1 Preconditions

- The MOST network is based on cPHY full duplex by design.
- MOST ports to be inspected must not be in power-down mode.
- The TimingMaster node contains an application to control the cPHY duplex network diagnosis.
- For TimingSlave nodes, the cPHY duplex network diagnosis is implemented in the MOST Network Interface Controller. This ensures that remote controlled nodes are included.
- Communication is based on MOST messages via the MOST control channel.
- To start the cPHY duplex network diagnosis, the TimingMaster needs to be in NetInterface Off state.
- The cPHY duplex network diagnosis does not require any additional hardware or wiring (e.g., ECL) for starting and during execution.

2.2 Behavior

The cPHY duplex network diagnosis is intended to perform the following actions:

- Signal the cPHY duplex network diagnosis state to every node in the MOST network by a flag in the administrative area of the network frame.
- Enable and disable MOST ports and assign logical node addresses.
- Find the nodes that are present in the network and map out the network topology.
- From all nodes, gather node-specific properties contained in the signature.
- If a node does not respond, trigger verification of the physical link to determine whether the issue is with the cable connection or the connected node.

2.3 Limitations

What the cPHY duplex network diagnosis is not designed for and the limitations:

- Performing cPHY duplex network diagnosis in a ring network topology is not possible (e.g., dual-simplex cPHY connections).
- The cPHY duplex network diagnosis cannot detect a weak link between two nodes.
- In a daisy chain topology, only the first defective link can be found.
- Cable link diagnostics cannot distinguish between an open cable and a short circuit.

2.4 Network Topologies

The cPHY duplex network diagnosis is performed separately on every branch. If the TimingMaster is a multiport MOST Network Interface Controller, as shown in Figure 1, the cPHY duplex network diagnosis cannot be executed on all network ports at once.

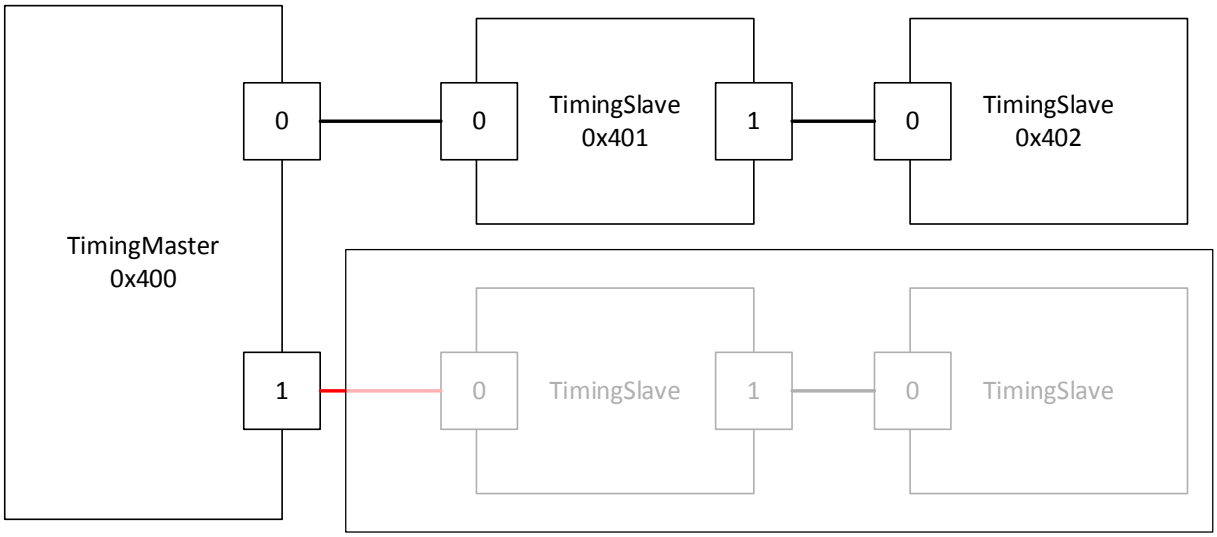

Figure 1 - Network topology example with multiple branches

In this case, the diagnosis process is performed network port by network port.

3 Diagnosis method

For the supported topologies, the cPHY duplex network diagnosis method is independent of the physical layout of a MOST network, that is, the number of nodes and the topology of node connections.

Communication over the MOST control channel is used to get information about connected nodes. In case of a not responding node, cable link diagnostics is executed. During the cPHY duplex network diagnosis, the network is reported as not available for application communication.

The TimingMaster distributes a diagnosis flag during the cPHY duplex network diagnosis in order to signal the cPHY duplex network diagnosis state. This flag ensures that a TimingSlave node detects that the MOST network is in the cPHY duplex network diagnosis state, immediately when it is able to lock to the MOST signal.

3.1 Initiate the diagnosis

Based on the System Integrator's specification, the diagnosis evaluator determines when the cPHY duplex network diagnosis is started.

When the cPHY duplex network diagnosis is started, the following actions are performed by the MOST Network Interface Controller:

- Distribute the diagnosis flag (TimingMaster node only)
- Enter the cPHY duplex network diagnosis state
- Disable its MOST ports. TimingSlaves do not disable the clock reference port (in the case of a multiport MOST Network Interface Controller)
- Indicate to the application that the MOST network is not available
- Set the node address to the reserved value `0x0FFE` indicating an uninitialized node

The diagnosis evaluator starts the cPHY duplex network diagnosis by issuing a Start_Diag (see 3.7.1 for details about request and reports) request to the diagnosis worker. The diagnosis worker requests Diagnosis_Initiate from the MOST Network Interface Controller to initiate the diagnosis process.

The local MOST Network Interface Controller is asked for the signature and its logical node address is set to `0x0F00`.

3.2 Explore the network

A branch consists of two or more MOST nodes. In a branch with more than two MOST nodes, these nodes are connected in a daisy chain.

If the TimingMaster is a multiport MOST Network Interface Controller, the diagnosis worker application enables each of the TimingMaster's MOST ports in succession and checks one branch after the other. In this way, a multiport network topology is reduced to multiple single-branch networks.

When the diagnosis flag is detected, the MOST ports that do not receive the TimingMaster's MOST signal (i.e., in a multiport Network Interface Controller scenario, those ports that are no clock reference ports) are disabled.

The diagnosis worker application in the TimingMaster node is responsible for enabling the disabled MOST port of a multiport TimingSlave node. Nodes join the branch one by one when a MOST port is enabled.

3.3 Identify nodes

3.3.1 General

One by one, the diagnosis worker requests the node-specific properties from the nodes that joined the network. The diagnosis worker reports the nodes with their node-specific properties to the diagnosis evaluator.

Based on this information, the diagnosis evaluator may perform a nominal-actual comparison between the reported nodes and a pre-configured list of nodes, to decide if the diagnosis process was passed or failed.

3.3.2 Process description

To request node-specific properties, the diagnosis worker sends the ExtendedNetworkControl (ENC) FBlock function `Hello.Get` to the blocking broadcast address. A TimingSlave responds with a `Hello.Status` message, including its node specific signature. After the TimingSlave node responds, the diagnosis worker sets that node's temporary logical node address by sending `ENC.Welcome.StartResult`, which is sent to the node position address of the TimingSlave. The TimingMaster adds the node to the network. After receiving the `ENC.Welcome.StartResult` message, the node no longer answers to `Hello.Get` broadcasts.

The logical node address of a TimingSlave is temporary because it is only valid for the duration of the cPHY duplex network diagnosis. The logical node addresses of TimingSlave nodes are set to `0x0F01`, `0x0F02`, … in ascending order.

The TimingSlave node compares the signature of the `ENC.Welcome.StartResult` message to its internal signature, which it reported in the `ENC.Hello.Status` message. Only if the signature matches, the node changes its node logical node address and responds with a `ENC.Welcome.Result` message. Otherwise error handling is triggered.

The diagnosis worker sends `ENC.EnablePort.StartResult` to the temporary logical node address (`0x0F01`, `0x0F02`, …) to enable the disabled MOST port of the TimingSlave.

The same steps are executed for all TimingSlave nodes one by one.

3.4 Evaluate intermediate results

Every successfully verified link is signaled by the diagnosis worker to the application with the signatures of the first and the second node of the link.

If the connection is interrupted as shown in Figure 2, sending `ENC.CableLinkDiagnosis.StartResult` using the temporary logical node address to the last reachable node in that branch will start cable link diagnostics on that specific MOST port (in this example this is the MOST port 1 of TimingSlave 0x0401). The result of cable link diagnostics may indicate a defective cable connection or a defective node.

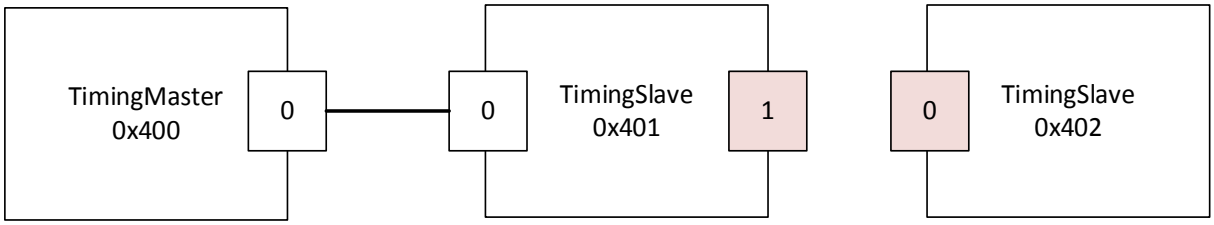

Figure 2 - Defective Link

Depending on the MOST network configuration, an erroneous connection status (e.g., such as an open cable) does not necessarily imply a defect but rather a port could be left unconnected on purpose.

If there are multiple defective links within a branch, only the first defect can be found. After the defect was fixed, a diagnosis needs to be started again in order to detect further link defects within that branch.
A defective link is signaled by the diagnosis worker to the diagnosis evaluator with the signature of the last reachable node and the result of cable link diagnostics.

If more than one node responds to the `ENC.Hello.Get` message, an error is signaled to the diagnosis evaluator. It is the responsibility of the diagnosis evaluator to perform proper error handling (e.g., to start the diagnosis of that branch from the very beginning).

The same is valid if a node enters NetInterface Off state unexpectedly. In this case, the node sets its logical node address to `0x0FFE`, indicating an uninitialized node.

3.5 Perform cable link diagnostics

Since the nodes in the MOST network are addressed by control messages, all connections between nodes that can be addressed are considered as failure-free with respect to their physical linkage. As a result, a weak link between two nodes cannot be detected by the cPHY duplex network diagnosis.

If a node does not answer to a control message within t_{Hello}, cable link diagnostics is executed to get further information about the physical connection status. Cable link diagnostics determines whether a cable issue (open or shorted) or a defective node (unpowered MOST port or the node's bypass is closed) is present. If a fully operational node is reported, the result of cable link diagnostics is inconclusive.

3.6 End the diagnosis process

The cPHY duplex network diagnosis is finished after all branches are examined or an erroneous event is found.

The cPHY duplex network diagnosis process may be interrupted by unpredictable errors (e.g., an unexpected NetInterface Off). An error ends the cPHY duplex network diagnosis process and needs further application handling (e.g., a restart of the diagnosis process).

The exploration of the MOST network continues until the last node of the last branch was found. The last node is found when:

- The last MOST Network Interface Controller that has joined the network has only one MOST port.
- The last MOST Network Interface Controller that has joined the network has no additional active MOST port.
- The last MOST Network Interface Controller that has joined the network has enabled an additional MOST port but no additional node answered to a communication request. In this case the cable link diagnostics process is started.
- An additional MOST port of the last MOST Network Interface Controller that has joined the network is not connected to full-duplex network topology, but e.g., to an oPHY network.

The diagnosis worker reports the end of the cPHY duplex network diagnosis to the diagnosis evaluator. The diagnosis evaluator performs the analysis of the gathered reports from the diagnosis worker and is responsible for the final verdict regarding the state of the MOST network.

After the cPHY duplex network diagnosis has ended, the MOST network stays in NetInterface Off state until a regular network startup is requested.

3.7 MSCs

3.7.1 Internal TimingMaster Communication

The following requests and reports are used to describe internal TimingMaster communication in the MSCs.

3.7.1.1 Diagnosis_Initiate

The `Diagnosis_Initiate` request is used by the diagnosis worker to instruct the Network Interface Controller to switch on the MOST signal and set the diagnosis flag. It contains no parameters.

```
Diagnosis_Initiate{
}
```

3.7.1.2 Diagnosis_Initiated

The `Diagnosis_Initiated` report is sent from the Network Interface Controller to the diagnosis worker to confirm that the MOST signal is switched on and the diagnosis flag is set.

```
Diagnosis_Initiated{
  Signature
}
```

3.7.1.2.1 **Signature**

The `Signature` parameter is identical to that of the `Hello` function, as specified in 5.1.1.3.

3.7.1.3 Diagnosis_End

The `Diagnosis_End` request is sent from the diagnosis worker to the Network Interface Controller to terminate the cPHY duplex network diagnosis process by switching off the MOST signal. It contains no parameters.

```
Diagnosis_End{
}
```

3.7.1.4 Diagnosis_Ended

The `Diagnosis_Ended` report is sent from the Network Interface Controller to the diagnosis worker to confirm that the MOST signal is switched off. It contains no parameters.

```
Diagnosis_Ended{
}
```

3.7.1.5 Diagnosis_Result

The Diagnosis_Result report is sent from the diagnosis worker to the diagnosis evaluator to provide the signature, MOST port number, and cPHY duplex network diagnosis result for a link on which cable link diagnosis was performed.

```
Diagnosis_Result{
  Signature
  PortNumber
  Result
}
```

3.7.1.5.1 **Signature**

The Signature parameter is identical to that of the Hello function, as specified in 5.1.1.3.

3.7.1.5.2 **PortNumber**

The PortNumber parameter is identical to that of the EnablePort function, as specified in 5.3.1.1.

3.7.1.5.3 **Result**

The Result parameter is identical to that of the CableLinkDiagnosis function, as specified in 5.4.1.2.

3.7.1.6 End_Diag

The End_Diag report is sent from the diagnosis worker to the diagnosis evaluator to indicate that the cPHY duplex network diagnosis process is completed. It contains no parameters.

```
End_Diag{
}
```

3.7.1.7 Node_Identified

The Node_Identified report is sent from the diagnosis worker to the diagnosis evaluator to indicate that the next node in the network provided its signature. Signature1 and Signature2 belong to the nodes on either end of the explored cable link, where Signature2 is the most recently added.

```
Node_Identified{
  Signature1
  Signature2
}
```

3.7.1.7.1 **Signature1**

The Signature1 parameter is identical to the Signature parameter of the Hello function, as specified in 5.1.1.3.

3.7.1.7.2 **Signature2**

The `Signature2` parameter is identical to to the `Signature` parameter of the `Hello` function, as specified in 5.1.1.3.

3.7.1.8 Start_Diag

The `Start_Diag` request is used by the diagnosis evaluator to instruct the diagnosis worker to begin the cPHY duplex network diagnosis process. It contains no parameters.

```
Start_Diag{
}
```

3.7.2 All Nodes Reachable

Figure 3 shows an example of a MOST network with three nodes, where every node is reachable during the cPHY duplex network diagnosis.

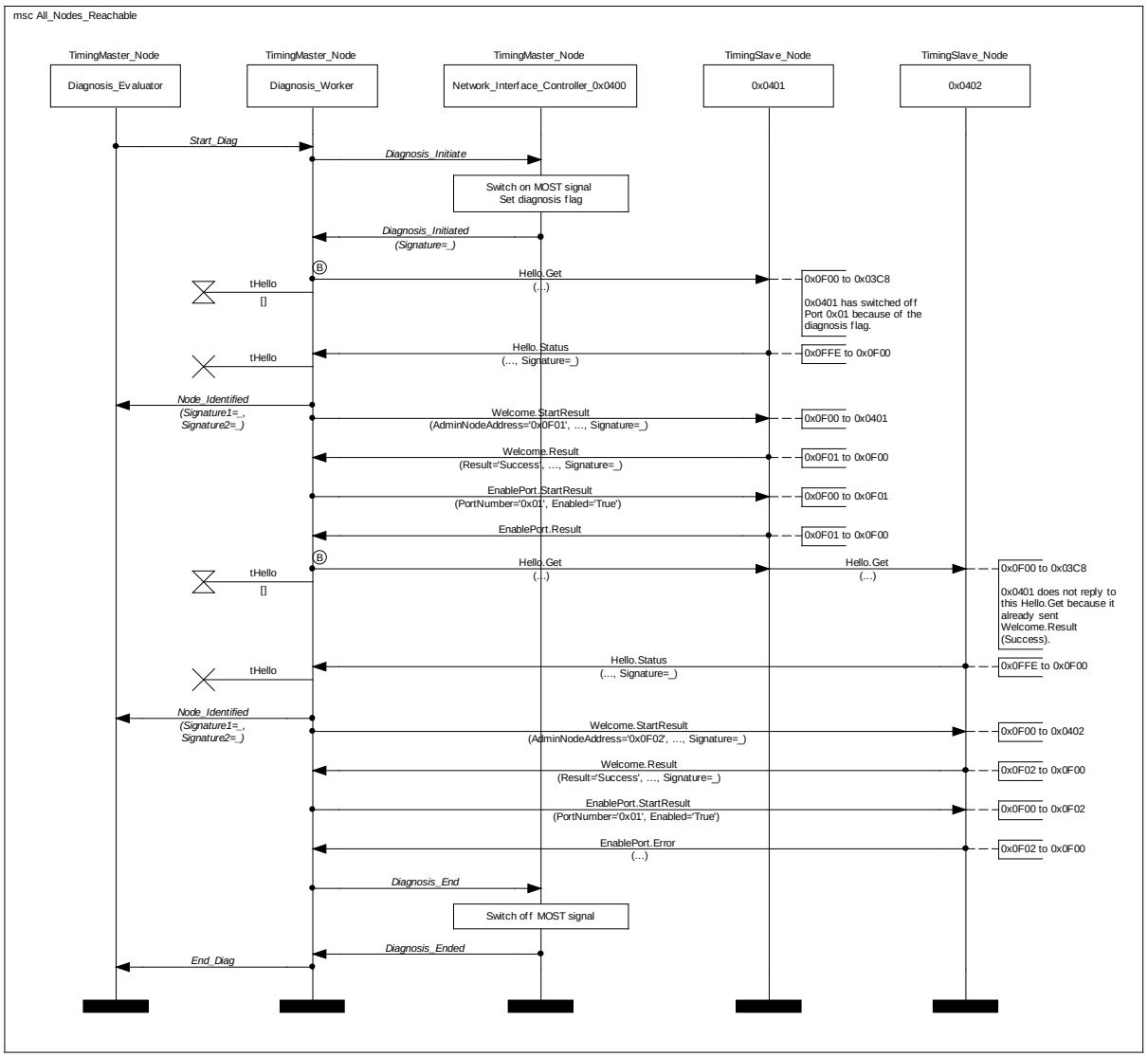

Figure 3 – All nodes reachable MSC

3.7.3 Not all Nodes Reachable

Figure 4 shows an example of a MOST network with three nodes, where one node is not reachable during the cPHY duplex network diagnosis and cable link diagnosis is started as a result.

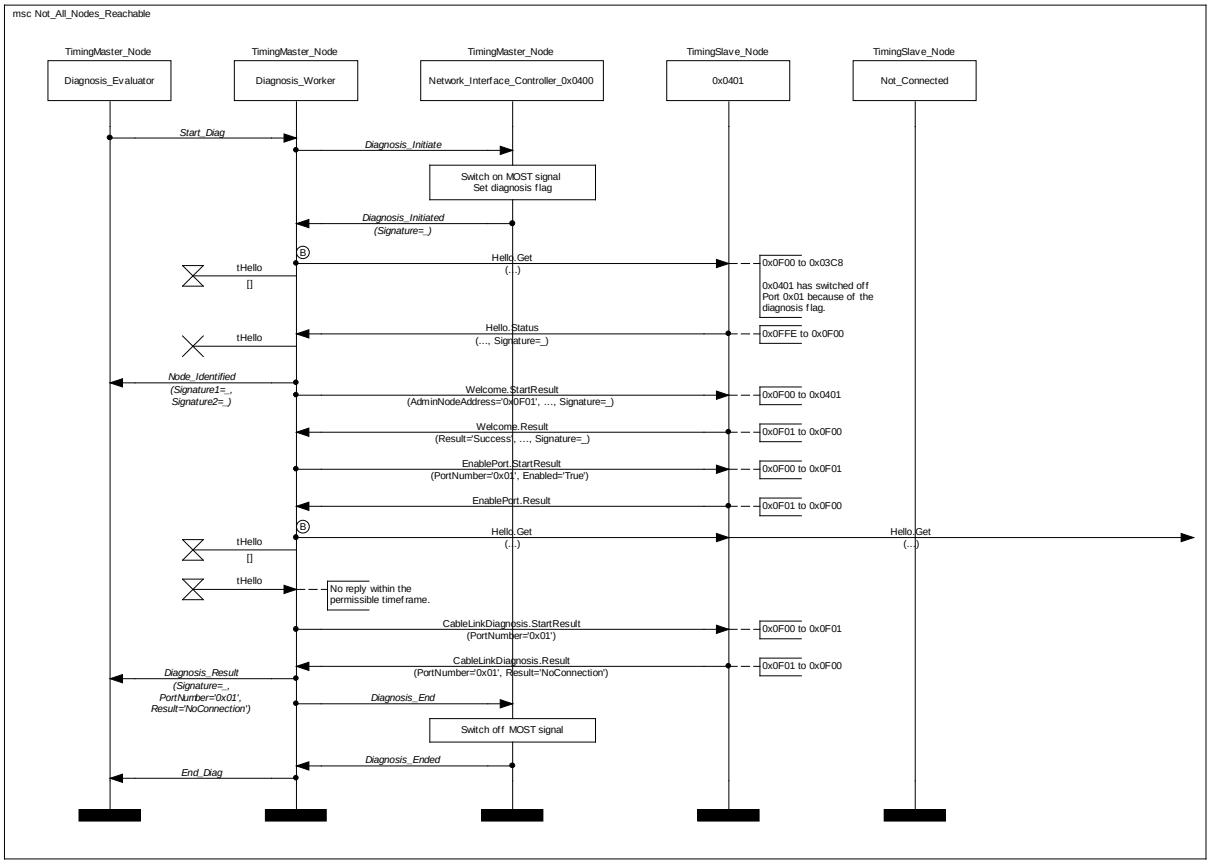

Figure 4 – Not all nodes reachable MSC

4 Timing definitions

Name	Min Value	Typ Value	Max Value	Unit	Purpose
t_{Hello}	System Integrator specific	-	System Integrator specific	ms	Time the diagnosis worker waits for a node to respond.

Table 4-1: Timer t_{Hello}

5 ExtendedNetworkControl FBlock functions

The following functions of FBlock ExtendedNetworkControl are used in the MOST150 cPHY duplex network diagnosis.

Function Name	FktID	Possible to Notify
Hello	0x200	No
Welcome	0x201	No
EnablePort	0x210	No
CableLinkDiagnosis	0x211	No

5.1 Hello (0x200)

This function is used to get the unique Signature of a node. The requesting node sends ExtendedNetworkControl.Hello.Get as a broadcast message to the MOST network.

Only those nodes that have not successfully sent ExtendedNetworkControl.Welcome.Result will answer with the ExtendedNetworkControl.Hello.Status message. The ExtendedNetworkControl.Hello.Status message is sent by using source address 0x0FFE.

FBlockID	FktID	OPType	Parameters
ExtendedNetwork-Control (0x0A)	Hello (0x200)	Get (0x1)	VersionLimit
		Status (0xC)	Version, Signature
		Error (0xF)	ErrorCode, ErrorInfo

5.1.1 Parameters

5.1.1.1 VersionLimit

Defines the maximum version of Signature the requesting node can handle.

Data Type	Exp.	Valid Values	Step	Unit
Unsigned Byte	0	1...255	1	None

5.1.1.2 Version

Defines the version of the Signature.

Data Type	Valid Values	Mnemonic	Description
Enum	0x01	v1	Version 1 of the Signature

5.1.1.3 Signature

Unique signature of a node.

Data Type	Number of Elements	Parameters
Stream	16	NodeAddress, GroupAddress, MACAddress_47to32, MACAddress_31to16, MACAddress_15to0, NodePositionAddress, DiagID, NumberOfPorts, ChipID, FWVersion_Major, FWVersion_Minor, FWVersion_Release, FWVersion_Build, CSVersion_Major, CSVersion_Minor, CSVersion_Release

5.1.1.4 NodeAddress

The logical node address can be customized.

Data Type	Exp.	Valid Values	Step	Unit
Unsigned Word	0	0x0100...0x013F	1	None
		0x0010...0x00FF, 0x0140...0x02FF, 0x0500...0x0EFF		
		0xFFFF	—	

5.1.1.5 GroupAddress

The group address can be customized.

Data Type	Exp.	Valid Values	Step	Unit
Unsigned Word	0	0x0300...0x03C7	1	None
		0x03C9...0x03FE		

5.1.1.6 MACAddress_47to32

Packet EUI-48 bits 47 to 32 of the MAC address. This parameter can be customized.

Data Type	Exp.	Valid Values	Step	Unit
Unsigned Word	0	0x0000...0xFFFF	1	None

5.1.1.7 MACAddress_31to16

Packet EUI-48 Bits 31 to 16 of the MAC address. This parameter can be customized.

Data Type	Exp.	Valid Values	Step	Unit
Unsigned Word	0	0x0000...0xFFFF	1	None

5.1.1.8 MACAddress_15to0

Packet EUI-48 Bits 15 to 0 of the MAC address. This parameter can be customized.

Data Type	Exp.	Valid Values	Step	Unit
Unsigned Word	0	0x0000...0xFFFF	1	None

5.1.1.9 NodePositionAddress

Current valid NodePositionAddress if network is available. 0x0400 is reported if the MOST network is not available.

Data Type	Exp.	Valid Values	Step	Unit
Unsigned Word	0	0x0400...0x043F	1	None

5.1.1.10 DiagID

The DiagID parameter can be customized.

Data Type	Exp.	Valid Values	Step	Unit
Unsigned Word	0	0x0000...0xFFFF	1	None

5.1.1.11 NumberOfPorts

Number of MOST ports available.

Data Type	Exp.	Valid Values	Step	Unit
Unsigned Byte	0	0x01...0x02	1	None

5.1.1.12 ChipID

Data Type	Exp.	Valid Values	Step	Unit
Unsigned Byte	0	0x00...0xFF	1	None

5.1.1.13 FWVersion_Major

Major version number of the firmware.

Data Type	Exp.	Valid Values	Step	Unit
Unsigned Byte	0	0x00...0xFF	1	None

5.1.1.14 FWVersion_Minor

Minor version number of the firmware.

Data Type	Exp.	Valid Values	Step	Unit
Unsigned Byte	0	0x00...0xFF	1	None

5.1.1.15 FWVersion_Release

Release version number of the firmware

Data Type	Exp.	Valid Values	Step	Unit
Unsigned Byte	0	0x00...0xFF	1	None

5.1.1.16 FWVersion_Build

Build version number of the firmware.

Data Type	Exp.	Valid Values	Step	Unit
Unsigned Long	0	0x00000000... 0xFFFFFFFF	1	None

5.1.1.17 CSVersion_Major

Major version number defined by the supplier.

Data Type	Exp.	Valid Values	Step	Unit
Unsigned Byte	0	0x00...0xFF	1	None

5.1.1.18 CSVersion_Minor

Minor version number defined by the supplier.

Data Type	Exp.	Valid Values	Step	Unit
Unsigned Byte	0	0x00...0xFF	1	None

5.1.1.19 CSVersion_Release

Release version number defined by the supplier.

Data Type	Exp.	Valid Values	Step	Unit
Unsigned Byte	0	0x00...0xFF	1	None

5.1.1.20 ErrorCode, ErrorInfo

ErrorCode	ErrorInfo		
	ErrorClass	ErrorID	Description of the Error
0x20	0x03	0x30	Node is already welcomed.

5.2 Welcome (0x201)

This function is used to welcome a node in the MOST network.

The function uses the Signature which was returned by the ExtendedNetworkControl.Hello.Status message. The receiving node compares the signature of the ExtendedNetworkControl.Welcome.StartResult message with its own signature sent by the ExtendedNetworkControl.Hello.Status message.

If the signature matches and the AdminNodeAddress is in the range of 0x0F00...0x0FEF, the NodeAddress is set to the AdminNodeAddress. The message is always sent to the NodePositionAddress.

FBlockID	FktID	OPType	Parameters
ExtendedNetwork-Control (0x0A)	Welcome (0x201)	StartResult (0x2)	AdminNodeAddress, Version, Signature
		Result (0xC)	Result, Version, Signature
		Error (0xF)	ErrorCode, ErrorInfo

5.2.1 Parameters

5.2.1.1 AdminNodeAddress

The node address used during the cPHY duplex network diagnosis.

Data Type	Exp.	Valid Values	Step	Unit
Unsigned Word	0	0x0F00...0x0FEF	1	None
		0xFFFF		

5.2.1.2 Version

Defines the version of the Signature.

Data Type	Valid Values	Mnemonic	Description
Enum	0x01	v1	Version 1 of the Signature

5.2.1.3 Signature

Contains the Signature values as returned by the ExtendedNetworkControl.Hello.Status message.

Data Type	Number of Elements	Parameters
Stream	16	NodeAddress, GroupAddress, MACAddress_47to32, MACAddress_31to16, MACAddress_15to0, NodePositionAddress, DiagID, NumberOfPorts, ChipID, FWVersion_Major, FWVersion_Minor, FWVersion_Release, FWVersion_Build, CSVersion_Major, CSVersion_Minor, CSVersion_Release

5.2.1.4 NodeAddress

Contains the NodeAddress as returned by parameter Signature in the ExtendedNetworkControl.Hello.Status message.

Data Type	Exp.	Valid Values	Step	Unit
Unsigned Word	0	0x0100...0x013F	1	None
		0x0010...0x00FF, 0x0140...0x02FF, 0x0500...0x0EFF		
		0xFFFF	—	

5.2.1.5 GroupAddress

Contains the GroupAddress as returned by parameter Signature in the ExtendedNetworkControl.Hello.Status message.

Data Type	Exp.	Valid Values	Step	Unit
Unsigned Word	0	0x0300...0x03C7	1	None
		0x03C9...0x03FE		

5.2.1.6 MACAddress_47to32

Contains bits 47 to 32 of the MAC address as returned by parameter Signature in the ExtendedNetworkControl.Hello.Status message.

Data Type	Exp.	Valid Values	Step	Unit
Unsigned Word	0	0x0000...0xFFFF	1	None

5.2.1.7 MACAddress_31to16

Contains bits 31 to 16 of the MAC address as returned by parameter Signature in the ExtendedNetworkControl.Hello.Status message.

Data Type	Exp.	Valid Values	Step	Unit
Unsigned Word	0	0x0000...0xFFFF	1	None

5.2.1.8 MACAddress_15to0

Contains bits 15 to 0 of the MAC address as returned by parameter Signature in the ExtendedNetworkControl.Hello.Status message.

Data Type	Exp.	Valid Values	Step	Unit
Unsigned Word	0	0x0000...0xFFFF	1	None

5.2.1.9 NodePositionAddress

Contains the NodePositionAddress as returned by parameter Signature in the ExtendedNetworkControl.Hello.Status message.

Data Type	Exp.	Valid Values	Step	Unit
Unsigned Word	0	0x0400...0x043F	1	None

5.2.1.10 DiagID

Contains the DiagID as returned by parameter Signature in the ExtendedNetworkControl.Hello.Status message.

Data Type	Exp.	Valid Values	Step	Unit
Unsigned Word	0	0x0000...0xFFFF	1	None

5.2.1.11 NumberOfPorts

Contains the NumberOfPorts as returned by parameter Signature in the ExtendedNetworkControl.Hello.Status message.

Data Type	Exp.	Valid Values	Step	Unit
Unsigned Byte	0	0x01...0x02	1	None

5.2.1.12 ChipID

Contains the ChipID as returned by parameter Signature in the ExtendedNetworkControl.Hello.Status message.

Data Type	Exp.	Valid Values	Step	Unit
Unsigned Byte	0	0x00...0xFF	1	None

5.2.1.13 FWVersion_Major

Contains the major firmware version as returned by parameter Signature in the ExtendedNetworkControl.Hello.Status message.

Data Type	Exp.	Valid Values	Step	Unit
Unsigned Byte	0	0x00...0xFF	1	None

5.2.1.14 FWVersion_Minor

Contains the minor firmware version as returned by parameter `Signature` in the `ExtendedNetworkControl.Hello.Status` message.

Data Type	Exp.	Valid Values	Step	Unit
Unsigned Byte	0	0x00...0xFF	1	None

5.2.1.15 FWVersion_Release

Contains the release firmware version as returned by parameter `Signature` in the `ExtendedNetworkControl.Hello.Status` message.

Data Type	Exp.	Valid Values	Step	Unit
Unsigned Byte	0	0x00...0xFF	1	None

5.2.1.16 FWVersion_Build

Build version number of the firmware.

Data Type	Exp.	Valid Values	Step	Unit
Unsigned Long	0	0x00000000... 0xFFFFFFFF	1	None

5.2.1.17 CSVersion_Major

Contains the supplier-defined major version as returned by parameter `Signature` in the `ExtendedNetworkControl.Hello.Status` message.

Data Type	Exp.	Valid Values	Step	Unit
Unsigned Byte	0	0x00...0xFF	1	None

5.2.1.18 CSVersion_Minor

Contains the supplier-defined minor version as returned by parameter `Signature` in the `ExtendedNetworkControl.Hello.Status` message.

Data Type	Exp.	Valid Values	Step	Unit
Unsigned Byte	0	0x00...0xFF	1	None

5.2.1.19 CSVersion_Release

Contains the supplier-defined release version as returned by parameter Signature in the ExtendedNetworkControl.Hello.Status message.

Data Type	Exp.	Valid Values	Step	Unit
Unsigned Byte	0	0x00...0xFF	1	None

5.2.1.20 Result

Result of the Signature compare.

Data Type	Valid Values	Mnemonic	Description
Enum	0x00	Success	Compare of the signature was successful.
	0x01	NoSuccess	Compare of the signature was not successful.

5.2.1.21 ErrorCode, ErrorInfo

ErrorCode	ErrorInfo		
	ErrorClass	ErrorID	Description of the Error
0x20	0x02	0x20	- RBD is active, or - physical layer test is running, or - ForcedNA was set.
	0x03	0x31	Node has not yet received an ExtendedNetworkControl.Hello.Get message.
		0x32	Node has already successfully received an ExtendedNetworkControl.Welcome.StartResult message.
		0x33	AdminNodeAddress is 0xFFFF. This value is not supported during the cPHY duplex network diagnosis.
		0x35	The configured value of NodeAddress is not valid. The value is either 0xFFFF or in the dynamic address range while AdminNodeAddress is 0xFFFF.
	0x04	0x40	MOST Network Interface Controller is already in cPHY duplex network diagnosis state.

5.3 EnablePort (0x210)

This function is used to enable a particular MOST port.

FBlockID	FktID	OPType	Parameters
ExtendedNetwork-Control (0x0A)	EnablePort (0x210)	StartResult (0x2)	PortNumber, Enabled
		Result (0xC)	—
		Error (0xF)	ErrorCode, ErrorInfo

5.3.1 Parameters

5.3.1.1 PortNumber

Number of MOST Port.

Data Type	Exp.	Valid Values	Step	Unit
Unsigned Byte	0	0x00...0x01	1	None

5.3.1.2 Enabled

Indicates if the MOST port is enabled or disabled

Data Type	Bit #	Valid Values	Description
Boolean	0	False (0)	MOST port is disabled.
		True (1)	MOST port is enabled.

5.3.1.3 ErrorCode, ErrorInfo

ErrorCode	ErrorInfo		
	ErrorClass	ErrorID	Description of the Error
0x20	0x03	0x30	Node is not participating in the cPHY duplex network diagnosis and therefore no MOST port can be enabled or disabled.
		0x31	Another MOST port has been already enabled. – TimingMaster: Only one MOST port is allowed to be enabled at the same time. – TimingSlave: Only the MOST port and the clock reference MOST port are allowed to be enabled at the same time.
		0x32	The slave's clock reference MOST port cannot be enabled or disabled.
		0x33	The MOST port is not used.
	0x04	0x40	The MOST port is not configured in full duplex coaxial mode.

5.4 CableLinkDiagnosis (0x211)

This function is used to start the cable link diagnostics. Cable link diagnostics is used to verify a full duplex MOST port cable link.

FBlockID	FktID	OPType	Parameters
ExtendedNetwork-Control (0x0A)	CableLinkDiagnosis (0x211)	StartResult (0x2)	PortNumber
		Result (0xC)	PortNumber, Result
		Error (0xF)	ErrorCode, ErrorInfo

5.4.1 Parameters

5.4.1.1 PortNumber

PortNumber specifies the MOST port that is connected to the cable link to be diagnosed.

Data Type	Exp.	Valid Values	Step	Unit
Unsigned Byte	0	0x00...0x01	1	None

5.4.1.2 Result

Result of the cable link diagnosis.

Cable link diagnosis was run without any test interruption.

Data Type	Valid Values	Mnemonic	Description
Enum	0x00	NoConnection	The cable is unconnected or has a short between inner and outer conductor.
	0x01	TerminatedConnection	The cable end is correctly terminated with an appropriate impedance, but there is no responding signal from the receiving node. This points towards a properly connected cable with a receiving node that is not powered.
	0x02	PassiveConnection	The receiving node is available, but its bypass is closed, e.g., the receiving node is in reset.
	0x03	ActiveConnection	The receiving node is available and the NetInterface is fully operational.

493 Cable link diagnosis was run with test interruption.

Data Type	Valid Values	Mnemonic	Description
Enum	0x81	DebugInt0	Test was aborted since the Debug Header was used at the same time.
	0x91	DebugInt1	
	0x80	Failure0	Test was aborted due a common processing failure.
	0x82	Failure1	
	0x83	Failure2	
	0x87	Failure3	
	0x90	Failure4	
	0x92	Failure5	
	0x95	Failure6	

494 ### 5.4.1.3 ErrorCode, ErrorInfo

ErrorCode	ErrorInfo		
	ErrorClass	ErrorID	Description of the Error
0x20	0x01	0x10	- The last ENC.CableLinkDiagnosis call has not yet been finished.
	0x02	0x20	- RBD is active or - ForcedNA was set.
	0x03	0x30	Node is not participating in the cPHY duplex network diagnosis and therefore cable link diagnosis cannot be triggered.
		0x31	The respective MOST port is not enabled.
		0x32	Cable link diagnosis cannot be triggered on the slave's clock reference MOST port.

Notes: