MOST

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

MOST50 bPHY Half-Duplex Network Diagnosis
Rev. 3.1
10/2018

MOSTCO CONFIDENTIAL

See page 3 for the terms of disclosure

MOST® COOPERATION

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

MOST50 Half-Duplex Network Diagnosis Rev. 3.1

Change Ref.	Section	Changes
3V1_001	All	Initial document.

Bibliography

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

Number	Document	Revision
[1]	MOST Specification	3.1
[2]	MOST Lean Specification	4.0
[3]	MOST50 bPHY Automotive Sub-Specification	1.0

Table 1: Document references

Glossary

Table 2 lists commonly used MOST terms and their definition.

Term	Definition
BKD TimingMaster	"TimingMaster in backward direction" is a role that is assumed by the subject or the observer during half-duplex network diagnosis.
Diagnosis evaluator	In the root, the diagnosis evaluator starts the MOST50 bPHY half-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_Result`	This report is sent from the diagnosis worker to the diagnosis evaluator to provide the information about a link on which MOST50 bPHY half-duplex network diagnosis was performed.
Diagnosis worker	In the root, the diagnosis worker communicates with the nodes in the MOST network through the MOST network 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 MOST50 bPHY half-duplex network diagnosis process is completed.
Observer	The observer is the node that determines the examination result for the link to the subject.
Participant	A node that participates in a MOST network and does not contain the TimingMaster in default communication direction, which is forward.
Relay	A node between the root and the observer in default communication direction, which is forward.
Root	The node that contains the TimingMaster in default communication direction, which is forward.
`Signature`	The signature of a node contains the `DiagID` and other relevant information.
`Start_Diag`	This request is used by the diagnosis evaluator to instruct the diagnosis worker to begin the half-duplex network diagnosis process.
Subject	The subject is the node with the node position of the observer incremented by one — downstream in forward direction.

Table 2: Glossary entries

1 Introduction

1.1 Purpose

This document specifies the diagnosis concept for a MOST50 bPHY network [3].

The purpose of the MOST50 bPHY half-duplex network diagnosis is to provide a possibility to diagnose the MOST network state with respect to a ring break — or a broken link in general — between two nodes in a half-duplex network structure (ring structure).

The MOST50 bPHY half-duplex network diagnosis obtains the diagnosis information without relying on an external trigger mechanism.

The MOST50 bPHY half-duplex network diagnosis introduces the capability to operate in backward direction. The default MOST communication in forward direction (FWD) and communication in backward direction (BKD) are mutually exclusive. The backward direction can be activated when there is no network activity in forward direction and vice versa.

1.2 Scope

This document contains the specification of the diagnosis method.

1.3 Motivation

The motivation and advantages of using the MOST50 bPHY half-duplex network diagnosis are:

- Rely on half-duplex communication.
- Establish a unique addressing scheme, independent of the node position address.
- Initiate and execute diagnosis without any additional hardware or wiring (e.g., electrical control line).
- Initiate diagnosis without the need for a trigger outside the MOST network.
- Perform diagnosis without the necessity of reaching NetInterface Normal Operation.
- Determine the physical position of a node in the network.
- Gather the signatures from the nodes that are present in the network.
- Gather the state of the network links that connect the nodes.

2 Overview

2.1 Preconditions

For the MOST50 bPHY half-duplex network diagnosis, the following preconditions have to be met:

- The TimingMaster contains an application to control the diagnosis.
- TimingSlaves implement the diagnosis procedure in the MOST network controller. This ensures that nodes without implementation of a network services library are included.
- To start the diagnosis, the TimingMaster is in the NetInterface Off state.
- Communication is based on messages on the MOST control channel.

2.2 Behavior

The MOST50 bPHY half-duplex network diagnosis is intended to perform the following actions:

- Indicate the diagnosis state to every node in the MOST network by the diagnosis flag in the administrative area of the network frame.
- Assign logical node addresses for temporary use during diagnosis.
- Find the nodes that are present in the network and gather node-specific information contained in the signature.

2.3 Limitations

What the MOST50 bPHY half-duplex network diagnosis is not designed for and the limitations:

- Only the first defective link can be found.
- To determine link qualitity, in addition to the MNC, other circuitry is necessary.

3 Diagnosis method

3.1 General

For full-duplex networks, the communication link between two nodes is available in both directions at all times. For half-duplex networks, the direction of the data stream has to be switched.

The MOST50 bPHY half-duplex network diagnosis defines diagnostic capabilities needed to analyze network failures in a half-duplex network. The logic to control the diagnosis procedure resides in the TimingMaster and is divided between the diagnosis worker, which is part of the network services, and the diagnosis evaluator, which is part of the application or the network supervisor.

The network services library triggers the MOST50 bPHY half-duplex network diagnosis and gathers all information. The diagnosis results are passed to the diagnosis evaluator for further analysis.

For diagnosis, the same link is used in forward direction and backward direction. Switching between the forward direction and the backward direction is handled by a set of well-defined timers. The timer allow for sharing of the same physical line and prevention of collisions. The diagnosis worker uses the control channel to distribute the timer values.

In MOST50 bPHY half-duplex network diagnosis, a distinction is made between the TimingMaster, which is referred to as the root, and the BKD TimingMaster. The root provides the network clock in default communication direction, which is forward. This role is always assumed by the same node, the one that acts as TimingMaster outside diagnosis when NetInterface Normal Operation can be reached. The BKD TimingMaster is the TimingMaster in backward direction. This role is assumed by the subject or the observer during diagnosis.

The observer is the node that determines the examination result for the link to the subject. The subject is the following node downstream in forward direction. The observer reports the result to the diagnosis worker.

3.2 Node identification

The network services library in the root controls the identification process, that is, it collects the signatures of the particiants. The participants, that is, all other nodes, act autonomously without any need for an application that is associated with the MOST network controller.

All participants are able to send the information to the root without the help of any additional diagnosis line.

3.3 Basic Concept

During the diagnosis phase, the links between adjacent MOST nodes are examined one by one. In order to test a certain link, one MOST node assumes the role of the observer and the next node downstream the role of the subject.

The link to be examined is the link between the observer and the subject. The subject is the node that is connected to the Tx of the observer in forward direction.

To control, which node becomes the observer and which node becomes the subject, the root sends `ExtendedNetworkControl.ReverseRequest.StartResult` to the blocking broadcast address. The receiving nodes evaluate their current node position when the message is received.

If a node does not receive the message, it cannot assume its role during the diagnosis process.

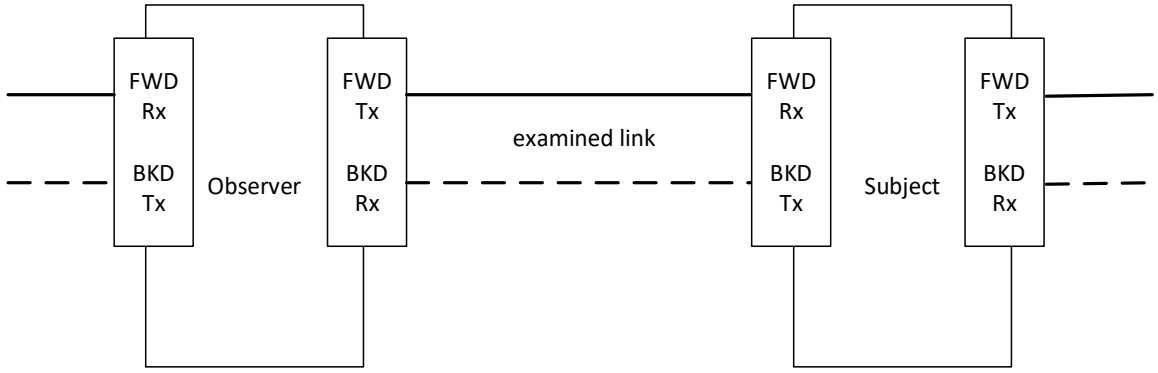

Figure 1: Link examination

3.3.1 Subject

If the subject receives `ExtendedNetworkControl.ReverseRequest.StartResult` in forward direction, it switches to backward direction when t_{BKD} expires and switches on the MOST output.

By default, the node address of the subject is $0FFE_{16}$.

3.3.2 Observer

If the observer receives `ExtendedNetworkControl.ReverseRequest.StartResult` in forward direction, it switches to backward direction when t_{BKD} expires. The observer changes its logical node address to the contained `ObserverAddress`; this address is used when responding to the command. The observer then waits for network activity.

If the subject switches on the MOST output before t_{Wait} expires, the observer sends a message that indicates the detection of network activity to the blocking broadcast address after $t_{DiagSend}$ expires. If the subject does not switch on the MOST output before t_{Wait} expires, the observer configures itself as BKD TimingMaster, switches on the MOST output in backward direction and sends a message that indicates failure to detect network activity to the blocking broadcast address after $t_{DiagSend}$ expires.

Therefore, the diagnosis procedure, using a link-by-link examination approach, ensures that the root always receives a valid value in the `ObserverResult` parameter (`SlaveOK`, `SlaveWrongNodePosition`, `MasterNoRxSignal`, `MasterRxLock` or `NoResult`; see 6.1.1.2.10).

If the root does not receive any message from the observer, the diagnosis phase was interrupted irregularly. The diagnosis worker cancels the diagnosis. The diagnosis evaluator may start a new diagnosis session.

3.3.3 Relay

All nodes between the root and the observer act as relays. They switch between forward and backward direction, based on the corresponding timers, and relay any message. A node becomes a relay if its node position is less than the observer's node position and its logical node address is valid (i.e., not equal to $0FFE_{16}$) because it previously acted as observer.

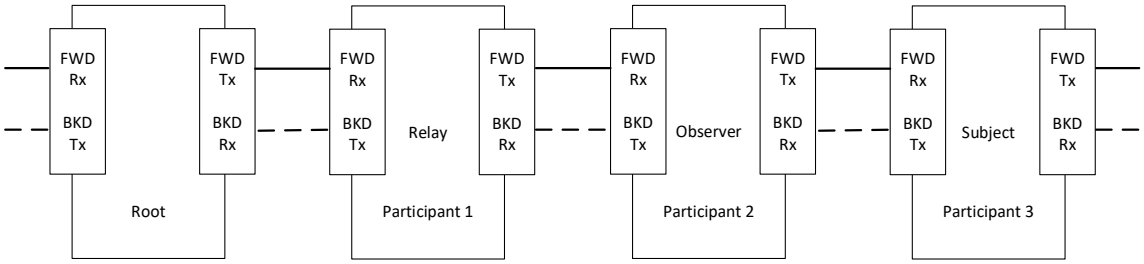

Figure 2: System testing

3.3.4 Other nodes

The root, the observer, the subject, and the relays have a role during MOST50 bPHY half-duplex network diagnosis. All other nodes, which have no role, are not considered and must not disturb the diagnosis.

3.4 API

3.4.1 Functions

The diagnosis worker initiates the node identification by sending the following API command to its MOST network controller (the NetInterface has to be in the NetInterface Off state):

- `MNC.NetworkDiagnosisHalfDuplex.StartResult`

If the MOST network controller is ready to perform diagnosis, a result message is returned to the diagnosis worker:

- `MNC.NetworkDiagnosisHalfDuplex.Result`

To switch on the MOST output, the diagnosis worker sends the following command to the MOST network controller:

- `ExtendedNetworkControl.EnableTX.StartResult`

If the function was executed successfully by switching on the MOST output and setting the diagnosis flag, a result message is returned to the diagnosis worker:

- `ExtendedNetworkControl.EnableTX.Result`

The participants are initialized by sending out a message to the blocking broadcast address:

- `ExtendedNetworkControl.ReverseRequest.StartResult(SubjectPosition, tBKD, tSend, tFWD, RequestID, RequestList)`

If the function was executed successfully, a result message is sent from the observer to the blocking broadcast address:

- ExtendedNetworkControl.ReverseRequest.Result(RequestID, ResultList)

The diagnosis worker terminates node identification by sending the following API command to the MOST network controller (has to be in diagnosis state):

- NetworkDiagnosisHalfDuplexEnd.StartResult

If the function was executed successfully, a result message is returned to diagnosis worker:

- NetworkDiagnosisHalfDuplexEnd.Result

3.4.2 Root-internal communication

The following requests and reports are used to describe root-internal communication between the diagnosis evaluator and the diagnosis worker.

3.4.2.1 Start_Diag

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

```
Start_Diag{
}
```

3.4.2.2 Diagnosis_Result

The Diagnosis_Result report is sent from the diagnosis worker to the diagnosis evaluator to provide information about a link that was examined as well as the signature of the observer.

```
Diagnosis_Result{
  ObserverResult
  LQResult
  Signature
}
```

The ObserverResult parameter may contain one of the values that are specified in 6.1.1.2.10.
LQResult contains the result of the link quality diagnosis performed by the observer. The values are supplier specific.
The Signature parameter contains the information listed in 6.1.1.2.12.

3.4.2.3 End_Diag

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

```
End_Diag{
}
```

3.5 Process description

The diagnosis starts with verifying the link between the root (observer in that case) and the first participant (participant 1; subject in that case).

If this link is verified as fully functional (the subject has switched on the MOST output in backward direction), the next link is examined by shifting the observer role and subject role to the next two nodes (participant 1 becomes the observer, participant 2 becomes the subject). Since the first link is verified as functional with respect to communication in forward and backward direction, the root can rely on receiving a message from the observer (participant 1). If the second link is also verified as fully functional, the observer role and subject role are shifted to the next two nodes, and so on.

Thus, the communication path from the current observer to the root is always verified before an additional link is examined.

The diagnosis process ends either if

- a subject does not switch on the MOST output (ring break detected) and therefore a message that indicates failure to detect network activity is sent from the observer to the root or
- the observer when configured as BKD TimingMaster reaches stable lock (ring closed)
- timeout of $t_{NextSubject}$ without receiving `ReverseRequest.Result`

3.5.1 Implementation

The diagnosis worker in the root starts the diagnosis by sending `NetworkDiagnosisHalfDuplex.StartResult` to its MOST network controller.

If the NetInterface is in NetInterface Off state, `NetworkDiagnosisHalfDuplex.Result` indicates successful execution. In order to enable the Tx output, the diagnosis worker sends `ExtendedNetworkControl.EnableTx.StartResult` to its MOST network controller.

The following actions are performed by the MOST network controller in the root on `ExtendedNetworkControl.EnableTx.StartResult`:

- Enable Tx (switch on the MOST output)
- Distribute the diagnosis flag
- Ignore network activity

The root or any participant in the ring — or ring segment if the ring is broken — recognizes the diagnosis state if it detects network activity with the diagnosis flag set.

The following actions are performed by the participant when network activity is detected and the diagnosis flag is set:

- Enable Tx
- The NetInterface remains in NetInterface Init until the diagnosis process is completed. (NetInterface Normal Operation is not reached. Message exchange is performed by the MOST network controller.)
- Wait for further MOST message reception

Which node acts as a observer and which node acts as a subject is defined by the `ExtendedNetworkControl.ReverseRequest.StartResult` message which contains a parameter `SubjectPosition`:

- The node whose node position matches the value of parameter `SubjectPosition` becomes the subject (acting as the BKD TimingMaster).
- The node whose node position matches `SubjectPosition - 1` becomes the observer.

The observer sets its logical node address to the value contained in the parameter `ObserverAddress`.

All nodes with a valid logical node address (`ObserverAddress`) which are not observer and not subject act as relays.

Observer:

- Waits t_{Wait} for network activity in backward direction
- Sends `ExtendedNetworkControl.ReverseRequest.Result` in backward direction, indicating success or failure to detect network activity

Subject:

- Switches on the MOST output in backward direction

Relay:

- Only nodes already initialized with a valid `ObserverAddress` .
- Switches back and forth with correct timing.

3.5.2 Process flow examples

The following illustrations and MSCs provide some examples for different diagnosis scenarios.

3.5.2.1 Broken link at root Tx

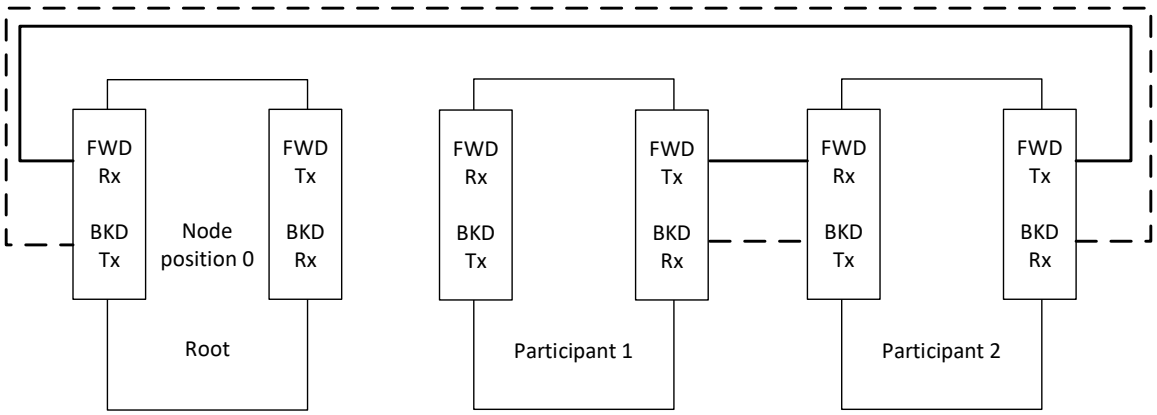

Figure 3: Broken link between root and participant 1

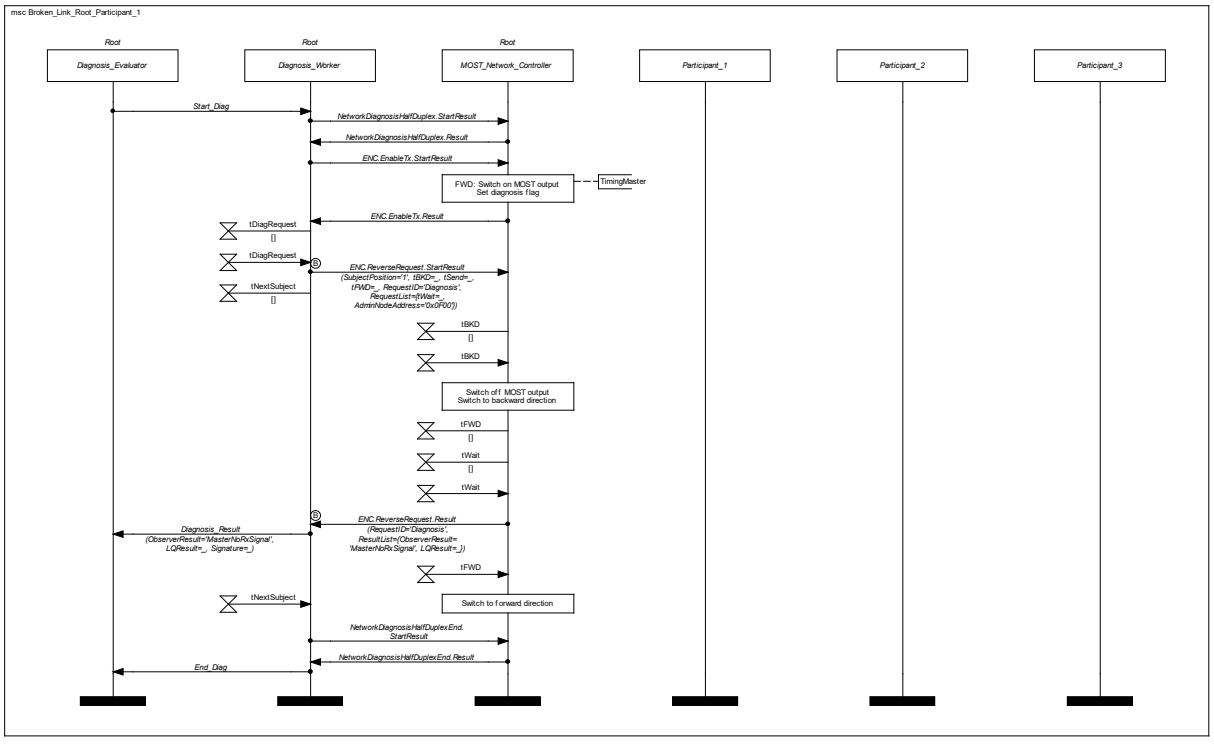

Figure 4: MSC – broken link between root and participant 1

3.5.2.2 Broken link between two participants

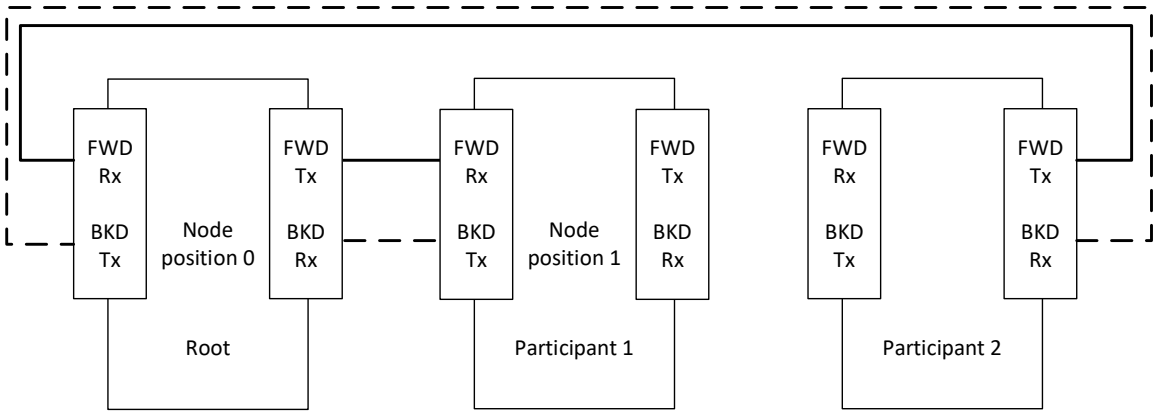

Figure 5: Broken link between participant 1 and 2

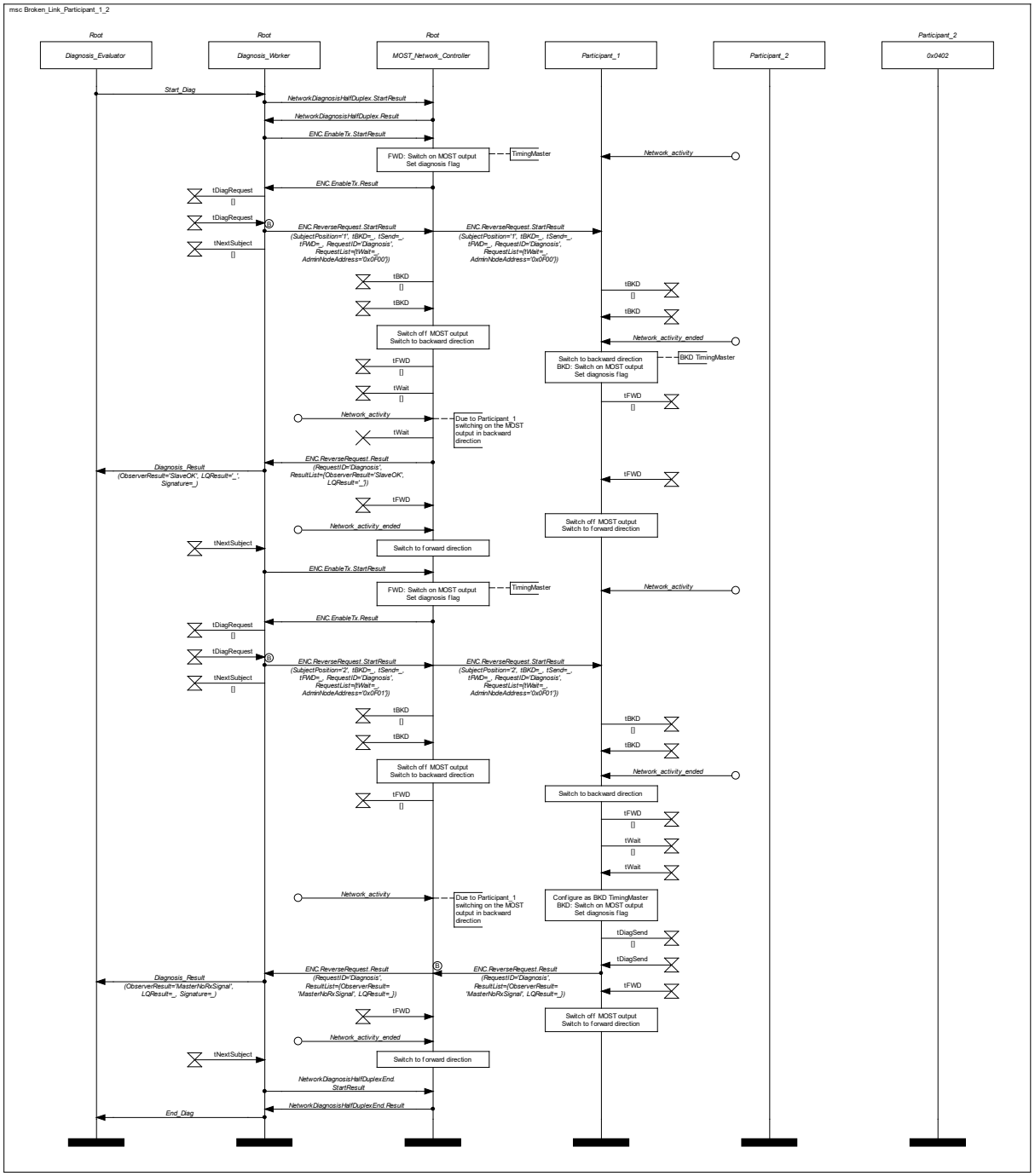

Figure 6: MSC – broken link between participant 1 and 2

3.5.2.3 Broken link at root Rx

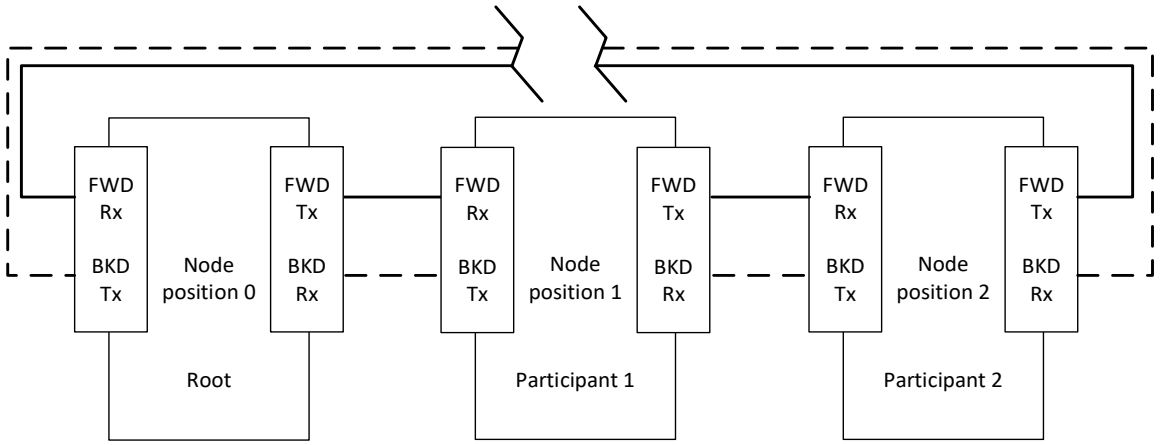

Figure 7: Broken link between participant 2 and root

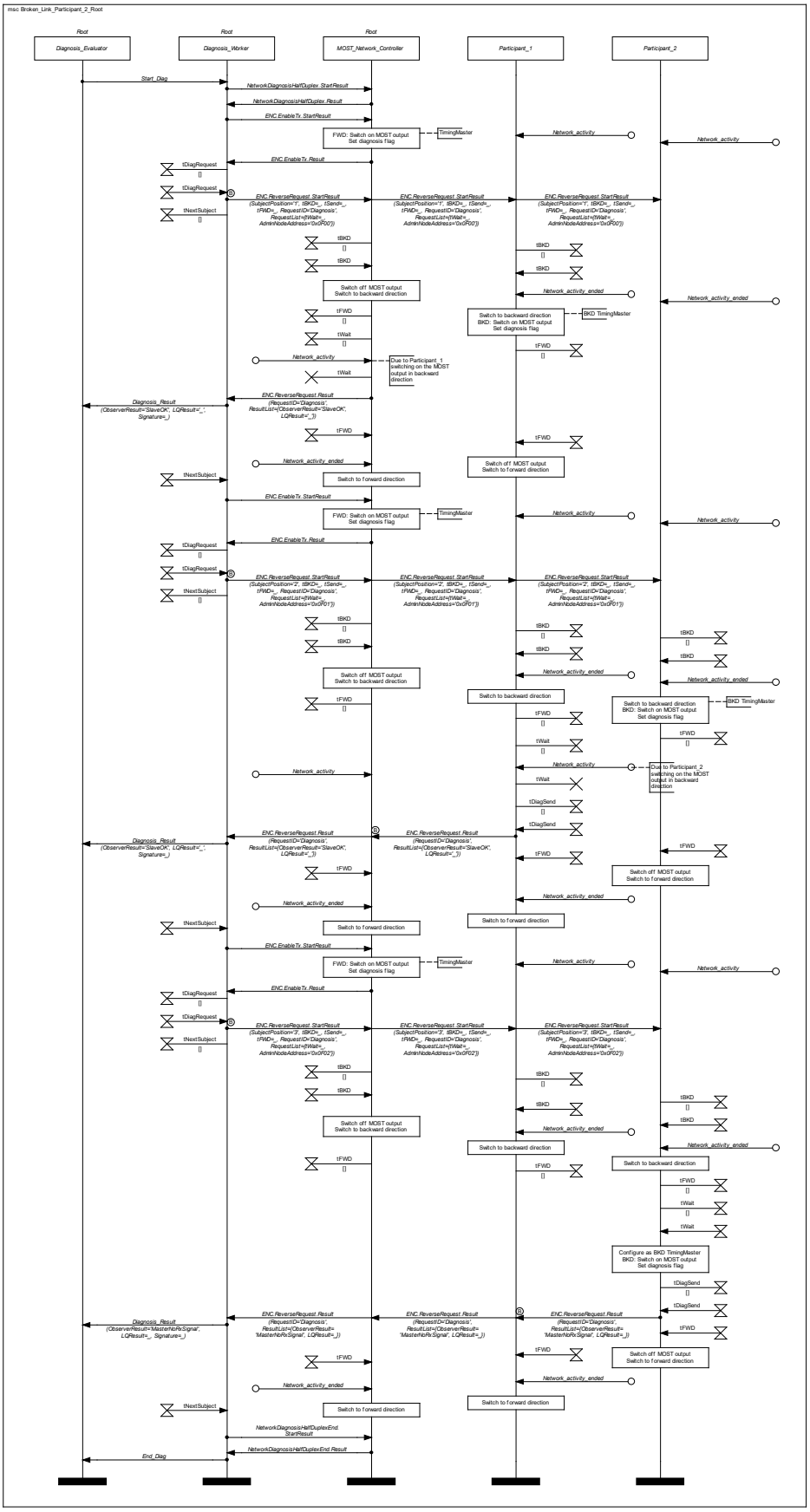

Figure 8: MSC – broken link between participant 2 and root

3.5.2.4 Closed ring

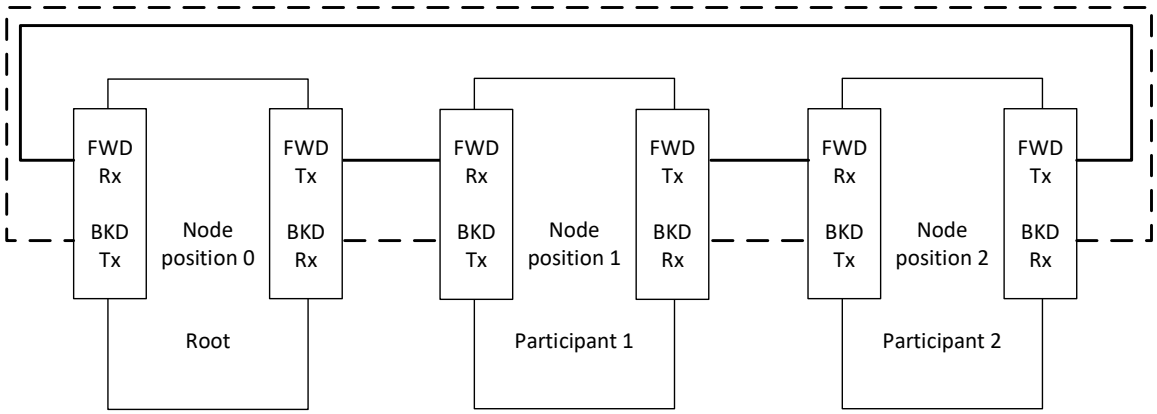

Figure 9: Closed ring

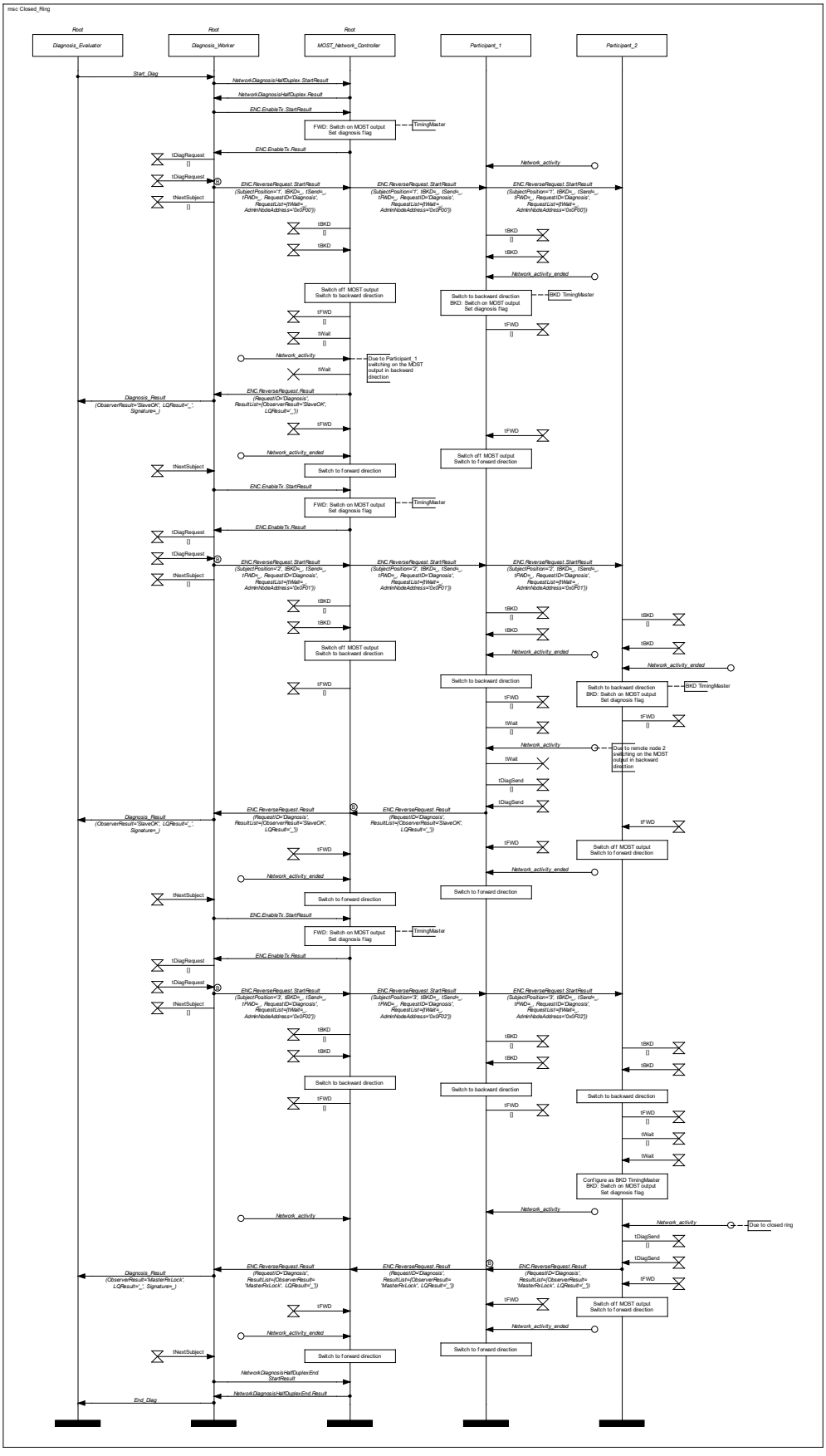

Figure 10: MSC – Closed ring

3.5.2.5 Reset during diagnosis execution

If the diagnosis is disturbed, for example, by a reset of a relay, observer or subject, the root gets no information in backward direction at all; the diagnosis worker detects that no message is received, which indicates an erroneous diagnosis phase. The diagnosis phase is cancelled and may be started again.

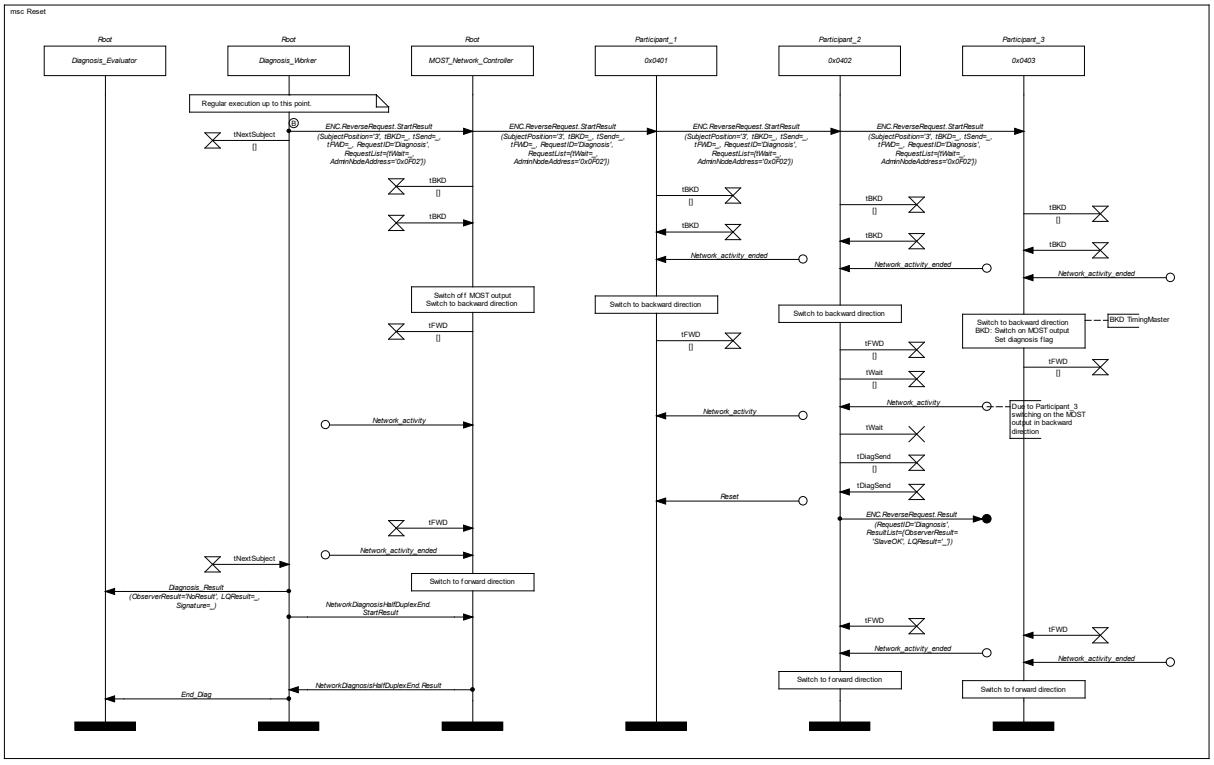

Figure 11: MSC – Reset during diagnosis execution

A node reset may occur while the network operates in forward or backward direction. In both cases, a node coming from reset does neither act as a relay (no valid `ObserverAddress`), nor as observer or subject (if the reset happens after sending out the `ExtendedNetworkControl.ReverseRequest.StartResult` message). This mechanism ensures a failure during the diagnosis, which can be detected by the root.

3.5.3 Ending the MOST50 bPHY half-duplex network diagnosis

To end the diagnosis, the diagnosis worker performs a network startup with cleared diagnosis flag. Network startup might not be successful due to a network failure.

4 Timing definitions

Table 3 lists the timers that are relevant for MOST50 bPHY half-duplex network diagnosis. Examples of timer values are given in 4.7.

Name	Value	Unit	Purpose
t_{Wait}	System integrator specific	ms	Time the observer waits for network activity in backward direction before sending a message that indicated failure to detect network activity.
t_{BKD}	System integrator specific	ms	Time a node waits before switching to backward direction.
t_{FWD}	System integrator specific	ms	Time a node waits before switching to forward direction again. $t_{FWD} > t_{Wait} + t_{DiagSend}$
$t_{DiagRequest}$	System integrator specific	ms	Ensures that all TimingSlaves have enough time to detect network activity before the diagnosis worker sends `ReverseRequest.StartResult`.
$t_{DiagSend}$	System integrator specific	ms	Ensures that all TimingSlaves have enough time to detect network activity before `ReverseRequest.Result` is sent.
$t_{NextSubject}$	System integrator specific	ms	Triggers the next diagnosis step from the current link to the following link or ends the diagnosis. Time the root waits before sending `Diagnosis_Result(NoResult)`. $t_{NextSubject} > t_{BKD} + t_{FWD}$

Table 3: Timers

4.1 Timer t_{Wait}

t_{Wait} is distributed in `ExtendedNetworkControl.ReverseRequest.StartResult`. t_{Wait} controls how long an observer waits for network activity in backward direction before sending a message that indicates failure to detect network activity.

Start and Stop conditions

An observer shall start t_{Wait} when t_{BKD} expires.

An observer shall stop t_{Wait} when it detects network activity.

Timer expiration

When t_{Wait} expires, the observer shall send `ExtendedNetworkControl.ReverseRequest.Result` to the root.

4.2 Timer t_{BKD}

t_{BKD} is is distributed in `ExtendedNetworkControl.ReverseRequest.StartResult`. t_{BKD} controls how long a node waits before switching to backward direction.

Start and Stop conditions

A node shall start t_{BKD} when it receives `ExtendedNetworkControl.ReverseRequest.StartResult`.

Timer expiration

When t_{BKD} expires, after the end of network activity, the node shall switch to backward direction.

4.3 Timer t_{FWD}

t_{FWD} is distributed along with `ExtendedNetworkControl.ReverseRequest.StartResult`. It controls how long a node waits before switching to forward direction.

Start and Stop conditions

A node shall start t_{FWD} when t_{BKD} expires.

Timer expiration

When t_{FWD} expires, after the end of network activity, the node shall switch to forward direction.

4.4 Timer $t_{DiagSend}$

$t_{DiagSend}$ ensures that all TimingSlaves have enough time to detect network activity before `ReverseRequest.Result` is sent by the observer.

Start and Stop conditions

An observer that is not the root shall start $t_{DiagSend}$ when it detects network activity.

An observer that is not the root shall start $t_{DiagSend}$ when it switches on the MOST output in backward direction after configuring itself as BKD TimingMaster.

Timer expiration

When $t_{DiagSend}$ expires, the observer shall send `ReverseRequest.Result` to the blocking broadcast address.

4.5 Timer $t_{DiagRequest}$

$t_{DiagRequest}$ ensures that all TimingSlaves have enough time to detect network activity before `ReverseRequest.StartResult` is sent by the diagnosis worker.

Start and Stop conditions

The diagnosis worker shall start $t_{DiagRequest}$ when it receives `EnableTx.Result`.

Timer expiration

When $t_{DiagRequest}$ expires, the diagnosis worker shall send `ReverseRequest.StartResult` to the blocking broadcast address.

4.6 Timer $t_{NextSubject}$

$t_{NextSubject}$ triggers the next diagnosis step from the current link to the following link or ends the diagnosis.

Start and Stop conditions

The diagnosis worker shall start $t_{NextSubject}$ when it sends `ReverseRequest.StartResult`.

Timer expiration

When $t_{NextSubject}$ expires after reception of `ReverseRequest.Result` with `MasterRxLock`, `MasterRxNoSignal`, `SlaveWrongNodePosition` or `NoResult`, the diagnosis worker shall send `NetworkDiagnosisHalfDuplexEnd.StartResult` to the MOST network controller.

When $t_{NextSubject}$ expires after reception of `ReverseRequest.Result` with `SlaveOK`, the diagnosis worker shall send `EnableTx.StartResult` to the MOST network controller.

When $t_{NextSubject}$ expires without receiving `ReverseRequest.Result`, the diagnosis worker shall report `NoResult` to the diagnosis evaluator and send `NetworkDiagnosisHalfDuplexEnd.StartResult` to the MOST network controller.

4.7 Example values for timers

Table 4 provides examples for reasonable, aligned timer values that are compatible with the described process.

Name	Value	Unit	Purpose
t_{Wait}	300	ms	Time the observer waits for network activity in backward direction before sending a message that indicated failure to detect network activity.
t_{BKD}	100	ms	Time a node waits before switching to backward direction.
t_{FWD}	500	ms	Time a node waits before switching to forward direction again.
$t_{DiagRequest}$	200	ms	Ensures that all TimingSlaves have enough time to detect network activity before the diagnosis worker sends `ReverseRequest.StartResult`.
$t_{DiagSend}$	100	ms	Ensures that all TimingSlaves have enough time to detect network activity before `ReverseRequest.Result` is sent.
$t_{NextSubject}$	700	ms	Triggers the next diagnosis step from the current link to the following link or ends the diagnosis. Time the root waits before sending `ReverseRequest.Result(NoResult)`.

Table 4: Example values for timers

5 MOST network controller functions

The following functions of FBlock MOST network controller (FBlockID 00_{16}) are used in the MOST50 bPHY half-duplex network diagnosis. The listed parameters serve the purpose of providing the minimum information that is required to perform an operation. In an acutal implementation, the parameter list might differ in extent and parameter position.

5.1.1 NetworkDiagnosisHalfDuplex ($52E_{16}$)

Occurrence: Mandatory

This function is used to start the MOST50 bPHY half-duplex network diagnosis.

5.1.1.1 Format of Function

Function classes: Unclassified Method

FBlock	Function	OPType	Parameter
MNC (00_{16})	NetworkDiagnosis HalfDuplex ($52E_{16}$)	StartResult	-
		Result	-
		Error	ErrorCode, ErrorInfo

5.1.1.2 Parameter

5.1.1.2.1 ErrorCode

Basis data type	Range of values	Code	Symbolic Name	Description
Enum	01_{16}...20_{16}	01_{16}	FBlockIDNotAvailable	FBlockID not available
		02_{16}	InstIDNotAvailable	InstID not available
		03_{16}	FktIDNotAvailable	FktID not available
		04_{16}	OPTypeNotAvailable	OPType not available
		05_{16}	InvalidLength	Invalid length
		06_{16}	ParamWrong_OutOfRange	Parameter wrong or out of range
		$0C_{16}$	SegmentationError	Segmentation error
		20_{16}	FunctionSpecificError	Function specific error

5.1.1.2.2 ErrorInfo

Basis data type	Length	Condition	Description
Stream		-	Content: ErrorData (repeated)

For ErrorCode 20_{16}, ErrorData[1] is interpreted as follows:

Value	Description
22_{16}	The node is not in NetInterface Off state.
$3A_{16}$	Functionality is not available.

5.1.1.2.3 ErrorData

Additional error information.

Basis data type	Exp.	Range of values	Step	Unit
Unsigned Byte			1	none

5.1.2 NetworkDiagnosisHalfDuplexEnd ($52F_{16}$)

Occurrence: Mandatory

5.1.2.1 Format of Function

Function classes: Unclassified Method

FBlock	Function	OPType	Parameter
MNC (00_{16})	NetworkDiagnosis HalfDuplexEnd ($52F_{16}$)	StartResult	-
		Result	-
		Error	ErrorCode, ErrorInfo

5.1.2.2 Parameter

5.1.2.2.1 ErrorCode

Basis data type	Range of values	Code	Symbolic Name	Description
Enum	01_{16}...20_{16}	01_{16}	FBlockIDNotAvailable	FBlockID not available
		02_{16}	InstIDNotAvailable	InstID not available
		03_{16}	FktIDNotAvailable	FktID not available
		04_{16}	OPTypeNotAvailable	OPType not available
		05_{16}	InvalidLength	Invalid length
		06_{16}	ParamWrong_OutOfRange	Parameter wrong or out of range
		$0C_{16}$	SegmentationError	Segmentation error
		20_{16}	FunctionSpecificError	Function specific error

5.1.2.2.2 ErrorInfo

Basis data type	Length	Condition	Description
Stream		-	Content: ErrorData (repeated)

For ErrorCode 20_{16}, ErrorData[1] is interpreted as follows:

Value	Description
22_{16}	Network is not in MOST50 bPHY half-duplex diagnosis mode.
$3A_{16}$	Functionality is not available.

5.1.2.2.3 ErrorData

Additional error information.

Basis data type	Exp.	Range of values	Step	Unit
Unsigned Byte			1	none

6 ExtendedNetworkControl FBlock functions

The following functions of FBlock ExtendedNetworkControl (FBlockID $0A_{16}$) are used in the MOST50 bPHY half-duplex network diagnosis. The listed parameters serve the purpose of providing the minimum information that is required to perform an operation. In an acutal implementation, the parameter list might differ in extent and parameter position.

6.1.1 ReverseRequest (222_{16})

Occurrence: Mandatory

This function is used to retrieve diagnosis results pertaining to the subject from the observer. The request is sent as a broadcast message in forward direction. The Result is returned in backward direction.

6.1.1.1 Format of Function

Function classes: Unclassified Method

FBlock	Function	OPType	Parameter
ExtendedNetwork Control ($0A_{16}$)	ReverseRequest (222_{16})	StartResult	SubjectPosition, tBKD, tSend, tFWD, RequestID, RequestList
		Result	RequestID, ResultList
		Error	ErrorCode, ErrorInfo

6.1.1.2 Parameter

6.1.1.2.1 SubjectPosition

Indicates the node position of the BKD TimingMaster. Determines which node is the observer (node position = SubjectPosition-1) and which is the subject (node position = SubjectPosition).

Basis data type	Exp.	Range of values	Step	Unit
Unsigned Byte			1	none

6.1.1.2.2 tBKD

Defines the time in [ms] for switching the message direction, after an ExtendedNetworkControl. ReverseRequest has been received. This is the time the root, the observer, the subject, and the relays wait before switching to the backward direction.

Basis data type	Exp.	Range of values	Step	Unit
Unsigned Word			1	ms

6.1.1.2.3 tSend

Defines the time the node has to wait with communication after the message direction has been switched. This is the time the observer waits before sending in backward direction. The time specifies when the result is sent back at the earliest.

Basis data type	Exp.	Range of values	Step	Unit
Unsigned Word			1	ms

6.1.1.2.4 tFWD

For `RequestID Diagnosis`, this parameter defines the time the node resides in backward communication direction before it switches back to forward direction.

Basis data type	Exp.	Range of values	Step	Unit
Unsigned Word			1	ms

6.1.1.2.5 RequestID

Specifies the request type of the `ExtendedNetworkControl.ReverseRequest` message. The values that are greater than 00_{16} may be defined by the MNC supplier.

Basis data type	Range of values	Code	Symbolic Name	Description
Enum	00_{16}...FF_{16}	00_{16}	`Diagnosis`	Diagnosis requested

6.1.1.2.6 RequestList

Lists the parameters depending on the `RequestID`.

Basis data type	Length	Condition	Description
Stream		RequestID = 00_{16}	Content: tWait, ObserverAddress

6.1.1.2.7 tWait

Defines the time observer waits for network activity before switching itself to BKD TimingMaster mode.

Basis data type	Exp.	Range of values	Step	Unit
Unsigned Word			1	ms

6.1.1.2.8 ObserverAddress

The logical node address used during MOST50 bPHY half-duplex network diagnosis.

Basis data type	Exp.	Range of values	Step	Unit
Unsigned Word			1	none

6.1.1.2.9 ResultList

Lists the parameters depending on the `RequestID`.

Basis data type	Length	Condition	Description
Stream		RequestID = 00_{16}	Content: ObserverResult, LQResult, Signature

6.1.1.2.10 ObserverResult

The result regarding network activity, obtained by the observer.

Basis data type	Range of values	Code	Symbolic Name	Description
Enum	00_{16}...FF_{16}	00_{16}	`SlaveOk`	The observer is configured as TimingSlave in backward direction and detects network activity. It is in node position 1.
		01_{16}	`SlaveWrongNodePosition`	The observer is configured as TimingSlave in backward direction and detects network activity. It is not in node position 1.

Basis data type	Range of values	Code	Symbolic Name	Description
		10_{16}	MasterNoRxSignal	The observer is configured as BKD TimingMaster and does not detect network activity.
		11_{16}	MasterRxLock	The observer is configured as BKD TimingMaster and reaches stable lock. The ring is closed.
		FF_{16}	NoResult	No diagnosis result

6.1.1.2.11 LQResult

The result of the link quality diagnosis performed by the observer. The values are supplier specific.

Basis data type	Range of values	Code	Symbolic Name	Description
Enum	00_{16}...FF_{16}	–	–	–

6.1.1.2.12 Signature

Unique signature of a node.

Basis data type	Length	Condition	Description
Stream			Content: NodeAddress, GroupAddress, MACAddress_47to32, MACAddress_31to16, MACAddress_15to0, NodePositionAddress, DiagID, NumberOfPorts

6.1.1.2.13 NodeAddress

The logical node address of the observer.

Basis data type	Exp.	Range of values	Step	Unit
Unsigned Word			1	none

6.1.1.2.14 GroupAddress

The group address of the observer.

Basis data type	Exp.	Range of values	Step	Unit
Unsigned Word			1	none

6.1.1.2.15 MACAddress_47to32

Packet EUI-48 Bits 47:32 of the MAC address as entered in the identification string.

Basis data type	Exp.	Range of values	Step	Unit
Unsigned Word			1	none

6.1.1.2.16 MACAddress_31to16

Packet EUI-48 Bits 31:16 of the MAC address as entered in the identification string.

Basis data type	Exp.	Range of values	Step	Unit
Unsigned Word			1	none

6.1.1.2.17 MACAddress_15to0

Packet EUI-48 Bits 15:0 of the MAC address as entered in the identification string.

Basis data type	Exp.	Range of values	Step	Unit
Unsigned Word			1	none

6.1.1.2.18 NodePositionAddress

The node position address of the observer during diagnosis.

Basis data type	Exp.	Range of values	Step	Unit
Unsigned Word			1	none

6.1.1.2.19 DiagID

The diagnosis identifier of the observer.

Basis data type	Exp.	Range of values	Step	Unit
Unsigned Word			1	none

6.1.1.2.20 NumberOfPorts

The number of ports that the observer has available.

Basis data type	Exp.	Range of values	Step	Unit
Unsigned Byte			1	none

6.1.1.2.21 ErrorCode

Basis data type	Range of values	Code	Symbolic Name	Description
Enum	01_{16}...20_{16}	01_{16}	FBlockIDNotAvailable	FBlockID not available
		02_{16}	InstIDNotAvailable	InstID not available
		03_{16}	FktIDNotAvailable	FktID not available
		04_{16}	OPTypeNotAvailable	OPType not available
		05_{16}	InvalidLength	Invalid length
		06_{16}	ParamWrong_OutOfRange	Parameter wrong or out of range
		$0C_{16}$	SegmentationError	Segmentation error
		20_{16}	FunctionSpecificError	Function specific error

6.1.1.2.22 ErrorInfo

Basis data type	Length	Condition	Description
Stream		-	Content: ErrorData (repeated)

For ErrorCode 20_{16}, ErrorData[1] is interpreted as follows:

Value	Description
10_{16}	The last ExtendedNetworkControl.ReverseRequest call has not yet been finished.
30_{16}	The node is not in MOST50 bPHY half-duplex network diagnosis mode.
31_{16}	The node cannot process the request.
32_{16}	The node has an invalid AdminNodeAddress.
33_{16}	The node is located behind the subject.
34_{16}	The node is at the wrong node position.

6.1.1.2.23 ErrorData

Additional error information.

Basis data type	Exp.	Range of values	Step	Unit
Unsigned Byte		00_{16}...FF_{16}	1	none

6.1.2 EnableTx (223_{16})

Occurrence: Mandatory

This function is used to switch on the MOST output in the TimingMaster during the MOST50 bPHY half-duplex network diagnosis.

6.1.2.1 Format of Function

Function classes: Unclassified Method

FBlock	Function	OPType	Parameter
ExtendedNetwork Control ($0A_{16}$)	EnableTx (223_{16})	StartResult	PortNumber
		Result	-
		Error	ErrorCode, ErrorInfo

6.1.2.2 Parameter

6.1.2.2.1 PortNumber

Number of network port.

Basis data type	Exp.	Range of values	Step	Unit
Unsigned Byte			1	none

6.1.2.2.2 ErrorCode

Basis data type	Range of values	Code	Symbolic Name	Description
Enum	01_{16}...20_{16}	01_{16}	FBlockIDNotAvailable	FBlockID not available
		02_{16}	InstIDNotAvailable	InstID not available
		03_{16}	FktIDNotAvailable	FktID not available
		04_{16}	OPTypeNotAvailable	OPType not available
		05_{16}	InvalidLength	Invalid length
		06_{16}	ParamWrong_OutOfRange	Parameter wrong or out of range
		$0C_{16}$	SegmentationError	Segmentation error
		20_{16}	FunctionSpecificError	Function specific error

6.1.2.2.3 ErrorInfo

Basis data type	Length	Condition	Description
Stream		-	Content: ErrorData

For ErrorCode 20_{16}, ErrorData[1] is interpreted as follows:

Value	Description
30_{16}	The node is not in MOST50 bPHY half-duplex network diagnosis mode.
31_{16}	The network port is not in use.
32_{16}	The node is not – a TimingMaster or – in the correct NetInterface state.
39_{16}	The wrong PortNumber is used.

6.1.2.2.4 ErrorData

Basis data type	Exp.	Range of values	Step	Unit
Unsigned Byte		00_{16}...FF_{16}	1	none

Notes: